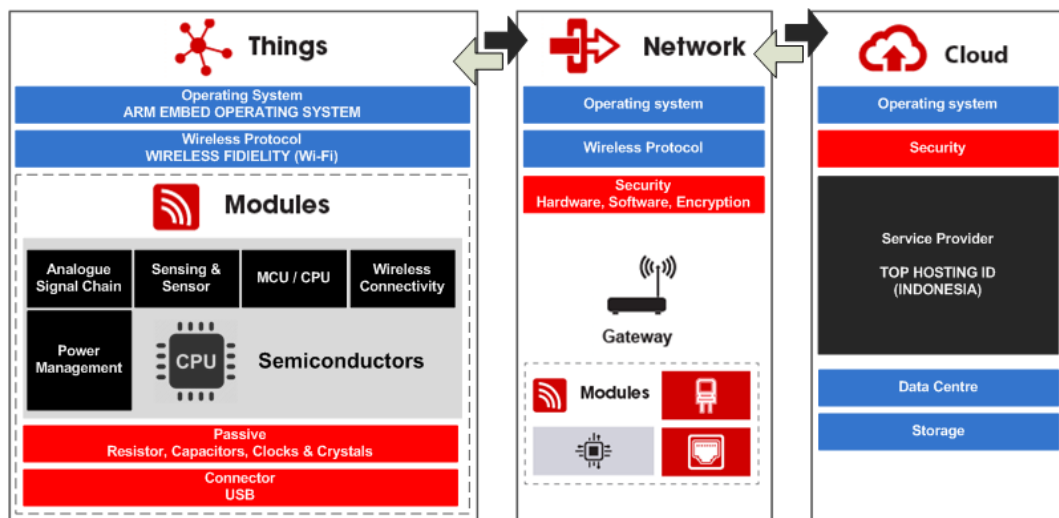


BAB III PERANCANGAN SISTEM

3.1 Arsitektur Sistem

Secara umum, sistem ini terbagi menjadi dua sistem, yaitu perangkat lunak (*software*) dan perangkat keras (*hardware*).

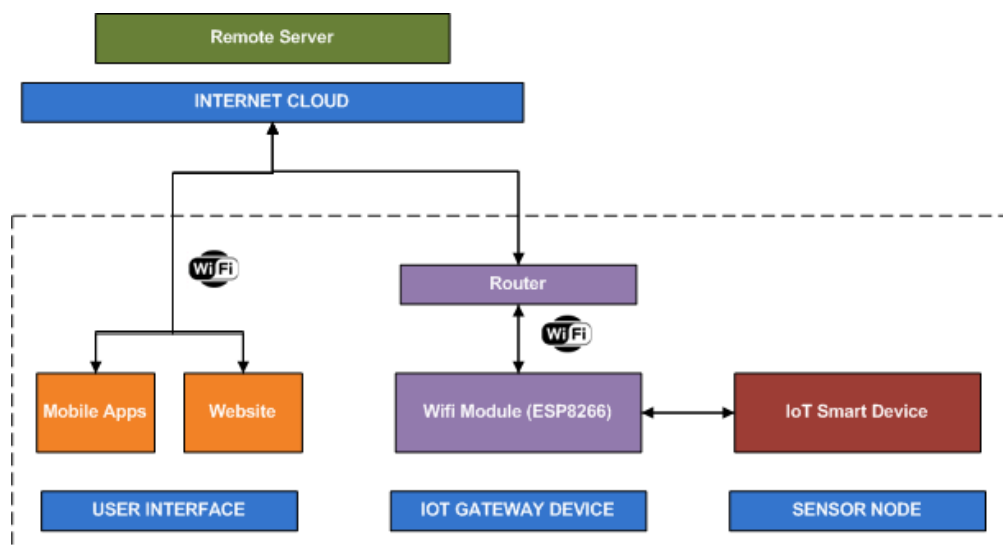


Gambar III-1 Arsitektur Sistem

Berikut ini arsitektural sistem digambarkan dalam diagram blok sistem terdiri dari tiga level :

