



**RANCANG BANGUN APLIKASI PENDISTRIBUSIAN
BAHAN BAKAR MINYAK PADA SPBU PERTAMINA
WILAYAH JEMBER MENGGUNAKAN ALGORITMA DIJKSTRA
BERBASIS WEB**

SKRIPSI

Oleh

Affan Taruna Santoso

NIM 122410101001

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2016



**RANCANG BANGUN APLIKASI PENDISTRIBUSIAN
BAHAN BAKAR MINYAK PADA SPBU PERTAMINA
WILAYAH JEMBER MENGGUNAKAN ALGORITMA DIJKSTRA
BERBASIS WEB**

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

Affan Taruna Santoso

NIM 122410101001

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2016

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Ayahanda Hari Santoso dan Ibunda Sri Hayuningsih tercinta;
2. Adik-adikku Firman Rizki Santoso, Amirah Zain Risanti, Alya Maudi Rahmawati, dan Rafa Adelia Rizki;
3. Keluarga besar H. Soewarso;
4. Sahabatku bersama doa dan dukungannya;
5. Guru-guruku sejak taman kanak-kanak sampai dengan perguruan tinggi;
6. Almamater Program Studi Sistem Informasi Universitas Jember;

MOTO

“Niscaya Allah akan mengangkat (derajat) orang-orang yang beriman diantaramu dan orang-orang yang diberi ilmu beberapa derajat”. (QS. Al-Mujadalah : 11)¹



¹ Kementerian Agama Republik Indonesia. 2014. *Ummul Mukminin : Al Qur'an dan Terjemahannya untuk Wanita*. Jakarta: OASIS Terrace Resident

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Affan Taruna Santoso

NIM : 122410101001

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Rancang Bangun Aplikasi Pendistribusian Bahan Bakar Minyak Pada SPBU Pertamina Wilayah Jember Menggunakan Algoritma Dijkstra Berbasis Web”, 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, Oktober 2016

Yang menyatakan,

Affan Taruna Santoso

NIM 122410101001

SKRIPSI

**RANCANG BANGUN APLIKASI PENDISTRIBUSIAN
BAHAN BAKAR MINYAK PADA SPBU PERTAMINA
WILAYAH JEMBER MENGGUNAKAN ALGORITMA DIJKSTRA
BERBASIS WEB**

Oleh

Affan Taruna Santoso

NIM 122410101001

Pembimbing :

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

Dosen Pembimbing Pemdamping : Winda Eka Yulia Retnani, S.Kom., M.T

PENGESAHAN PEMBIMBING

Skripsi berjudul “Rancang Bangun Aplikasi Pendistribusian Bahan Bakar Minyak Pada SPBU Pertamina Wilayah Jember Menggunakan Algoritma Dijkstra Berbasis Web”, telah diuji dan disahkan pada:

hari, tanggal :

tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

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

Windi Eka Yulia Retnani, S.Kom, MT
NIP 198403052010122002

PENGESAHAN

Skripsi berjudul “Rancang Bangun Aplikasi Pendistribusian Bahan Bakar Minyak Pada SPBU Pertamina Wilayah Jember Menggunakan Algoritma Dijkstra Berbasis Web”, telah diuji dan disahkan pada:

hari, tanggal :

tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji,

Penguji I,

Penguji II,

Drs. Antonius Cahya P, M.App., Sc., Ph.D
NIP. 196909281993021001

Diah Ayu Retnani W, ST., M.Eng
NIP. 198603052014042001

Mengesahkan
Ketua Program Studi,

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

RINGKASAN

Rancang Bangun Aplikasi Pendistribusian Bahan Bakar Minyak Pada SPBU Pertamina Wilayah Jember Menggunakan Algoritma Dijkstra Berbasis Web.

Proses pendistribusian bahan bakar minyak saat ini di wilayah Kabupaten Jember sudah ada sistem untuk pelacakan truk tangki, sehingga tidak ada lagi bentuk kecurangan yang terjadi akibat pendistribusian bahan bakar minyak. Masalah yang muncul selanjutnya adalah pada saat perjalanan truk tangki dari depot menuju SPBU yang dituju. Proses pemilihan jalur menuju SPBU masih manual dilakukan oleh Sopir truk tangki tersebut, sehingga masih belum memikirkan efisiensi jalur terpendek yang seharusnya dapat dilalui oleh truk tangki tersebut, maka diperlukan suatu sistem untuk menentukan jalur terpendek dalam pendistribusian bahan bakar minyak dengan menggunakan algoritma yang tepat sehingga pengiriman dapat berjalan tepat waktu.

Salah satu algoritma yang dapat digunakan untuk pemilihan jalur terpendek dalam pendistribusian bahan bakar minyak adalah algoritma Dijkstra, karena dengan algoritma ini pengguna hanya menginputkan titik awal dan titik tujuan, sehingga memudahkan pengguna dalam menentukan jalur.

Aplikasi ini menggunakan algoritma Dijkstra yang berfungsi untuk menunjukkan rute terpendek dalam distribusi BBM yang telah dibandingkan dengan rute yang lain. Rute terpendek yang dihasilkan juga telah direpresentasikan dalam graf yang dipengaruhi oleh beberapa hal, yaitu titik awal pengiriman, titik tujuan pengiriman, *cost*, dan reverse cost setiap jalan yang akan dilewati.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Rancang Bangun Aplikasi Pendistribusian Bahan Bakar Minyak Pada SPBU Pertamina Wilayah Jember Menggunakan Algoritma Dijkstra Berbasis Web”. 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. Allah S.W.T. dan Rosullullah S.A.W dan Para Sahabat.
2. Ayahanda Hari Santoso dan Ibunda Sri Hayuningsih yang selalu mendoakan dan mendukung;
3. Prof. Drs. Slamain, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
4. Prof. Drs. Slamain, M.Comp.Sc., Ph.D., selaku Dosen Pembimbing Utama dan Windi Eka Yulia Retnani S.Kom., MT., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
5. Prof. Drs. Slamain, M.Comp.Sc., Ph.D., sebagai dosen pembimbing akademik, yang telah mendampingi penulis sebagai mahasiswa;
6. seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
7. Alm. H. Soewarso beserta keluarga besar;
8. teman-teman seperjuanganku FORMATION angkatan 2012 dan semua mahasiswa Program Studi Sistem Informasi yang telah menjadi keluarga kecil bagi penulis selama menempuh pendidikan S1;
9. keluarga besar Himpunan Mahasiswa Sistem Informasi (HIMASIF) periode 2014-2015 yang tidak dapat disebutkan satu persatu;

10. keluarga besar asisten laboratorium Rekayasa Perangkat Lunak (RPL) tahun 2015-2016 yang tidak dapat disebutkan satu-persatu;
11. sahabat-sahabatku Next Planz yang selalu mendukung dan mendoakan;
12. Rizki Vadilla, Mohammad Ainul Yaqin, dan Muhammad Fariz Dwi Fadlilah yang telah banyak membantu dalam skripsi ini;
13. Savira Oktari dan Safitri Febryanti Astuti adik-adik angkatanku yang selalu memberi semangat dan senyuman;
14. untuk kamu yang sedang menunggu untuk dihalalkan;
15. semua pihak yang tidak dapat disebutkan satu-persatu.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu penulis mengharapkan adanya masukan yang bersifat membangun dari semua pihak. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember, Oktober 2016

Penulis

DAFTAR ISI

SKRIPSI.....	i
SKRIPSI.....	i
PERSEMBAHAN.....	ii
MOTO.....	iii
PERNYATAAN.....	iv
SKRIPSI.....	v
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN.....	vii
RINGKASAN.....	viii
PRAKATA.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xvi
DAFTAR TABEL.....	xviii
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	Distribusi	6
2.3	Bahan Bakar Minyak	6
2.4	Stasiun Pengisian Bahan Bakar Umum	6
2.6	Algoritma Dijkstra	8
2.7	Model <i>Waterfall</i>	13
BAB 3. METODOLOGI PENELITIAN		14
3.1	Tahapan Penelitian	14
3.1.1	Tahapan Analisis Kebutuhan	14
3.1.2	Tahapan Desain Sistem	18
3.1.3	Tahapan Implementasi Sistem	19
3.1.4	Tahapan Pengujian Sistem	20
3.1.5	Tahapan Pemeliharaan Sistem	21
BAB 4. PENGEMBANGAN SISTEM		22
4.1	Analisis Kebutuhan Sistem	22
4.1.2	Kebutuhan Non-Fungsional	22
4.2	Desain Sistem	23
4.2.1	<i>Business Process</i>	23
4.2.2	<i>Use Case Diagram</i>	24
4.2.3	<i>Use Case Scenario</i>	26
4.2.4	<i>Sequence Diagram</i>	28
4.2.5	<i>Activity Diagram</i>	30
4.2.6	<i>Class Diagram</i>	31
4.2.7	<i>Entity Relationship Diagram (ERD)</i>	35

4.3	Penulisan Kode Program	36
4.3.1	Kode Program Pengelolaan Data SPBU	36
4.3.2	Kode Program Melihat Data Distribusi BBM.....	36
4.3.3	Kode Program Pengelolaan Data Akun	36
4.3.4	Kode Program Pengelolaan Data Distribusi	36
4.3.5	Kode Program Melihat SPBU	41
4.3.6	Kode Program Melihat Peta	41
4.3.7	Kode Program Verifikasi Data Distribusi BBM	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.....		48
5.1	Hasil Penelitian.....	48
5.1.1	Penentuan Jalur Terpendek Distribusi BBM.....	48
5.1.2	Perhitungan Metode Dijkstra dalam Pendistribusian BBM.....	50
5.2	Hasil Pembangunan Aplikasi	53
5.2.1	Halaman <i>Login</i>	53
5.2.2	Halaman Dashboard.....	54
5.2.3	Halaman Manajemen SPBU	55
5.2.4	Halaman Distribusi BBM.....	56
5.2.5	Halaman Akun	56
5.2.6	Halaman Tambah Distribusi	57
5.2.7	Halaman Data SPBU.....	58

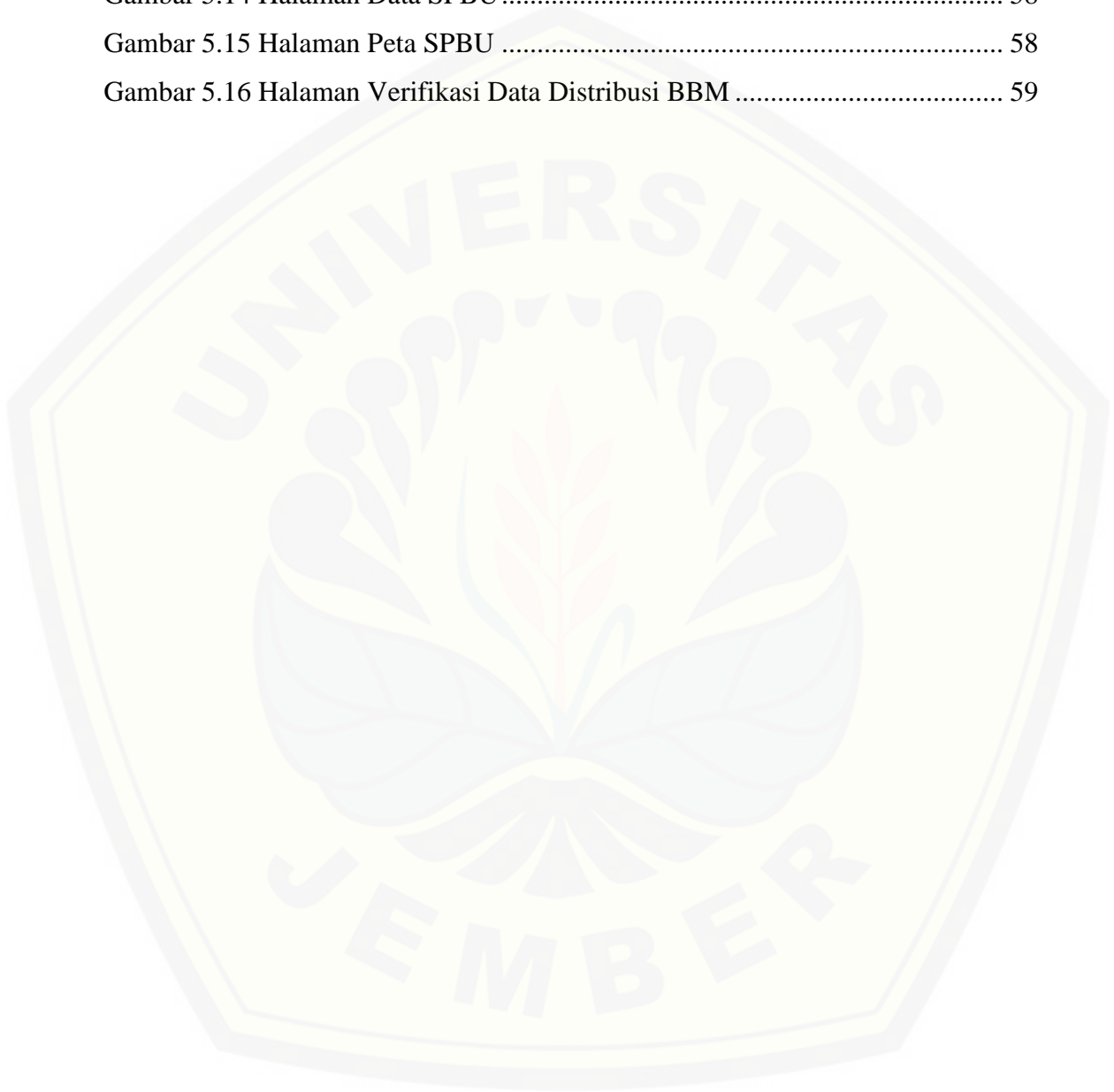
5.2.8 Halaman Peta SPBU	58
5.2.9 Halaman Verifikasi Data Distribusi BBM	59
5.3 Pembahasan	59
5.3.1 Analisis Algoritma Dijkstra	59
5.3.2 Analisis Model <i>Waterfall</i>	59
BAB 6. PENUTUP	61
6.1 Kesimpulan	61
6.2 Saran	62
DAFTAR PUSTAKA	63
LAMPIRAN	65
A. <i>Use Case Scenario</i>	65
A.1 Skenario <i>Use Case</i> Pengelolaan Data SPBU	65
A.2 Skenario <i>Use Case</i> Melihat Data Distribusi BBM	68
A.3 Skenario <i>Use Case</i> Pengelolaan Akun	69
A.4 Skenario <i>Use Case</i> Melihat Data SPBU	72
A.5 Skenario <i>Use Case</i> Melihat Peta	73
A.6 Skenario <i>Use Case</i> Verifikasi Data Distribusi BBM	73
B. <i>Sequence Diagram</i>	74
B.1. <i>Sequence Diagram</i> Pengelolaan Data SPBU	74
B.2 <i>Sequence Diagram</i> Pengelolaan Data Akun	75
B.3 <i>Sequence Diagram</i> Melihat Distribusi BBM	76
B.4 <i>Sequence Diagram</i> Melihat Data SPBU	76
B.5 <i>Sequence Diagram</i> Melihat Peta	76

B.6	<i>Sequence Diagram</i> Verifikasi Data Distribusi BBM	77
C.	<i>Activity Diagram</i>	78
C.1	<i>Activity Diagram</i> Pengelolaan Data SPBU	78
C.2	<i>Activity Diagram</i> Melihat Data Distribusi.....	79
C.3	<i>Activity Diagram</i> Pengelolaan Data Akun	80
C.4	<i>Activity Diagram</i> Melihat SPBU	81
C.5	<i>Activity Diagram</i> Melihat Peta	81
C.6	<i>Activity Diagram</i> Verifikasi Data Distribusi BBM	82
D.	Kode Program.....	83
D.1	Kode Program Pengelolaan Data SPBU	83
D.2	Kode Program Melihat Data Distribusi BBM.....	96
D.3	Kode Program Pengelolaan Data Akun	100
D.4	Kode Program Melihat SPBU	112
D.5	Kode Program Melihat Peta	126
E.	Pengujian <i>White Box</i>	137
E.1	Kelas Controller.....	137
E.2	Kelas Model.....	140
F.	Pengujian <i>Black Box</i>	141
F.1	Fitur Pengelolaan Data Akun.....	141
F.2	Fitur Pengelolaan Data SPBU	142
F.3	Fitur Melihat Data Distribusi BBM	144
F.4	Fitur Melihat Data SPBU.....	144
F.5	Fitur Melihat Peta	144

DAFTAR GAMBAR

Gambar 2. 1 Diagram Alir Algoritma Dijkstra	10
Gambar 2. 2 Contoh Graf penentuan jalur terpendek	11
Gambar 2. 3 Model Waterfall (Sommerville, 2003)	13
Gambar 3. 1 Diagram tahap analisis data.....	16
Gambar 3. 2 Diagram alir penentuan jalur terpendek	17
Gambar 4. 1 Business Process	23
Gambar 4. 2 Use Case Diagram	24
Gambar 4. 3 Sequence Diagram Pengelolaan Distribusi BBM	29
Gambar 4. 4 Activity Diagram Pengelolaan Distribusi BBM.....	31
Gambar 4. 5 Classs Diagram.....	32
Gambar 4. 6 Entity Relation Diagram	35
Gambar 4. 7 Diagram Alir Function _construct.....	42
Gambar 4. 8 Diagram Alir Function cobaakun()	43
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 Penentuan Rute Terpendek.....	49
Gambar 5. 2 Kode Program Fungsi pgr_dijkstra	51
Gambar 5. 3 Representasi Graf Sistem	51
Gambar 5. 4 Marker SPBU	52
Gambar 5. 5 Hasil Pencarian Jalur Terpendek.....	53
Gambar 5.6 Halaman Utama Aplikasi	54
Gambar 5.7 Halaman Dashboard Admin	54
Gambar 5.8 Halaman Dashboard Sopir	55
Gambar 5.9 Halaman Dashboard SPBU	55
Gambar 5.10 Halaman Manajemen SPBU.....	56
Gambar 5.11 Halaman Distribusi.....	56

Gambar 5.12 Halaman Akun.....	57
Gambar 5.13 Halaman Tambah Distribusi.....	57
Gambar 5.14 Halaman Data SPBU.....	58
Gambar 5.15 Halaman Peta SPBU	58
Gambar 5.16 Halaman Verifikasi Data Distribusi BBM	59



DAFTAR TABEL

Tabel 2. 1 Penyelesaian Graf dengan menggunakan Algoritma Dijkstra	11
Tabel 4. 1 Definisi Aktor.....	25
Tabel 4. 2 Definisi Use Case.....	25
Tabel 4. 3 Skenario Use Case Pengelolaan Distribusi BBM (Sopir).....	27
Tabel 4. 4 Kode Program Controller Distribusi	36
Tabel 4. 5 Kode Program Kelas Controller Route	39
Tabel 4. 6 Kode Program Kelas Distribusi Model.....	40
Tabel 4. 7 Kode Program Kelas Route Model	41
Tabel 4. 9 Test Case Function _construct()	42
Tabel 4. 10 Test Case Function cobaakun().....	43
Tabel 4. 11 Test Case Function profil()	44
Tabel 4. 12 Test Case Function detail(\$id_user)	45
Tabel 4. 13 Test Case Function add()	46
Tabel 4. 14 Pengujian Black Box Pengelolaan Data Distribusi BBM (Sopir)	47
Tabel 5. 1 Relasi Antara Tabel Jalan dan Tabel Jenis Kendaraan	49
Tabel 5. 2 Tabel Jalan	50
Tabel 5. 3 Perhitungan Representasi Graf dari Algoritma Dijkstra	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

Kabupaten Jember sebagai salah satu kabupaten yang memiliki tingkat perkembangan infrastruktur yang pesat. Penyebab perkembangan ini erat kaitannya dengan mobilitas masyarakat Kabupaten Jember yang semakin berkembang pesat. Dengan perkembangan mobilitas masyarakat Kabupaten Jember saat ini, maka peran Stasiun Pengisian Bahan bakar Umum (SPBU) sangat dibutuhkan untuk memenuhi pasokan bahan bakar minyak masyarakat Jember tiap harinya. Pendistribusian bahan bakar minyak harus dilakukan secara cepat dan tepat agar tidak terjadi penurunan kinerja masyarakat Jember akibat keterlambatan pendistribusian bahan bakar minyak.

Proses pendistribusian bahan bakar minyak saat ini di wilayah Kabupaten Jember sudah ada sistem untuk pelacakan truk tangki, sehingga tidak ada lagi bentuk kecurangan yang terjadi akibat pendistribusian bahan bakar minyak. Masalah yang muncul selanjutnya adalah pada saat perjalanan truk tangki dari depot menuju SPBU yang dituju. Proses pemilihan jalur menuju SPBU masih manual dilakukan oleh sopir truk tangki tersebut, sehingga masih belum memikirkan efisiensi jalur terpendek yang seharusnya dapat dilalui oleh truk tangki tersebut. Oleh karena itu, sangat diperlukan suatu sistem untuk menentukan jalur terpendek dalam pendistribusian bahan bakar minyak dengan menggunakan algoritma yang tepat sehingga pengiriman dapat berjalan tepat waktu.

Salah satu algoritma yang dapat digunakan untuk pemilihan jalur terpendek dalam pendistribusian bahan bakar minyak adalah algoritma Dijkstra. Karena dengan algoritma ini pengguna hanya menginputkan titik awal dan titik tujuan (Arifianto, 2012), sehingga memudahkan pengguna dalam menentukan jalur.

Penelitian sebelumnya yang pernah dilakukan dengan mengimplementasikan Algoritma Dijkstra yaitu dengan judul “Implementasi Algoritma Dijkstra Dalam Aplikasi Untuk Menentukan Lintasan Terpendek Jalan Darat Antar Kota Di Sumatera Bagian Selatan” yang dilakukan oleh Fitria dan Apri Triansyah dari Fakultas Ilmu Komputer Universitas Sriwijaya pada tahun 2013. Pada penelitian ini dijelaskan bagaimana menentukan lintasan terpendek antar kota karena terdapat jalan yang bercabang-cabang dan penerapan algoritma Dijkstra dalam kalkulasi kemungkinan bobot terkecil dari setiap titik. Hasil dari penelitian ini adalah algoritma Dijkstra dapat digunakan untuk mencari rute terpendek secara optimal karena dapat mempercepat dalam menentukan rute terpendek dan penyusunan peta yang dinamik (Fitria & Apri Triansyah, 2013).

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).

Berdasarkan penelitian-penelitian tersebut dapat disimpulkan bahwa penggunaan algoritma Dijkstra sangat mendukung dalam implementasi program yang

akan saya buat karena berdasarkan peraturan dari Pertamina sendiri untuk pendistribusian BBM hanya untuk satu SPBU saja dan tidak dapat satu pengiriman BBM untuk beberapa SPBU. Algoritma Dijkstra sangat mendukung untuk pencarian jalur terpendek dengan satu tujuan tempat. .

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 pendistribusian bahan bakar minyak terpendek pada SPBU Pertamina wilayah Jember menggunakan algoritma Dijkstra ?
2. Bagaimana merancang aplikasi pendistribusian bahan bakar minyak pada SPBU Pertamina wilayah Jember menggunakan algoritma dijkstra berbasis web ?

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 dala rute pendistribusian bahan bakar minyak pada SPBU Pertamina wilayah Kabupaten Jember.
2. Merancang aplikasi pendistribusian bahan bakar minyak pada SPBU Pertamina wilayah Jember menggunakan algoritma Dijkstra berbasis web.

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 jalur distribusi menuju stasiun pengisian bahan bakar umum wilayah Kabupaten Jember.

2. Terbatas hanya untuk penentuan jalur terpendek pendistribusian bahan bakar minyak pada SPBU wilayah Kabupaten Jember.
3. Jenis truk tangki yang digunakan sesuai dengan aturan dari pihak Pertamina dan BBM yang akan didistribusikan berdasarkan pesanan SPBU.
4. Pemilihan jalur terdekat dengan menghitung cost dan reverse cost yang ada pada setiap jalan.

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

Penelitian sebelumnya yang pernah dilakukan dengan mengimplementasikan algoritma dijkstra yaitu dengan judul “Implementasi Algoritma Dijkstra Dalam Aplikasi Untuk Menentukan Lintasan Terpendek Jalan Darat Antar Kota Di Sumatera Bagian Selatan” yang dilakukan oleh Fitria dan Apri Triansyah dari Fakultas Ilmu Komputer Universitas Sriwijaya pada tahun 2013. Pada penelitian ini dijelaskan bagaimana menentukan lintasan terpendek antar kota karena terdapat jalan yang bercabang-cabang dan penerapan algoritma Dijkstra dalam kalkulasi kemungkinan bobot terkecil dari setiap titik. Hasil dari penelitian ini adalah algoritma Dijkstra dapat digunakan untuk mencari rute terpendek secara optimal karena dapat mempercepat dalam menentukan rute terpendek dan penyusunan peta yang dinamik (Fitria & Apri Triansyah, 2013).

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 Distribusi

Pengertian distribusi menurut (Sembiring, 1991) “Distribusi adalah penyaluran barang dari suatu tempat ketempat lainnya atau dari produsen ke konsumen untuk dimanfaatkan”.

2.3 Bahan Bakar Minyak

Bahan bakar merupakan salah satu kebutuhan manusia yang sangat penting. Menurut (Muin, 1998) pengertian dari bahan bakar yaitu segala sesuatu yang dapat dibakar. Bahan-bahan yang dapat dibakar misalnya: kertas, kayu, kain, minyak tanah, bensin, gas, batu bara dan lain-lain.

Minyak adalah bahan bakar lain yang ditemukan dalam batuan bumi. Minyak terbuat dari makhluk laut kecil yang hidup jutaan tahun lampau. Ketika makhluk hidup itu mati, dia terbenam di dasar laut dan terkubur dalam pasir dan lumpur. Jutaan tahun kemudian, sisa-sisa makhluk hidup tersebut berubah menjadi tetes-tetes minyak. Bahan bakar minyak atau BBM adalah salah satu jenis bahan bakar. Menurut (Muin, 1998) jenis bahan bakar ada tiga, yaitu:

1. Bahan bakar padat (solid fuel), contoh: kayu, batu bara, kokas dan lain-lain.
2. Bahan bakar cair (liquid fuel), contoh: bensin, minyak tanah dan minyak bakar.
3. Bahan bakar gas (gasues fuel), contoh: gas methan, gas LNG, gas LPG dan lain-lain.

2.4 Stasiun Pengisian Bahan Bakar Umum

Stasiun Pengisian Bahan Bakar untuk Umum merupakan prasarana umum yang disediakan oleh PT. Pertamina untuk masyarakat luas guna memenuhi kebutuhan

bahan bakar. Pada umumnya SPBU menjual bahan bakar sejenis premium, solar, pertamax dan pertamax plus (Pertamina, 2014).

Stasiun pengisian bahan bakar umum sendiri dibedakan menjadi 3 jenis usaha, yaitu:

a. *Company Operation Company Owner (COCO)*

Merupakan SPBU yang di miliki dan di kelola oleh Pertamina. Dalam hal ini yang mengelola adalah PT. Pertamina Retail sebagai anak perusahaan. Saat ini sudah banyak tersebar SPBU coco di Indonesia.

b. *Dealer Operation Dealer Owner (DODO)*

Merupakan SPBU murni milik swasta atau perorangan. Jadi segala hal mengenai manajemen perusahaan di kelola oleh perorangan atau badan usaha.

c. *Company Operation Dealer Owner (CODO)*

Merupakan SPBU milik swasta atau perorangan yang bekerjasama dengan PT. Pertamina Retail.

Produk-produk yang ada pada stasiun pengisian bahan bakar umum:

- a. Premium adalah bahan bakar minyak jenis distilat berwarna kekuningan yang jernih. Warna kuning tersebut akibat adanya zat pewarna tambahan (*dye*). Penggunaan premium pada umumnya adalah untuk bahan bakar kendaraan bermotor bermesin bensin, seperti mobil, sepeda motor, motor tempel dan lain-lain. Bahan bakar ini sering juga disebut motor *gasoline* atau *petrol*.
- b. Pertalite dapat dikategorikan sebagai bahan bakar kendaraan yang memenuhi syarat dasar durability atau ketahanan, dimana bahan bakar kendaraan ini tidak akan menimbulkan gangguan serta kerusakan mesin, karena kandungan oktan 90 lebih sesuai dengan perbandingan kompresi kebanyakan kendaraan bermotor yang beredar di Indonesia.
- c. Pertamax adalah motor *gasoline* tanpa timbal dengan kandungan aditif lengkap generasi mutakhir yang akan Sopirihkan *Intake Valve Port Fuel*

Injector dan ruang bakar dari *carbon* deposit dan mempunyai *Research Octane Number* (RON) 92. Pertamax merupakan bahan bakar ramah lingkungan (*unleaded*) dan beroktan tinggi.

- d. Pertamax Plus adalah bahan bakar superior Perusahaan Publik dengan kandungan energi tinggi dan ramah lingkungan, diproduksi menggunakan bahan baku pilihan berkualitas tinggi sebagai hasil penyempurnaan formula terhadap produk Perusahaan Publik sebelumnya. Produk ini ditujukan untuk kendaraan yang berteknologi mutakhir yang mempersyaratkan penggunaan bahan bakar beroktan tinggi dan ramah lingkungan.
- e. Pertamina DEX merupakan bahan bakar mesin *diesel modern* yang telah memenuhi dan mencapai standar emisi gas buang EURO 2, memiliki angka performa tinggi dengan *cetane number* 53 ke atas (HSD mempunyai *cetane number* 45), memiliki kualitas tinggi dengan kandungan sulfur di bawah 300 ppm, direkomendasikan untuk mesin diesel teknologi terbaru (*Diesel Common Rail System*), sehingga pemakaian bahan bakar akan lebih irit dan ekonomis serta menghasilkan tenaga yang lebih besar.
- f. Bio Solar adalah bahan bakar campuran untuk mesin diesel yang terdiri dari minyak hayati non fosil (*bio fuel*) – sebesar lima persen minyak kelapa sawit atau *Crude Palm Oil* (CPO) yang telah dibentuk menjadi *Fatty Acid Methyl Ester* (FAME) dan 95 persen solar murni bersubsidi. Bahan bakar ini secara bertahap akan mengurangi peran solar.

2.6 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

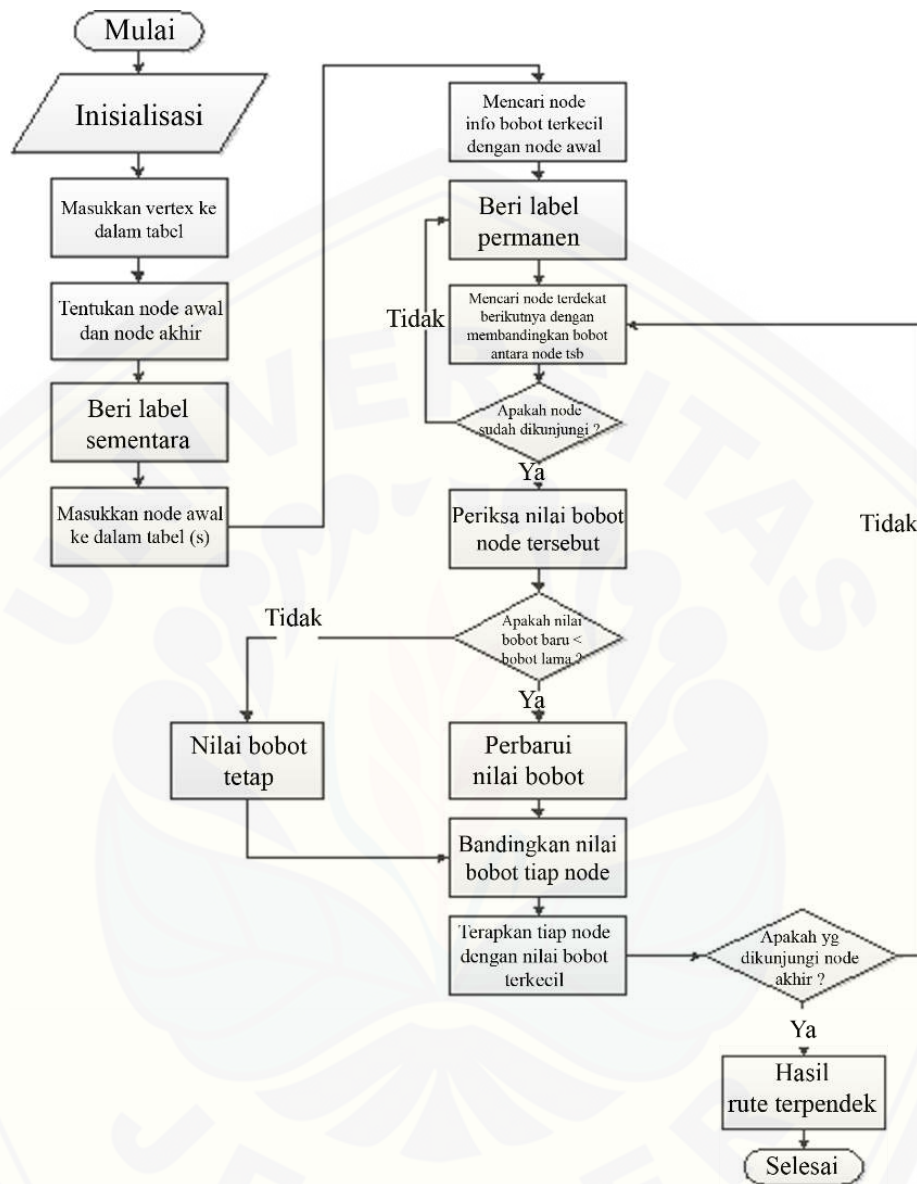
terpendek. Algoritma ini sering digunakan pada *routing*. Algoritma Dijkstra mencari lintasan terpendek dalam sejumlah langkah (Fakhri, 2008).