3.1.1 Diagram Blok Level 1

Pada diagram blok level 1, akan menampilkan model diagram sistem secara umum, menjelaskan hubungan antar sub-sistem sistem bsmart. Terdapat tiga elemen sistem, yaitu perangkat bsmart (*device*) yang berfungsi untuk mendeteksi posisi, kecepatan, dan pergerakan bus, *cloud* server berfungsi sebagai pemroses dan penyimpanan data, serta *mobile application*, berfungsi sebagai *user interface* untuk mengakses informasi dan reservasi tiket bus *online*. Berikut ini gambar model diagram blok sistem bsmart level 1 :



Gambar III-2 Diagram Blok Level 1

Setiap elemen tersebut dijelaskan dalam tabel keterangan dibawah ini :

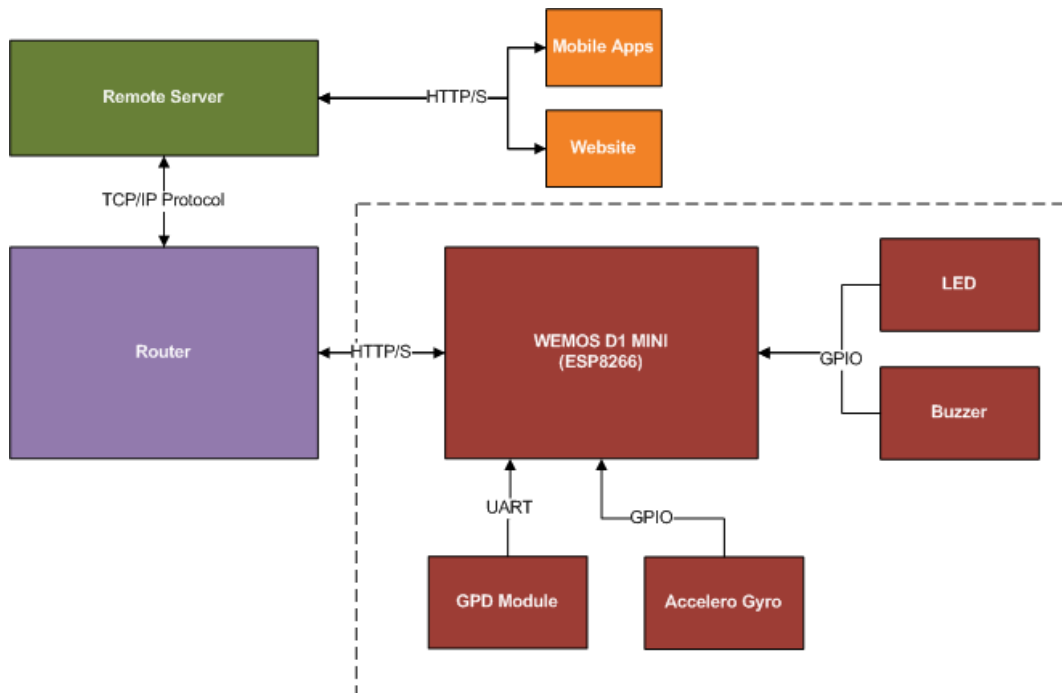
Tabel III-1 Keterangan Diagram Blok Level 1

No	Nama Elemen	Keterangan
1.	User Interface	Sistem Bsmart memiliki <i>user interface</i> berupa halaman website yang digunakan sebagai pusat informasi dan penyimpanan data, serta aplikasi <i>mobile</i> .
2.	IoT Gateway	Berfungsi sebagai sub sistem yang memiliki akses internet untuk melakukan pengiriman data ke server (<i>wirelees network</i>).
3.	Sensor Node	Sensor berfungsi untuk mengubah besaran fisik menjadi elektrik.

3.1.2 Diagram Blok Level 2

Diagram blok sistem level 2, menguraikan kembali lebih detail model diagram blok sistem level 1. Pada level ini, elemen *device* diuraikan menjadi

beberapa komponen penyusunnya, seperti sensor-sensor dan modul. Digambarkan keterkaitan, dan bagaimana masing-masing komponen dapat saling terhubung.



Gambar III-3 Diagram Blok Level 2

Berikut ini keterangan dari Gambar III-3 diatas :

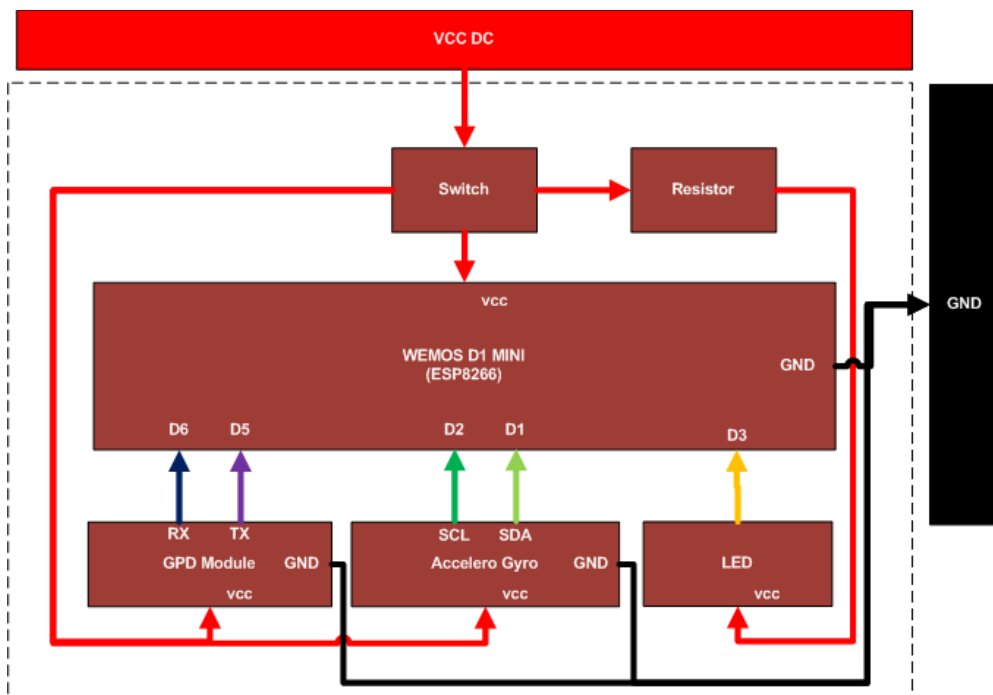
Tabel III-2 Keterangan Diagram Blok Level 2

No	Nama Elemen	Keterangan
1.	GPS Module	Modul elektronika yang memiliki fungsi utama untuk mendeteksi posisi dan kecepatan laju bus.
2.	Accelero Gyro	Modul elektronika yang memiliki fungsi untuk mendeteksi pergerakan dan akselerasi bus.
3.	LED	Lampu <i>light emitted diode</i> (LED) yang berfungsi sebagai indikator on/off perangkat.
4.	Buzzer	Indikator suara untuk mendeteksi apakah pergerakan bus normal atau berpotensi terjadi kecelakaan.
5.	Wemos D1 Mini	Mikrokontrolller yang sudah memiliki modul esp8266 untuk melakukan komunikasi dengan server melalui koneksi internet. Komponen ini berfungsi sebagai

No	Nama Elemen	Keterangan
		pemroses data yang diterima dari sensor dan modul, untuk selanjutnya dikirimkan ke server utama
6	Cloud Database / Remote Server	Mikrokontroler akan terhubung dengan <i>database</i> utama melalui komunikasi <i>wireless</i> internet.
7.	Router	Perangkat jaringan yang menghubungkan sistem dengan server.
8.	Mobile Apps / Website	Berfungsi sebagai antarmuka sistem, website sebagai pusat informasi dan penyimpanan data, serta <i>mobile apps</i> sebagai antarmuka pendukung bagi masyarakat umum.

3.1.3 Diagram Blok Level 3

Diagram blok level tiga akan menggambarkan lebih detail mengenai konektifitas komponen elektronika yang terdapat pada perangkat bsmart, lengkap dengan konfigurasi pin serta pengkabelan.