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).

Penggunaan algoritma Dijkstra sangat cocok dengan alur kerja dari proses pendistribusian BBM oleh Pertamina karena berdasarkan peraturan dari Pertamina sendiri untuk pendistribusian BBM hanya untuk satu SPBU saja dan tidak dapat satu pengiriman BBM untuk beberapa SPBU. Algoritma Dijkstra sangat mendukung untuk pencarian jalur terpendek dengan satu tujuan tempat.

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 2.1



Gambar 2. 1 Diagram Alir Algoritma Dijkstra
(Sumber: (Gusmao & dkk, 2013))

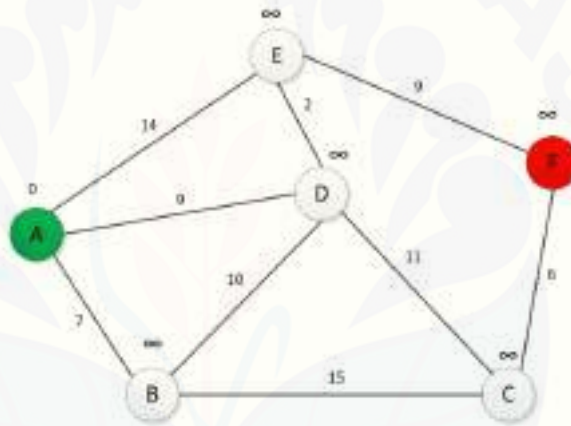
Sebuah graf berarah yang berbobot (*weighted directed graph*) G dan sumber *vertices* dalam G dan V merupakan himpunan semua *vertices* dalam *graph* G , dalam hal ini graf berarah yang berbobot dan sumber *vertices* dapat disimpulkan sebagai inputan dari algoritma Dijkstra. Setiap sisi dari graf ini 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.

$$W: E \rightarrow [0, \infty] \dots\dots\dots 1$$

Contoh bentuk pengimplementasian algoritma Dijkstra pada graf untuk menentukan jalur terpendek dari *node* A ke F pada Gambar 2.2 dan penyelesaiannya pada tabel 2.1.



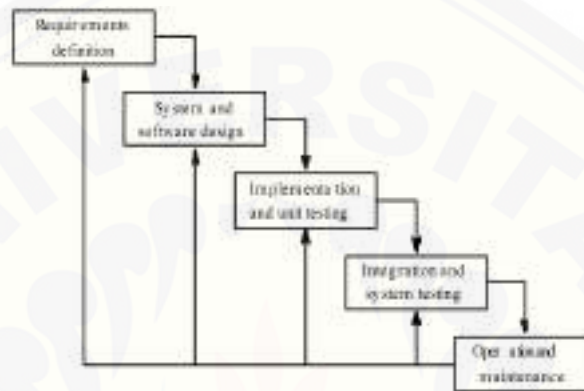
Gambar 2. 2 Contoh Graf penentuan jalur terpendek (Sumber: (Roqib, 2014)

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	$E \rightarrow F$ $11 + 9 = 20$	

2.7 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 2.3.



Gambar 2.3 Model *Waterfall* (Sommerville, 2003)

Berdasarkan Gambar 2.3 dapat dilihat bahwa dalam penggunaan model *Waterfall* dilakukan secara bertahap dan apabila ada kesalahan maka harus kembali ke tahap sesuai pada Gambar 2.3. 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 coding 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 Tahapan Penelitian

Penelitian mengenai rancang bangun aplikasi pendistribusian bahan bakar minyak pada SPBU Pertamina wilayah Jember menggunakan algoritma dijkstra berbasis web ini menggunakan *System Development Life Cycle (SDLC) waterfall* seperti yang telah dijelaskan pada bab tinjauan pustaka subbab 2.7. Pengembangan aplikasi ini menggunakan model *waterfall* karena disesuaikan dengan kebutuhan aplikasi. Aplikasi yang akan dibangun merupakan aplikasi dalam skala yang kecil, Sumber Daya Manusia (SDM) yang membangun aplikasi hanya 1 orang, dan menyesuaikan dengan *user* yang akan menjadi pengguna aplikasi.

3.1.1 Tahapan Analisis Kebutuhan

Penelitian yang dilakukan pada kantor Pertamina TBBM Tanjung Wangi yang berlokasi di jalan Gatot Subroto No. 72 Banyuwangi dan beberapa SPBU di kota Jember. Penelitian yang dilaksanakan dalam waktu empat bulan dimulai pada bulan Oktober 2015 sampai dengan bulan Januari 2016. 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 Pertamina, 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 pendistribusian bahan bakar minyak. Para narasumber yang dimaksud, yaitu staff pendistribusian BBM, sopir truk tangki Pertamina, dan manajer SPBU Kota Jember. Dari wawancara yang telah dilakukan diketahui bahwa jadwal pendistribusian BBM dijadwalkan dari malam sampai pagi dan alur kerja dari pendistribusian BBM tersebut.

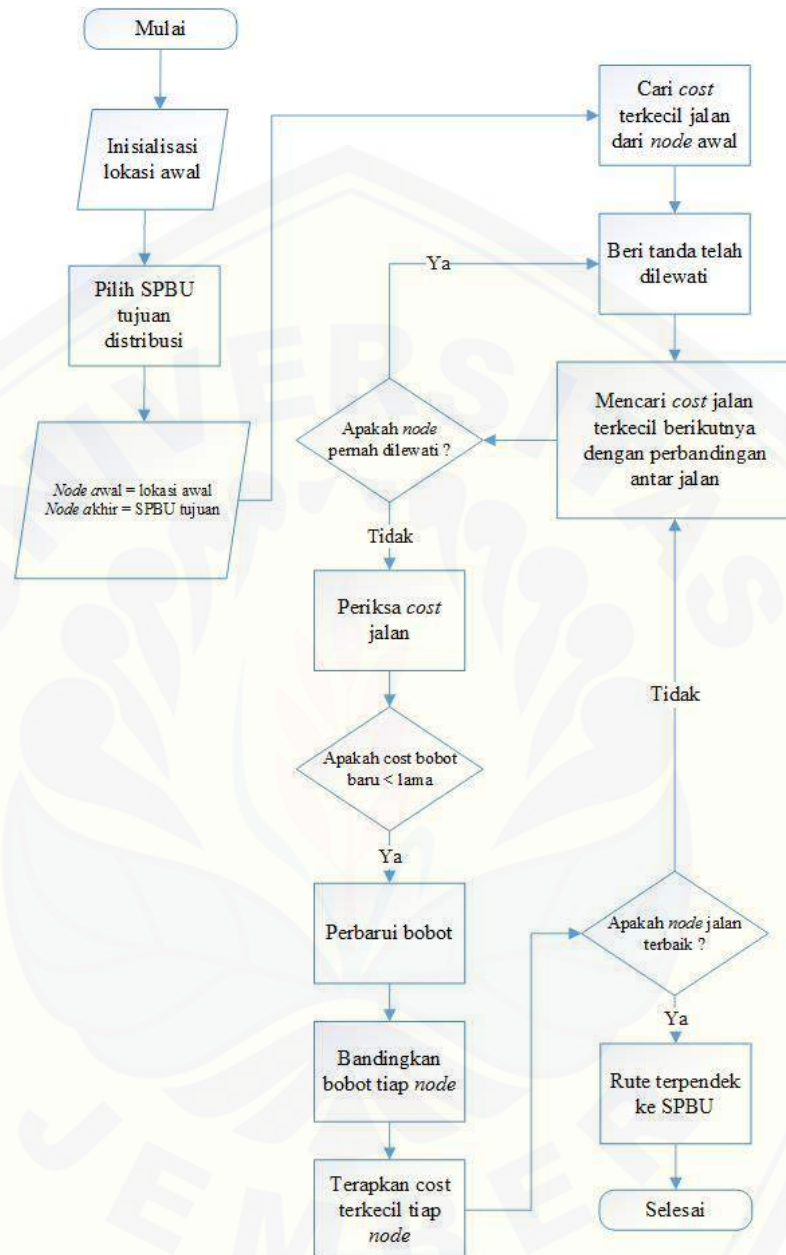
2. Tahap Analisis Data

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 SPBU dan status data jalan yang ada di Kabupaten Jember. Data *input* untuk algoritma dijkstra ada dua macam, yaitu lokasi titik awal dan lokasi SPBU tujuan. Output dari proses algoritma Dijkstra ini yaitu jalur terpendek dari titik lokasi awal menuju SPBU yang telah menjadi 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 pendistribusian bahan bakar minyak pada SPBU Pertamina wilayah Jember dengan menggunakan algoritma Dijkstra berbasis web. Aplikasi ini dibangun dengan maksud mempermudah pendistribusian bahan

bakar minyak pada saat truk tangki akan menuju SPBU yang sudah memesan bahan bakar minyak. Dengan aplikasi ini akan menentukan jalur terpendek yang dapat dilalui oleh truk tangki sehingga dapat sampai ke lokasi tepat waktu. 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 SPBU wilayah Jember, dan SPBU yang akan verifikasi data distribusi BBM.

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 distribusi BBM, data SPBU, data jalan, data titik awal posisi, data jenis kendaraan. *Output* yang dihasilkan oleh sistem berupa data jalur terpendek menuju SPBU, data rekap distribusi BBM, 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*, SPBU dan Sopir. Fitur yang disediakan oleh aplikasi yaitu fitur login, pengelolaan data *akun*, pengelolaan data distribusi BBM, pengelolaan data SPBU, verifikasi data distribusi BBM dan rekomendasi jarak terdekat dalam pendistribusian BBM.

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, kontroller, 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 (*coding*) 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 3.1.

$$CC = (E - N) + 2 \quad \dots (3.1)$$

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. *User* disini yaitu *admin*, sopir, dan SPBU.

3.1.5 Tahapan Pemeliharaan Sistem

Sistem yang sudah 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 aplikasi pendistribusian bahan bakar minyak pada SPBU Pertamina wilayah Jember dengan menggunakan algoritma dijkstra berbasis web. 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 akun meliputi (lihat, tambah, ubah, dan hapus).
2. Sistem mampu mengelola data distribusi BBM meliputi (lihat dan tambah).
3. Sistem mampu mengelola data SPBU meliputi (lihat, tambah, ubah, dan hapus).
4. Sistem mampu menampilkan hasil jarak terdekat dalam pendistribusian BBM.
5. Sistem mampu memverifikasi data distribusi BBM.

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.

15.2.1 Business Process

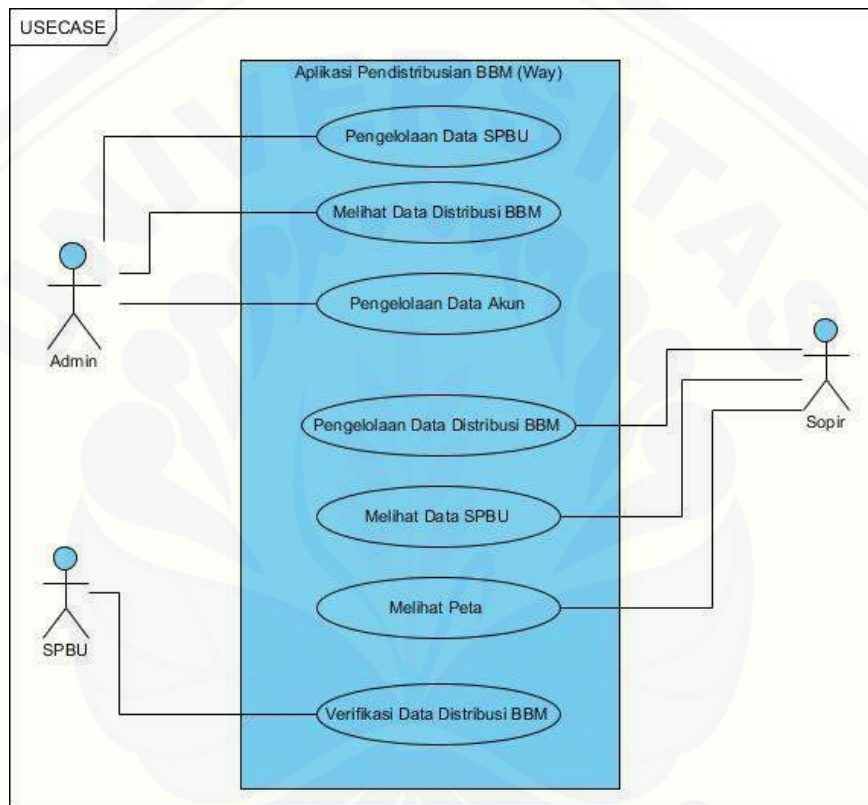
Business Process adalah suatu kumpulan aktivitas yang terstruktur untuk mencapai suatu tujuan tertentu atau untuk menghasilkan sebuah produk. Ada beberapa komponen di dalamnya, meliputi data yang menjadi masukan (*input*), data masukan yang kemudian diolah menjadi data keluaran (*output*), media yang 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 aplikasi pendistribusian bahan bakar minyak pada SPBU Pertamina wilayah 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 aplikasi pendistribusian bahan bakar minyak pada SPBU Pertamina wilayah Jember yang akan dibangun. Terdapat tiga aktor seperti yang dijelaskan pada Tabel 4.1.

Tabel 4. 1 Definsi 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 Sopir, data SPBU, dan data distribusi BBM.
2.	Sopir	Sopir digeneralisasi sebagai responden yang memiliki hak akses untuk mengisi data pendistribusian BBM serta pengelolaan jalur terpendek. Sopir juga memiliki hak akses melihat data distribusi BBM dan data SPBU.
3.	SPBU	SPBU digeneralisasi sebagai aktor yang berada di lokasi tujuan distribusi. Memiliki hak akses untuk melakukan verifikasi pada saat pendistribusian BBM.

2. Definisi *Use Case*

Definisi *use case* merupakan penjelasan dari masing-masing *use case* atau fitur-fitur dari aplikasi pendistribusian bahan bakar minyak pada SPBU Pertamina wilayah 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.	Pengelolaan Data SPBU	Menggambarkan proses menambah, mengedit, menghapus, dan melihat data SPBU.

2.	Melihat Data Distribusi BBM	Menggambarkan proses mengedit, menghapus, dan melihat data distribusi BBM pada pengguna <i>Admin</i> .
3.	Pengelolaan Data Akun	Menggambarkan proses menambah, mengedit, menghapus, dan melihat data pengguna.
4.	Pengelolaan Data Distribusi	Menggambarkan proses menambah data distribusi BBM untuk Sopir.
5.	Melihat SPBU	Menggambarkan proses melihat data SPBU
6.	Melihat Peta	Menggambarkan proses melihat peta yang terdapat marker lokasi-lokasi SPBU.
7.	Verifikasi Data Distribusi BBM	Menggambarkan proses verifikasi data distribusi BBM.

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* Pengelolaan Data SPBU

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

2. Skenario *Use Case* Pengelolaan Data Distribusi

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* pengelolaan data distribusi BBM dapat dilihat pada Tabel

4.3.

Tabel 4. 3 Skenario *Use Case* Pengelolaan Distribusi BBM (Sopir)

Nama Use Case	Pengelolaan Distribusi BBM
Aktor	Sopir
Deskripsi Singkat	Sopir akan memproses data pendistribusian BBM untuk selanjutnya akan diproses dengan menampilkan sebuah peta yang sudah dihubungkan dengan metode Dijkstra
Prekondisi	Data Distribusi BBM
Prakondisi	Data Distribusi BBM berhasil diproses
<i>Flow Events</i>	
Skenario Normal: Rute Distribusi BBM	
Aksi Aktor	Reaksi Sistem
1. Klik Menu Distribusi BBM	
	2. Menampilkan halaman form distribusi BBM meliputi peta spbu, nama spbu, id spbu, latitude, longitude, jenis BBM, volume, tanggal, jam, dan Sopir.
3. Mengisi data distribusi BBM	
4. Klik tombol rute	
	5. Menampilkan jalur pada peta yang digunakan dalam perjalanan BBM
Skenario Normal: Tambah Distribusi BBM	
Aksi Aktor	Reaksi Sistem
4. Klik tombol simpan	
	5. Menyimpan ke database
Skenario Alternatif: Form Data Tidak Lengkap	

Aksi Aktor	Reaksi Sistem
4a. Klik tombol simpan	
	5a. Menampilkan warning “ <i>please fill out this field</i> ”

3. Skenario *Use Case* Pengelolaan Sopir

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* pengelolaan Sopir dapat dilihat pada lampiran A.

4. Skenario *Use Case* Pengelolaan Data Distribusi

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

5. Skenario *Use Case* Melihat SPBU

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

6. Skenario *Use Case* Melihat Peta

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

7. Skenario *Use Case* Verifikasi Data Distribusi BBM

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal, skenario *use case* verifikasi data distribusi BBM 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 aplikasi pendistribusian bahan bakar minyak pada SPBU Pertamina wilayah Jember.

5. *Sequence Diagram* Melihat SPBU

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

6. *Sequence Diagram* Melihat Peta

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

7. *Sequence Diagram* Verifikasi Data Distribusi BBM

Penggambaran *sequence diagram* verifikasi data distribusi BBM 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 aplikasi pendistribusian bahan bakar minyak pada SPBU Pertamina wilayah Jember.

1. *Activity Diagram* Pengelolaan Data SPBU

Activity diagram pengelolaan data SPBU dapat dilihat pada lampiran C.

2. *Activity Diagram* Melihat Data Distribusi BBM

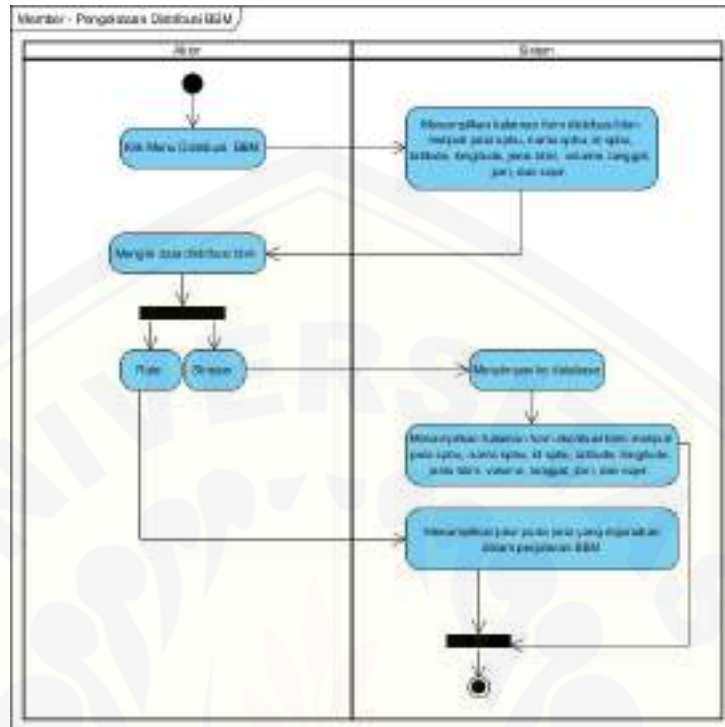
Activity diagram melihat data distribusi BBM dapat dilihat pada lampiran C.

3. *Activity Diagram* Pengelolaan Data Akun

Activity diagram pengelolaan data akun dapat dilihat pada lampiran C.

4. *Activity Diagram* Pengelolaan Distribusi BBM

Activity diagram memasukkan distribusi BBM dapat dilihat pada Gambar 4.4



Gambar 4. 4 *Activity Diagram* Pengelolaan Distribusi BBM

5. *Activity Diagram* Melihat SPBU

Activity diagram melihat SPBU dapat dilihat pada lampiran C.

6. *Activity Diagram* Melihat Peta

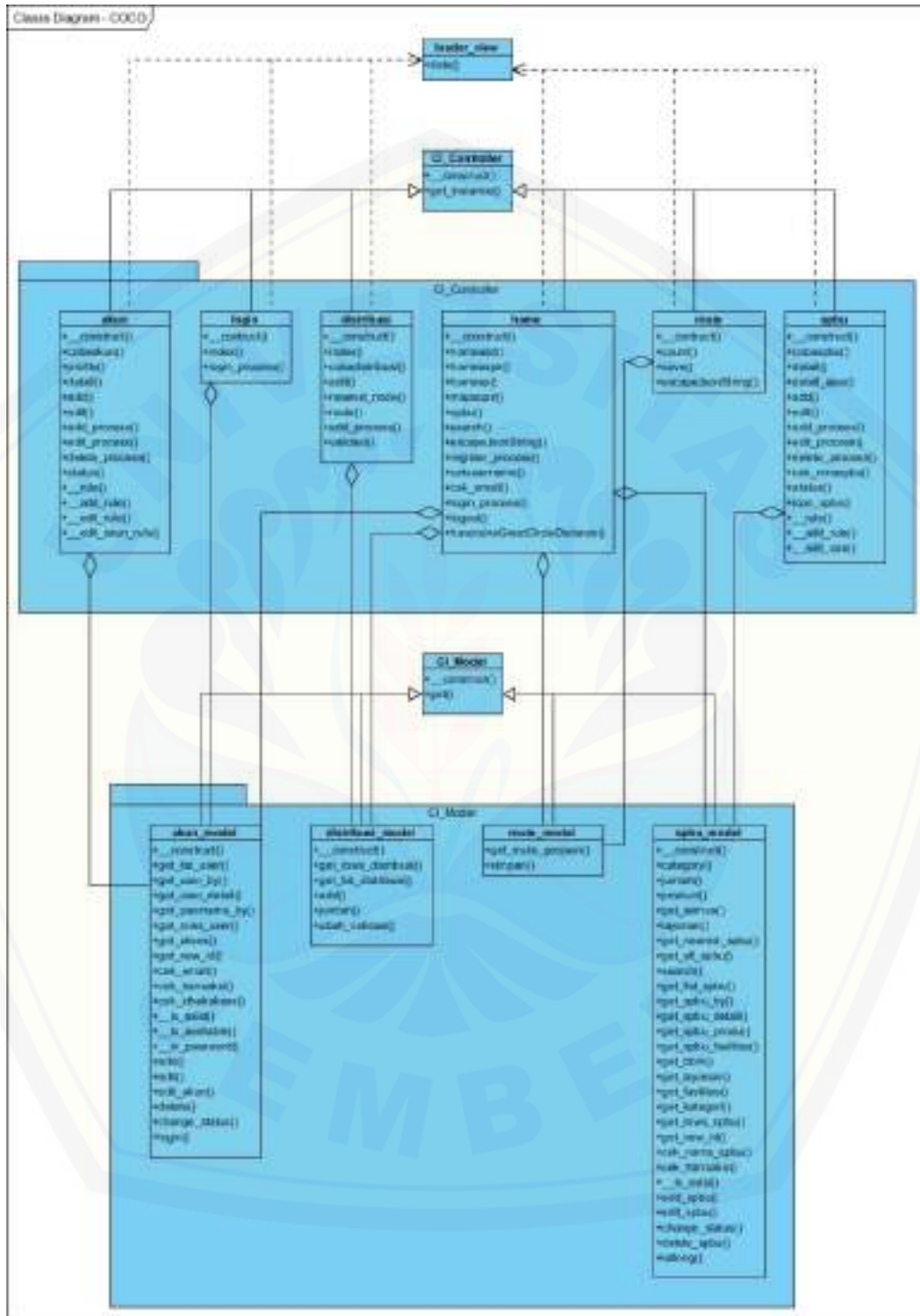
Activity diagram melihat peta dapat dilihat pada lampiran C.

7. *Activity Diagram* Verifikasi Data Distribusi BBM

Activity diagram verifikasi data distribusi BBM dapat dilihat pada lampiran C.

4.2.6 *Class Diagram*

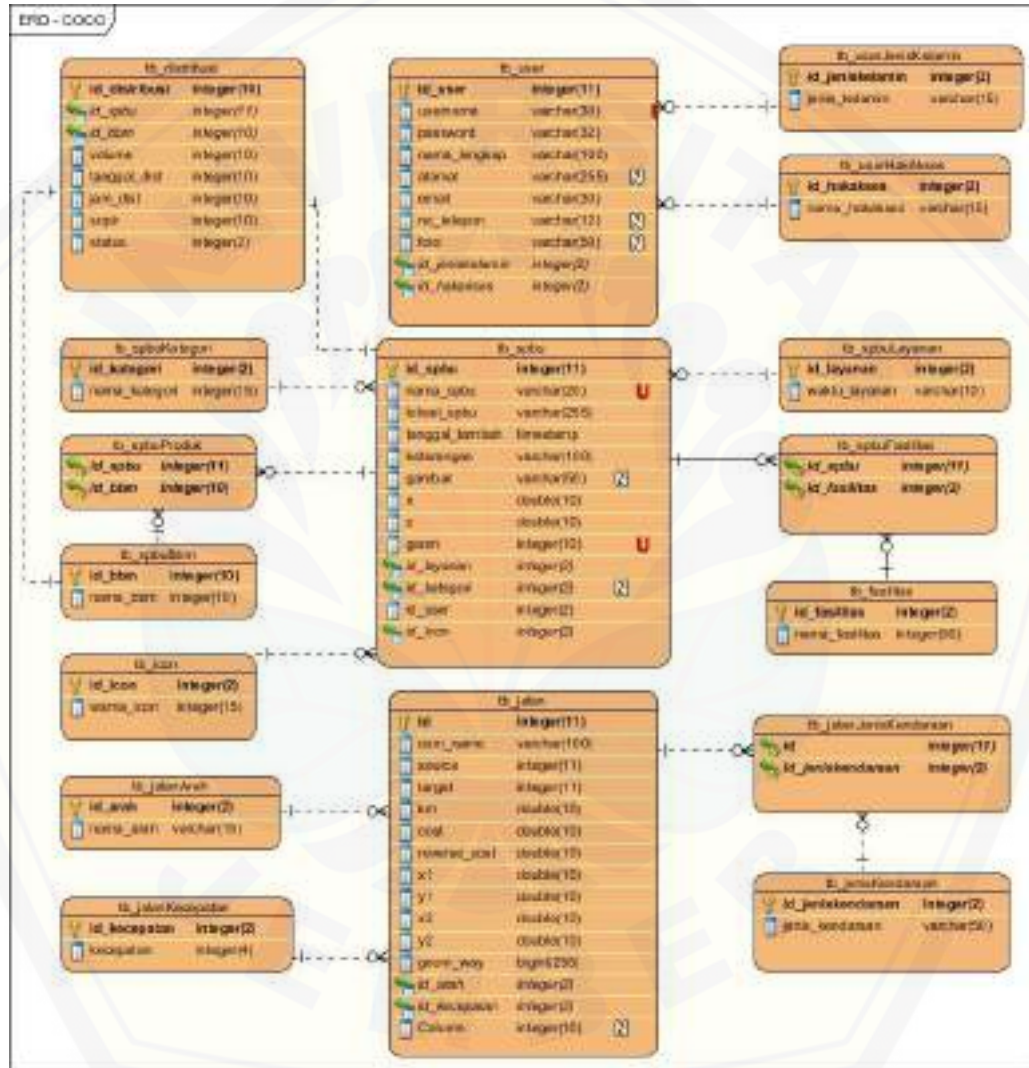
Class Diagram menggambarkan hubungan antarkelas yang digunakan untuk membangun suatu sistem. *Class Diagram* aplikasi pendistribusian bahan bakar minyak dapat dilihat pada Gambar 4.5.



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 aplikasi pendistribusian bahan bakar minyak dapat dilihat pada Gambar 4.6.