Gambar III-4 Diagram Blok Level 3

Berikut ini keterangan dari Gambar III-4 diatas :

Tabel III-3 Keterangan Diagarm Blok Level 3

No	Nama Elemen	Keterangan
1.	Power Supply 5 v	Sumber daya listrik utama
2.	Wemos D1 Mini	Mikrokontroler terhubung secara langsung dengan sumber daya listrik
3.	GPS Module	Modul GPS yang terdiri dari empat pin, terhubung ke VCC, GND, dan RX, TX mikrokontroler.
4.	Accelero-Gyro Module	Modul Accelero-Gyro, pin VCC dan GND terhubung ke sumber daya utama dan <i>ground</i> , serta pin SCSDA dan INT terhubung ke mikrokontroler Wemos.
5.	LED	LED yang digunakan sebagai indikator, terdiri dari dua pin, VCC terhubung ke sumber tegangan, sedangkan pin GND terhubung ke pin digital mikrokontroler Wemos (mode aktif <i>low</i>).
6.	Switch	Switch berfungsi untuk mengaktifkan / menonaktifkan perangkat.

3.2 Analisis Kebutuhan Sistem

Untuk mengetahui spesifikasi sistem, maka penulis akan menjabarkan beberapa subbab yang berisi mengenai perancangan sistem *software*, *hardware*, dan *database*, serta bagaimana perilaku sistem yang akan dimodelkan dalam bentuk diagram UML.

3.2.1 Analisis Kebutuhan Fungsional

Berikut ini beberapa kebutuhan sistem yang berkaitan dengan fungsional utama. Terbagi menjadi dua, yaitu kebutuhan *hardware* dan *software*.

3.2.1.1 Analisis Kebutuhan Fungsional Hardware

Sistem yang dirancang terdiri dari beberapa komponen *hardware* (sensor, mikrokontroller, komponen elektronika aktif dan pasif, serta sumber daya). Berikut ini spesifikasi dari setiap komponen yang digunakan.

Tabel III-4 Kebutuhan Perangkat Keras

No	Komponen	Spesifikasi	Fungsi
1.	Wemos D1 Mini	<ul style="list-style-type: none"> • Tegangan kerja 3.3 volt • Digital I/O Pin 11 • Analog input pin 1 (maks 3.2v) • Clock speed 80MHz/160MHz • Flash 4Mbytes • Panjang 68.6 mm • Lebar 53.4 mm • Berat 25gr 	Mikrokontroller berfungsi sebagai perangkat pemroses data, dan pengirim ke server
2.	MPU6050	<ul style="list-style-type: none"> • Chip MPU-6050 • Tegangan kerja 3-5 volt • Protokol komunikasi IIC • 16 bit AD converter • Rentang nilai Gyroscope +/- 250, 500, 1000, 2000 derajat/detik • Rentang nilai Accelerometer +/- 2g, 4g, 8g, 16g • Pin pitch 2.54 mm 	Sensor Accelerometer dan Gyroscope berfungsi mendeteksi pergerakan bus
3.	GPS NEO-6m Ublox	<ul style="list-style-type: none"> • GPS penerima produksi Ublox NEO-6M • Waktu <i>cold start</i> 38 detik • Waktu <i>hot start</i> 1 detik • Waktu <i>recapture</i> 0.1 detik • Tegangan kerja 3.3 – 5 volt • Antarmuka UART • Ukuran 31 x 24 mm 	Modul GPS berfungsi untuk mendeteksi posisi, dan kecepatan bus.
4.	Smartphone	<ul style="list-style-type: none"> • Sistem operasi android • Ram minimal 4gb 	Smartphone berfungsi sebagai perangkat validasi tiket penumpang bus

No	Komponen	Spesifikasi	Fungsi
5.	Power bank / lithium 4000mah	<ul style="list-style-type: none"> • Kapasitas 4000 mah • Berat 110 gr • Dimensi 118 x 68 x 9 mm • Tipe Lithium Polymer • USB • Port tegangan masuk DC 5 volt / 2 ampere • Port tegangan keluar DC5 volt / 2000 mah 	Power bank berfungsi sebagai sumber daya listrik perangkat
6.	<i>Light Emitted Diode</i> (LED)	<ul style="list-style-type: none"> • Tipe RC422-05 • Tinggi 25 mm • Dimeter 10 mm • Warna merah • Intensitas 52,000 mcd • Frekuensi warna 31 x 32 • Sudut cahaya 42 derajat • Tegangan kerja 1.9 – 2.1 volt • Arus kerja 20 mA 	Lampu LED digunakan sebagai indikator perangkat ON/OFF
7.	Switch TPS 10 PC Mini 2 Pin SPST	<ul style="list-style-type: none"> • Nama pabrikan TPS • Berat 100 gr • No. Model CNB-1032 • Part Number CNB-1032, CNB-1032 	Berfungsi sebagai penghubung dan pemutus daya listrik

3.2.1.2 Analisis Kebutuhan Fungsional Software

Pada perangkat lunak, terdiri dari dua bagian, yaitu sistem perancangan *website* dan aplikasi *mobile* Android. *Website* berperan sebagai *dashboard center* untuk menampilkan informasi bus. Sedangkan aplikasi *mobile* Android digunakan untuk pemesanan tiket bus *online*, serta verifikasi tiket penumpang.

3.2.1.2.1 Analisis Spesifikasi Perangkat

Berikut ini beberapa spesifikasi yang dibutuhkan agar *website* dan aplikasi *mobile* dapat berfungsi dengan baik pada perangkat (*smartphone* / *laptop* / *PC*). Data dibawah ini merupakan spesifikasi minimalis.

Tabel III-5 Kebutuhan Perangkat

No	Perangkat	Spesifikasi
----	-----------	-------------

No	Perangkat	Spesifikasi
1.	Memori PC	Minimum 2gb
2.	Sistem operasi PC	Windows/Linux/GUI Interface
3.	Koneksi Internet	Kecepatan minimum 10MBps
4.	Sistem Operasi <i>Smartphone</i>	Android versi Lollipop (minimum)
5.	Memori <i>Smartphone</i>	Minimum 1gb
6.	Resolusi layar	Minimum 480 x 960 pixels
7.	Layanan Hosting	Memiliki kapasitas minimum 2gb

3.2.1.2.2 Analisis Metode Geofencing

Metode geofencing digunakan dalam fitur navigasi / monitoring posisi bus. Secara sederhana, metode ini melakukan *filter* terhadap suatu objek berdasarkan radius antara suatu titik pusat dan objek tersebut. Pada sistem ini, metode Geofencing ditunjang menggunakan persamaan *havershine* untuk melakukan kalkulasi radius antara dua titik. Berikut ini analisa perhitungan *havershine* untuk menentukan radius :

Tabel III-6 Data Simulasi

No	Posisi <i>center of point</i>		Posisi objek	
	Latitude (lat1)	Longitude (long1)	Latitude (lat2)	Longitude (long2)
1.	-6.907612	107.649643	-6.906965	107.642792

$$\Delta latitude = lat2 - lat1 = -6.906965 - (-6.907612) = \dots$$

$$\Delta longitude = long2 - long1 = 107.642792 - 107.649643 = \dots$$

$$a = \sin^2 \frac{\Delta latitude}{2} + \cos(latitude1) \cos(latitude2) \sin^2 \left(\frac{\Delta longitude}{2} \right) = \dots$$

$$c = 2 \operatorname{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R \times c$$

3.2.2 Analisis Kebutuhan Non-Fungsional

Berikut ini beberapa kebutuhan sistem yang harus terpenuhi, namun diluar lingkup kerja dari fungsional utama, terbagi menjadi dua yaitu kebutuhan data dan pengguna.