Gambar 4. 6 Entity Relation Diagram

4.3 Penulisan Kode Program

Desain yang telah dibuat akan diimplementasikan 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 Pengelolaan Data SPBU

Kode program pengelolaan data SPBU terletak pada kelas *spbu_list*, *spbu_add*, *spbu_edit*, *spbu_detail*, *controller spbu*, dan *spbu_model*. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.2 Kode Program Melihat Data Distribusi BBM

Kode program melihat data distribusi terletak pada kelas *distribusi_list*, *controller distribusi*, dan *distribusi_model*. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.3 Kode Program Pengelolaan Data Akun

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 Pengelolaan Data Distribusi

Kode program pengelolaan data distribusi terletak pada kelas *controller distribusi*, *controller route*, *distribusi_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* Distribusi

1.	<?php
2.	if (!defined('BASEPATH'))
3.	exit('No direct script access allowed');
4.	class Distribusi extends CI_Controller {
5.	private \$data;
6.	function __construct() {
7.	parent::__construct();
8.	\$this->load->model('spbu_model');
9.	\$this->load->model('akun_model');

```
10. $this->load->model('routing_model');
11. $this->load->model('distribusi_model');
12. $this->data = array(
13. 'title' => NULL,
14. 'content' => NULL,
15. 'menu' => NULL,
16. 'akses' => $this->session->userdata('spbu_hakakses'),
17. 'username' => $this->session->userdata('spbu_username'),
18. );
19. }
20. function index() {
21. $this->list_distribusi();
22. }
23. function cobadistribusi(){
24. $this->data['menu'] = 1;
25. $this->data['title'] = 'Manajemen Distribusi';
26. $this->data['content'] = $this->load->view('distribusi_list',array(
27. 'data' => $this->distribusi_model->get_list_distribusi()
28. ),TRUE);
29. $this->load->view('template_baru',$this->data);
30. }
31. function cobaedit(){
32. $this->data['menu'] = 1;
33. $this->data['title'] = 'Manajemen Distribusi';
34. $this->data['content'] = $this->load->view('distribusi_edit',array(
35. 'data' => $this->distribusi_model->get_list_distribusi()
36. ),TRUE);
37. $this->load->view('template_baru',$this->data);
38. }
39. function add(){
40. $this->data['menu'] = 1;
41. $this->data['title'] = 'Tambah Distribusi';
42. $this->data['content'] = $this->load->view('distribusi_add',array(
43. 'produk' => $this->spbu_model->get_BBM(),
44. 'kategori' => $this->spbu_model->get_kategori(),
45. 'fasilitas' => $this->spbu_model->get_fasilitas(),
46. 'layanan' => $this->spbu_model->get_layanan(),
47. 'spbu' => $this->spbu_model->get_semua()
48. ),TRUE);
49. $this->load->view('template_baru',$this->data);
50. }
51. function nearest_node() {
52. $data = array(
53. 'x' => $this->input->post('x'),
54. 'y' => $this->input->post('y')
55. );
56. $result = $this->routing_model->get_nearest_coordinate($data);
57. $result2 = $this->routing_model->get_nearest_network($data);
58. if ($result2['osm_name'] != "") {
59. $nama_jalan = $result2['osm_name'];
60. } else {
61. $nama_jalan = 'No name';
62. }
63. echo $result['lat'] . ',' . $result['lng'];
64. echo '|';
65. echo $nama_jalan;
66. }
```

```
67. function route() {
68.     $data = array(
69.         'x1' => $this->input->post('x1'),
70.         'y1' => $this->input->post('y1'),
71.         'x2' => $this->input->post('x2'),
72.         'y2' => $this->input->post('y2'),
73.         'kendaraan' => $this->input->post('kendaraan'),
74.         'jalur' => $this->input->post('jalur'),
75.         'id' => $this->input->post('id')
76.     );
77.     $rs = $this->routing_model->get_route_geojson($data);
78.     $output = "";
79.     $rowOutput = "";
80.     foreach ($rs as $row) {
81.         $rowOutput = (strlen($rowOutput) > 0 ? ',' : '') . '{"type": "Feature", "geometry": ' .
82.             $row['geojson'] . ', "properties": {';
83.         $props = "";
84.         $id = "";
85.         foreach ($row as $key => $val) {
86.             if ($key != "geojson") {
87.                 $props .= (strlen($props) > 0 ? ',' : '') . '"" . $key . '":' . $this->escapeJsonString($val) .
88.                     ""';
89.             }
90.             if ($key == "id") {
91.                 $id .= ',"id":' . $this->escapeJsonString($val) . ""';
92.             }
93.         }
94.         $rowOutput .= $props . '}' . "\n";
95.         $rowOutput .= $id;
96.         $rowOutput .= '}' . "\n";
97.         $output .= $rowOutput;
98.     }
99.     $output = '{ "type": "FeatureCollection", "features": [ ' . $output . ' ] }';
100.    echo $output;
101. }
102. function add_process(){
103.     $data = array(
104.         'id_spbu' => $this->input->post('id_spbu'),
105.         'id_BBM' => $this->input->post('id_BBM'),
106.         'volume' => $this->input->post('volume'),
107.         'tanggal_dist' => $this->input->post('tanggal_dist'),
108.         'jam_dist' => $this->input->post('jam_dist'),
109.         'Sopir' => $this->input->post('Sopir')
110.     );
111.     $data2 = array(
112.         'lat' => $this->input->post('lat'),
113.         'long' => $this->input->post('long')
114.     );
115.     $hasil = $this->distribusi_model->add($data);
116.     if ($hasil) {
117.         redirect('home/mapsSopir');
118.     } else {
119.         echo 'gagal';
120.     }
121. }
122. function delete_process($id) {
123.     if ($this->data['akses'] != NULL) {
```


124.	\$sukses = \$this->distribusi_model->delete_distribusi(\$id);
125.	if (\$sukses == true) {
126.	\$this->session->set_userdata('message', '1,SPBU telah dihapus.');
127.	redirect('home/cobadistribusi');
128.	}
129.	} else {
130.	redirect('home');
131.	}
132.	}
133.	}

Tabel 4. 5 Kode Program Kelas *Controller* Route

1.	<?php
2.	if (!defined('BASEPATH'))
3.	exit('No direct script access allowed');
4.	class Route extends CI_Controller {
5.	public function count(){
6.	// x1 = longitude gumitir
7.	// y1 = Latitude gumitir
8.	\$this->load->model('route_model');
9.	\$data_map = array(
10.	'x1' => '113.92416817252524',
11.	'y1' => '-8.25025275443009',
12.	'x2' => \$this->input->post('x'),
13.	'y2' => \$this->input->post('y'),
14.	'id' => 0
15.);
16.	\$rs = \$this->route_model->get_route_geojson(\$data_map);
17.	\$output = "";
18.	\$rowOutput = "";
19.	foreach (\$rs as \$row) {
20.	\$rowOutput = (strlen(\$rowOutput) > 0 ? ',' : '') . '{"type": "Feature", "geometry": ' .
21.	\$row['geojson'] . ', "properties": {';
22.	\$props = "";
23.	\$id = "";
24.	foreach (\$row as \$key => \$val) {
25.	if (\$key != "geojson") {
26.	\$props .= (strlen(\$props) > 0 ? ',' : '') . "" . \$key . "":"" . \$this->escapeJsonString(\$val) .
27.	"";
28.	}
29.	if (\$key == "id") {
30.	\$id .= ","id":"" . \$this->escapeJsonString(\$val) . "";
31.	}
32.	}
33.	\$rowOutput .= \$props . '}';
34.	\$rowOutput .= \$id;
35.	\$rowOutput .= '}';
36.	\$output .= \$rowOutput;
37.	}
38.	\$output = '{ "type": "FeatureCollection", "features": [' . \$output . ']}';
39.	echo \$output;
40.	}
41.	public function save()
42.	{

```

43. $this->load->model('route_model');
44. $data_simpan = array(
45. 'id_spbu'      => $this->input->post('id_spbu'),
46. 'id_BBM'      => $this->input->post('id_BBM'),
47. 'volume'      => $this->input->post('volume'),
48. 'tanggal_dist' => $this->input->post('tanggal_dist'),
49. 'jam_dist'    => $this->input->post('jam_dist'),
50. 'Sopir'       => $this->input->post('Sopir')
51. );
52. if($this->route_model->simpan($data_simpan)){
53. echo "Sukses";
54. }else{
55. show_404();
56. }
57. }
58. function escapeJsonString($value) {
59. $escapers = array("\\", "/", "\", "\n", "\r", "\t", "\x08", "\x0c");
60. $replacements = array("\\\\", "\\/", "\\\"", "\\n", "\\r", "\\t", "\\f", "\\b");
61. $result = str_replace($escapers, $replacements, $value);
62. return $result;
63. }
64. }

```

Tabel 4. 6 Kode Program Kelas Distribusi Model

```

1. <?php
2. if (!defined('BASEPATH'))
3. exit('No direct script access allowed');
4. class Distribusi_Model extends CI_Model {
5. function __construct() {
6. parent::__construct();
7. }
8. function get_rows_distribusi() {
9. return $this->db->query("select id_distribusi from tb_distribusi")->num_rows();
10. }
11. function get_list_distribusi() {
12. $data = $this->db->query("select * from
13. tb_distribusi a join tb_spbu b on (a.id_spbu = b.id_spbu) join tb_spbuBBM c
14. on(a.id_BBM=c.id_BBM)
15. order by id_distribusi desc
16. ");
17. return $data->result_array();
18. }
19. function add($data){
20. $data = $this->db->query("INSERT INTO public.tb_distribusi(
21. id_spbu, id_BBM, volume, tanggal_dist, jam_dist,Sopir)
22. VALUES (
23. '" . $data['id_spbu'] . "',
24. '" . $data['id_BBM'] . "',
25. '" . $data['volume'] . "',
26. '" . $data['tanggal_dist'] . "',
27. '" . $data['jam_dist'] . "',
28. '" . $data['Sopir'] . "'");
29. return $data;
30. }

```

31.	function get_distribusi_edit(\$id) {
32.	\$data = \$this->db->query("select *
33.	from tb_distribusi where id_distribusi= \$id");
34.	return \$data->row_array();
35.	}
36.	}
37.	

Tabel 4. 7 Kode Program Kelas Route Model

1.	<?php
2.	if (!defined('BASEPATH'))
3.	exit('No direct script access allowed');
4.	class Route_model extends CI_Model {
5.	function get_route_geojson(\$input) {
6.	\$data = \$this->db->query("SELECT seq, st_asgeojson(geom_way) as geojson
7.	FROM pgr_dijkstra(
8.	'SELECT id, source, target, cost_len as cost, rcost_len as reverse_cost FROM tb_jalan j,
9.	tb_jalanjeniskendaraan jk
10.	where j.id = jk.id_jalan',
11.	(select source from tb_jalan j, tb_jalanjeniskendaraan jk where j.id = jk.id_jalan
12.	order by st_distance(st_makepoint(x1, y1), st_makepoint(\$input[x1], \$input[y1])) limit
13.	1),
14.	(select target from tb_jalan j, tb_jalanjeniskendaraan jk where j.id = jk.id_jalan
15.	order by st_distance(st_makepoint(x2, y2), st_makepoint(\$input[x2], \$input[y2])) limit
16.	1),
17.	false, true
18.) d,
19.	tb_jalan j
20.	WHERE d.id2 = j.id");
21.	return \$data->result_array();
22.	}
23.	function simpan(\$d){
24.	return \$this->db->query("INSERT INTO tb_distribusi(id_spbu, id_BBM, volume,
25.	tanggal_dist, jam_dist, Sopir) VALUES('\$d[id_spbu]', '\$d[id_BBM]', '\$d[volume]',
26.	'\$d[tanggal_dist]', '\$d[jam_dist]', '\$d[Sopir]");
27.	}
28.	}

4.3.5 Kode Program Melihat SPBU

Kode program melihat SPBU terletak pada kelas `spbu_list`, `controller spbu`, dan `spbu_model`. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.6 Kode Program Melihat Peta

Kode program melihat SPBU terletak pada kelas `spbu_map`, `controller home`, dan `spbu_model`. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.7 Kode Program Verifikasi Data Distribusi BBM

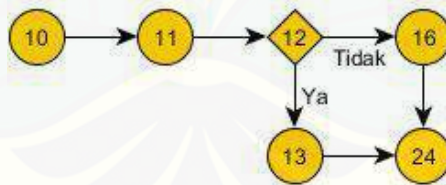
Kode program melihat SPBU terletak pada kelas *distribusi_list*, *controller* distribusi, dan *distribusi_model*. Penulisan kode program ini dapat dilihat pada lampiran D.

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 pendistribusian 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	Jika userdata kosong
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

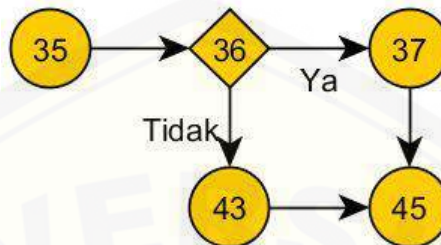


Gambar 4. 8 Diagram Alir *Function* cobaakun()

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

Tabel 4. 9 *Test Case Function* cobaakun()

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

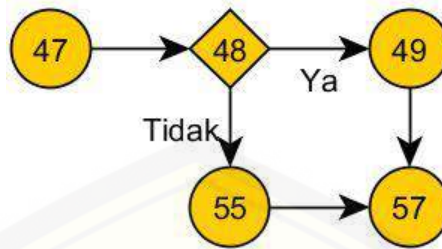


Gambar 4. 9 Diagram Alir *Function* profil()

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

Tabel 4. 10 *Test Case Function* profil()

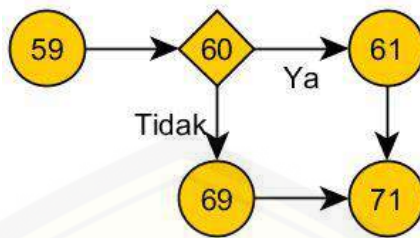
Test case	Jika berhasil memilih tombol detail
Target yang Diharapkan	Menampilkan profil akun yang dipilih
Hasil pengujian	Benar
Path/jalur	35-36-37-45
Test case	Jika tidak berhasil memilih tombol detail
Target yang Diharapkan	Kembali ke halaman home
Hasil pengujian	Benar
Path/jalur	35-36-43-45

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	Jika berhasil memilih tombol detail
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	Jika berhasil memilih tombol tambah data
Target yang Diharapkan	Menampilkan form tambah data akun
Hasil pengujian	Benar
Path/jalur	59-60-61-71
Test case	Jika tidak berhasil memilih tombol tambah data
Target yang Diharapkan	Kembali ke halaman home/cobakun
Hasil pengujian	Benar
Path/jalur	59-60-67-71

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 pengelolaan distribusi BBM dapat dilihat pada Tabel 4.14. Sedangkan untuk pengujian fitur yang lain dapat dilihat pada lampiran F.

Tabel 4. 13 Pengujian *Black Box* Pengelolaan Data Distribusi BBM (Sopir)

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Pengelolaan Distribusi BBM	Memilih menu Distribusi BBM	Menampilkan form tambah data distribusi BBM	Berhasil
		Memilih tombol Rute	Menampilkan jalur terpendek dalam peta.	Berhasil
		Memilih tombol Simpan	Menyimpan data ke <i>database</i>	Berhasil

BAB 5. HASIL DAN PEMBAHASAN

Bab ini menjelaskan tentang hasil pembangunan aplikasi pendistribusian bahan bakar minyak pada spbu Pertamina wilayah Jember menggunakan algoritma Dijkstra berbasis web.

5.1 Hasil Penelitian

Hasil penelitian ini akan membahas mengenai kesesuaian dengan tujuan awal penelitian yaitu penentuan jalur terpendek dalam pendistribusian bahan bakar minyak wilayah Jember. Pembahasan pada penentuan jalur terpendek dengan algoritma Dijkstra.

5.1.1 Penentuan Jalur Terpendek Distribusi BBM

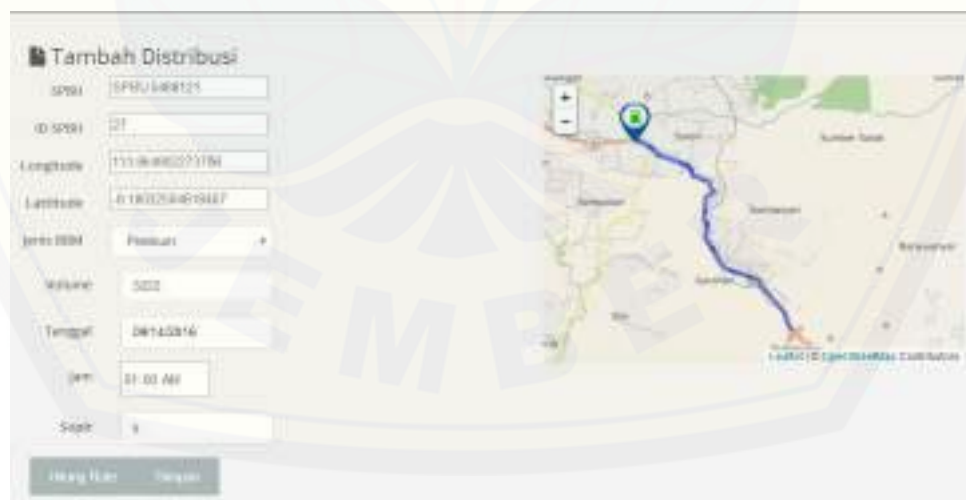
Menentukan jalur terpendek dilakukan dengan penghitungan panjang jalan antara titik awal rute dengan posisi SPBU yang dipilih oleh Sopir. Pemilihan posisi SPBU dilakukan pada saat Sopir memasukkan data distribusi. Sopir hanya memilih marker SPBU lalu menekan tombol rute, maka sistem akan menampilkan rute terpendek yang mungkin untuk dilewati oleh truk tangki Pertamina.

Proses dari penentuan jalur untuk pendistribusian yang telah disesuaikan dengan kondisi jalan dan kendaraan. Dimana kondisi jalan dibagi menjadi tiga jenis jalan yaitu, jalan kecil, jalan sedang, dan jalan besar. Sedangkan dalam proses pendistribusian BBM, kendaraan yang digunakan untuk proses distribusi telah disesuaikan dengan jalan yang telah ditentukan, seperti yang ada pada Tabel 5.1.

Tabel 5. 1 Relasi Antara Tabel Jalan dan Tabel Jenis Kendaraan

	id_jalan integer	id_jeniskendaraan integer
1	1174	3
2	1183	3
3	1184	3
4	5420	3
5	1439	3
6	1508	3
7	1514	3
8	6053	3
9	5665	3
10	5391	3
11	5603	3
12	2148	3
13	1023	3

Gambar 5.1 menjelaskan relasi antara tabel jalan dan tabel jenis kendaraan, dimana dari kedua tabel tersebut dapat dikondisikan kendaraan yang akan melakukan pendistribusian BBM harus melewati jalan yang mana. Sehingga dengan adanya tabel tersebut tidak dimungkinkan terjadinya kendaraan yang digunakan untuk pendistribusian BBM mengalami salah jalur atau jalan yang seharusnya tidak bisa dilewati. Hasil dari proses pendistribusian BBM yang sudah sesuai jalur yang telah ditentukan seperti yang terlihat pada Gambar 5.1.



Gambar 5. 1 Penentuan Rute Terpendek

5.1.2 Perhitungan Metode Dijkstra dalam Pendistribusian BBM

Tahapan-tahapan dalam proses perhitungan dengan menggunakan metode Dijkstra dalam pendistribusian BBM:

- a. Melakukan representasi graf yang dapat digunakan untuk menghitung jalur terpendek menggunakan algoritma Dijkstra. Berikut data jalan yang digunakan dalam pendistribusian BBM, dimana ada atribut *cost* dan *reverse_cost*. Atribut *cost* digunakan sebagai nilai dari panjang jalan, dan *reverse_cost* digunakan sebagai nilai dari panjang jalan dengan nilai terbalik. Berikut tabel jalan yang dapat dilihat pada Tabel 5.2.

Tabel 5. 2 Tabel Jalan

	id [PK] integer	osm_name character varying	source integer	target integer	len double precision	cost double precision	reverse_cost double precision
1	1		3055	1276	0.544731222214498	3.00312757306578548	0.00312757306578548
2	2		66	71	0.581078550712738	0.0048668866625284	0.0048668866625284
3	3		2690	107	0.424203748437291	3.00385403106807765	0.00385403106807765
4	4		5012	3135	0.159466340485195	3.0014420919973674	0.0014420919973674
5	5		56	3077	0.914599842570126	3.00825602244804688	0.00825602244804688
6	6		3057	208	0.023004280801828	0.000208704807331042	0.00208704807331042
7	7		3077	5047	0.803508888805732	3.00728672312831121	0.00728672312831121
8	8		5047	2843	0.376450186437829	0.00341405375888334	0.00341405375888334
9	9		3843	2840	0.148223162780479	3.00185268300827728	0.00185268300827728
10	10		3960	1057	0.283440159655826	3.00368188287566768	0.00368188287566768
11	11		5068	478	0.153115918879558	3.0014346936657828	0.0014346936657828
12	12	Salas Longsep	28	28	0.134160310198641	3.00121011937278483	0.00121011937278483
13	13		1022	1032	0.0487971306817447	3.0003695485907119	0.003695485907119
14	14		1243	1138	0.0383432175472912	3.000346804597539574	0.00346804597539574
15	15		1050	1527	1.07464521748384	3.00874291126251958	0.00874291126251958
16	16		1627	2850	0.246164081874801	3.0033280186287788	0.0033280186287788
17	17		4591	3034	0.426898179083195	3.0038604178186591	0.0038604178186591
18	18		3004	1274	0.426792893884285	3.003950202295276518	0.003950202295276518
19	19		1382	1179	0.053693711190157	3.000459947692812527	0.00459947692812527
20	20		20	21	2.51340137652895	0.0238701840977403	0.0238701840977403
21	21		22	22	0.6244046810932192	3.0056695848575147	0.0056695848575147

- b. Representasi *cost* dan *reverse_cost* dapat dilihat pada fungsi `pgr_dijkstra` yang digunakan untuk melakukan penentuan jalur terpendek pada database terletak pada model *route* yaitu pada *function* `get_route_geojson`. Kode program dapat dilihat pada Gambar 5.2.

```

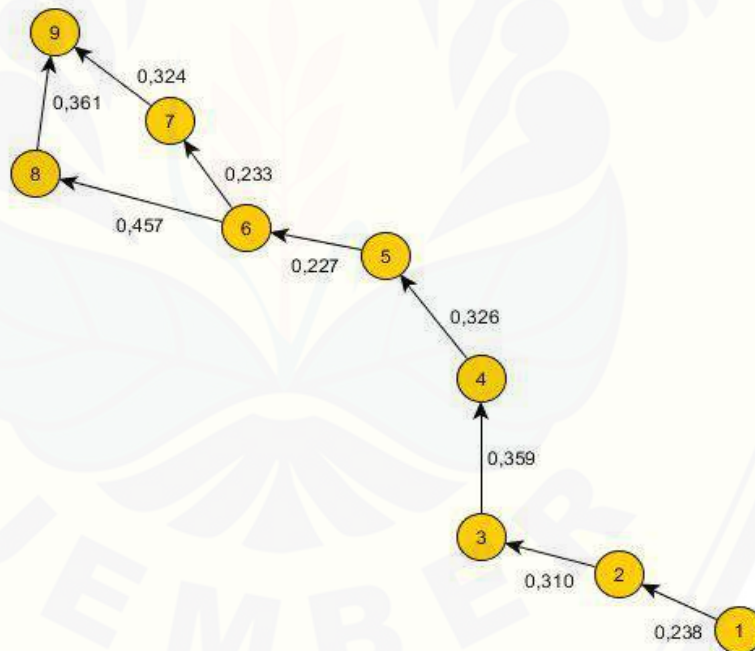
class Route_model extends CJ_Model {

function get_route_geosjon($laput) {
    $data = $this->db->query("SELECT src, st_sgeojion(geom_way) as geojion
FROM pgr_dijkstra(
    'SELECT id, source, target, cost, len as cost, rcost, len as reverse_cost FROM th_jalan j, th_jalanjendikendara
where j.id = jk.id_jalan',
    (select source from th_jalan j, th_jalanjendikendara jk where j.id = jk.id_jalan
order by st_distance(st_aksepot($laput[x1], $laput[y1]), st_aksepot($laput[x1], $laput[y1])) limit 1),
    (select target from th_jalan j, th_jalanjendikendara jk where j.id = jk.id_jalan
order by st_distance(st_aksepot($laput[x2], $laput[y2]), st_aksepot($laput[x2], $laput[y2])) limit 1),
    false, true
) 4,
    th_jalan j
where 0.102 = j.id");
    return $data->result_array();
}
}

```

Gambar 5. 2 Kode Program Fungsi pgr_dijkstra

- c. Contoh graf dari penentuan jalur terpendek dari hasil Gambar 5.1 dapat dilihat pada Gambar 5.3.



Gambar 5. 3 Representasi Graf Sistem

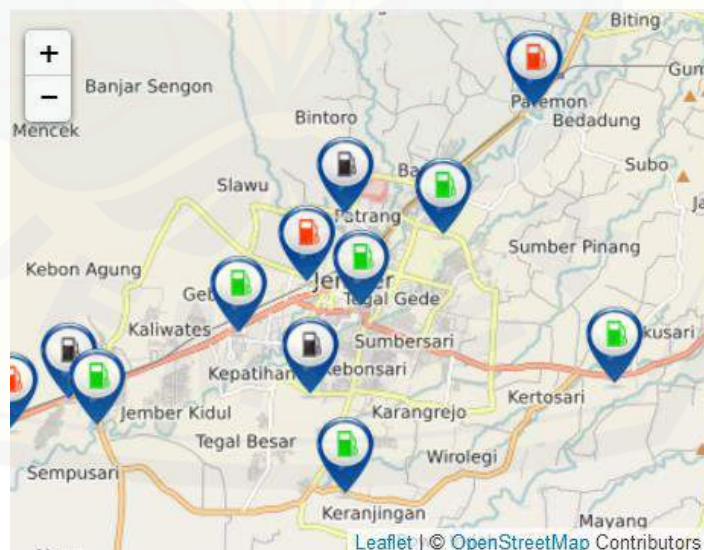
Dari contoh representasi graf Gambar 5.5, ditampilkan rute distribusi dari titik awal menuju titik tujuan yaitu, SPBU 5461089. Pemilihan rute ini mendukung karena terdapat cabang yang memiliki nilai *cost* yang berbeda sehingga dapat dilakukan perbandingan terhadap beberapa jalan yang ada. Berdasarkan Gambar 5.3,

menggunakan algoritma Dijkstra dapat menghasilkan rute terpendek 1-2-3-4-5-6-7-9. Penyelesaian perhitungan dapat dilihat pada Tabel 5.3.

Tabel 5. 3 Perhitungan Representasi Graf dari Algoritma Dijkstra

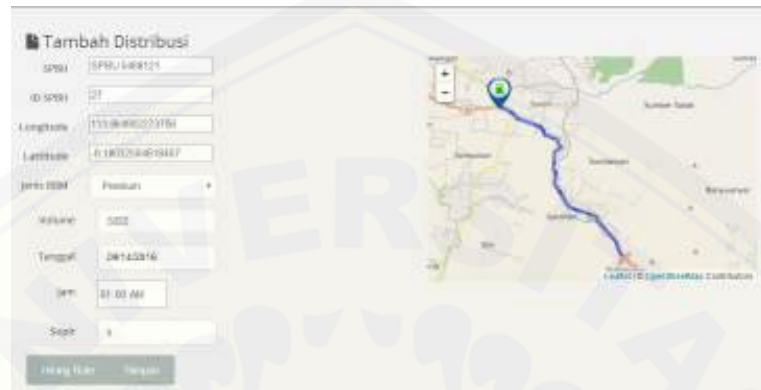
No.	Vertex Sumber	Jarak dengan Vertex Lain	
1.	1	1→2	0,238
2.	2	2→3	$0,238 + 0,310 = 0,548$
3.	3	3→4	$0,548 + 0,359 = 0,907$
4.	4	4→5	$0,907 + 0,326 = 1,233$
5.	5	5→6	$1,233 + 0,227 = 1,46$
6.	6	6→7	$1,46 + 0,233 = 1,693$
		6→8	$1,46 + 0,457 = 1,917$
8.	7	7→9	$1,693 + 0,324 = 2,017$

- d. Menentukan tujuan dalam pendistribusian BBM yang direpresentasikan dalam bentuk *marker*, yang dapat dilihat pada Gambar 5.4.



Gambar 5. 4 Marker SPBU

- e. Memilih tombol rute untuk selanjutnya akan ditampilkan hasil seperti Gambar 5.5.



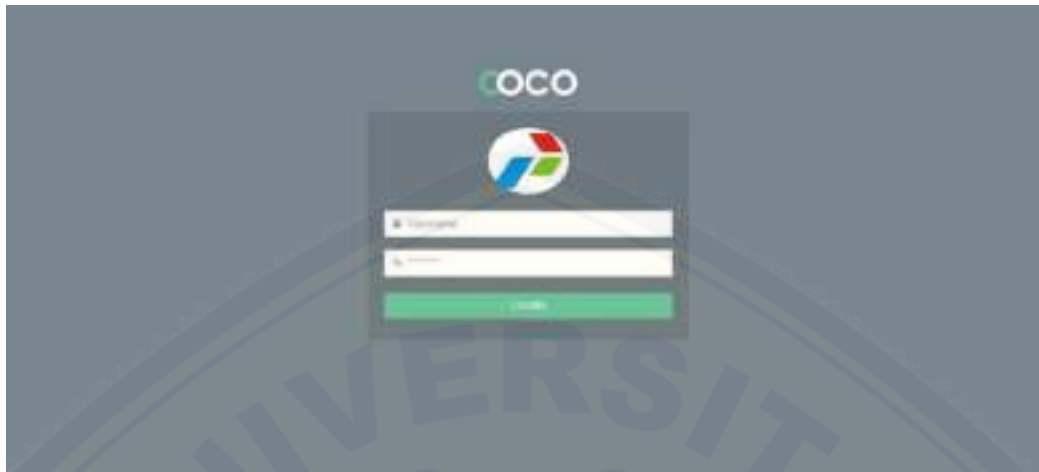
Gambar 5. 5 Hasil Pencarian Jalur Terpendek

5.2 Hasil Pembangunan Aplikasi

Hasil pembangunan aplikasi distribusi bahan bakar minyak ini memiliki beberapa fitur yang dapat diakses oleh tiga user, yaitu *Admin*, sopir, dan SPBU. Aplikasi distribusi bahan bakar minyak bertujuan untuk menentukan jalur terpendek dalam pendistribusian bahan bakar minyak dengan menggunakan algoritma Dijkstra sehingga pengiriman dapat berjalan tepat waktu. Tampilan setiap fitur akan dijelaskan pada sub-subbab di bawah.

5.2.1 Halaman *Login*

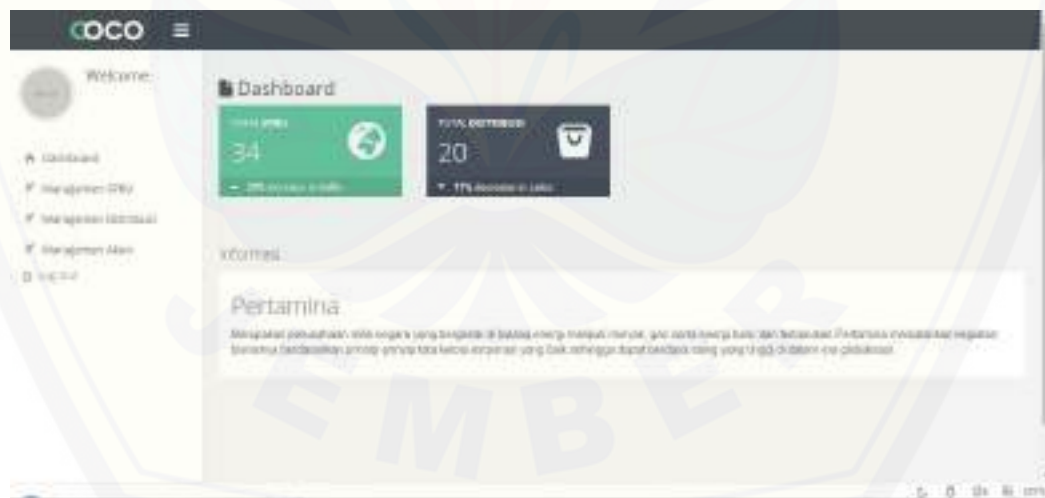
Halaman *login* merupakan halaman utama ketika semua aktor akan mengakses aplikasi. Aktor diharuskan untuk menginputkan *username* dan *password* untuk mengakses aplikasi sesuai dengan hak akses yang dimiliki. Halaman *login* pada aplikasi ini dapat dilihat pada Gambar 5.6.



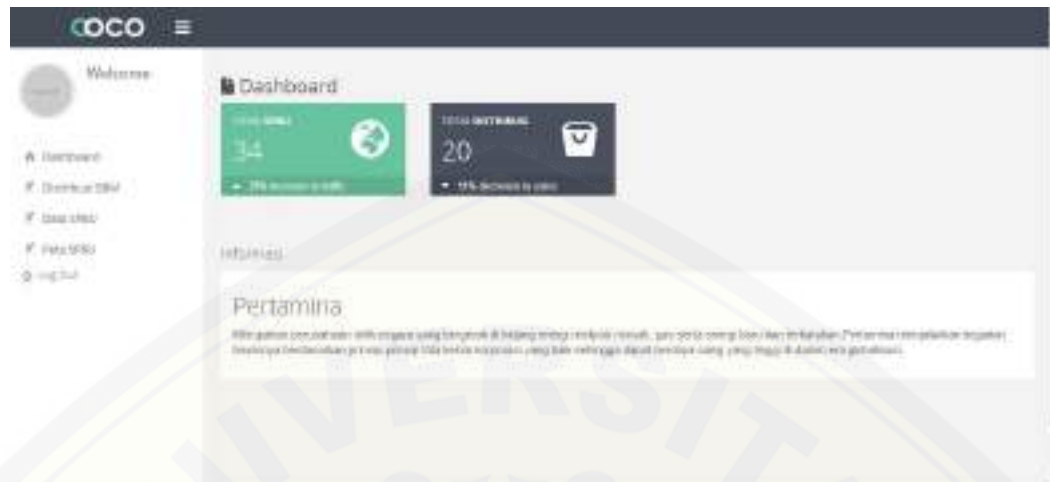
Gambar 5.6 Halaman Utama Aplikasi

5.2.2 Halaman Dashboard

Halaman dashboard merupakan halaman awal yang ditampilkan saat aktor melakukan *login* ke aplikasi. Halaman dashboard dapat dilihat pada Gambar 5.7 sampai dengan Gambar 5.9.



Gambar 5.7 Halaman Dashboard Admin



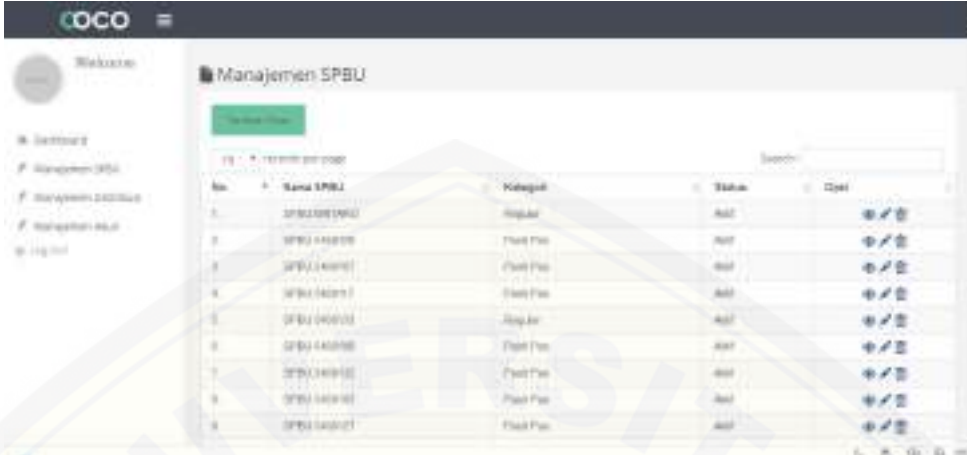
Gambar 5.8 Halaman Dashboard Sopir



Gambar 5.9 Halaman Dashboard SPBU

5.2.3 Halaman Manajemen SPBU

Halaman Manajemen SPBU dapat diakses oleh *Admin*, pada halaman Manajemen SPBU ini *Admin* dapat melakukan aktivitas untuk mengelola data SPBU yang dapat mengakses aplikasi. *Admin* dapat melakukan tambah data baru, mengedit, dan menghapus data. Halaman manajemen SPBU dapat dilihat pada Gambar 5.10.