3.2.2.1 Analisis Kebutuhan Data

Dalam perancangan sistem bus pintar, data yang diproses / diolah secara umum terdapat tiga jenis, yaitu data penumpang, data pemesanan tiket *online*, dan data monitoring bus. Data ini memiliki peran penting dalam sistem, terutama untuk proses navigasi bus, tiketting, serta manajemen penumpang. Berikut tabel perancangan data :

Tabel III-7 Kebutuhan Non-Fungsional

No	Nama Field	Tipe	Deskripsi
1.	Data penumpang	Teks	Data ini diambil dari <i>input</i> user saat mengisi form pendaftaran member bsmart. Form registrasi dapat diakses melalui <i>website</i> atau aplikasi android.
2.	Data pemesanan tiket <i>online</i>	Teks	Data ini didapatkan dari <i>input</i> user saat akan melakukan perjalanan, user harus melakukan pemesanan tiket terlebih dahulu. Pemesanan tiket <i>online</i> dapat dilakukan melalui <i>website</i> atau aplikasi Android.
3.	Data monitoring bus	Teks	Data monitoring posisi dan kecepatan bus didapatkan dari <i>input</i> perangkat bsmart yang terpasang pada bus. Data ini akan digunakan untuk menentukan bus

			terdekat (geofencing).
--	--	--	------------------------

3.2.2.2 Analisis Pengguna

Sistem Bsmart memiliki tiga jenis aktor / pengguna yang masing-masing memiliki hak akses dan peran yang berbeda terhadap sistem. Berikut ini penjelasan mengenai peran aktor-aktor terhadap sistem.

Tabel III-8 Peran aktor terhadap sistem

No	Aktor	Aplikasi / Platform	Fungsi
1.	Administrator	Website	Admin memiliki akses penuh terhadap konten, pengisian saldo pembayaran, dan sirkulasi data sistem aplikasi Bsmart yang diakses melalui <i>website</i> . Admin berhak melakukan management data (CRUD) keseluruhan.
2.	Dishub	Website	Dishub merupakan dinas yang terkait dalam operasional bus Damri. Aktor ini memiliki akses penuh terhadap management dan sirkulasi data melalui <i>website</i> , namun dibatasi tidak diberikan akses untuk management saldo pembayaran.
3.	Pengguna 1 (Masyarakat)	Aplikasi Android user dan Website	Masyarakat memiliki akses terhadap sistem melalui <i>website</i> dan aplikasi android khusus masyarakat. Pada <i>website</i> masyarakat hanya dapat melakukan pemesanan tiket, melihat informasi, dan memantau

No	Aktor	Aplikasi / Platform	Fungsi
			posisi bus. Sedangkan pada aplikasi android, digunakan untuk melakukan pemesanan tiket, melaporkan kejadian / kecelakaan, panggilan darurat, melihat berita, serta informasi operasional bus.
4.	Pengguna 2 (Pengendara Bus)	Aplikasi Android khusus <i>driver</i>	Sopir bus dapat menggunakan aplikasi android khusus <i>driver</i> yang digunakan untuk validasi tiket menggunakan sistem OTP, melihat peta digital, serta informasi operasional bus damri.

3.3 Perancangan Perangkat Keras

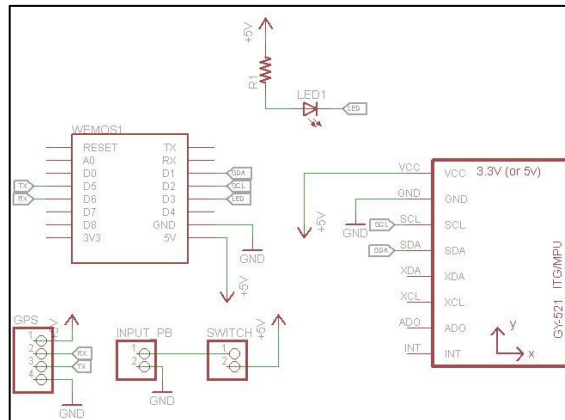
Perancangan perangkat keras membahas mengenai proses desain skematik rangkaian hingga menjadi *board circuit*, konfigurasi alokasi pin, serta desain *mockup* perangkat keras.

3.3.1 Perancangan Elektrikal

Perancangan elektrikal didesain menggunakan aplikasi Eagle. Aplikasi ini digunakan untuk merancang papan sirkuit (*circuit board*) yang digunakan untuk pemasangan komponen elektrikal, mikrokontroller, dan modul-modul.