No.	Nama SPBU	Kategori	Status	Oper.
1	SPBU 040010	Reguler	Aktif	
2	SPBU 040010	Peak Pico	Aktif	
3	SPBU 040010	Peak Pico	Aktif	
4	SPBU 040010	Peak Pico	Aktif	
5	SPBU 040010	Reguler	Aktif	
6	SPBU 040010	Peak Pico	Aktif	
7	SPBU 040010	Peak Pico	Aktif	
8	SPBU 040010	Peak Pico	Aktif	
9	SPBU 040010	Peak Pico	Aktif	

Gambar 5.10 Halaman Manajemen SPBU

5.2.4 Halaman Distribusi BBM

Halaman distribusi dapat diakses oleh *Admin*, pada halaman distribusi ini *Admin* dapat melakukan aktivitas untuk mengelola data distribusi. *Admin* dapat melakukan melihat data. Halaman data distribusi BBM dapat dilihat pada Gambar 5.11.



No.	Dept	Tabung	Jenis BBM	Volume	Tanggal	Jml
1	reg	SPBU 040010	Reguler	8000	2019-09-14	0021000
2	Wko	SPBU 040010	Pertamax	8000	2019-09-10	1250000
3	Wko Pico	SPBU 040010	Pertamax LEO	8000	2019-09-14	0030000
4	Wko	SPBU 040010	Pertamax	8000	2019-09-20	0030000
5	Wko	SPBU 040010	Pertamax	8000	2019-09-06	0030000
6	Wko	SPBU 040010	Pertamax	250	2019-08-31	0030000
7	Wko	SPBU 040010	Reguler	8000	2019-08-01	0030000
8	Wko	SPBU 040010	Pertamax	7000	2019-09-01	0030000
9	Wko	SPBU 040010	Pertamax	800	2019-09-01	0030000
10	Wko	SPBU 040010	Pertamax	8	2019-09-01	0030000

Gambar 5.11 Halaman Distribusi

5.2.5 Halaman Akun

Halaman akun dapat diakses oleh *Admin*, pada halaman akun ini *Admin* dapat melakukan aktivitas untuk mengelola data akun yang ada dalam sistem. *Admin* dapat

melakukan tambah data baru, mengedit, dan menghapus data. Halaman akun dapat dilihat pada Gambar 5.12.



Gambar 5.12 Halaman Akun

5.2.6 Halaman Tambah Distribusi

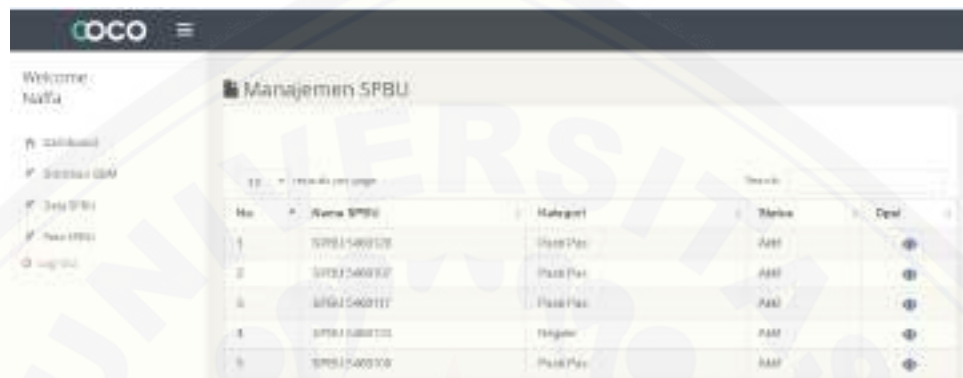
Halaman tambah distribusi dapat diakses oleh Sopir. Pada halaman tambah distribusi ini Sopir dapat melakukan aktivitas untuk memasukkan data distribusi dan pada halaman ini akan ditampilkan rute perjalanan setelah Sopir memasukkan data SPBU. Halaman tambah distribusi dapat dilihat pada Gambar 5.13.



Gambar 5.13 Halaman Tambah Distribusi

5.2.7 Halaman Data SPBU

Halaman data SPBU dapat diakses oleh Sopir, pada halaman data SPBU ini Sopir dapat melihat data SPBU yang ada di dalam sistem. Halaman data SPBU dapat dilihat pada Gambar 5.14.



No	Nama SPBU	Kategori	Status	Detail
1	SPBU 5400128	Stasiun Patis	Aktif	
2	SPBU 5400137	Stasiun Patis	Aktif	
3	SPBU 5400117	Stasiun Patis	Aktif	
4	SPBU 5400110	Stasiun Patis	Aktif	
5	SPBU 5400110	Stasiun Patis	Aktif	

Gambar 5.14 Halaman Data SPBU

5.2.8 Halaman Peta SPBU

Halaman peta SPBU dapat diakses oleh Sopir, pada halaman melihat peta ini Sopir dapat melakukan aktivitas untuk melihat peta penyebaran SPBU yang ada di wilayah kabupaten Jember. Halaman data peta SPBU dapat dilihat pada Gambar 5.15.



Gambar 5.15 Halaman Peta SPBU

5.2.9 Halaman Verifikasi Data Distribusi BBM

Halaman verifikasi data distribusi BBM dapat diakses oleh SPBU yang bersangkutan. Pada halaman ini SPBU dapat melakukan aktivitas untuk memverifikasi data distribusi BBM yang telah dilakukan oleh Sopir. Halaman verifikasi data distribusi BBM dapat dilihat pada Gambar 5.16.



No.	Sopir	Tujuan	Jenis BBM	Volume	Tanggal	Jam	Status	Aksi
1	1465	SPBU 1460718	Premium	10000	2019-05-30	14:11:00	Selesai Distribusi	✓
2	1465	SPBU 1460718	Premium	5000	2019-05-30	13:00:00	Terdistribusi	Selesai

Gambar 5.16 Halaman Verifikasi Data Distribusi BBM

5.3 Pembahasan

5.3.1 Analisis Algoritma Dijkstra

Algoritma Dijkstra yang dipakai sangat mendukung dalam pengembangan sistem penentuan jalur terpendek. Hasil yang didapatkan mampu mendapatkan jalur terpendek dalam rute pengiriman BBM oleh Pertamina dengan memperhatikan beberapa indikator yang telah diberikan. Indikator-indikator yang ada berupa panjang jalan, jenis kendaraan yang dipakai, lokasi awal, dan lokasi tujuan SPBU.

Optimasi pemilihan jalur yang digunakan dalam pendistribusian menggunakan algoritma Dijkstra disini yaitu penghitungan rata-rata *cost* dan *reverse cost* pada setiap jalan yang dilewati serta jenis kendaraan yang dipakai. Sehingga pemilihan jalur tidak akan salah dan akan sesuai dengan yang ada di dalam sistem.

5.3.2 Analisis Model *Waterfall*

Pengembangan sistem dengan menggunakan model *Waterfall* ternyata memiliki beberapa kendala yang ada. Model *Waterfall* yang dipilih dalam mengembangkan

sistem ini karena sumber daya manusia (SDM) hanya satu orang, aplikasi masih dalam skala kecil, dan menyesuaikan dengan user yang akan menjadi pengguna aplikasi ternyata tidak bisa dilakukan sesuai dengan yang terdapat dalam teori model *Waterfall*. Pengembangan sistem dilakukan sesuai dengan tahapan yang ada dalam model *Waterfall* tetapi ada beberapa urutan pengerjaan yang masih kurang konsisten. Urutan pengerjaan yang kurang konsisten disini terjadi pada saat setelah tahap pengimplementasian koding ke dalam sistem, terjadi perubahan kebutuhan yang mengakibatkan berubahnya aturan sistem. Dikarenakan perubahan kebutuhan tersebut seharusnya sesuai dengan model *Waterfall*, yaitu memulai kembali dari tahapan awal pada saat analisis kebutuhan menjadi hanya memperbaiki pada tahapan desain sistem. Sehingga membuat pengembangan sistem menjadi kurang rapi dan tidak menurut pada teori model *Waterfall*.

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

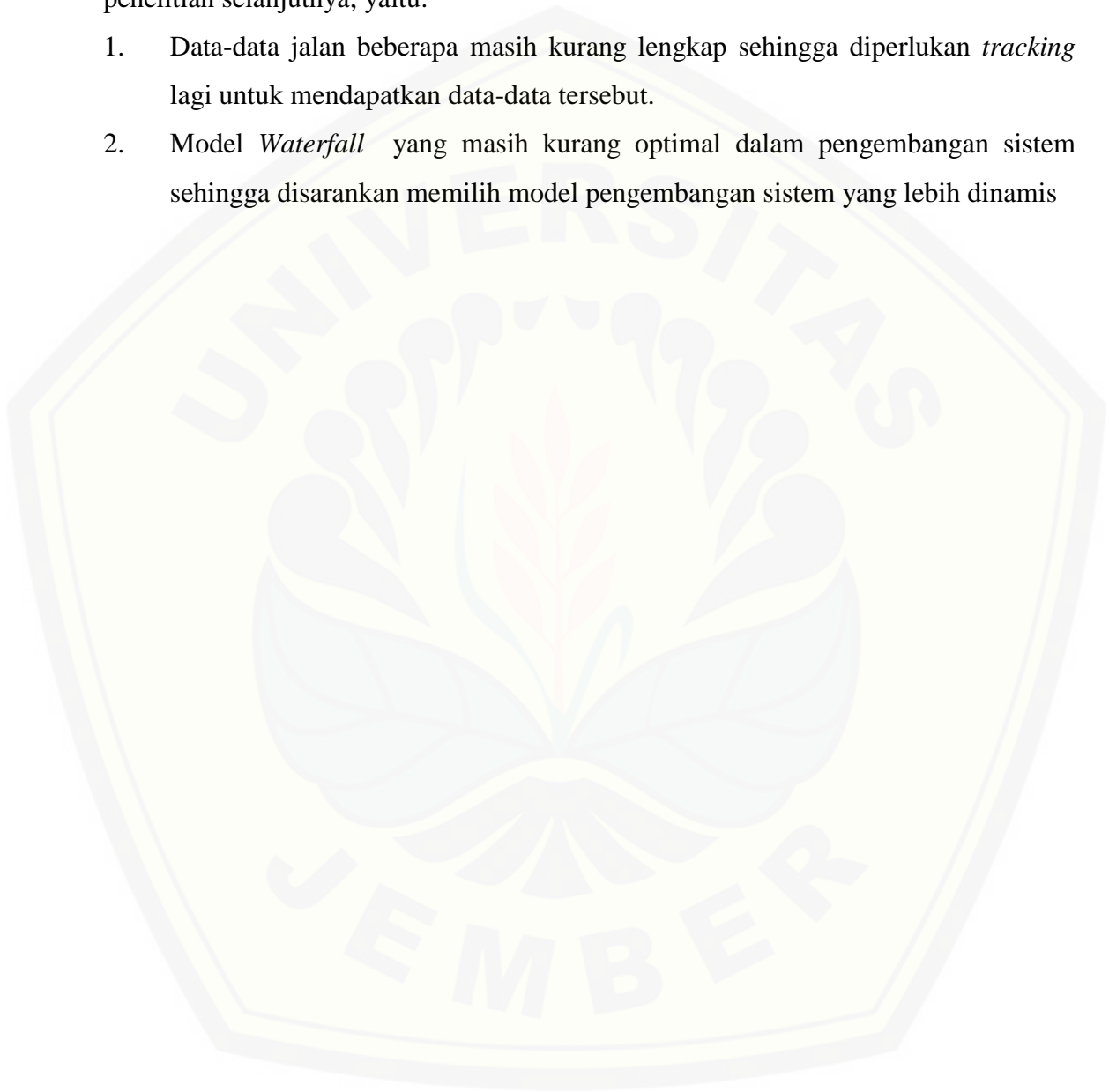
Kesimpulan dari penelitian yang dilakukan adalah:

1. Penelitian ini menghasilkan sebuah aplikasi penentu jarak terpendek dalam pendistribusian bahan bakar minyak. Aplikasi ini menggunakan algoritma *Dijkstra* yang berfungsi untuk menunjukkan rute terpendek dalam distribusi BBM yang telah dibandingkan dengan rute yang lain. Penggunaan algoritma *Dijkstra* dalam aplikasi ini berhasil mendapatkan jalur terpendek dalam rute pengiriman BBM oleh Pertamina. Berdasarkan hasil yang didapat, algoritma *Dijkstra* dapat mengoptimasi pemilihan jalur yang akan digunakan dalam pendistribusian. Optimasi disini yaitu penghitungan rata-rata *cost* dan *reverse cost* pada setiap jalan yang dilewati serta jenis kendaraan yang dipakai.
2. Pengembangan aplikasi menggunakan model *waterfall* yang pada penerapannya memiliki kelemahan yaitu terjadinya iterasi yang menyebabkan proses pengembangan kurang mengikuti urutan sekuensial seperti pada teori.
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 tangki.

6.2 Saran

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

1. Data-data jalan beberapa masih kurang lengkap sehingga diperlukan *tracking* lagi untuk mendapatkan data-data tersebut.
2. Model *Waterfall* yang masih kurang optimal dalam pengembangan sistem sehingga disarankan memilih model pengembangan sistem yang lebih dinamis



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.*
- Fakhri. (2008). Penerapan Algoritma Dijkstra Dalam Pencarian Solusi Maximum Flow Problem. *Makalah If2251 Strategi Algoritmik.*
- Fitria, & Apri Triansyah. (2013). Implementasi Algoritma Dijkstra dalam Aplikasi untuk Menentukan Lintasan Terpendek Jalan Darat Antar Kota di Sumatera Bagian Selatan.
- Gusmao, A., & dkk. (2013). Sistem Informasi Geografis Pariwisata Berbasis Web Dan Pencarian Jalur Terpendek Dengan Algoritma Dijkstra. *Jurnal EECCIS.*
- Muin, S. A. (1998). *Pesawat - Pesawat Konveksi Energi I (Ketel Uap)*. Jakarta: Rajawali Pers.
- Pertamina. (2014, April 13). Stasiun Pengisian Bahan Bakar Umum (SPBU). pp. [Online]. [http://www.pertamina.com/our-business/hilir/pemasaran -dan-niaga/produkdan-layanan/produk-konsumen/spbu/](http://www.pertamina.com/our-business/hilir/pemasaran-dan-niaga/produkdan-layanan/produk-konsumen/spbu/).
- 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.
- Sembiring, N. (1991). *Ekonomi Koperasi*. Jakarta: Erlangga.

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*.



LAMPIRAN

A. *Use Case Scenario*A.1 Skenario *Use Case* Pengelolaan Data SPBU

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* pengelolaan data SPBU dapat dilihat pada Tabel 1.

Tabel 1 Skenario *Use Case* Pengelolaan Data SPBU

Nama Use Case	Pengelolaan Manajemen SPBU
Aktor	<i>Admin</i>
Deskripsi Singkat	<i>Admin</i> akan mengelola data manajemen SPBU
Prekondisi	Data manajemen SPBU
Prakondisi	<ol style="list-style-type: none"> 1. Data manajemen SPBU berhasil ditambahkan 2. Data manajemen SPBU berhasil diubah 3. Data manajemen SPBU berhasil dihapus
Flow Events	
Skenario Normal: Melihat Data Manajemen SPBU	
Aksi Aktor	Reaksi Sistem
1. Klik menu Manajemen SPBU	2. Menampilkan halaman SPBU meliputi tabel data SPBU yang telah dimasukkan beserta tombol detail, ubah dan hapus
Skenario Normal: Melihat Detail Data Manajemen SPBU	
Aksi Aktor	Reaksi Sistem

3. Klik tombol Detail	
	4. Menampilkan halaman detail SPBU meliputi nama SPBU, kategori, alamat, produk BBM, fasilitas, jumlah pompa, layanan, dan keterangan.
Skenario Normal: Tambah Data Manajemen SPBU	
Aksi Aktor	Reaksi Sistem
3. Klik tombol Tambah Data	
	4. Menampilkan form tambah data SPBU meliputi nama SPBU, alamat, kategori, layanan, produk, fasilitas, gambar, jumlah pompa, latitude, longitude, keterangan, dan peta SPBU.
5. Klik tombol simpan	
	6. Menampilkan halaman manajemen SPBU.
Skenario Alternatif: Form Data Tidak Lengkap	
Aksi Aktor	Reaksi Sistem
5a. Klik tombol Simpan	
	6a. Menampilkan warning “please fill out this field”
Skenario Normal: Batal Tambah Data Manajemen SPBU	
Aksi Aktor	Reaksi Sistem
5a. Klik tombol Batal	
	6a. Menampilkan halaman manajemen SPBU.

Skenario Normal: Ubah Data Manajemen SPBU	
Aksi Aktor	Reaksi Sistem
3. Klik tombol Ubah	
	4. Menampilkan form ubah data SPBU meliputi nama SPBU, alamat, kategori, layanan, produk, fasilitas, gambar, jumlah pompa, latitude, longitude, keterangan, dan peta SPBU.
5. Mengubah form ubah data manajemen SPBU	
6. Klik tombol Simpan	
	7. Menyimpan ke database
	8. Menampilkan halaman manajemen SPBU
Skenario Alternatif: Form Data Tidak Lengkap	
Aksi Aktor	Reaksi Sistem
6a. Klik tombol Simpan	
	7a. Menampilkan warning "please fill out this field"
Skenario Normal: Memilih Tombol Batal	
Aksi Aktor	Reaksi Sistem
6b. Klik tombol Batal	
	7b. Menampilkan kembali halaman manajemen SPBU
Skenario Normal: Hapus Data Manajemen SPBU	
Aksi Aktor	Reaksi Sistem
3. Klik tombol Hapus	

	4. Menghapus data pada database
	5. Menampilkan halaman manajemen SPBU

A.2 Skenario *Use Case* Melihat Data Distribusi BBM

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

Tabel 2 Skenario *Use Case* Melihat Data Distribusi BBM

Nama Use Case	Melihat Data Distribusi BBM
Aktor	<i>Admin</i>
Deskripsi Singkat	<i>Admin</i> akan mengelola data manajemen distribusi
Prekondisi	Data distribusi BBM
Prakondisi	1. Data distribusi BBM berhasil diubah 2. Data distribusi BBM berhasil dihapus
Flow Events	
Skenario Normal: Melihat Data Distribusi BBM	
Aksi Aktor	Reaksi Sistem
1. Klik menu Manajemen Distribusi	
	2. Menampilkan halaman distribusi meliputi tabel data distribusi yang telah dimasukkan beserta tombol ubah dan hapus

A.3 Skenario *Use Case* Pengelolaan Akun

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

Tabel 3 Skenario *Use Case* Pengelolaan Akun

Nama Use Case	Pengelolaan Manajemen Akun	
Aktor	<i>Admin</i>	
Deskripsi Singkat	<i>Admin</i> akan mengelola data manajemen akun	
Prekondisi	Data manajemen akun	
Prakondisi	3. Data manajemen akun berhasil ditambahkan 4. Data manajemen akun berhasil diubah 5. Data manajemen akun berhasil dihapus	
Flow Events		
Skenario Normal: Melihat Data Manajemen Akun		
Aksi Aktor	Reaksi Sistem	
1. Klik menu Manajemen Akun		
	2. Menampilkan halaman akun meliputi tabel data akun yang telah dimasukkan beserta tombol detail, ubah dan hapus	
Skenario Normal: Melihat Detail Data Manajemen Akun		
Aksi Aktor	Reaksi Sistem	
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.
Skenario Normal: Tambah Data Manajemen Akun	
Aksi Aktor	Reaksi Sistem
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 database
	7. Menampilkan halaman manajemen akun.
Skenario Alternatif: Form Data Tidak Lengkap	
Aksi Aktor	Reaksi Sistem
5a. Klik tombol Simpan	
	6a. Menampilkan warning “please fill out this field”
Skenario Normal: Batal Tambah Data Manajemen Akun	
Aksi Aktor	Reaksi Sistem
5a. Klik tombol Batal	
	6a. Menampilkan halaman manajemen akun.
Skenario Normal: Ubah Data Manajemen Akun	

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

	5. Menampilkan halaman manajemen akun
--	---------------------------------------

A.4 Skenario *Use Case* Melihat Data SPBU

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal, skenario *use case* melihat SPBU dapat dilihat pada Tabel 4.

Tabel 4 Skenario *Use Case* Melihat Data SPBU

Nama <i>Use Case</i>	Melihat Data SPBU
Aktor	Sopir
Deskripsi Singkat	Sopir akan melihat data SPBU
Prekondisi	Data SPBU
Prakondisi	Data SPBU berhasil dilihat
<i>Flow Events</i>	
Skenario Normal: Melihat Data SPBU	
Aksi Aktor	Reaksi Sistem
1. Klik menu Data SPBU	
	2. Menampilkan halaman data SPBU meliputi tabel data SPBU yang telah diinputkan dengan tombol detail.
3. Klik tombol Detail	
	4. Menampilkan detail data SPBU secara lengkap meliputi foto, nama spbu, kategori, alamat, produk BBM, fasilitas, jumlah pompa, layanan, keterangan, dan rating.

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

Nama <i>Use Case</i>	Melihat Peta
Aktor	Sopir
Deskripsi Singkat	Sopir akan melihat peta SPBU
Prekondisi	Peta SPBU
Prakondisi	Peta SPBU berhasil dilihat
<i>Flow Events</i>	
Skenario Normal: Melihat Peta SPBU	
Aksi Aktor	Reaksi Sistem
1. Klik menu Peta SPBU	
	2. Menampilkan halaman peta SPBU beserta marker.

A.6 Skenario *Use Case* Verifikasi Data Distribusi BBM

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

Tabel 6 Skenario *Use Case* Verifikasi Data Distribusi BBM

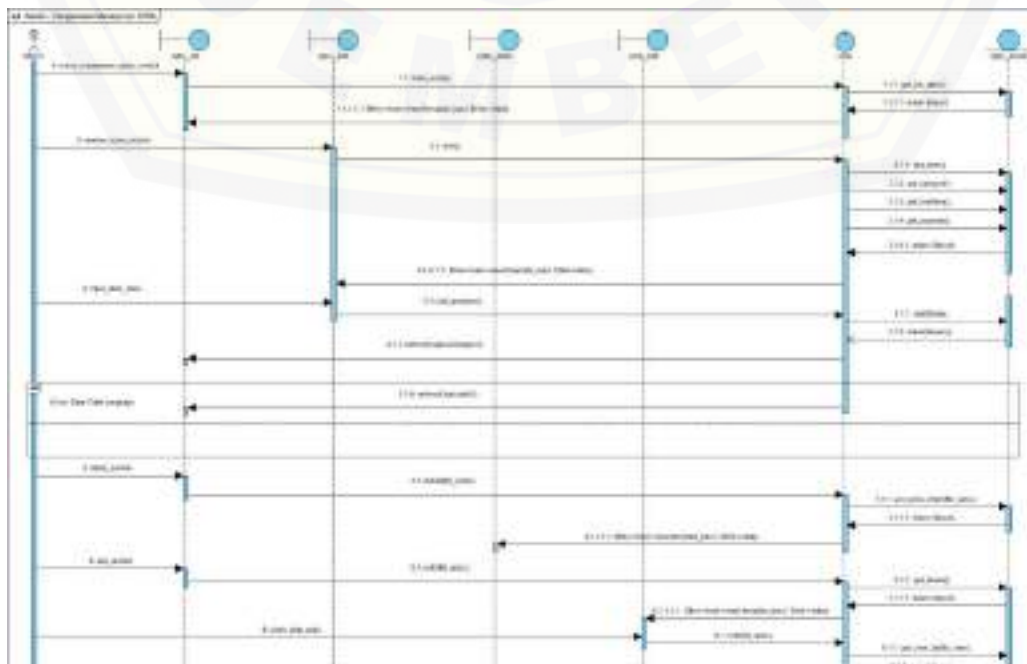
Nama <i>Use Case</i>	Verifikasi Data Distribusi BBM
Aktor	SPBU
Deskripsi Singkat	SPBU akan melakukan verifikasi data distribusi BBM
Prekondisi	Data distribusi BBM belum terverifikasi

Prakondisi	Data distribusi BBM berhasil diverifikasi
<i>Flow Events</i>	
Skenario Normal: Verifikasi Data Distribusi BBM	
Aksi Aktor	Reaksi Sistem
1. Klik menu Verifikasi Distribusi	
	2. Menampilkan halaman data distribusi BBM, meliputi tabel data distribusi, dan tombol verifikasi
3. Klik tombol Verifikasi pada data distribusi yang dipilih	
	4. Menampilkan data distribusi BBM yang telah terverifikasi

B. Sequence Diagram

B.1. Sequence Diagram Pengelolaan Data SPBU

Penggambaran *sequence diagram* pengelolaan data SPBU digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 1.



B.2 *Sequence Diagram* Pengelolaan Data Akun

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



Gambar 2 *Sequence Diagram* Pengelolaan Data Akun

B.3 Sequence Diagram Melihat Distribusi BBM

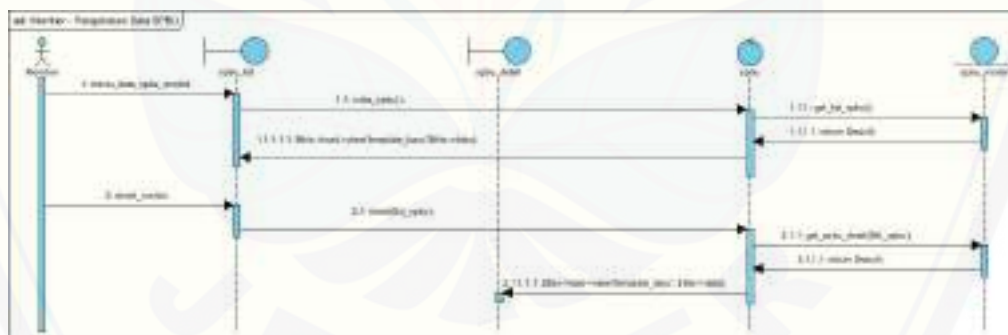
Penggambaran *sequence diagram* melihat distribusi BBM digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 3.



Gambar 3 Sequence Diagram Melihat Data Distribusi

B.4 Sequence Diagram Melihat Data SPBU

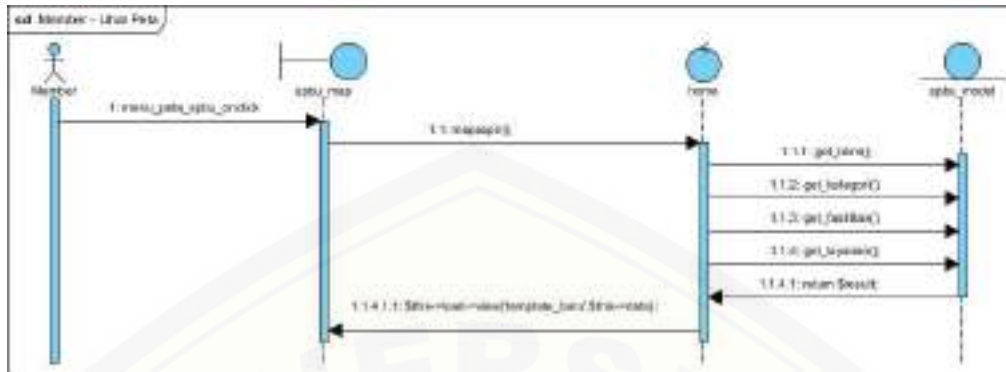
Penggambaran *sequence diagram* melihat data SPBU digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 4.



Gambar 4 Sequence Diagram Melihat SPBU

B.5 Sequence Diagram Melihat Peta

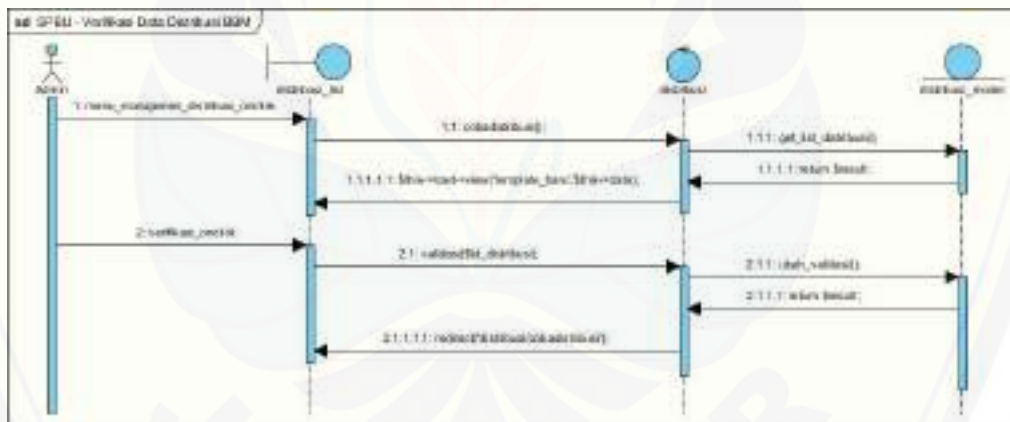
Penggambaran *sequence diagram* melihat peta digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 5.



Gambar 5 Sequence Diagram Melihat Peta

B.6 Sequence Diagram Verifikasi Data Distribusi BBM

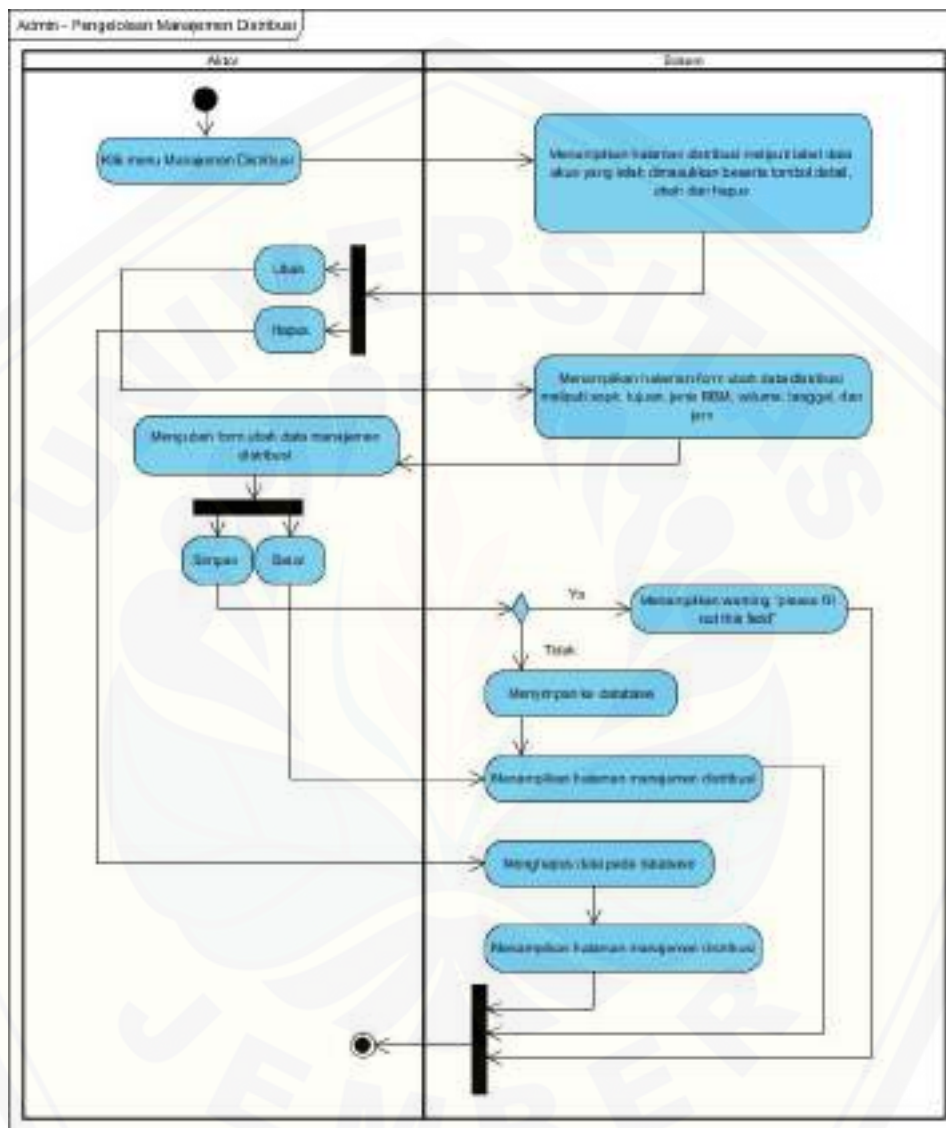
Penggambaran *sequence diagram* verifikasi data distribusi BBM digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 6.



Gambar 6 Sequence Diagram Verifikasi Data Distribusi BBM

C.2 Activity Diagram Melihat Data Distribusi

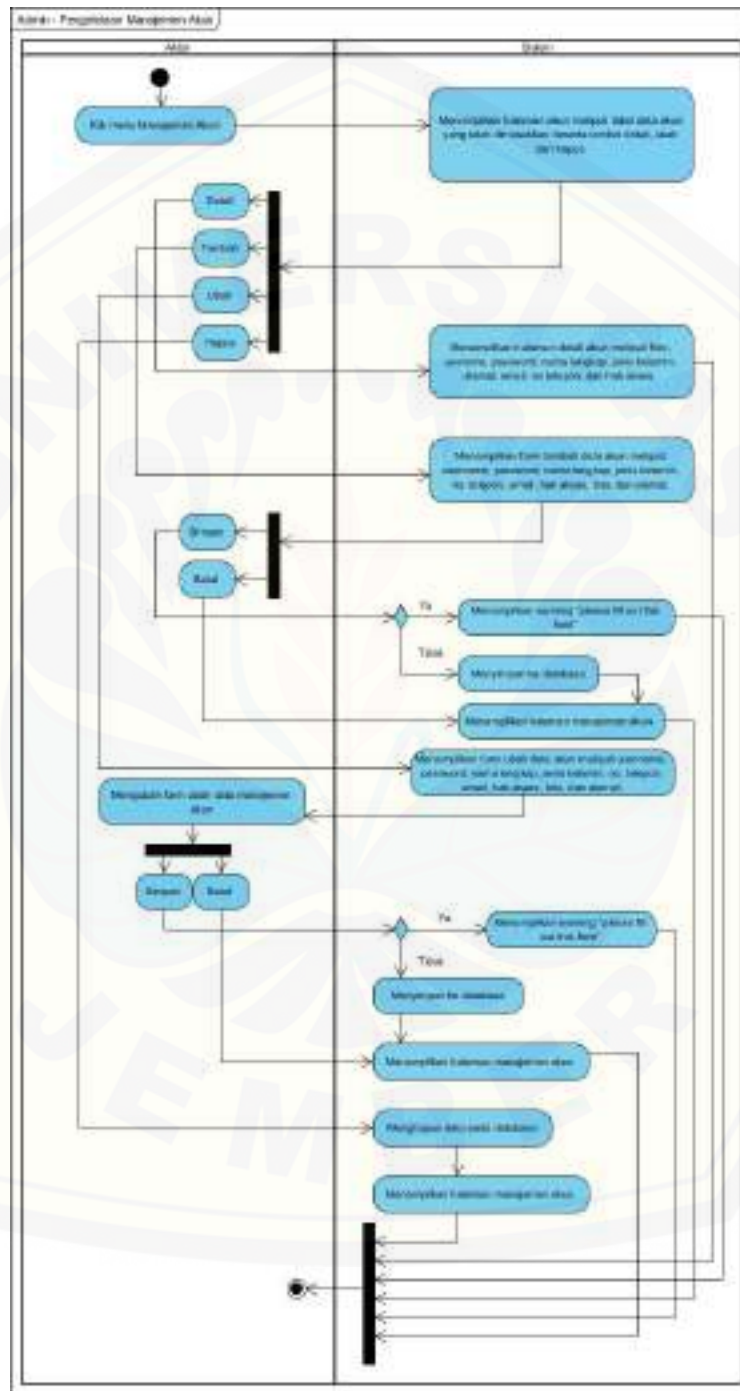
Activity diagram melihat data distribusi dapat dilihat pada Gambar 8.



Gambar 8 Activity Diagram Pengelolaan Data Distribusi

C.3 Activity Diagram Pengelolaan Data Akun

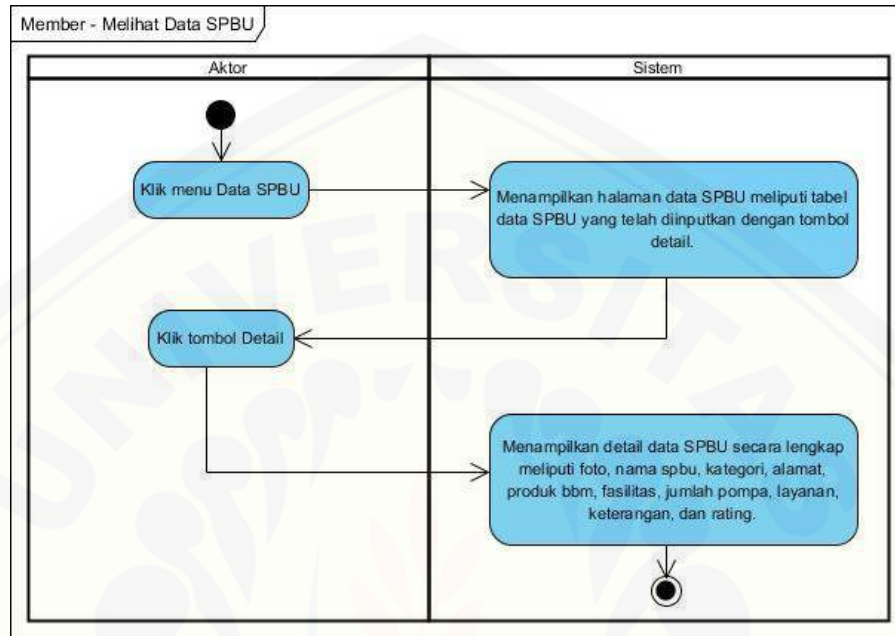
Activity diagram pengelolaan data akun dapat dilihat pada Gambar 9.



Gambar 9 Activity Diagram Pengelolaan Data Akun

C.4 Activity Diagram Melihat SPBU

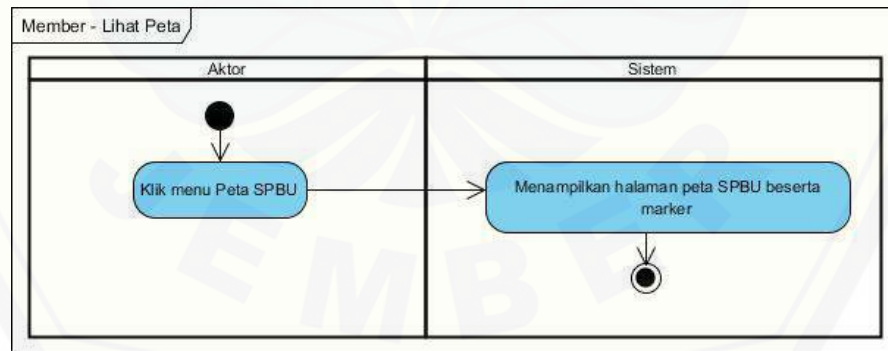
Activity diagram melihat spbu dapat dilihat pada Gambar 10.



Gambar 10 Activity Diagram Melihat SPBU

C.5 Activity Diagram Melihat Peta

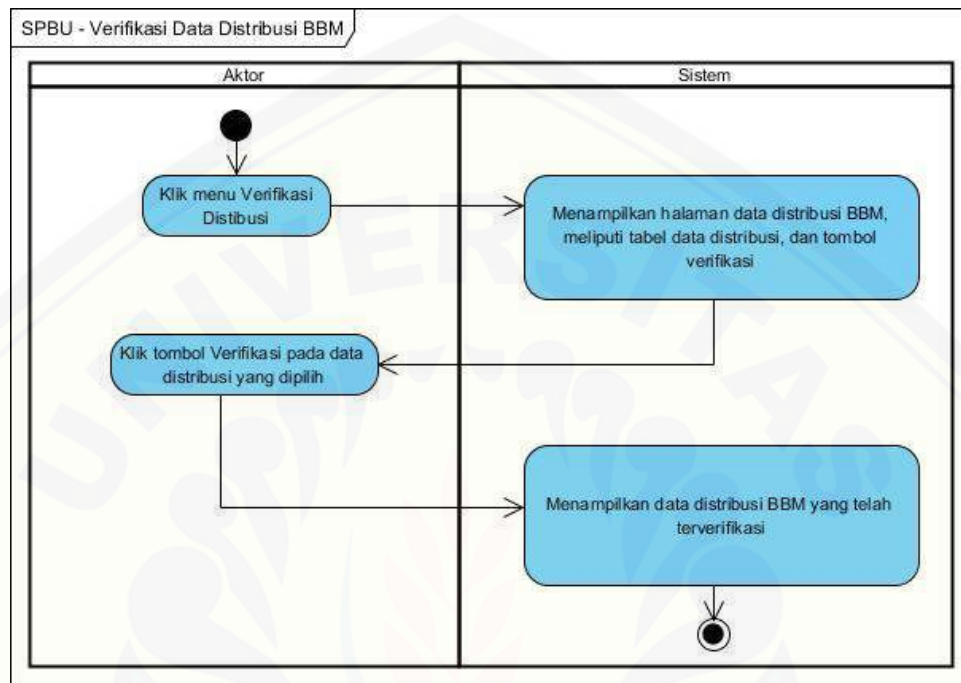
Activity diagram melihat peta dapat dilihat pada Gambar 11.



Gambar 11 Activity Diagram Melihat Peta

C.6 Activity Diagram Verifikasi Data Distribusi BBM

Activity diagram verifikasi data distribusi BBM dapat dilihat pada Gambar 12.



Gambar 12 Activity Diagram Verifikasi Data Distribusi BBM

D. Kode Program

D.1 Kode Program Pengelolaan Data SPBU

Kode program pengelolaan data SPBU terletak pada kelas spbu dan spbu_model.

1. Kelas spbu

Penulisan kode program spbu dapat dilihat pada Tabel 7.

Tabel 7 Kode Program Kelas spbu

1.	<?php
2.	if (!defined('BASEPATH'))
3.	exit('No direct script access allowed');
4.	class Spbu extends CI_Controller {
5.	private \$data;
6.	function __construct() {
7.	parent::__construct();
8.	\$this->load->model('spbu_model');
9.	\$this->load->model('akun_model');
10.	\$this->data = array(
11.	'title' => NULL,
12.	'content' => NULL,
13.	'menu' => NULL,
14.	'akses' => \$this->session->userdata('spbu_hakakses'),
15.	'username' => \$this->session->userdata('spbu_username'),
16.	'nama_lengkap' => \$this->session->userdata('nama_lengkap')
17.);
18.	}
19.	// memanggil function list spbu untuk ditampilkan di halaman awal manajemen spbu
20.	function cobaspbu(){
21.	\$this->data['menu'] = 1;
22.	\$this->data['title'] = 'Manajemen SPBU';
23.	\$this->data['content'] = \$this->load->view('spbu_list',array(
24.	'data' => \$this->spbu_model->get_list_spbu(
25.),TRUE);
26.	\$this->load->view('template_baru',\$this->data);
27.	}
28.	// menampilkan detail ketika user mengklik icon spbu
29.	function detail(\$id) {
30.	\$data_spbu = \$this->spbu_model->get_spbu_by(\$id);
31.	\$this->data['title'] = \$data_spbu['nama_spbu'];
32.	\$this->data['content'] = \$this->load->view(
33.	'spbu_detail', array(
34.	'id_spbu' => \$id,
35.	'data' => \$data_spbu,
36.	'data_produk' => \$this->spbu_model->get_spbu_produk(\$id),
37.	'data_fasilitas' => \$this->spbu_model->get_spbu_fasilitas(\$id)
38.), TRUE);
39.	\$this->load->view('template_baru', \$this->data);
40.	}
41.	// fungsi ajax untuk menampilkan detail spbu
42.	function detail_ajax() {
43.	\$id = \$this->input->post('id');

```
44. $data_spbu = $this->spbu_model->get_spbu_by($id);
45. $produk = $this->spbu_model->get_spbu_produk($id);
46. $fasilitas = $this->spbu_model->get_spbu_fasilitas($id);
47. if ($data_spbu['gambar'] == "") {
48.     $gambar = 'no-image.png';
49. } else {
50.     $gambar = $data_spbu['gambar'];
51. }
52. echo '||';
54. echo '<h1>' . $data_spbu['nama_spbu'] . '</h1>';
55. echo 'Alamat : ' . $data_spbu['lokasi_spbu'] . '<br/>';
56. echo 'Kategori : ' . $data_spbu['nama_kategori'] . '<br/>';
57. echo 'Produk BBM : ';
58. $produk_str = "";
59. foreach ($produk as $value) {
60.     $produk_str .= $value['nama_BBM'] . ', ';
61. }
62. echo substr($produk_str, 0, -2);
63. echo '<br/>';
64. echo 'Fasilitas : ';
65. $fasilitas_str = "";
66. foreach ($fasilitas as $value) {
67.     $fasilitas_str .= $value['nama_fasilitas'] . ', ';
68. }
69. echo substr($fasilitas_str, 0, -2);
70. echo '<br/>';
71. echo 'Layanan : ' . $data_spbu['waktu_layanan'] . '<br/>';
72. echo 'Keterangan : Tersedia ' . $data_spbu['jumlah_pompa'] . ' pompa. ' .
73. $data_spbu['keterangan'] . '<br/>';
74. echo '||' . $rate;
75. }
76. //menampilkan halaman input spbu
77. function add() {
78.     if ($this->data['akses'] != NULL) {
79.         $this->data['title'] = 'Tambah SPBU';
80.         $this->data['content'] = $this->load->view(
81.             'spbu_add', array(
82.                 'produk' => $this->spbu_model->get_BBM(),
83.                 'kategori' => $this->spbu_model->get_kategori(),
84.                 'fasilitas' => $this->spbu_model->get_fasilitas(),
85.                 'layanan' => $this->spbu_model->get_layanan()
86.             ), TRUE);
87.         $this->load->view('template_baru', $this->data);
88.     } else {
89.         redirect('home');
90.     }
91. }
92. //menampilkan halaman edit, dengan mengambil id spbu
93. function edit($id) {
94.     if ($this->data['akses'] != NULL) {
95.         $this->data['title'] = 'Edit SPBU';
96.         $this->data['content'] = $this->load->view('spbu_edit', array(
97.             'data' => $this->spbu_model->get_spbu_detail($id),
98.             'BBM' => $this->spbu_model->get_BBM(),
99.             'produk' => $this->spbu_model->get_spbu_produk($id),
100.            'kategori' => $this->spbu_model->get_kategori(),
```

```
101. 'fasilitas' => $this->spbu_model->get_fasilitas(),
102. 'spbu_fasilitas' => $this->spbu_model->get_spbu_fasilitas($id),
103. 'layanan' => $this->spbu_model->get_layanan()
104. ), TRUE);
105. $this->load->view('template_baru', $this->data);
106. } else {
107. redirect('home');
108. }
109. }
110. // fungsi yang dijalankan dalam proses menambah spbu
111. function add_process() {
112. if ($this->data['akses'] != NULL) {
113. if ($this->__add_rule()) {
114. if ($_FILES['image']['name'] == "") {
115. $gambar = 1;
116. } else {
117. $gambar = $_FILES['image']['name'];
118. }
119. $data = array(
120. 'nama_spbu' => strtoupper($this->input->post('nama_spbu')),
121. 'lokasi' => $this->input->post('lokasi'),
122. 'id_kategori' => $this->input->post('kategori'),
123. 'gambar' => $gambar,
124. 'jumlah_pompa' => $this->input->post('jumlah_pompa'),
125. 'lat' => $this->input->post('lat'),
126. 'long' => $this->input->post('long'),
127. 'produk' => $this->input->post('produk'),
128. 'fasilitas' => $this->input->post('fasilitas'),
129. 'id_layanan' => $this->input->post('layanan'),
130. 'keterangan' => $this->input->post('keterangan'),
131. 'id_icon' => $this->input->post('produk')
132. );
133. if ($data['gambar'] == 1) {
134. $sukses = $this->spbu_model->add_spbu($data);
135. if ($sukses) {
136. $this->session->set_userdata('message', '1,SPBU telah ditambahkan.');
```

```
158. } else {
159. $this->session->set_userdata('message', '2,SPBU gagal ditambahkan.');
```

160. redirect('home/cobaspbu/');
161. }
162. } else {
163. \$this->session->set_userdata('message', '2,Upload gambar gagal.');

164. redirect('spbu/add');
165. }
166. }
167. } else {
168. \$this->add();
169. }
170. } else {
171. redirect('home');

172. }
173. }
174. // fungsi menyimpan perubahan dalam proses edit
175. function edit_process() {
176. if (\$this->data['akses'] != NULL) {
177. if (\$this->__edit_rule()) {
178. if (\$_FILES['image']['name'] == "") {
179. \$gambar = 1;
180. } else {
181. \$gambar = \$_FILES['image']['name'];
182. }
183. \$data = array(
184. 'id_spbu' => \$this->input->post('id_spbu'),
185. 'nama_spbu' => strtoupper(\$this->input->post('nama_spbu')),
186. 'lokasi' => \$this->input->post('lokasi'),
187. 'id_kategori' => \$this->input->post('kategori'),
188. 'gambar' => \$gambar,
189. 'jumlah_pompa' => \$this->input->post('jumlah_pompa'),
190. 'lat' => \$this->input->post('lat'),
191. 'long' => \$this->input->post('long'),
192. 'produk' => \$this->input->post('produk'),
193. 'fasilitas' => \$this->input->post('fasilitas'),
194. 'id_layanan' => \$this->input->post('layanan'),
195. 'keterangan' => \$this->input->post('keterangan'),
196. 'id_icon' => \$this->icon_spbu(\$this->input->post('produk'))
197.);
198. if (\$data['gambar'] == 1) {
199. \$sukses = \$this->spbu_model->edit_spbu(\$data);
200. if (\$sukses) {
201. \$this->session->set_userdata('message', '1,SPBU telah diupdate.');

202. redirect('spbu/cobaspbu/' . \$this->input->post('id_spbu'));
203. } else {
204. \$this->session->set_userdata('message', '2,SPBU gagal diupdate.');

205. redirect('spbu/edit/' . \$this->input->post('id_spbu'));
206. }
207. } else {
208. \$config_image = array(
209. 'upload_path' => './images/',
210. 'allowed_types' => 'gif|jpg|png|jpeg',
211. 'max_size' => '500000',
212. 'overwrite' => TRUE,
213. 'remove_spaces' => FALSE,
214. 'file_name' => \$_FILES['image']['name']


```
215. );
216. $this->load->library('upload');
217. $this->upload->initialize($config_image);
218. if ($this->upload->do_upload('image')) {
219.     $sukses = $this->spbu_model->edit_spbu($data);
220.     if ($sukses) {
221.         $this->session->set_userdata('message', '1,SPBU telah diupdate.');
```

```
222.         redirect('spbu/edit/' . $this->input->post('id_spbu'));
223.     } else {
224.         $this->session->set_userdata('message', '2,SPBU gagal diupdate.');
```

```
225.         redirect('spbu/edit/' . $this->input->post('id_spbu'));
226.     }
227. } else {
228.     $this->session->set_userdata('message', '2,Upload gambar gagal.');
```

```
229.     redirect('spbu/edit/' . $this->input->post('id_spbu'));
230. }
231. }
232. } else {
233.     $this->edit($this->input->post('id_spbu'));
234. }
235. } else {
236.     redirect('home');
```

```
237. }
238. }
239. // proses mennghapus id berdasarkan spbu yang dipilih
240. function delete_process($id) {
241.     if ($this->data['akses'] != NULL) {
242.         if ($this->spbu_model->cek_transaksi($id) == TRUE) {
243.             $sukses = $this->spbu_model->delete_spbu($id);
244.             if ($sukses == true) {
245.                 $this->session->set_userdata('message', '1,SPBU telah dihapus.');
```

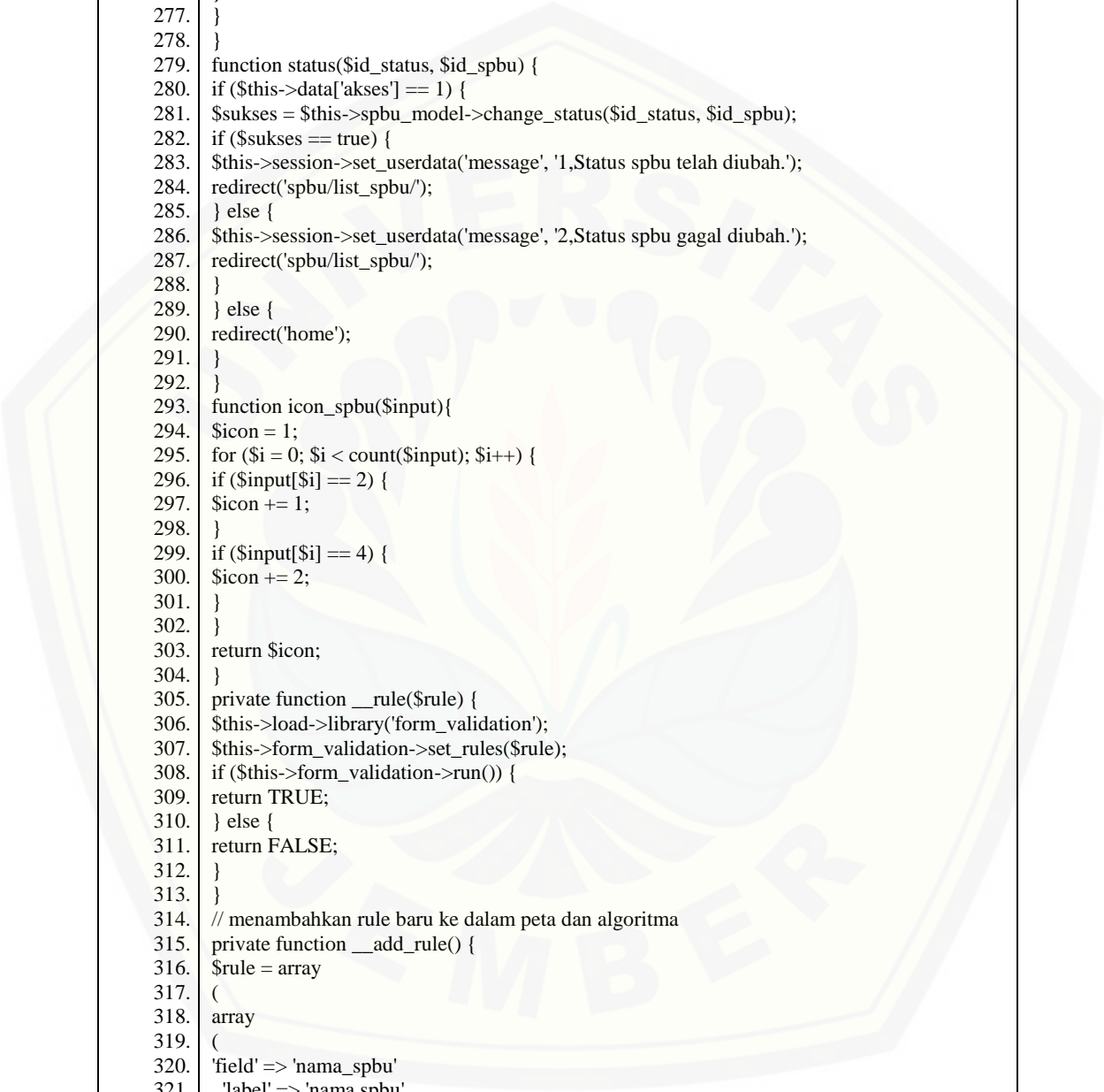
```
246.                 redirect('spbu/cobaspbu');
247.             } else {
248.                 $this->session->set_userdata('message', '2,SPBU gagal dihapus.');
```

```
249.                 redirect('spbu/list_spbu/');
250.             }
251.         } else {
252.             $this->session->set_userdata('message', '2,Telah ada transaksi dengan tabel lain SPBU tidak
253.             boleh dihapus.');
```

```
254.             redirect('spbu/list_spbu/');
255.         }
256.     } else {
257.         redirect('home');
```

```
258.     }
259. }
260. // fungsi dalam tambah spbu, untuk mengecek adakah nama yang sama
261. function cek_namaspbu() {
262.     $validateValue = $_REQUEST['fieldValue'];
263.     $validateId = $_REQUEST['fieldId'];
264.     $arrayToJs = array();
265.     $arrayToJs[0] = $validateId;
266.     $cek = $this->spbu_model->cek_namaspbu($validateValue);
267.     if ($cek == true) {
268.         $arrayToJs[1] = true;
269.         echo json_encode($arrayToJs);
270.     } else {
271.         for ($x = 0; $x < 1000000; $x++) {
```

```
272. if ($x == 990000) {
273.   $arrayToJs[1] = false;
274.   echo json_encode($arrayToJs);
275. }
276. }
277. }
278. }
279. function status($id_status, $id_spbu) {
280.   if ($this->data['akses'] == 1) {
281.     $sukses = $this->spbu_model->change_status($id_status, $id_spbu);
282.     if ($sukses == true) {
283.       $this->session->set_userdata('message', '1,Status spbu telah diubah.');
```



```
284.       redirect('spbu/list_spbu/');
285.     } else {
286.       $this->session->set_userdata('message', '2,Status spbu gagal diubah.');
```

```
287.       redirect('spbu/list_spbu/');
288.     }
289.   } else {
290.     redirect('home');
291.   }
292. }
293. function icon_spbu($input){
294.   $icon = 1;
295.   for ($i = 0; $i < count($input); $i++) {
296.     if ($input[$i] == 2) {
297.       $icon += 1;
298.     }
299.     if ($input[$i] == 4) {
300.       $icon += 2;
301.     }
302.   }
303.   return $icon;
304. }
305. private function __rule($rule) {
306.   $this->load->library('form_validation');
307.   $this->form_validation->set_rules($rule);
308.   if ($this->form_validation->run()) {
309.     return TRUE;
310.   } else {
311.     return FALSE;
312.   }
313. }
314. // menambahkan rule baru ke dalam peta dan algoritma
315. private function __add_rule() {
316.   $rule = array
317.   (
318.     array
319.     (
320.       'field' => 'nama_spbu'
321.       , 'label' => 'nama spbu'
322.       , 'rules' => 'required'
323.     )
324.     , array
325.     (
326.       'field' => 'lat'
327.       , 'label' => 'latitude'
328.       , 'rules' => 'trim|required'
```

```
329. )
330. , array
331. (
332. 'field' => 'long'
333. , 'label' => 'longitude'
334. , 'rules' => 'trim|required'
335. )
336. , array
337. (
338. 'field' => 'lokasi'
339. , 'label' => 'lokasi'
340. , 'rules' => 'trim|required'
341. )
342. );
343. return $this->__rule($rule);
344. }
345. // fungsi edit rule, ketika spbu diedit
346. private function __edit_rule() {
347. $rule = array
348. (
349. array
350. (
351. 'field' => 'nama_spbu'
352. , 'label' => 'nama spbu'
353. , 'rules' => 'required'
354. )
355. , array
356. (
357. 'field' => 'lat'
358. , 'label' => 'latitude'
359. , 'rules' => 'trim|required'
360. )
361. , array
362. (
363. 'field' => 'long'
364. , 'label' => 'longitude'
365. , 'rules' => 'trim|required'
366. )
367. , array
368. (
369. 'field' => 'lokasi'
370. , 'label' => 'lokasi'
371. , 'rules' => 'trim|required'
372. )
373. );
374. return $this->__rule($rule);
375. }
376. }
377. ?>
```

2. Kelas spbu_model

Penulisan kode program spbu_model dapat dilihat pada Tabel 8.