3.3.1.1 Skema Rangkaian

Skema rangkaian merupakan *blueprint* papan sirkuit yang akan dibuat. Isinya berupa layout yang berisi simbol-simbol komponen beserta konfigurasi pin yang akan digunakan. Gambar dibawah ini adalah desain skematik rangkaian yang telah dibuat :



Gambar III-5 Skematik Rangkaian

3.3.1.2 Alokasi Pin I/O Mikrokontroler

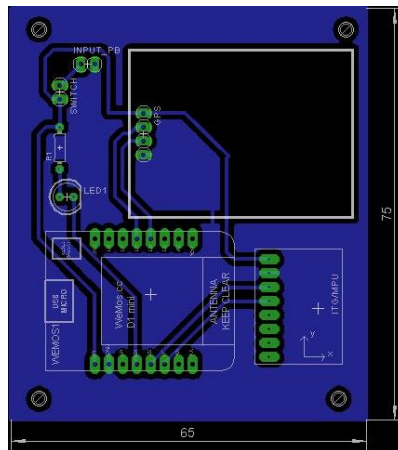
Mikrokontroler Wemos D1 Mini memiliki 9 pin digital, 1 pin analog, sepasang pin RX-TX, serta 2 pin tegangan sumber 5 volt dan 3.3 volt. Berikut ini konfigurasi pin yang digunakan :

Tabel III-9 Konfigurasi Pin

No.	Nama Pin	Keterangan	Cheklis
1.	A0	Pin Analog	
2.	D0	Pin Digital	
3.	D1	Pin digital yang terhubung dengan pin SDA pada modul MPU	✓
4.	D2	Pin digital; yang terhubung dengan pin SCL pada modul MPU	✓
5.	D3	Pin digital yang terhubung dengan kaki LED	✓
6.	D4	Pin Digital	
7.	D5	Pin digital yang terhubung pada kaki TX GPS	✓
8.	D6	Pin digital yang terhubung dengan kaki RX GPS	✓
9.	D7	Pin Digital	
10.	D8	Pin Digital	
11.	TX	Pin Transmitter	
12.	RX	Pin Receiver	

3.3.1.3 Desain Board PCB

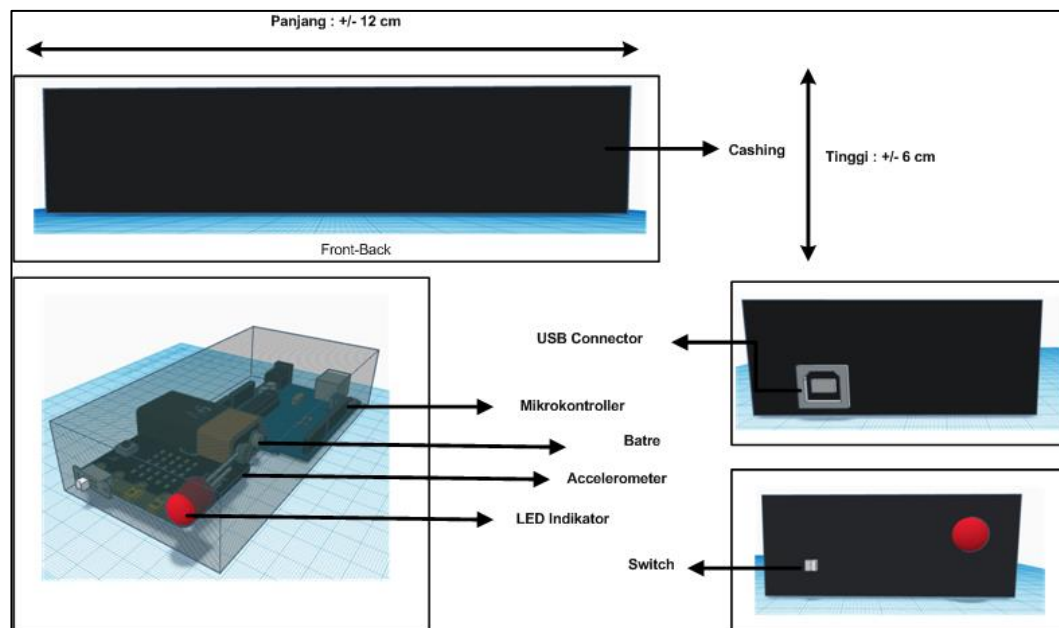
Hasil perancangan skematik rangkaian, dapat di-generate menjadi papan sirkuit elektrik. Pada tahap akhir, dicetak menggunakan *printer* khusus dan campuran beberapa bahan cairan kimia. Berikut ini hasil papan sirkuit yang telah dirancang :