Tabel 8 Kode Program Kelas spbu_model

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

```

56. and st_distance_spheroid(geom, st_setsrid(st_makepoint($input[x],$input[y]), 4326),
57. 'SPHEROID[\"WGS84\",6378137,298.25728])/1000 <= $input[radius] $kategori
58. $produk $layanan
59. order by st_distance_spheroid(geom, st_setsrid(st_makepoint($input[x],$input[y]),
60. 4326), 'SPHEROID[\"WGS84\",6378137,298.25728])/1000 asc");
61. return $data->result_array();
62. }
63. // menampilkan semua koordinat spbu
64. function get_all_spbu($input) {
65. $kategori = $this->category($input['kategori']);
66. $produk = $this->product($input['produk']);
67. $layanan = $this->layanannya($input['layanannya']);
68. $data = $this->db->query("select distinct(tb_szu.id_spbu), nama_spbu, x, y,
69. tb_spbu.id_kategori, gambar, id_icon, st_asgeojson(geom) as geojson
70. from tb_spbu, tb_spbuKategori, tb_spbuBBM, tb_spbuProduk
71. where tb_spbu.id_kategori = tb_spbuKategori.id_kategori
72. and tb_spbu.id_spbu = tb_spbuProduk.id_spbu
73. and tb_spbuBBM.id_BBM = tb_spbuProduk.id_BBM
74. and tb_spbu.id_status = 2
75. $kategori $produk $layanannya");
76. return $data->result_array();
77. }
78. function search($input) {
79. $kategori = $this->category($input['kategori']);
80. $produk = $this->product($input['produk']);
81. $layanan = $this->layanannya($input['layanannya']);
82. $data = $this->db->query("select distinct(tb_spbu.id_spbu), nama_spbu, x, y,
83. tb_spbu.id_kategori, gambar, id_icon, st_asgeojson(geom) as geojson
84. from tb_spbu, tb_spbuKategori, tb_spbuBBM, tb_spbuProduk
85. where tb_spbu.id_kategori = tb_spbuKategori.id_kategori
86. and tb_spbu.id_spbu = tb_spbuProduk.id_spbu
87. and tb_spbuBBM.id_BBM = tb_spbuProduk.id_BBM
88. and tb_spbu.id_status = 2
89. and nama_spbu ilike '%$input[keyword]%' $kategori $produk $layanannya");
90. return $data->result_array();
91. }
92. // get
93. function get_list_spbu() {
94. $data = $this->db->query("select * from
95. tb_spbu, tb_spbuKategori, tb_status
96. where tb_spbu.id_kategori = tb_spbuKategori.id_kategori
97. and tb_spbu.id_status = tb_status.id_status
98. order by id_spbu desc");
99. return $data->result_array();
100. }
101. function get_spbu_by($id) {
102. $data = $this->db->query("select *
103. from tb_spbu, tb_spbuKategori, tb_spbulayanan
104. where tb_spbu.id_kategori = tb_spbuKategori.id_kategori
105. and tb_spbu.id_layanan = tb_spbulayanan.id_layanan
106. and id_spbu = $id");
107. return $data->row_array();
108. }
109. function get_spbu_form() {
110. $data = $this->db->query("select * from tb_spbu");
111. return $data->result_array();
112. }

```

```
113. function get_spbu_detail($id) {
114.     $data = $this->db->query("select *
115.     from tb_spbu, tb_spbuKategori, tb_spbulayanan
116.     where tb_spbu.id_kategori = tb_spbuKategori.id_kategori
117.     and tb_spbu.id_layanan = tb_spbulayanan.id_layanan
118.     and id_spbu = $id");
119.     return $data->row_array();
120. }
121. function get_spbu_produk($id) {
122.     $data = $this->db->query("select *
123.     from tb_spbuProduk, tb_spbuBBM
124.     where tb_spbuProduk.id_BBM = tb_spbuBBM.id_BBM
125.     and id_spbu = $id");
126.     return $data->result_array();
127. }
128. function get_spbu_fasilitas($id) {
129.     $data = $this->db->query("select *
130.     from tb_spbufasilitas, tb_fasilitas
131.     where tb_spbufasilitas.id_fasilitas = tb_fasilitas.id_fasilitas
132.     and id_spbu = $id");
133.     return $data->result_array();
134. }
135. function get_BBM() {
136.     return $this->db->query("select * from tb_spbuBBM")->result_array();
137. }
138. function get_layanan() {
139.     return $this->db->query("select * from tb_spbulayanan")->result_array();
140. }
141.
142. function get_fasilitas() {
143.     return $this->db->query("select * from tb_fasilitas")->result_array();
144. }
145. function get_kategori() {
146.     return $this->db->query("select * from tb_spbuKategori")->result_array();
147. }
148. function get_rows_spbu() {
149.     return $this->db->query("select id_spbu from tb_spbu")->num_rows();
150. }
151. function get_new_id() {
152.     $data = $this->db->query("select max(id_spbu) as id from tb_spbu");
153.     $last = $data->row_array();
154.     if ($last['id'] == 0 || $last['id'] == NULL) {
155.         return 1;
156.     } else {
157.         return ($last['id'] + 1);
158.     }
159. }
160. // cek
161. function cek_namaspbu($nama){
162.     $data = $this->db->query("select * from tb_spbu where nama_spbu =
163.     '".strtoupper($nama)."'");
164.     if($data->num_rows() > 0){
165.         return FALSE;
166.     } else{
167.         return TRUE;
168.     }
169. }
```

```
170. function cek_transaksi($id_spbu){
171.   $data1 = $this->db->query("select * from tb_rating where id_spbu = $id_spbu");
172.   $data2 = $this->db->query("select * from tb_review where id_spbu = $id_spbu");
173.   if($data1->num_rows() > 0 || $data2->num_rows() > 0){
174.     return FALSE;
175.   } else{
176.     return TRUE;
177.   }
178. }
179. function __is_exist($id) {
180.   $this->db->where('id_spbu', $id);
181.   $data = $this->db->get('tb_spbu');
182.   if ($data->num_rows() > 0) {
183.     return TRUE;
184.   } else {
185.     return FALSE;
186.   }
187. }
188. // set
189. function add_spbu($input) {
190.   $new_id = $this->get_new_id();
191.   $user = $this->session->userdata('spbu_iduser');
192.   $this->db->trans_begin();
193.   if ($input['gambar'] == 1) {
194.     $this->db->query("
195.     insert into tb_spbu (
196.     id_spbu,
197.     nama_spbu,
198.     lokasi_spbu,
199.     id_kategori,
200.     id_layanan,
201.     keterangan,
202.     id_user,
203.     id_status,
204.     tanggal_tambah,
205.     jumlah_pompa,
206.     id_icon,
207.     x,
208.     y,
209.     geom
210.     ) values (
211.     "" . $new_id . "",
212.     "" . $input['nama_spbu'] . "",
213.     "" . $input['lokasi'] . "",
214.     "" . $input['id_kategori'] . "",
215.     "" . $input['id_layanan'] . "",
216.     "" . $input['keterangan'] . "",
217.     "" . $user . "",
218.     2,
219.     now(),
220.     $input[jumlah_pompa],
221.     $input[id_icon],
222.     $input[long],
223.     $input[lat],
224.     ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
225.     )
226.     ");
```

```
227.     } else {
228.     $this->db->query("
229.     insert into tb_spbu (
230.     id_spbu,
231.     nama_spbu,
232.     lokasi_spbu,
233.     id_kategori,
234.     id_layanan,
235.     keterangan,
236.     id_user,
237.     id_status,
238.     gambar,
239.     tanggal_tambah,
240.     jumlah_pompa,
241.     id_icon,
242.     x,
243.     y,
244.     geom
245.     ) values (
246.     "" . $new_id . "",
247.     "" . $input['nama_spbu'] . "",
248.     "" . $input['lokasi'] . "",
249.     "" . $input['id_kategori'] . "",
250.     "" . $input['id_layanan'] . "",
251.     "" . $input['keterangan'] . "",
252.     "" . $user . "",
253.     2,
254.     "" . $input['gambar'] . "",
255.     now(),
256.     $input[jumlah_pompa],
257.     $input[id_icon],
258.     $input[long],
259.     $input[lat],
260.     ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
261.     )
262.     ");
263.     }
264.     for ($i = 0; $i < count($input['produk']); $i++) {
265.     $id_BBM = $input['produk'][$i];
266.     if ($id_BBM != 0) {
267.     $this->db->query("insert into tb_spbuProduk (id_spbu, id_BBM) values ($new_id,
268.     $id_BBM)");
269.     }
270.     }
271.     for ($i = 0; $i < count($input['fasilitas']); $i++) {
272.     $id_fasilitas = $input['fasilitas'][$i];
273.     if ($id_fasilitas != 0) {
274.     $this->db->query("insert into tb_spbufasilitas (id_spbu, id_fasilitas) values ($new_id,
275.     $id_fasilitas)");
276.     }
277.     }
278.     if ($this->db->trans_status() === FALSE) {
279.     $this->db->trans_rollback();
280.     return FALSE;
281.     } else {
282.     $this->db->trans_commit();
283.     return TRUE;
```



```
284.     }
285.     }
286.     function edit_spbu($input) {
287.     $this->db->trans_begin();
288.     if ($input['gambar'] == 1) {
289.     $this->db->query("
290.     update tb_spbu set
291.     nama_spbu = '$input[nama_spbu]',
292.     lokasi_spbu = '$input[lokasi]',
293.     id_kategori = $input[id_kategori],
294.     id_layanan = $input[id_layanan],
295.     keterangan = '$input[keterangan]',
296.     jumlah_pompa = $input[jumlah_pompa],
297.     id_icon = $input[id_icon],
298.     x = $input[long],
299.     y = $input[lat],
300.     geom = ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
301.     where id_spbu = $input[id_spbu]
302.     ");
303.     } else {
304.     $this->db->query("
305.     update tb_spbu set
306.     nama_spbu = '$input[nama_spbu]',
307.     lokasi_spbu = '$input[lokasi]',
308.     id_kategori = $input[id_kategori],
309.     id_layanan = $input[id_layanan],
310.     keterangan = '$input[keterangan]',
311.     jumlah_pompa = $input[jumlah_pompa],
312.     gambar = '$input[gambar]',
313.     id_icon = $input[id_icon],
314.     x = $input[long],
315.     y = $input[lat],
316.     geom = ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
317.     where id_spbu = $input[id_spbu]
318.     ");
319.     }
320.     $this->db->query("delete from tb_spbuProduk where id_spbu = $input[id_spbu]");
321.     for ($i = 0; $i < count($input['produk']); $i++) {
322.     $id_BBM = $input['produk'][$i];
323.     if ($id_BBM != 0) {
324.     $this->db->query("insert into tb_spbuProduk (id_spbu, id_BBM) values
325.     ($input[id_spbu], $id_BBM)");
326.     }
327.     }
328.     $this->db->query("delete from tb_spbufasilitas where id_spbu = $input[id_spbu]");
329.     for ($i = 0; $i < count($input['fasilitas']); $i++) {
330.     $id_fasilitas = $input['fasilitas'][$i];
331.     if ($id_fasilitas != 0) {
332.     $this->db->query("insert into tb_spbufasilitas (id_spbu, id_fasilitas) values
333.     ($input[id_spbu], $id_fasilitas)");
334.     }
335.     }
336.     if ($this->db->trans_status() === FALSE) {
337.     $this->db->trans_rollback();
338.     return FALSE;
339.     } else {
340.     $this->db->trans_commit();
```

```

341. return TRUE;
342. }
343. }
344. function change_status($id_status, $id_spbu) {
345. $cek = $this->db->query("update tb_spbu set id_status = $id_status where id_spbu =
346. $id_spbu");
347. if ($cek == TRUE) {
348. return TRUE;
349. } else {
350. return FALSE;
351. }
352. }
353. function delete_spbu($id) {
354. if ($this->__is_exist($id)) {
355. $this->db->trans_begin();
356. $this->db->query("delete from tb_spbufasilitas where id_spbu = $id");
357. $this->db->query("delete from tb_spbuproduk where id_spbu = $id");
358. $this->db->query("delete from tb_spbu where id_spbu = $id");
359. if ($this->db->trans_status() === FALSE) {
360. $this->db->trans_rollback();
361. return FALSE;
362. } else {
363. $this->db->trans_commit();
364. return TRUE;
365. }
366. } else {
367. return FALSE;
368. }
369. }
370. function latlong(){
371. $data = $this->db->query("select x as lng, y as lat, nama_spbu from tb_spbu");
372. return $data->result_array();
373. }
}
?>

```

D.2 Kode Program Melihat Data Distribusi BBM

Kode program melihat data distribusi BBM terletak pada kelas distribusi dan distribusi_model.

1. Kelas distribusi

Penulisan kode program distribusi dapat dilihat pada Tabel 9.

Tabel 9 Kode Program Kelas distribusi

1.	<?php
2.	if (!defined('BASEPATH'))
3.	exit('No direct script access allowed');
4.	class Distribusi extends CI_Controller {
5.	private \$data;

```
6. function __construct() {
7.     parent::__construct();
8.     $this->load->model('spbu_model');
9.     $this->load->model('akun_model');
10.    $this->load->model('routing_model');
11.    $this->load->model('distribusi_model');
12.    $this->data = array(
13.        'title' => NULL,
14.        'content' => NULL,
15.        'menu' => NULL,
16.        'akses' => $this->session->userdata('spbu_hakakses'),
17.        'username' => $this->session->userdata('spbu_username'),
18.    );
19.    }
20.    function index() {
21.        $this->list_distribusi();
22.    }
23.    function cobadistribusi(){
24.        $this->data['menu'] = 1;
25.        $this->data['title'] = 'Manajemen Distribusi';
26.        $this->data['content'] = $this->load->view('distribusi_list',array(
27.            'data' => $this->distribusi_model->get_list_distribusi()
28.        ),TRUE);
29.        $this->load->view('template_baru',$this->data);
30.    }
31.    function cobaedit(){
32.        $this->data['menu'] = 1;
33.        $this->data['title'] = 'Manajemen Distribusi';
34.        $this->data['content'] = $this->load->view('distribusi_edit',array(
35.            'data' => $this->distribusi_model->get_list_distribusi()
36.        ),TRUE);
37.        $this->load->view('template_baru',$this->data);
38.    }
39.    function add(){
40.        $this->data['menu'] = 1;
41.        $this->data['title'] = 'Tambah Distribusi';
42.        $this->data['content'] = $this->load->view('distribusi_add',array(
43.            'produk' => $this->spbu_model->get_BBM(),
44.            'kategori' => $this->spbu_model->get_kategori(),
45.            'fasilitas' => $this->spbu_model->get_fasilitas(),
46.            'layanan' => $this->spbu_model->get_layanan(),
47.            'spbu' => $this->spbu_model->get_semua()
48.        ),TRUE);
49.        $this->load->view('template_baru',$this->data);
50.    }
51.    function nearest_node() {
52.        $data = array(
53.            'x' => $this->input->post('x'),
54.            'y' => $this->input->post('y')
55.        );
56.        $result = $this->routing_model->get_nearest_coordinate($data);
57.        $result2 = $this->routing_model->get_nearest_network($data);
58.        if ($result2['osm_name'] != "") {
59.            $nama_jalan = $result2['osm_name'];
60.        } else {
61.            $nama_jalan = 'No name';
62.        }
}
```

```
63. echo $result['lat'] . ',' . $result['lng'];
64. echo '|';
65. echo $nama_jalan;
66. }
67. function route() {
68. $data = array(
69. 'x1' => $this->input->post('x1'),
70. 'y1' => $this->input->post('y1'),
71. 'x2' => $this->input->post('x2'),
72. 'y2' => $this->input->post('y2'),
73. 'kendaraan' => $this->input->post('kendaraan'),
74. 'jalur' => $this->input->post('jalur'),
75. 'id' => $this->input->post('id')
76. );
77. $rs = $this->routing_model->get_route_geojson($data);
78. $output = "";
79. $rowOutput = "";
80. foreach ($rs as $row) {
81. $rowOutput = (strlen($rowOutput) > 0 ? ',' : '') . '{"type": "Feature", "geometry": ' .
82. $row['geojson'] . ', "properties": {';
83. $props = "";
84. $id = "";
85. foreach ($row as $key => $val) {
86. if ($key != "geojson") {
87. $props .= (strlen($props) > 0 ? ',' : '') . '"" . $key . '":' . $this->escapeJsonString($val) .
88. '""';
89. }
90. if ($key == "id") {
91. $id .= ',"id":' . $this->escapeJsonString($val) . '""';
92. }
93. }
94. $rowOutput .= $props . ' }';
95. $rowOutput .= $id;
96. $rowOutput .= ' }';
97. $output .= $rowOutput;
98. }
99. $output = '{ "type": "FeatureCollection", "features": [ ' . $output . ' ] }';
100. echo $output;
101. }
102. function add_process(){
103. $data = array(
104. 'id_spbu' => $this->input->post('id_spbu'),
105. 'id_BBM' => $this->input->post('id_BBM'),
106. 'volume' => $this->input->post('volume'),
107. 'tanggal_dist' => $this->input->post('tanggal_dist'),
108. 'jam_dist' => $this->input->post('jam_dist'),
109. 'Sopir' => $this->input->post('Sopir')
110. );
111. $data2 = array(
112. 'lat' => $this->input->post('lat'),
113. 'long' => $this->input->post('long')
114. );
115. $hasil = $this->distribusi_model->add($data);
116. if ($hasil) {
117. redirect('home/mapsSopir');
118. } else {
119. echo 'gagal';
```

```

120. }
121. }
122. function edit($id) {
123. if ($this->data['akses'] != NULL) {
124. $this->data['title'] = 'Edit SPBU';
125. $this->data['content'] = $this->load->view('distribusi_edit',
126. 'data' => $this->distribusi_model->get_distribusi_edit($id));
127. $this->load->view('template_baru', $this->data);
128. } else {
129. redirect('home');
130. }
131. }
132. function delete_process($id) {
133. if ($this->data['akses'] != NULL) {
134. $sukses = $this->distribusi_model->delete_distribusi($id);
135. if ($sukses == true) {
136. $this->session->set_userdata('message', '1,SPBU telah dihapus. ');
137. redirect('home/cobadistribusi');
138. }
139. } else {
140. redirect('home');
141. }
142. }
143. }

```

2. Kelas distribusi_model

Penulisan kode program distribusi_model dapat dilihat pada Tabel 10.

Tabel 10 Kode Program Kelas distribusi_model

```

1. <?php
2. if (!defined('BASEPATH'))
3. exit('No direct script access allowed');
4. class Distribusi_Model extends CI_Model {
5. function __construct() {
6. parent::__construct();
7. }
8. function get_rows_distribusi() {
9. return $this->db->query("select id_distribusi from tb_distribusi")->num_rows();
10. }
11. function get_list_distribusi() {
12. $data = $this->db->query("select * from
13. tb_distribusi a join tb_spbu b on (a.id_spbu = b.id_spbu) join tb_spbuBBM c
14. on(a.id_BBM=c.id_BBM)
15. order by id_distribusi desc
16. ");
17. return $data->result_array();
18. }
19. function add($data){
20. $data = $this->db->query("INSERT INTO public.tb_distribusi(
21. id_spbu, id_BBM, volume, tanggal_dist, jam_dist,Sopir)
22. VALUES (
23. "" . $data['id_spbu'] . "",

```

```

24.     "" . $data['id_BBM'] . "",
25.     "" . $data['volume'] . "",
26.     "" . $data['tanggal_dist'] . "",
27.     "" . $data['jam_dist'] . "",
28.     "" . $data['Sopir'] . ""));
29.     return $data;
30.     }
31.     function get_distribusi_edit($id) {
32.     $data = $this->db->query("select *
33.     from tb_distribusi where id_distribusi= $id");
34.     return $data->row_array();
35.     }
36.     function jumlah(){
37.     return $this->db->query ("select COUNT(id_distribusi) as jumlah from tb_distribusi");
38.     }
39.     }

```

D.3 Kode Program Pengelolaan Data Akun

Kode program pengelolaan data akun 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

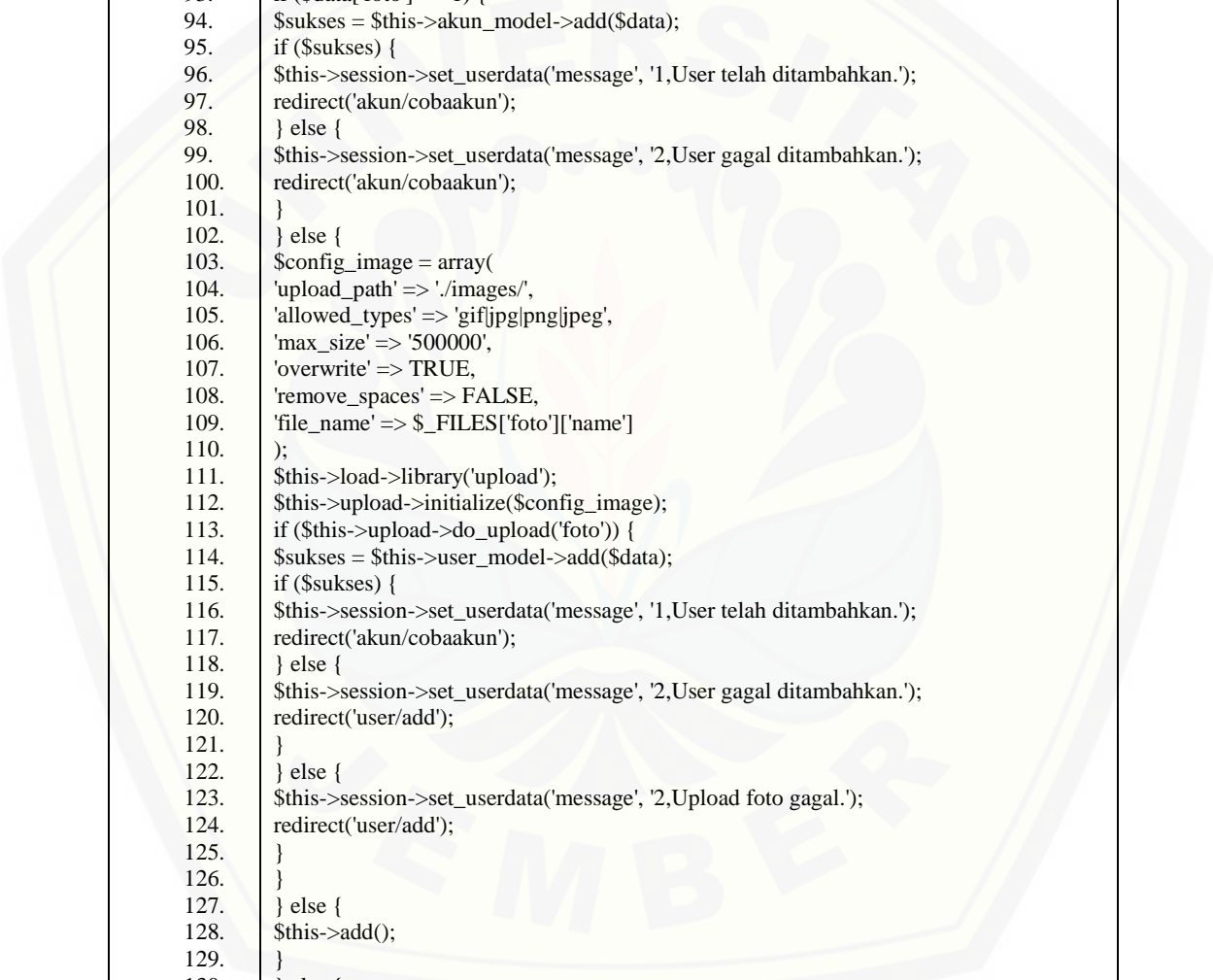
```

1.     <?php
2.     if (!defined('BASEPATH'))
3.     exit('No direct script access allowed');
4.     class Akun extends CI_Controller {
5.     private $data;
6.     function __construct() {
7.     parent::__construct();
8.     if ($this->session->userdata('spbu_hakakses') == NULL) {
9.     redirect('home');
10.    }
11.    $this->load->model('akun_model');
12.    $this->data = array(
13.    'title' => NULL,
14.    'content' => NULL,
15.    'menu' => NULL,
16.    'akses' => $this->session->userdata('spbu_hakakses'),
17.    'username' => $this->session->userdata('spbu_username'),
18.    );
19.    }
20.    function cobaakun(){
21.    $this->data['menu'] = 1;
22.    $this->data['title'] = 'Manajemen Akun';
23.    $this->data['content'] = $this->load->view('akun_list',array(
24.    'data' => $this->akun_model->get_list_user()

```

```
25.     ),TRUE);
26.     $this->load->view('template_baru',$this->data);
27.     }
28.     function profile() {
29.     if ($this->data['akses'] != NULL) {
30.     $this->data['title'] = 'Profil';
31.     $this->data['content'] = $this->load->view(
32.     'akun_detail', array('data' => $this->akun_model->get_user_detail($this->session-
33.     >userdata('spbu_iduser'))), TRUE);
34.     $this->load->view('template', $this->data);
35.     } else {
36.     redirect('home');
37.     }
38.     }
39.     function detail($id_user) {
40.     if ($this->data['akses'] != NULL) {
41.     $this->data['title'] = 'Detail Akun';
42.     $this->data['content'] = $this->load->view(
43.     'akun_detail', array('data' => $this->akun_model->get_user_detail($id_user)), TRUE);
44.     $this->load->view('template_baru', $this->data);
45.     } else {
46.     redirect('home');
47.     }
48.     }
49.     function add() {
50.     if ($this->data['akses'] == 1) {
51.     $this->data['title'] = 'Tambah Akun';
52.     $this->data['content'] = $this->load->view(
53.     'akun_add', array(
54.     'akses' => $this->akun_model->get_akses(), TRUE);
55.     $this->load->view('template_baru', $this->data);
56.     } else {
57.     redirect('home/cobaakun');
58.     }
59.     }
60.     function edit($id_user) {
61.     if ($this->data['akses'] != NULL) {
62.     $this->data['title'] = 'Edit Akun';
63.     $this->data['content'] = $this->load->view(
64.     'akun_edit', array(
65.     'kode' => NULL,
66.     'akses' => $this->akun_model->get_akses(),
67.     'data' => $this->akun_model->get_user_by($id_user), TRUE);
68.     $this->load->view('template_baru', $this->data);
69.     } else {
70.     redirect('home');
71.     }
72.     }
73.     function add_process() {
74.     if ($this->data['akses'] != NULL) {
75.     if ($this->__add_rule()) {
76.     if ($_FILES['foto']['name'] == "") {
77.     $foto = 1;
78.     } else {
79.     $foto = $_FILES['foto']['name'];
80.     }
81.     $data = array(
```

```
82. 'id_user' => "",
83. 'username' => $this->input->post('username'),
84. 'password' => $this->input->post('password'),
85. 'nama_lengkap' => $this->input->post('nama_lengkap'),
86. 'id_jeniskelamin' => $this->input->post('jenis_kelamin'),
87. 'no_telepon' => $this->input->post('no_telepon'),
88. 'email' => $this->input->post('email'),
89. 'alamat' => $this->input->post('alamat'),
90. 'id_hakakses' => $this->input->post('hak_akses'),
91. 'foto' => $foto
92. );
93. if ($data['foto'] == 1) {
94. $sukses = $this->akun_model->add($data);
95. if ($sukses) {
96. $this->session->set_userdata('message', '1,User telah ditambahkan.');
```



```
97. redirect('akun/cobaakun');
98. } else {
99. $this->session->set_userdata('message', '2,User gagal ditambahkan.');
```

```
100. redirect('akun/cobaakun');
101. }
102. } else {
103. $config_image = array(
104. 'upload_path' => './images/',
105. 'allowed_types' => 'gif|jpg|png|jpeg',
106. 'max_size' => '500000',
107. 'overwrite' => TRUE,
108. 'remove_spaces' => FALSE,
109. 'file_name' => $_FILES['foto']['name']
110. );
111. $this->load->library('upload');
112. $this->upload->initialize($config_image);
113. if ($this->upload->do_upload('foto')) {
114. $sukses = $this->user_model->add($data);
115. if ($sukses) {
116. $this->session->set_userdata('message', '1,User telah ditambahkan.');
```

```
117. redirect('akun/cobaakun');
118. } else {
119. $this->session->set_userdata('message', '2,User gagal ditambahkan.');
```

```
120. redirect('user/add');
121. }
122. } else {
123. $this->session->set_userdata('message', '2,Upload foto gagal.');
```

```
124. redirect('user/add');
125. }
126. }
127. } else {
128. $this->add();
129. }
130. } else {
131. redirect('home');
```

```
132. }
133. }
134. function edit_process($kode = NULL) {
135. if ($this->data['akses'] != NULL) {
136. if ($this->__edit_rule()) {
137. if ($_FILES['foto']['name'] == "") {
138. $foto = 1;
```



```
139.     } else {
140.     $foto = $_FILES['foto']['name'];
141.     }
142.     $data = array(
143.     'id_user' => $this->input->post('id_user'),
144.     'nama_lengkap' => $this->input->post('nama_lengkap'),
145.     'id_jeniskelamin' => $this->input->post('jenis_kelamin'),
146.     'no_telepon' => $this->input->post('no_telepon'),
147.     'email' => $this->input->post('email'),
148.     'alamat' => $this->input->post('alamat'),
149.     'id_hakakses' => $this->input->post('hak_akses'),
150.     'foto' => $foto
151.     );
152.
153.     if ($data['foto'] == 1) {
154.     $sukses = $this->akun_model->edit($data);
155.     if ($kode == NULL) {
156.     if ($sukses) {
157.     $this->session->set_userdata('message', '1,User telah diupdate. ');
158.     redirect('akun/cobaakun' . $this->input->post('id_user'));
159.     } else {
160.     $this->session->set_userdata('message', '2,User gagal diupdate. ');
161.     redirect('akun/cobaakun' . $this->input->post('id_user'));
162.     }
163.     } else {
164.     if ($sukses) {
165.     $this->session->set_userdata('message', '1,Profil telah diupdate. ');
166.     redirect('akun/edit_account/');
167.     } else {
168.     $this->session->set_userdata('message', '2,Profil gagal diupdate. ');
169.     redirect('akun/edit_account/');
170.     }
171.     }
172.     } else {
173.     $config_image = array(
174.     'upload_path' => './images/',
175.     'allowed_types' => 'gif|jpg|png|jpeg',
176.     'max_size' => '500000',
177.     'overwrite' => TRUE,
178.     'remove_spaces' => FALSE,
179.     'file_name' => $_FILES['foto']['name']
180.     );
181.     $this->load->library('upload');
182.     $this->upload->initialize($config_image);
183.     if ($this->upload->do_upload('foto')) {
184.     $sukses = $this->user_model->edit($data);
185.     if ($kode == NULL) {
186.     if ($sukses) {
187.     $this->session->set_userdata('message', '1,User telah diupdate. ');
188.     redirect('akun/cobaakun' . $this->input->post('id_user'));
189.     } else {
190.     $this->session->set_userdata('message', '2,User gagal diupdate. ');
191.     redirect('akun/cobaakun' . $this->input->post('id_user'));
192.     }
193.     } else {
194.     if ($sukses) {
195.     $this->session->set_userdata('message', '1,Profil telah diupdate. ');
```

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



```
199. redirect('akun/edit_account/');
200. }
201. }
202. } else {
203. $this->session->set_userdata('message', '2,Upload foto gagal.');
```

```
204. if ($kode == NULL) {
205. redirect('akun/edit/' . $this->input->post('id_user'));
206. } else {
207. redirect('akun/edit_account/');
```

```
208. }
209. }
210. }
211. } else {
212. if ($kode == NULL) {
213. $this->edit($this->input->post('id_user'));
214. } else {
215. $this->edit_account();
216. }
217. }
218. } else {
219. redirect('home');
```

```
220. }
221. }
222. function edit_password_process() {
223. if ($this->data['akses'] != NULL) {
224. $data = array(
225. 'id_user' => $this->input->post('id_user'),
226. 'username' => $this->input->post('username'),
227. 'password' => $this->input->post('password'),
228. 'password_baru' => $this->input->post('password_baru'),
229. 'password_konfirmasi' => $this->input->post('password_konfirmasi'),
230. );
231. $is_password = $this->user_model->__is_password($data['id_user'], $data['password']);
232. if ($is_password) {
233. if ($data['password_baru'] == $data['password_konfirmasi']) {
234. $sukses = $this->user_model->edit_account($data);
235. if ($sukses) {
236. $this->session->set_userdata('message', '1,Password telah diupdate.');
```

```
237. redirect('akun/edit_akun/');
238. } else {
239. $this->session->set_userdata('message', '2,Password gagal diupdate.');
```

```
240. redirect('akun/edit_akun/');
241. }
242. } else {
243. $this->session->set_userdata('message', '2,Konfirmasi password salah.');
```

```
244. redirect('akun/edit_akun/');
245. }
246. } else {
247. $this->session->set_userdata('message', '2,Password salah.');
```

```
248. redirect('akun/edit_akun/');
249. }
250. } else {
251. redirect('home');
```

```
252. }
```

```
253.     }
254.     function delete_process($id_user) {
255.     if ($this->data['akses'] == 1) {
256.     if ($this->akun_model->cek_idhakakses($id_user) != 1) {
257.     if ($this->akun_model->cek_transaksi($id_user) == TRUE) {
258.     $sukses = $this->akun_model->delete($id_user);
259.     if ($sukses == true) {
260.     $this->session->set_userdata('message', '1,User telah dihapus. ');
261.     redirect('akun/cobaakun/');
262.     } else {
263.     $this->session->set_userdata('message', '2,User gagal dihapus. ');
264.     redirect('akun/cobaakun/');
265.     }
266.     } else {
267.     $this->session->set_userdata('message', '2,Telah ada transaksi dengan tabel lain user
268.     tidak boleh dihapus. ');
269.     redirect('akun/cobaakun/');
270.     }
271.     } else {
272.     $this->session->set_userdata('message', '2,User administrator tidak boleh dihapus. ');
273.     redirect('akun/cobaakun/');
274.     }
275.     } else {
276.     redirect('home');
277.     }
278.     }
279.     function status($id_status, $id_user) {
280.     if ($this->data['akses'] == 1) {
281.     if ($this->user_model->cek_idhakakses($id_user) != 1) {
282.     $sukses = $this->user_model->change_status($id_status, $id_user);
283.     if ($sukses == true) {
284.     $this->session->set_userdata('message', '1,Status user telah diubah. ');
285.     redirect('user/list_user/');
286.     } else {
287.     $this->session->set_userdata('message', '2,Status user gagal diubah. ');
288.     redirect('user/list_user/');
289.     }
290.     } else {
291.     $this->session->set_userdata('message', '2,User administrator tidak boleh
292.     dinontaktifkan. ');
293.     redirect('user/list_user/');
294.     }
295.     } else {
296.     redirect('home');
297.     }
298.     }
299.     private function __rule($rule) {
300.     $this->load->library('form_validation');
301.     $this->form_validation->set_rules($rule);
302.     if ($this->form_validation->run()) {
303.     return TRUE;
304.     } else {
305.     return FALSE;
306.     }
307.     }
308.     private function __add_rule() {
309.     $rule = array
```

```
310. (
311. array
312. (
313. 'field' => 'username'
314. , 'label' => 'Username'
315. , 'rules' => 'required'
316. )
317. , array
318. (
319. 'field' => 'password'
320. , 'label' => 'Password'
321. , 'rules' => 'required'
322. )
323. , array
324. (
325. 'field' => 'nama_lengkap'
326. , 'label' => 'Nama Lengkap'
327. , 'rules' => 'required'
328. )
329. , array
330. (
331. 'field' => 'jenis_kelamin'
332. , 'label' => 'Jenis Kelamin'
333. , 'rules' => 'required'
334. )
335. , array
336. (
337. 'field' => 'alamat'
338. , 'label' => 'Alamat'
339. , 'rules' => 'required'
340. )
341. , array
342. (
343. 'field' => 'hak_akses'
344. , 'label' => 'Hak Akses'
345. , 'rules' => 'required'
346. )
347. , array
348. (
349. 'field' => 'no_telepon'
350. , 'label' => 'No Telepon'
351. , 'rules' => 'trim|required|numeric'
352. )
353. , array
354. (
355. 'field' => 'email'
356. , 'label' => 'Email'
357. , 'rules' => 'required|email'
358. )
359. );
360. return $this->__rule($rule);
361. }
362. private function __edit_rule() {
363. $rule = array
364. (
365. array
366. (
```

```
367.     'field' => 'nama_lengkap'
368.     , 'label' => 'Nama Lengkap'
369.     , 'rules' => 'required'
370.     )
371.     , array
372.     (
373.     'field' => 'jenis_kelamin'
374.     , 'label' => 'Jenis Kelamin'
375.     , 'rules' => 'required'
376.     )
377.     , array
378.     (
379.     'field' => 'alamat'
380.     , 'label' => 'Alamat'
381.     , 'rules' => 'required'
382.     )
383.     , array
384.     (
385.     'field' => 'hak_akses'
386.     , 'label' => 'Hak Akses'
387.     , 'rules' => 'required'
388.     )
389.     , array
390.     (
391.     'field' => 'no_telepon'
392.     , 'label' => 'No Telepon'
393.     , 'rules' => 'trim|required|numeric'
394.     )
395.     , array
396.     (
397.     'field' => 'email'
398.     , 'label' => 'Email'
399.     , 'rules' => 'required|email'
400.     )
401. );
402. return $this->__rule($rule);
403. }
404. private function __edit_akun_rule() {
405.     $rule = array
406.     (
407.     array
408.     (
409.     'field' => 'password'
410.     , 'label' => 'Old Password'
411.     , 'rules' => 'required'
412.     )
413.     , array
414.     (
415.     'field' => 'password_baru'
416.     , 'label' => 'New Password'
417.     , 'rules' => 'required'
418.     )
419.     , array
420.     (
421.     'field' => 'password_konfirmasi'
422.     , 'label' => 'Password Confirm'
423.     , 'rules' => 'required'
```

424.)
425.);
426.	return \$this->__rule(\$rule);
427.	}
428.	}
429.	?>

2. Kelas akun_model

Penulisan kode program akun_model dapat dilihat pada Tabel 12.

Tabel 12 Kode Program Kelas akun_model

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

```
42. $last = $data->row_array();
43. if ($last['id'] == 0 || $last['id'] == NULL) {
44.     return 1;
45. } else {
46.     return ($last['id'] + 1);
47. }
48. }
49. // cek
50. function cek_email($email){
51.     $data = $this->db->query("select * from tb_user where email = '$email'");
52.     if($data->num_rows() > 0){
53.         return FALSE;
54.     } else{
55.         return TRUE;
56.     }
57. }
58. function cek_transaksi($id_user){
59.     $data1 = $this->db->query("select * from tb_rating where id_user = $id_user");
60.     $data2 = $this->db->query("select * from tb_review where id_user = $id_user");
61.     $data3 = $this->db->query("select * from tb_lalulintas where id_user = $id_user");
62.     if($data1->num_rows() > 0 || $data2->num_rows() > 0 || $data3->num_rows() > 0){
63.         return FALSE;
64.     } else{
65.         return TRUE;
66.     }
67. }
68. function cek_idhakakses($id_user){
69.     $data = $this->db->query("select id_hakakses from tb_user where id_user = $id_user");
70.     return $data->row()->id_hakakses;
71. }
72. function __is_exist($id_user) {
73.     $this->db->where('id_user', $id_user);
74.     $data = $this->db->get('tb_user');
75.     if ($data->num_rows() > 0) {
76.         return TRUE;
77.     } else {
78.         return FALSE;
79.     }
80. }
81. function __is_available($username) {
82.     $data = $this->db->query("select * from tb_user where username = '$username'");
83.     if ($data->num_rows() > 0) {
84.         return FALSE;
85.     } else {
86.         return TRUE;
87.     }
88. }
89. function __is_password($id_user, $password) {
90.     $data = $this->db->query("select * from tb_user where id_user = $id_user and password
91. = '' . md5($password) . ''");
92.     if ($data->num_rows() > 0) {
93.         return TRUE;
94.     } else {
95.         return FALSE;
96.     }
97. }
98. // set
```

```
99. function add($user) {
100.   if ($this->__is_available($user['username'])) {
101.     $id = $this->get_new_id();
102.     if ($user['foto'] == 1) {
103.       $this->db->query("
104.         insert into tb_user (
105.           id_user,
106.           username,
107.           password,
108.           nama_lengkap,
109.           alamat,
110.           email,
111.           no_telepon,
112.           tgl_register,
113.           id_jeniskelamin,
114.           id_hakakses,
115.           id_status
116.         ) values (
117.           '$id',
118.           '$user[username]' . "",
119.           '$user[password]' . "",
120.           '$user[nama_lengkap]' . "",
121.           '$user[alamat]' . "",
122.           '$user[email]' . "",
123.           '$user[no_telepon]' . "",
124.           now(),
125.           $user[id_jeniskelamin],
126.           $user[id_hakakses],
127.           2
128.         )
129.       ");
130.     } else {
131.       $this->db->query("
132.         insert into tb_user (
133.           id_user,
134.           username,
135.           password,
136.           nama_lengkap,
137.           alamat,
138.           email,
139.           no_telepon,
140.           tgl_register,
141.           id_jeniskelamin,
142.           id_hakakses,
143.           id_status,
144.           foto
145.         ) values (
146.           '$id',
147.           '$user[username]' . "",
148.           '$user[password]' . "",
149.           '$user[nama_lengkap]' . "",
150.           '$user[alamat]' . "",
151.           '$user[email]' . "",
152.           '$user[no_telepon]' . "",
153.           now(),
154.           $user[id_jeniskelamin],
155.           $user[id_hakakses],
```



```
156.     2,  
157.     "" . $user['foto'] . ""  
158.     )  
159.     ");  
160.     }  
161.     return TRUE;  
162.     } else {  
163.     return FALSE;  
164.     }  
165.     }  
166.     function edit($user) {  
167.     if ($this->__is_exist($user['id_user'])) {  
168.     if ($user['foto'] == 1) {  
169.     $this->db->query(" update tb_user set  
170.     nama_lengkap = '$user[nama_lengkap]',  
171.     alamat = '$user[alamat]',  
172.     email = '$user[email]',  
173.     no_telepon = '$user[no_telepon]',  
174.     id_jeniskelamin = '$user[id_jeniskelamin]',  
175.     id_hakakses = $user[id_hakakses]  
176.     where id_user = $user[id_user]  
177.     ");  
178.     } else {  
179.     $this->db->query(" update tb_user set  
180.     nama_lengkap = '$user[nama_lengkap]',  
181.     alamat = '$user[alamat]',  
182.     email = '$user[email]',  
183.     no_telepon = '$user[no_telepon]',  
184.     id_jeniskelamin = '$user[id_jeniskelamin]',  
185.     id_hakakses = $user[id_hakakses],  
186.     foto = '$user[foto]'  
187.     where id_user = $user[id_user]  
188.     ");  
189.     }  
190.     return TRUE;  
191.     } else {  
192.     return FALSE;  
193.     }  
194.     }  
195.     function edit_akun($user) {  
196.     if ($this->__is_exist($user['id_user'])) {  
197.     $this->db->query("update tb_user set  
198.     password = "" . md5($user['password_baru']) . ""  
199.     where id_user = '$user[id_user]'  
200.     ");  
201.     return TRUE;  
202.     } else {  
203.     return FALSE;  
204.     }  
205.     }  
206.     function delete($id_user) {  
207.     if ($this->__is_exist($id_user)) {  
208.     $this->db->trans_begin();  
209.     $this->db->query("delete from tb_review where id_user = $id_user");  
210.     $this->db->query("delete from tb_lalulintas where id_user = $id_user");  
211.     $this->db->query("delete from tb_rating where id_user = $id_user");  
212.     $this->db->query("delete from tb_user where id_user = $id_user");
```

```

213. if ($this->db->trans_status() === FALSE) {
214.     $this->db->trans_rollback();
215.     return FALSE;
216. } else {
217.     $this->db->trans_commit();
218.     return TRUE;
219. }
220. } else {
221.     return FALSE;
222. }
223. }
224. function change_status($id_status, $id_user) {
225.     $cek = $this->db->query("update tb_user set id_status = $id_status where id_user =
226.     $id_user");
227.     if ($cek === TRUE) {
228.         return TRUE;
229.     } else {
230.         return FALSE;
231.     }
232. }
233. function login($input) {
234.     $data = $this->db->query("select * from tb_user, tb_userHakAkses, tb_status
235.     where tb_user.id_hakAkses = tb_userHakAkses.id_hakAkses
236.     and tb_user.id_status = tb_status.id_status
237.     and tb_user.id_status = 2
238.     and username = '$input[username]'
239.     and password = '" . md5($input['password']) . "'");
240.     if ($data->num_rows() > 0) {
241.         $this->session->set_userdata('spbu_username', $data->row()->username);
242.         $this->session->set_userdata('spbu_nama', $data->row()->nama_lengkap);
243.         $this->session->set_userdata('spbu_hakakses', $data->row()->id_hakakses);
244.         $this->session->set_userdata('spbu_iduser', $data->row()->id_user);
245.         $this->db->query("update tb_user set tgl_login = now() where username =
246.         '$input[username]'");
247.         return TRUE;
248.     } else {
249.         return FALSE;
250.     }
251. }
252. }
253. ?>

```

D.4 Kode Program Melihat SPBU

Kode program melihat SPBU terletak pada spbu dan spbu_model.

1. Kelas spbu

Penulisan kode program spbu dapat dilihat pada Tabel 13.

Tabel 13 Kode Program Kelas spbu

1.	<?php
----	-------

```
2.     if (!defined('BASEPATH'))
3.     exit('No direct script access allowed');
4.     class Spbu extends CI_Controller {
5.     private $data;
6.     function __construct() {
7.     parent::__construct();
8.     $this->load->model('spbu_model');
9.     $this->load->model('akun_model');
10.    $this->data = array(
11.    'title' => NULL,
12.    'content' => NULL,
13.    'menu' => NULL,
14.    'akses' => $this->session->userdata('spbu_hakakses'),
15.    'username' => $this->session->userdata('spbu_username'),
16.    'nama_lengkap' => $this->session->userdata('nama_lengkap')
17.    );
18.    }
19.    // memanggil function list spbu untuk ditampilkan di halaman awal manajemen spbu
20.    function cobaspbu(){
21.    $this->data['menu'] = 1;
22.    $this->data['title'] = 'Manajemen SPBU';
23.    $this->data['content'] = $this->load->view('spbu_list',array(
24.    'data' => $this->spbu_model->get_list_spbu(
25.    ),TRUE);
26.    $this->load->view('template_baru',$this->data);
27.    }
28.    // menampilkan detail ketika user mengklik icon spbu
29.    function detail($id) {
30.    $data_spbu = $this->spbu_model->get_spbu_by($id);
31.    $this->data['title'] = $data_spbu['nama_spbu'];
32.    $this->data['content'] = $this->load->view(
33.    'spbu_detail', array(
34.    'id_spbu' => $id,
35.    'data' => $data_spbu,
36.    'data_produk' => $this->spbu_model->get_spbu_produk($id),
37.    'data_fasilitas' => $this->spbu_model->get_spbu_fasilitas($id)
38.    ), TRUE);
39.    $this->load->view('template_baru', $this->data);
40.    }
41.    // fungsi ajax untuk menampilkan detail spbu
42.    function detail_ajax() {
43.    $id = $this->input->post('id');
44.    $data_spbu = $this->spbu_model->get_spbu_by($id);
45.    $produk = $this->spbu_model->get_spbu_produk($id);
46.    $fasilitas = $this->spbu_model->get_spbu_fasilitas($id);
47.    if ($data_spbu['gambar'] == "") {
48.    $gambar = 'no-image.png';
49.    } else {
50.    $gambar = $data_spbu['gambar'];
51.    }
52.    echo '||';
54.    echo '<h1>' . $data_spbu['nama_spbu'] . '</h1>';
55.    echo 'Alamat : ' . $data_spbu['lokasi_spbu'] . '<br/>';
56.    echo 'Kategori : ' . $data_spbu['nama_kategori'] . '<br/>';
57.    echo 'Produk BBM : ';
58.    $produk_str = '';
```

```
59. foreach ($produk as $value) {
60.     $produk_str .= $value['nama_BBM'] . ' ';
61. }
62. echo substr($produk_str, 0, -2);
63. echo '<br/>';
64. echo 'Fasilitas : ';
65. $fasilitas_str = "";
66. foreach ($fasilitas as $value) {
67.     $fasilitas_str .= $value['nama_fasilitas'] . ' ';
68. }
69. echo substr($fasilitas_str, 0, -2);
70. echo '<br/>';
71. echo 'Layanan : ' . $data_spbu['waktu_layanan'] . '<br/>';
72. echo 'Keterangan : Tersedia ' . $data_spbu['jumlah_pompa'] . ' pompa. ' .
73. $data_spbu['keterangan'] . '<br/>';
74. echo '|' . $rate;
75. }
76. //menampilkan halaman input spbu
77. function add() {
78.     if ($this->data['akses'] != NULL) {
79.         $this->data['title'] = 'Tambah SPBU';
80.         $this->data['content'] = $this->load->view(
81.             'spbu_add', array(
82.                 'produk' => $this->spbu_model->get_BBM(),
83.                 'kategori' => $this->spbu_model->get_kategori(),
84.                 'fasilitas' => $this->spbu_model->get_fasilitas(),
85.                 'layanan' => $this->spbu_model->get_layanan()
86.             ), TRUE);
87.         $this->load->view('template_baru', $this->data);
88.     } else {
89.         redirect('home');
90.     }
91. }
92. //menampilkan halaman edit, dengan mengambil id spbu
93. function edit($id) {
94.     if ($this->data['akses'] != NULL) {
95.         $this->data['title'] = 'Edit SPBU';
96.         $this->data['content'] = $this->load->view('spbu_edit', array(
97.             'data' => $this->spbu_model->get_spbu_detail($id),
98.             'BBM' => $this->spbu_model->get_BBM(),
99.             'produk' => $this->spbu_model->get_spbu_produk($id),
100.            'kategori' => $this->spbu_model->get_kategori(),
101.            'fasilitas' => $this->spbu_model->get_fasilitas(),
102.            'spbu_fasilitas' => $this->spbu_model->get_spbu_fasilitas($id),
103.            'layanan' => $this->spbu_model->get_layanan()
104.        ), TRUE);
105.        $this->load->view('template_baru', $this->data);
106.    } else {
107.        redirect('home');
108.    }
109. }
110. // fungsi yang dijalankan dalam proses menambah spbu
111. function add_process() {
112.     if ($this->data['akses'] != NULL) {
113.         if ($this->__add_rule()) {
114.             if ($_FILES['image']['name'] == "") {
115.                 $gambar = 1;
```

```
116. } else {
117.     $gambar = $_FILES['image']['name'];
118. }
119. $data = array(
120.     'nama_spbu' => strtoupper($this->input->post('nama_spbu')),
121.     'lokasi' => $this->input->post('lokasi'),
122.     'id_kategori' => $this->input->post('kategori'),
123.     'gambar' => $gambar,
124.     'jumlah_pompa' => $this->input->post('jumlah_pompa'),
125.     'lat' => $this->input->post('lat'),
126.     'long' => $this->input->post('long'),
127.     'produk' => $this->input->post('produk'),
128.     'fasilitas' => $this->input->post('fasilitas'),
129.     'id_layanan' => $this->input->post('layanan'),
130.     'keterangan' => $this->input->post('keterangan'),
131.     'id_icon' => $this->icon_spbu($this->input->post('produk'))
132. );
133. if ($data['gambar'] == 1) {
134.     $sukses = $this->spbu_model->add_spbu($data);
135.     if ($sukses) {
136.         $this->session->set_userdata('message', '1,SPBU telah ditambahkan.');
```



```
137.         redirect('spbu/cobaspbu');
138.     } else {
139.         $this->session->set_userdata('message', '2,SPBU gagal ditambahkan.');
```

```
140.         redirect('spbu/add');
141.     }
142. } else {
143.     $config_image = array(
144.         'upload_path' => './images/',
145.         'allowed_types' => 'gif|jpg|png|jpeg',
146.         'max_size' => '500000',
147.         'overwrite' => TRUE,
148.         'remove_spaces' => FALSE,
149.         'file_name' => $_FILES['image']['name']
150.     );
151.     $this->load->library('upload');
152.     $this->upload->initialize($config_image);
153.     if ($this->upload->do_upload('image')) {
154.         $sukses = $this->spbu_model->add_spbu($data);
155.         if ($sukses) {
156.             $this->session->set_userdata('message', '1,SPBU telah ditambahkan.');
```

```
157.             redirect('home/cobaspbu/');
158.         } else {
159.             $this->session->set_userdata('message', '2,SPBU gagal ditambahkan.');
```

```
160.             redirect('home/cobaspbu/');
161.         }
162.     } else {
163.         $this->session->set_userdata('message', '2,Upload gambar gagal.');
```

```
164.         redirect('spbu/add');
165.     }
166. }
167. } else {
168.     $this->add();
169. }
170. } else {
171.     redirect('home');
```

```
172. }
```

```
173.     }
174.     // fungsi menyimpan perubahan dalam proses edit
175.     function edit_process() {
176.     if ($this->data['akses'] != NULL) {
177.     if ($this->__edit_rule()) {
178.     if ($_FILES['image']['name'] == "") {
179.     $gambar = 1;
180.     } else {
181.     $gambar = $_FILES['image']['name'];
182.     }
183.     $data = array(
184.     'id_spbu' => $this->input->post('id_spbu'),
185.     'nama_spbu' => strtoupper($this->input->post('nama_spbu')),
186.     'lokasi' => $this->input->post('lokasi'),
187.     'id_kategori' => $this->input->post('kategori'),
188.     'gambar' => $gambar,
189.     'jumlah_pompa' => $this->input->post('jumlah_pompa'),
190.     'lat' => $this->input->post('lat'),
191.     'long' => $this->input->post('long'),
192.     'produk' => $this->input->post('produk'),
193.     'fasilitas' => $this->input->post('fasilitas'),
194.     'id_layanan' => $this->input->post('layanan'),
195.     'keterangan' => $this->input->post('keterangan'),
196.     'id_icon' => $this->icon_spbu($this->input->post('produk'))
197.     );
198.     if ($data['gambar'] == 1) {
199.     $sukses = $this->spbu_model->edit_spbu($data);
200.     if ($sukses) {
201.     $this->session->set_userdata('message', '1,SPBU telah diupdate.');
```

```
230.     }
231.     }
232.     } else {
233.     $this->edit($this->input->post('id_spbu'));
234.     }
235.     } else {
236.     redirect('home');
237.     }
238.     }
239.     // proses mennghapus id berdasarkan spbu yang dipilih
240.     function delete_process($id) {
241.     if ($this->data['akses'] != NULL) {
242.     if ($this->spbu_model->cek_transaksi($id) == TRUE) {
243.     $sukses = $this->spbu_model->delete_spbu($id);
244.     if ($sukses == true) {
245.     $this->session->set_userdata('message', '1,SPBU telah dihapus. ');
246.     redirect('spbu/cobaspbu');
247.     } else {
248.     $this->session->set_userdata('message', '2,SPBU gagal dihapus. ');
249.     redirect('spbu/list_spbu/');
250.     }
251.     } else {
252.     $this->session->set_userdata('message', '2,Telah ada transaksi dengan tabel lain SPBU
253.     tidak boleh dihapus. ');
254.     redirect('spbu/list_spbu/');
255.     }
256.     } else {
257.     redirect('home');
258.     }
259.     }
260.     // fungsi dalam tambah spbu, untuk mengecek adakah nama yang sama
261.     function cek_namaspbu() {
262.     $validateValue = $_REQUEST['fieldValue'];
263.     $validateId = $_REQUEST['fieldId'];
264.     $arrayToJs = array();
265.     $arrayToJs[0] = $validateId;
266.     $cek = $this->spbu_model->cek_namaspbu($validateValue);
267.     if ($cek == true) {
268.     $arrayToJs[1] = true;
269.     echo json_encode($arrayToJs);
270.     } else {
271.     for ($x = 0; $x < 1000000; $x++) {
272.     if ($x == 990000) {
273.     $arrayToJs[1] = false;
274.     echo json_encode($arrayToJs);
275.     }
276.     }
277.     }
278.     }
279.     function status($id_status, $id_spbu) {
280.     if ($this->data['akses'] == 1) {
281.     $sukses = $this->spbu_model->change_status($id_status, $id_spbu);
282.     if ($sukses == true) {
283.     $this->session->set_userdata('message', '1,Status spbu telah diubah. ');
284.     redirect('spbu/list_spbu/');
285.     } else {
286.     $this->session->set_userdata('message', '2,Status spbu gagal diubah. ');
```

```
287. redirect('spbu/list_spbu/');
288. }
289. } else {
290. redirect('home');
291. }
292. }
293. function icon_spbu($input){
294. $icon = 1;
295. for ($i = 0; $i < count($input); $i++) {
296. if ($input[$i] == 2) {
297. $icon += 1;
298. }
299. if ($input[$i] == 4) {
300. $icon += 2;
301. }
302. }
303. return $icon;
304. }
305. private function __rule($rule) {
306. $this->load->library('form_validation');
307. $this->form_validation->set_rules($rule);
308. if ($this->form_validation->run()) {
309. return TRUE;
310. } else {
311. return FALSE;
312. }
313. }
314. // menambahkan rule baru ke dalam peta dan algoritma
315. private function __add_rule() {
316. $rule = array
317. (
318. array
319. (
320. 'field' => 'nama_spbu'
321. , 'label' => 'nama spbu'
322. , 'rules' => 'required'
323. )
324. , array
325. (
326. 'field' => 'lat'
327. , 'label' => 'latitude'
328. , 'rules' => 'trim|required'
329. )
330. , array
331. (
332. 'field' => 'long'
333. , 'label' => 'longitude'
334. , 'rules' => 'trim|required'
335. )
336. , array
337. (
338. 'field' => 'lokasi'
339. , 'label' => 'lokasi'
340. , 'rules' => 'trim|required'
341. )
342. );
343. return $this->__rule($rule);
```



```

344.     }
345.     // fungsi edit rule, ketika spbu diedit
346.     private function __edit_rule() {
347.         $rule = array
348.         (
349.             array
350.             (
351.                 'field' => 'nama_spbu'
352.                 , 'label' => 'nama spbu'
353.                 , 'rules' => 'required'
354.             )
355.             , array
356.             (
357.                 'field' => 'lat'
358.                 , 'label' => 'latitude'
359.                 , 'rules' => 'trim|required'
360.             )
361.             , array
362.             (
363.                 'field' => 'long'
364.                 , 'label' => 'longitude'
365.                 , 'rules' => 'trim|required'
366.             )
367.             , array
368.             (
369.                 'field' => 'lokasi'
370.                 , 'label' => 'lokasi'
371.                 , 'rules' => 'trim|required'
372.             )
373.         );
374.         return $this->__rule($rule);
375.     }
376. }
377. }
378. ?>

```

2. Kelas spbu_model

Penulisan kode program spbu_model dapat dilihat pada Tabel 14.

Tabel 14 Kode Program Kelas spbu_model

```

1. <?php
2. if (!defined('BASEPATH'))
3.     exit('No direct script access allowed');
4. class Spbu_Model extends CI_Model {
5.     function __construct() {
6.         parent::__construct();
7.     }
8.     function category($id) {
9.         if ($id != 0) {
10.            $kategori = " and tb_spbu.id_kategori = $id ";
11.        } else {
12.            $kategori = "";

```

```
13. }
14. return $kategori;
15. }
16. function jumlah(){
17. return $this->db->query ("select COUNT(id_spbu) as jumlah from tb_spbu where
18. id_status = 2");
19. }
20. function product($id) {
21. if ($id != 0) {
22. $produk = " and tb_spbuBBM.id_BBM = $id ";
23. } else {
24. $produk = "";
25. }
26. return $produk;
27. }
28. function get_semua(){
29. return $this->db->query("select * from tb_spbu")->result_array();
30. }
31. function layanan($id) {
32. if ($id != 0) {
33. $layanan = " and tb_spbu.id_layanan = $id ";
34. } else {
35. $layanan = "";
36. }
37. return $layanan;
38. }
39. // fungsi untuk menampilkan pom terdekat
40. function get_nearest_spbu($input) {
41. $kategori = $this->category($input['kategori']);
42. $produk = $this->product($input['produk']);
43. $layanan = $this->layanan($input['layanan']);
44. $data = $this->db->query("select distinct(tb_spbu.id_spbu), nama_spbu, x, y,
45. tb_spbu.id_kategori, gambar, id_icon, st_distance_spheroid(geom,
46. st_setsrid(st_makepoint($input[x],$input[y]), 4326),
47. 'SPHEROID[\"WGS84\",6378137,298.25728]') / 1000 as jarak, st_asgeojson(geom) as
48. geojson
49. from tb_spbu, tb_spbuKategori, tb_spbuBBM, tb_spbuProduk
50. where tb_spbu.id_kategori = tb_spbuKategori.id_kategori
51. and tb_spbu.id_spbu = tb_spbuProduk.id_spbu
52. and tb_spbuBBM.id_BBM = tb_spbuProduk.id_BBM
53. and tb_spbu.id_status = 2
54. and st_distance_spheroid(geom, st_setsrid(st_makepoint($input[x],$input[y]), 4326),
55. 'SPHEROID[\"WGS84\",6378137,298.25728]')/1000 <= $input[radius] $kategori
56. $produk $layanan
57. order by st_distance_spheroid(geom, st_setsrid(st_makepoint($input[x],$input[y]),
58. 4326), 'SPHEROID[\"WGS84\",6378137,298.25728]')/1000 asc");
59. return $data->result_array();
60. }
61. // menampilkan semua koordinat spbu
62. function get_all_spbu($input) {
63. $kategori = $this->category($input['kategori']);
64. $produk = $this->product($input['produk']);
65. $layanan = $this->layanan($input['layanan']);
66. $data = $this->db->query("select distinct(tb_szu.id_spbu), nama_spbu, x, y,
67. tb_spbu.id_kategori, gambar, id_icon, st_asgeojson(geom) as geojson
68. from tb_spbu, tb_spbuKategori, tb_spbuBBM, tb_spbuProduk
69. where tb_spbu.id_kategori = tb_spbuKategori.id_kategori
```

```
70. and tb_spbu.id_spbu = tb_spbuProduk.id_spbu
71. and tb_spbuBBM.id_BBM = tb_spbuProduk.id_BBM
72. and tb_spbu.id_status = 2
73. $kategori $produk $layanan");
74. return $data->result_array();
75. }
76. function search($input) {
77.     $kategori = $this->category($input['kategori']);
78.     $produk = $this->product($input['produk']);
79.     $layanan = $this->layanannya($input['layanannya']);
80.     $data = $this->db->query("select distinct(tb_spbu.id_spbu), nama_spbu, x, y,
81.     tb_spbu.id_kategori, gambar, id_icon, st_asgeojson(geom) as geojson
82.     from tb_spbu, tb_spbuKategori, tb_spbuBBM, tb_spbuProduk
83.     where tb_spbu.id_kategori = tb_spbuKategori.id_kategori
84.     and tb_spbu.id_spbu = tb_spbuProduk.id_spbu
85.     and tb_spbuBBM.id_BBM = tb_spbuProduk.id_BBM
86.     and tb_spbu.id_status = 2
87.     and nama_spbu ilike '%$input[keyword]%' $kategori $produk $layanannya");
88.     return $data->result_array();
89. }
90. // get
91. function get_list_spbu() {
92.     $data = $this->db->query("select * from
93.     tb_spbu, tb_spbuKategori, tb_status
94.     where tb_spbu.id_kategori = tb_spbuKategori.id_kategori
95.     and tb_spbu.id_status = tb_status.id_status
96.     order by id_spbu desc");
97.     return $data->result_array();
98. }
99. function get_spbu_by($id) {
100.     $data = $this->db->query("select *
101.     from tb_spbu, tb_spbuKategori, tb_spbulayanannya
102.     where tb_spbu.id_kategori = tb_spbuKategori.id_kategori
103.     and tb_spbu.id_layanannya = tb_spbulayanannya.id_layanannya
104.     and id_spbu = $id");
105.     return $data->row_array();
106. }
107. function get_spbu_form() {
108.     $data = $this->db->query("select * from tb_spbu");
109.     return $data->result_array();
110. }
111. function get_spbu_detail($id) {
112.     $data = $this->db->query("select *
113.     from tb_spbu, tb_spbuKategori, tb_spbulayanannya
114.     where tb_spbu.id_kategori = tb_spbuKategori.id_kategori
115.     and tb_spbu.id_layanannya = tb_spbulayanannya.id_layanannya
116.     and id_spbu = $id");
117.     return $data->row_array();
118. }
119. function get_spbu_produk($id) {
120.     $data = $this->db->query("select *
121.     from tb_spbuProduk, tb_spbuBBM
122.     where tb_spbuProduk.id_BBM = tb_spbuBBM.id_BBM
123.     and id_spbu = $id");
124.     return $data->result_array();
125. }
126. function get_spbu_fasilitas($id) {
```

```
127. $data = $this->db->query("select *
128. from tb_spuafasilitas, tb_fasilitas
129. where tb_spuafasilitas.id_fasilitas = tb_fasilitas.id_fasilitas
130. and id_spu = $id");
131. return $data->result_array();
132. }
133. function get_BBM() {
134. return $this->db->query("select * from tb_spuBBM")->result_array();
135. }
136. function get_layanan() {
137. return $this->db->query("select * from tb_spuayanan")->result_array();
138. }
139.
140. function get_fasilitas() {
141. return $this->db->query("select * from tb_fasilitas")->result_array();
142. }
143. function get_kategori() {
144. return $this->db->query("select * from tb_spuKategori")->result_array();
145. }
146. function get_rows_spu() {
147. return $this->db->query("select id_spu from tb_spu")->num_rows();
148. }
149. function get_new_id() {
150. $data = $this->db->query("select max(id_spu) as id from tb_spu");
151. $last = $data->row_array();
152. if ($last['id'] == 0 || $last['id'] == NULL) {
153. return 1;
154. } else {
155. return ($last['id'] + 1);
156. }
157. }
158. // cek
159. function cek_namaspu($nama){
160. $data = $this->db->query("select * from tb_spu where nama_spu =
161. ".strtoupper($nama).""");
162. if($data->num_rows() > 0){
163. return FALSE;
164. } else{
165. return TRUE;
166. }
167. }
168. function cek_transaksi($id_spu){
169. $data1 = $this->db->query("select * from tb_rating where id_spu = $id_spu");
170. $data2 = $this->db->query("select * from tb_review where id_spu = $id_spu");
171. if($data1->num_rows() > 0 || $data2->num_rows() > 0){
172. return FALSE;
173. } else{
174. return TRUE;
175. }
176. }
177. function __is_exist($id) {
178. $this->db->where('id_spu', $id);
179. $data = $this->db->get('tb_spu');
180. if ($data->num_rows() > 0) {
181. return TRUE;
182. } else {
183. return FALSE;
```

```
184.     }
185.     }
186.     // set
187.     function add_spbu($input) {
188.         $new_id = $this->get_new_id();
189.         $user = $this->session->userdata('spbu_iduser');
190.         $this->db->trans_begin();
191.         if ($input['gambar'] == 1) {
192.             $this->db->query("
193. insert into tb_spbu (
194. id_spbu,
195. nama_spbu,
196. lokasi_spbu,
197. id_kategori,
198. id_layanan,
199. keterangan,
200. id_user,
201. id_status,
202. tanggal_tambah,
203. jumlah_pompa,
204. id_icon,
205. x,
206. y,
207. geom
208. ) values (
209. " . $new_id . ",
210. " . $input['nama_spbu'] . ",
211. " . $input['lokasi'] . ",
212. " . $input['id_kategori'] . ",
213. " . $input['id_layanan'] . ",
214. " . $input['keterangan'] . ",
215. " . $user . ",
216. 2,
217. now(),
218. $input[jumlah_pompa],
219. $input[id_icon],
220. $input[long],
221. $input[lat],
222. ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
223. )
224. ");
225.     } else {
226.         $this->db->query("
227. insert into tb_spbu (
228. id_spbu,
229. nama_spbu,
230. lokasi_spbu,
231. id_kategori,
232. id_layanan,
233. keterangan,
234. id_user,
235. id_status,
236. gambar,
237. tanggal_tambah,
238. jumlah_pompa,
239. id_icon,
240. x,
```

```
241. y,
242. geom
243. ) values (
244. "" . $new_id . "",
245. "" . $input['nama_spbu'] . "",
246. "" . $input['lokasi'] . "",
247. "" . $input['id_kategori'] . "",
248. "" . $input['id_layanan'] . "",
249. "" . $input['keterangan'] . "",
250. "" . $user . "",
251. 2,
252. "" . $input['gambar'] . "",
253. now(),
254. $input[jumlah_pompa],
255. $input[id_icon],
256. $input[long],
257. $input[lat],
258. ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
259. )
260. ");
261. }
262. for ($i = 0; $i < count($input['produk']); $i++) {
263. $id_BBM = $input['produk'][$i];
264. if ($id_BBM != 0) {
265. $this->db->query("insert into tb_spbuProduk (id_spbu, id_BBM) values ($new_id,
266. $id_BBM)");
267. }
268. }
269. for ($i = 0; $i < count($input['fasilitas']); $i++) {
270. $id_fasilitas = $input['fasilitas'][$i];
271. if ($id_fasilitas != 0) {
272. $this->db->query("insert into tb_spbufasilitas (id_spbu, id_fasilitas) values ($new_id,
273. $id_fasilitas)");
274. }
275. }
276. if ($this->db->trans_status() === FALSE) {
277. $this->db->trans_rollback();
278. return FALSE;
279. } else {
280. $this->db->trans_commit();
281. return TRUE;
282. }
283. }
284. function edit_spbu($input) {
285. $this->db->trans_begin();
286. if ($input['gambar'] == 1) {
287. $this->db->query("
288. update tb_spbu set
289. nama_spbu = '$input[nama_spbu]',
290. lokasi_spbu = '$input[lokasi]',
291. id_kategori = $input[id_kategori],
292. id_layanan = $input[id_layanan],
293. keterangan = '$input[keterangan]',
294. jumlah_pompa = $input[jumlah_pompa],
295. id_icon = $input[id_icon],
296. x = $input[long],
297. y = $input[lat],
```

```
298. geom = ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
299. where id_spbu = $input[id_spbu]
300. ");
301. } else {
302. $this->db->query("
303. update tb_spbu set
304. nama_spbu = '$input[nama_spbu]',
305. lokasi_spbu = '$input[lokasi]',
306. id_kategori = $input[id_kategori],
307. id_layanan = $input[id_layanan],
308. keterangan = '$input[keterangan]',
309. jumlah_pompa = $input[jumlah_pompa],
310. gambar = '$input[gambar]',
311. id_icon = $input[id_icon],
312. x = $input[long],
313. y = $input[lat],
314. geom = ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
315. where id_spbu = $input[id_spbu]
316. ");
317. }
318. $this->db->query("delete from tb_spbuProduk where id_spbu = $input[id_spbu]");
319. for ($i = 0; $i < count($input['produk']); $i++) {
320. $id_BBM = $input['produk'][$i];
321. if ($id_BBM != 0) {
322. $this->db->query("insert into tb_spbuProduk (id_spbu, id_BBM) values
323. ($input[id_spbu], $id_BBM)");
324. }
325. }
326. $this->db->query("delete from tb_spbufasilitas where id_spbu = $input[id_spbu]");
327. for ($i = 0; $i < count($input['fasilitas']); $i++) {
328. $id_fasilitas = $input['fasilitas'][$i];
329. if ($id_fasilitas != 0) {
330. $this->db->query("insert into tb_spbufasilitas (id_spbu, id_fasilitas) values
331. ($input[id_spbu], $id_fasilitas)");
332. }
333. }
334. if ($this->db->trans_status() === FALSE) {
335. $this->db->trans_rollback();
336. return FALSE;
337. } else {
338. $this->db->trans_commit();
339. return TRUE;
340. }
341. }
342. function change_status($id_status, $id_spbu) {
343. $cek = $this->db->query("update tb_spbu set id_status = $id_status where id_spbu =
344. $id_spbu");
345. if ($cek == TRUE) {
346. return TRUE;
347. } else {
348. return FALSE;
349. }
350. }
351. function delete_spbu($id) {
352. if ($this->__is_exist($id)) {
353. $this->db->trans_begin();
354. $this->db->query("delete from tb_spbufasilitas where id_spbu = $id");
```

```

355. $this->db->query("delete from tb_spbuproduk where id_spbu = $id");
356. $this->db->query("delete from tb_spbu where id_spbu = $id");
357. if ($this->db->trans_status() === FALSE) {
358. $this->db->trans_rollback();
359. return FALSE;
360. } else {
361. $this->db->trans_commit();
362. return TRUE;
363. }
364. } else {
365. return FALSE;
366. }
367. }
368. function latlong(){
369. $data = $this->db->query("select x as lng, y as lat, nama_spbu from tb_spbu");
370. return $data->result_array();
371. }
372. }
373. ?>

```

D.5 Kode Program Melihat Peta

Kode program melihat peta terletak pada kelas home dan spbu_model.

1. Kelas home

Penulisan kode program home dapat dilihat pada Tabel 15.

Tabel 15 Kode Program Kelas home

```

1. <?php
2. if (!defined('BASEPATH'))
3. exit('No direct script access allowed');
4. class Home extends CI_Controller {
5. function __construct() {
6. parent::__construct();
7. $this->load->model('spbu_model');
8. $this->load->model('akun_model');
9. $this->load->model('distribusi_model');
10. $this->load->model('routing_model');
11. }
12. function homead(){
13. $this->data['menu'] = 1;
14. $this->data['title'] = 'Dashboard';
15. $this->data['content'] = $this->load->view('dashboard',array(
16. 'produk' => $this->spbu_model->get_BBM(),
17. 'angka' => $this->spbu_model->jumlah()->result_array()[0]['jumlah'],
18. 'angka2' => $this->distribusi_model->jumlah()->result_array()[0]['jumlah']
19. ),TRUE);
20. $this->load->view('template_baru',$this->data);
21. }
22. function homeSopir(){
23. $this->data['menu'] = 1;

```



```

24. $this->data['title'] = 'Dashboard';
25. $this->data['content'] = $this->load->view('dashboard',array(
26. 'produk' => $this->spbu_model->get_BBM(),
27. 'spbu' => $this->spbu_model->get_semua(),
28. 'angka' => $this->spbu_model->jumlah()->result_array()[0]['jumlah'],
29. 'angka2' => $this->distribusi_model->jumlah()->result_array()[0]['jumlah']
30. ),TRUE);
31. $this->load->view('template_baru',$this->data);
32. }
33. function mapsSopir(){
34. $this->data['menu'] = 1;
35. $this->data['title'] = 'Peta SPBU';
36. $this->data['content'] = $this->load->view(
37. 'spbu_map', array(
38. 'produk' => $this->spbu_model->get_BBM(),
39. 'kategori' => $this->spbu_model->get_kategori(),
40. 'fasilitas' => $this->spbu_model->get_fasilitas(),
41. 'layanan' => $this->spbu_model->get_layanan()
42. ), TRUE);
43. $this->load->view('template_baru', $this->data);
44. }
45. // halaman awal
46. // fungsi filter spbu berdasarkan kategori, produk dan layanan
47. function spbu() {
48. if ($this->input->post('x') == 'x') {
49. $data = array(
50. 'kategori' => $this->input->post('kategori'),
51. 'produk' => $this->input->post('produk'),
52. 'layanan' => $this->input->post('layanan')
53. );
54. $rs = $this->spbu_model->get_all_spbu($data);
55. } else {
56. $data = array(
57. 'x' => $this->input->post('x'),
58. 'y' => $this->input->post('y'),
59. 'radius' => $this->input->post('radius'),
60. 'kategori' => $this->input->post('kategori'),
61. 'produk' => $this->input->post('produk'),
62. 'layanan' => $this->input->post('layanan')
63. );
64. $rs = $this->spbu_model->get_nearest_spbu($data);
65. }
66. $output = "";
67. $rowOutput = "";
68. foreach ($rs as $row) {
69. $rowOutput = (strlen($rowOutput) > 0 ? ',' : '') . '{"type": "Feature", "geometry": ' .
70. $row['geojson'] . ', "properties": {';
71. $props = "";
72. $id = "";
73. foreach ($row as $key => $val) {
74. if ($key != "geojson") {
75. $props .= (strlen($props) > 0 ? ',' : '') . '."' . $key . '":' . $this->escapeJsonString($val) . '""';
76. }
77. if ($key == "id") {
78. $id .= ',"id":' . $this->escapeJsonString($val) . '""';
79. }
80. }

```

```

81. $rowOutput .= $props . '>';
82. $rowOutput .= $id;
83. $rowOutput .= '>';
84. $output .= $rowOutput;
85. }
86. $output = '{ "type": "FeatureCollection", "features": [ ' . $output . ' ]}';
87. echo $output;
88. }
89. // fungsi pencarian, menampilkan spbu
90. function search() {
91. $data = array(
92. 'keyword' => $this->input->post('keyword'),
93. 'kategori' => $this->input->post('kategori'),
94. 'produk' => $this->input->post('produk'),
95. 'layanan' => $this->input->post('layanan')
96. );
97. $rs = $this->spbu_model->search($data);
98. $output = "";
99. $rowOutput = "";
100. foreach ($rs as $row) {
101. $rowOutput = (strlen($rowOutput) > 0 ? ',' : '') . '{ "type": "Feature", "geometry": ' .
102. $row['geojson'] . ', "properties": {';
103. $props = "";
104. $id = "";
105. foreach ($row as $key => $val) {
106. if ($key != "geojson") {
107. $props .= (strlen($props) > 0 ? ',' : '') . '"" . $key . '":' . $this->escapeJsonString($val) . '""';
108. }
109. if ($key == "id") {
110. $id .= ', "id":' . $this->escapeJsonString($val) . '""';
111. }
112. }
113. $rowOutput .= $props . '>';
114. $rowOutput .= $id;
115. $rowOutput .= '>';
116. $output .= $rowOutput;
117. }
118. $output = '{ "type": "FeatureCollection", "features": [ ' . $output . ' ]}';
119. echo $output;
120. }
121. function escapeJsonString($value) {
122. $escapers = array("\\", "/", "\", "\n", "\r", "\t", "\x08", "\x0c");
123. $replacements = array("\\\\", "\\/", "\\\"", "\\n", "\\r", "\\t", "\\f", "\\b");
124. $result = str_replace($escapers, $replacements, $value);
125. return $result;
126. }
127. // proses registrasi user baru
128. function register_process() {
129. if ($this->input->post('register')) {
130. $data = array(
131. 'id_user' => "",
132. 'username' => $this->input->post('username'),
133. 'password' => $this->input->post('password'),
134. 'password_confirm' => $this->input->post('password_confirm'),
135. 'nama_lengkap' => NULL,
136. 'no_telepon' => NULL,
137. 'email' => $this->input->post('email'),

```

```
138. 'alamat' => ",
139. 'id_jeniskelamin' => 1,
140. 'id_hakakses' => 2,
141. 'foto' => 1
142. );
143. if ($data['password'] == $data['password_confirm']) {
144. $hasil = $this->user_model->add($data);
145. if ($hasil) {
146. echo 'sukses';
147. } else {
148. echo 'gagal';
149. }
150. } else {
151. echo 'tdk_sama';
152. }
153. } else {
154. redirect('home');
155. }
156. }
157. // fungsi untuk mengecek username, apakah sudah ada di db apa belum
158. function cek_username() {
159. $validateValue = $_REQUEST['fieldValue'];
160. $validateId = $_REQUEST['fieldId'];
161. $arrayToJs = array();
162. $arrayToJs[0] = $validateId;
163. $cek = $this->user_model->__is_available($validateValue);
164. if ($cek == true) {
165. $arrayToJs[1] = true;
166. echo json_encode($arrayToJs);
167. } else {
168. for ($x = 0; $x < 1000000; $x++) {
169. if ($x == 990000) {
170. $arrayToJs[1] = false;
171. echo json_encode($arrayToJs);
172. }
173. }
174. }
175. }
176. // cek email sudah terdaftar apa belum
177. function cek_email() {
178. $validateValue = $_REQUEST['fieldValue'];
179. $validateId = $_REQUEST['fieldId'];
180. $arrayToJs = array();
181. $arrayToJs[0] = $validateId;
182. $cek = $this->user_model->cek_email($validateValue);
183. if ($cek == true) {
184. $arrayToJs[1] = true;
185. echo json_encode($arrayToJs);
186. } else {
187. for ($x = 0; $x < 1000000; $x++) {
188. if ($x == 990000) {
189. $arrayToJs[1] = false;
190. echo json_encode($arrayToJs);
191. }
192. }
193. }
194. }
```

```

195. // proses login
196. function login_process() {
197. if ($this->input->post('login')) {
198. $data = array(
199. 'username' => $this->input->post('username'),
200. 'password' => $this->input->post('password')
201. );
202. $hasil = $this->user_model->login($data);
203. if ($hasil) {
204. echo 'sukses';
205. } else {
206. echo 'gagal';
207. }
208. } else {
209. redirect('home');
210. }
211. }
212. // proses logout
213. function logout() {
214. $this->session->sess_destroy();
215. redirect('login/index');
216. }
217. function haversineGreatCircleDistance(
218. $latitudeFrom, $longitudeFrom, $latitudeTo, $longitudeTo) {
219. // convert from degrees to radians
220. $latFrom = deg2rad($latitudeFrom);
221. $lonFrom = deg2rad($longitudeFrom);
222. $latTo = deg2rad($latitudeTo);
223. $lonTo = deg2rad($longitudeTo);
224. $earthRadius = 6378137;
225. $latDelta = $latTo - $latFrom;
226. $lonDelta = $lonTo - $lonFrom;
227. $angle = 2 * asin(sqrt(pow(sin($latDelta / 2), 2) +
228. cos($latFrom) * cos($latTo) * pow(sin($lonDelta / 2), 2)));
229. return $angle * $earthRadius;
230. }
231. }
232. ?>

```

2. Kelas spbu_model

Penulisan kode program spbu_model dapat dilihat pada Tabel 16.

Tabel 16 Kode Program Kelas spbu_model

```

1. <?php
2. if (!defined('BASEPATH'))
3. exit('No direct script access allowed');
4. class Spbu_Model extends CI_Model {
5. function __construct() {
6. parent::__construct();
7. }
8. function category($id) {
9. if ($id != 0) {

```

```

10. $kategori = " and tb_spbu.id_kategori = $id ";
11. } else {
12. $kategori = "";
13. }
14. return $kategori;
15. }
16. function jumlah(){
17. return $this->db->query ("select COUNT(id_spbu) as jumlah from tb_spbu where id_status =
18. 2");
19. }
20. function product($id) {
21. if ($id != 0) {
22. $produk = " and tb_spbuBBM.id_BBM = $id ";
23. } else {
24. $produk = "";
25. }
26. return $produk;
27. }
28. function get_semua(){
29. return $this->db->query("select * from tb_spbu")->result_array();
30. }
31. function layanan($id) {
32. if ($id != 0) {
33. $layanan = " and tb_spbu.id_layanan = $id ";
34. } else {
35. $layanan = "";
36. }
37. return $layanan;
38. }
39. // fungsi untuk menampilkan pom terdekat
40. function get_nearest_spbu($input) {
41. $kategori = $this->category($input['kategori']);
42. $produk = $this->product($input['produk']);
43. $layanan = $this->layanan($input['layanan']);
44. $data = $this->db->query("select distinct(tb_spbu.id_spbu), nama_spbu, x, y,
45. tb_spbu.id_kategori, gambar, id_icon, st_distance_spheroid(geom,
46. st_setsrid(st_makepoint($input[x],$input[y]), 4326),
47. 'SPHEROID[\"WGS84\",6378137,298.25728])/1000 as jarak, st_asgeojson(geom) as geojson
48. from tb_spbu, tb_spbuKategori, tb_spbuBBM, tb_spbuProduk
49. where tb_spbu.id_kategori = tb_spbuKategori.id_kategori
50. and tb_spbu.id_spbu = tb_spbuProduk.id_spbu
51. and tb_spbuBBM.id_BBM = tb_spbuProduk.id_BBM
52. and tb_spbu.id_status = 2
53. and st_distance_spheroid(geom, st_setsrid(st_makepoint($input[x],$input[y]), 4326),
54. 'SPHEROID[\"WGS84\",6378137,298.25728])/1000 <= $input[radius] $kategori $produk
55. $layanan
56. order by st_distance_spheroid(geom, st_setsrid(st_makepoint($input[x],$input[y]), 4326),
57. 'SPHEROID[\"WGS84\",6378137,298.25728])/1000 asc");
58. return $data->result_array();
59. }
60. // menampilkan semua koordinat spbu
61. function get_all_spbu($input) {
62. $kategori = $this->category($input['kategori']);
63. $produk = $this->product($input['produk']);
64. $layanan = $this->layanan($input['layanan']);
65. $data = $this->db->query("select distinct(tb_szu.id_spbu), nama_spbu, x, y,
66. tb_spbu.id_kategori, gambar, id_icon, st_asgeojson(geom) as geojson

```

```
67. from tb_spbu, tb_spbuKategori, tb_spbuBBM, tb_spbuProduk
68. where tb_spbu.id_kategori = tb_spbuKategori.id_kategori
69. and tb_spbu.id_spbu = tb_spbuProduk.id_spbu
70. and tb_spbuBBM.id_BBM = tb_spbuProduk.id_BBM
71. and tb_spbu.id_status = 2
72. $kategori $produk $layanan");
73. return $data->result_array();
74. }
75. function search($input) {
76. $kategori = $this->category($input['kategori']);
77. $produk = $this->product($input['produk']);
78. $layanan = $this->layanannya($input['layanannya']);
79. $data = $this->db->query("select distinct(tb_spbu.id_spbu), nama_spbu, x, y,
80. tb_spbu.id_kategori, gambar, id_icon, st_asgeojson(geom) as geojson
81. from tb_spbu, tb_spbuKategori, tb_spbuBBM, tb_spbuProduk
82. where tb_spbu.id_kategori = tb_spbuKategori.id_kategori
83. and tb_spbu.id_spbu = tb_spbuProduk.id_spbu
84. and tb_spbuBBM.id_BBM = tb_spbuProduk.id_BBM
85. and tb_spbu.id_status = 2
86. and nama_spbu ilike '%$input[keyword]%' $kategori $produk $layanannya");
87. return $data->result_array();
88. }
89. // get
90. function get_list_spbu() {
91. $data = $this->db->query("select * from
92. tb_spbu, tb_spbuKategori, tb_status
93. where tb_spbu.id_kategori = tb_spbuKategori.id_kategori
94. and tb_spbu.id_status = tb_status.id_status
95. order by id_spbu desc");
96. return $data->result_array();
97. }
98. function get_spbu_by($id) {
99. $data = $this->db->query("select *
100. from tb_spbu, tb_spbuKategori, tb_spbulayanannya
101. where tb_spbu.id_kategori = tb_spbuKategori.id_kategori
102. and tb_spbu.id_layanannya = tb_spbulayanannya.id_layanannya
103. and id_spbu = $id");
104. return $data->row_array();
105. }
106. function get_spbu_form() {
107. $data = $this->db->query("select * from tb_spbu");
108. return $data->result_array();
109. }
110. function get_spbu_detail($id) {
111. $data = $this->db->query("select *
112. from tb_spbu, tb_spbuKategori, tb_spbulayanannya
113. where tb_spbu.id_kategori = tb_spbuKategori.id_kategori
114. and tb_spbu.id_layanannya = tb_spbulayanannya.id_layanannya
115. and id_spbu = $id");
116. return $data->row_array();
117. }
118. function get_spbu_produk($id) {
119. $data = $this->db->query("select *
120. from tb_spbuProduk, tb_spbuBBM
121. where tb_spbuProduk.id_BBM = tb_spbuBBM.id_BBM
122. and id_spbu = $id");
123. return $data->result_array();
```

```
124. }
125. function get_spbu_fasilitas($id) {
126. $data = $this->db->query("select *
127. from tb_spbu_fasilitas, tb_fasilitas
128. where tb_spbu_fasilitas.id_fasilitas = tb_fasilitas.id_fasilitas
129. and id_spbu = $id");
130. return $data->result_array();
131. }
132. function get_BBM() {
133. return $this->db->query("select * from tb_spbuBBM")->result_array();
134. }
135. function get_layanan() {
136. return $this->db->query("select * from tb_spbu_layanan")->result_array();
137. }
138.
139. function get_fasilitas() {
140. return $this->db->query("select * from tb_fasilitas")->result_array();
141. }
142. function get_kategori() {
143. return $this->db->query("select * from tb_spbu_kategori")->result_array();
144. }
145. function get_rows_spbu() {
146. return $this->db->query("select id_spbu from tb_spbu")->num_rows();
147. }
148. function get_new_id() {
149. $data = $this->db->query("select max(id_spbu) as id from tb_spbu");
150. $last = $data->row_array();
151. if ($last['id'] == 0 || $last['id'] == NULL) {
152. return 1;
153. } else {
154. return ($last['id'] + 1);
155. }
156. }
157. // cek
158. function cek_namaspbu($nama){
159. $data = $this->db->query("select * from tb_spbu where nama_spbu = '".strtoupper($nama)."'");
160. if($data->num_rows() > 0){
161. return FALSE;
162. } else{
163. return TRUE;
164. }
165. }
166. function cek_transaksi($id_spbu){
167. $data1 = $this->db->query("select * from tb_rating where id_spbu = $id_spbu");
168. $data2 = $this->db->query("select * from tb_review where id_spbu = $id_spbu");
169. if($data1->num_rows() > 0 || $data2->num_rows() > 0){
170. return FALSE;
171. } else{
172. return TRUE;
173. }
174. }
175. function __is_exist($id) {
176. $this->db->where('id_spbu', $id);
177. $data = $this->db->get('tb_spbu');
178. if ($data->num_rows() > 0) {
179. return TRUE;
180. } else {
```

```
181. return FALSE;
182. }
183. }
184. // set
185. function add_spbu($input) {
186. $new_id = $this->get_new_id();
187. $user = $this->session->userdata('spbu_iduser');
188. $this->db->trans_begin();
189. if ($input['gambar'] == 1) {
190. $this->db->query("
191. insert into tb_spbu (
192. id_spbu,
193. nama_spbu,
194. lokasi_spbu,
195. id_kategori,
196. id_layanan,
197. keterangan,
198. id_user,
199. id_status,
200. tanggal_tambah,
201. jumlah_pompa,
202. id_icon,
203. x,
204. y,
205. geom
206. ) values (
207. " . $new_id . ",
208. " . $input['nama_spbu'] . ",
209. " . $input['lokasi'] . ",
210. " . $input['id_kategori'] . ",
211. " . $input['id_layanan'] . ",
212. " . $input['keterangan'] . ",
213. " . $user . ",
214. 2,
215. now(),
216. $input[jumlah_pompa],
217. $input[id_icon],
218. $input[long],
219. $input[lat],
220. ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
221. )
222. ");
223. } else {
224. $this->db->query("
225. insert into tb_spbu (
226. id_spbu,
227. nama_spbu,
228. lokasi_spbu,
229. id_kategori,
230. id_layanan,
231. keterangan,
232. id_user,
233. id_status,
234. gambar,
235. tanggal_tambah,
236. jumlah_pompa,
237. id_icon,
```



```
238. x,
239. y,
240. geom
241. ) values (
242. "" . $new_id . "",
243. "" . $input['nama_spbu'] . "",
244. "" . $input['lokasi'] . "",
245. "" . $input['id_kategori'] . "",
246. "" . $input['id_layanan'] . "",
247. "" . $input['keterangan'] . "",
248. "" . $user . "",
249. 2,
250. "" . $input['gambar'] . "",
251. now(),
252. $input[jumlah_pompa],
253. $input[id_icon],
254. $input[long],
255. $input[lat],
256. ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
257. )
258. ");
259. }
260. for ($i = 0; $i < count($input['produk']); $i++) {
261. $id_BBM = $input['produk'][$i];
262. if ($id_BBM != 0) {
263. $this->db->query("insert into tb_spbuProduk (id_spbu, id_BBM) values ($new_id,
264. $id_BBM)");
265. }
266. }
267. for ($i = 0; $i < count($input['fasilitas']); $i++) {
268. $id_fasilitas = $input['fasilitas'][$i];
269. if ($id_fasilitas != 0) {
270. $this->db->query("insert into tb_spbufasilitas (id_spbu, id_fasilitas) values ($new_id,
271. $id_fasilitas)");
272. }
273. }
274. if ($this->db->trans_status() === FALSE) {
275. $this->db->trans_rollback();
276. return FALSE;
277. } else {
278. $this->db->trans_commit();
279. return TRUE;
280. }
281. }
282. function edit_spbu($input) {
283. $this->db->trans_begin();
284. if ($input['gambar'] == 1) {
285. $this->db->query("
286. update tb_spbu set
287. nama_spbu = '$input[nama_spbu]',
288. lokasi_spbu = '$input[lokasi]',
289. id_kategori = $input[id_kategori],
290. id_layanan = $input[id_layanan],
291. keterangan = '$input[keterangan]',
292. jumlah_pompa = $input[jumlah_pompa],
293. id_icon = $input[id_icon],
294. x = $input[long],
```

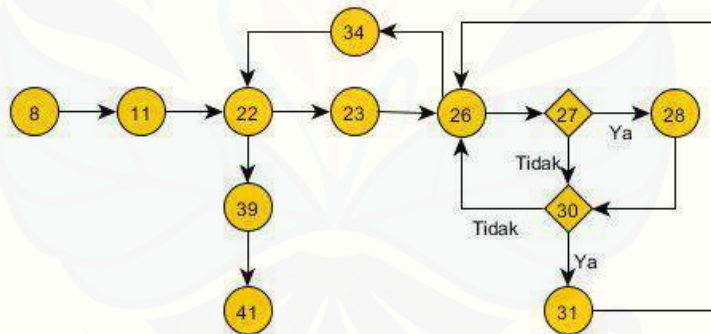
```
295. y = $input[lat],
296. geom = ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
297. where id_spbu = $input[id_spbu]
298. ");
299. } else {
300. $this->db->query("
301. update tb_spbu set
302. nama_spbu = '$input[nama_spbu]',
303. lokasi_spbu = '$input[lokasi]',
304. id_kategori = $input[id_kategori],
305. id_layanan = $input[id_layanan],
306. keterangan = '$input[keterangan]',
307. jumlah_pompa = $input[jumlah_pompa],
308. gambar = '$input[gambar]',
309. id_icon = $input[id_icon],
310. x = $input[long],
311. y = $input[lat],
312. geom = ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
313. where id_spbu = $input[id_spbu]
314. ");
315. }
316. $this->db->query("delete from tb_spbuProduk where id_spbu = $input[id_spbu]");
317. for ($i = 0; $i < count($input['produk']); $i++) {
318. $id_BBM = $input['produk'][$i];
319. if ($id_BBM != 0) {
320. $this->db->query("insert into tb_spbuProduk (id_spbu, id_BBM) values ($input[id_spbu],
321. $id_BBM)");
322. }
323. }
324. $this->db->query("delete from tb_spbufasilitas where id_spbu = $input[id_spbu]");
325. for ($i = 0; $i < count($input['fasilitas']); $i++) {
326. $id_fasilitas = $input['fasilitas'][$i];
327. if ($id_fasilitas != 0) {
328. $this->db->query("insert into tb_spbufasilitas (id_spbu, id_fasilitas) values ($input[id_spbu],
329. $id_fasilitas)");
330. }
331. }
332. if ($this->db->trans_status() === FALSE) {
333. $this->db->trans_rollback();
334. return FALSE;
335. } else {
336. $this->db->trans_commit();
337. return TRUE;
338. }
339. }
340. function change_status($id_status, $id_spbu) {
341. $cek = $this->db->query("update tb_spbu set id_status = $id_status where id_spbu =
342. $id_spbu");
343. if ($cek == TRUE) {
344. return TRUE;
345. } else {
346. return FALSE;
347. }
348. }
349. function delete_spbu($id) {
350. if ($this->__is_exist($id)) {
351. $this->db->trans_begin();
```

```

352. $this->db->query("delete from tb_spbufasilitas where id_spbu = $id");
353. $this->db->query("delete from tb_spbuproduk where id_spbu = $id");
354. $this->db->query("delete from tb_spbu where id_spbu = $id");
355. if ($this->db->trans_status() === FALSE) {
356. $this->db->trans_rollback();
357. return FALSE;
358. } else {
359. $this->db->trans_commit();
360. return TRUE;
361. }
362. } else {
363. return FALSE;
364. }
365. }
366. function latlong(){
367. $data = $this->db->query("select x as lng, y as lat, nama_spbu from tb_spbu");
368. return $data->result_array();
369. }
370. }
371. ?>
    
```

E. Pengujian White Box

E.1 Kelas Controller



Gambar 13 Diagram Alir *Function count()*

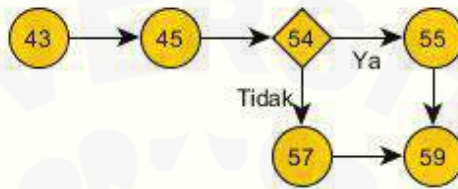
$$CC = E - N + 2 = 15 - 12 + 2 = 5$$

Tabel 17 *Test Case Function count()*

Test case	Jika ada data biasa masuk
Target yang Diharapkan	Menampilkan jalur pada map

Hasil pengujian	Benar
Path/jalur	8-11-22-39-41
Test case	Jika data terdapat data geoJson
Target yang Diharapkan	Menampilkan jalur pada map
Hasil pengujian	Benar
Path/jalur	8-11-22-23-26-27-28-30-31-26-34-22-39-41
Test case	Jika penguraian data tidak berhasil
Target yang Diharapkan	Tidak ditampilkan jalur
Hasil pengujian	Benar
Path/jalur	8-11-22-23-26-27-28-30-26-34-22-39-41
Test case	Jika penguraian data berhasil
Target yang Diharapkan	Menampilkan jalur pada map
Hasil pengujian	Benar
Path/jalur	8-11-22-23-26-27-30-31-26-34-22-39-41
Test case	Jika userdata tidak kosong
Target yang Diharapkan	Tidak ditampilkan jalur

Hasil pengujian	Benar
Path/jalur	8-11-22-23-26-27-30-26-34-22-39-41



Gambar 14 Diagram Alir *Function* save()

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

Tabel 18 *Test Case Function* save()

Test case	Jika berhasil menyimpan data distribusi
Target yang Diharapkan	Menampilkan pesan sukses
Hasil pengujian	Benar
Path/jalur	43-45-54-55-59
Test case	Jika tidak berhasil menyimpan data distribusi
Target yang Diharapkan	Menampilkan 404
Hasil pengujian	Benar
Path/jalur	43-45-54-57-59



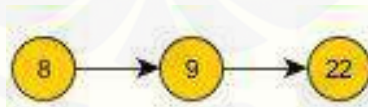
Gambar 15 Diagram Alir *Function* escapeJsonString()

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

Tabel 19 *Test Case Function* escapeJsonString ()

Test case	Jika data berhasil di <i>generate</i>
Target yang Diharapkan	Pertukaran data dalam fungsi count()
Hasil pengujian	Benar
Path/jalur	61-62-66

E.2 Kelas Model



Gambar 16 Diagram Alir *Function* get_route_geojson()

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

Tabel 20 *Test Case Function* get_route_geojson ()

Test case	Jika berhasil menambah data distribusi
-----------	---

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 21 *Test Case Function* simpan()

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

F. Pengujian *Black Box*

F.1 Fitur Pengelolaan Data Akun

Tabel 22 Pengujian *Black Box* Pengelolaan Data Akun

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

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

F.2 Fitur Pengelolaan Data SPBU

Tabel 23 Pengujian *Black Box* Pengelolaan Data SPBU

No.	Fitur	Aksi	Hasil	Kesimpulan
-----	-------	------	-------	------------

1.	Melihat data SPBU	Memilih menu Manajemen SPBU	Menampilkan tabel daftar SPBU	Berhasil
2.	Menambah data SPBU	Memilih tombol data baru	Menampilkan form tambah SPBU	Berhasil
		Memilih tombol simpan	a. Menyimpan data ke <i>database</i> b. Menampilkan halaman SPBU kembali	Berhasil
3.	Mengubah data SPBU	Memilih tombol edit	Menampilkan form edit SPBU	Berhasil
		Memilih tombol simpan	a. Menyimpan data ke <i>database</i> b. Menampilkan halaman SPBU kembali	Berhasil
4.	Menghapus data SPBU	Memilih tombol hapus	Menghapus data dari <i>database</i>	Berhasil
5.	Detail data SPBU	Memilih tombol detail	Menampilkan data detail SPBU yang dipilih	Berhasil

F.3 Fitur Melihat Data Distribusi BBM

Tabel 24 Pengujian *Black Box* Melihat Data Distribusi BBM

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Melihat data distribusi BBM	Memilih menu Manajemen distribusi	Menampilkan tabel daftar distribusi	Berhasil

F.4 Fitur Melihat Data SPBU

Tabel 25 Pengujian *Black Box* Melihat data SPBU

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

F.5 Fitur Melihat Peta

Tabel 26 Pengujian *Black Box* Melihat Peta

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Melihat Peta	Memilih menu Peta SPBU	Menampilkan peta SPBU	Berhasil

F.6 Fitur Verifikasi Data Distribusi BBM

Tabel 27 Pengujian *Black Box* Pengelolaan Data Akun

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