Gambar III-6 Circuit Board

3.3.2 Perancangan Casing

Perancangan *casing* dibutuhkan dalam sistem, hal ini untuk mencegah fungsional alat tidak bekerja dengan baik dikarenakan faktor lingkungan, seperti debu. Bahan baku yang digunakan untuk pembuatan *casing* adalah plastik, teksturnya yang kuat dan ringan, cocok untuk mendukung mobilitas perangkat saat digunakan pada kendaraan. Berikut ini hasil perancangan desain 3D dengan menggunakan *software 3D design* Tinkercad dari berbagaimacam perspektif :



Gambar III-7 3D Design Device

3.4 Perancangan Perangkat Lunak

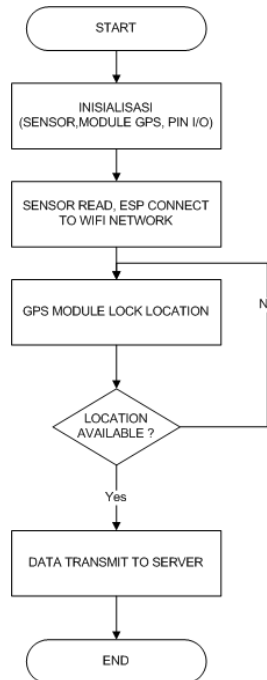
Perancangan perangkat lunak akan membahas bagaimana proses sistem perangkat lunak dibangun. Mulai dari memodelkan sistem dalam diagram UML, algoritma fungsional sistem, hingga perancangan antarmuka. Perangkat lunak yang dibangun yaitu sistem *website* dan aplikasi *mobile*.

3.4.1 Perangkat Lunak Terstruktur

Dalam perancangan perangkat lunak terstruktur digunakan dalam proses perancangan program mikrokontroler. Perancangan algoritma program bertujuan untuk memastikan bahwa fungsional, *output* program sesuai dengan tujuan penelitian. Program mikrokontroler dibuat dengan menggunakan bahasa C, dan *compiler* Arduino, serta *tools* Arduino IDE. Arduino dipilih karena penggunaanya yang mudah dan umum digunakan, sehingga referensi belajar / tutorial mudah didapatkan.

3.4.1.1 Diagram Alir Program Mikrokontroller

Berikut ini diagram alir yang menggambarkan fungsional mikrokontroller dan sensor-sensor :



Gambar III-8 Diagram Alir Mikrokontroller

Berikut di bawah ini penjelasan mengenai diagram alir di atas :

Tabel III-10 Keterangan Diagram Alir

Blok	Keterangan
START	Memulai proses kerja mikrokontroller
INISIALISASI	Tahap pertama, mikrokontroller akan melakukan inisialisasi sensor, modul, dan komponen elektronika yang terhubung melalui pin I/O.
SENSOR READ, ESP CONNECT TO NETWORK	Sensor mulai melakukan proses <i>sensing</i> (membaca parameter fisik dan mengubah dalam besaran elektrik), serta Wemos terkoneksi dengan jaringan <i>wireless</i> sesuai SSID yang telah ditentukan.

GPS LOCK LOCATION	Jika GPS berhasil <i>lock</i> posisi, maka data siap dikirimkan ke server, namun apabila data lokasi <i>null</i> , maka GPS akan <i>request</i> kembali hingga mendapatkan data lokasi (<i>latitude – longitude</i>)
DATA TRANSMIT TO SERVER	Data akan dikirim ke server
END	Proses Selesai

3.4.1.2 Format Data

Mikrokontroler akan mengirimkan data sensor-sensor menggunakan format data yang telah ditentukan, sebagai berikut :

Tabel III-11 Format Pengiriman Data

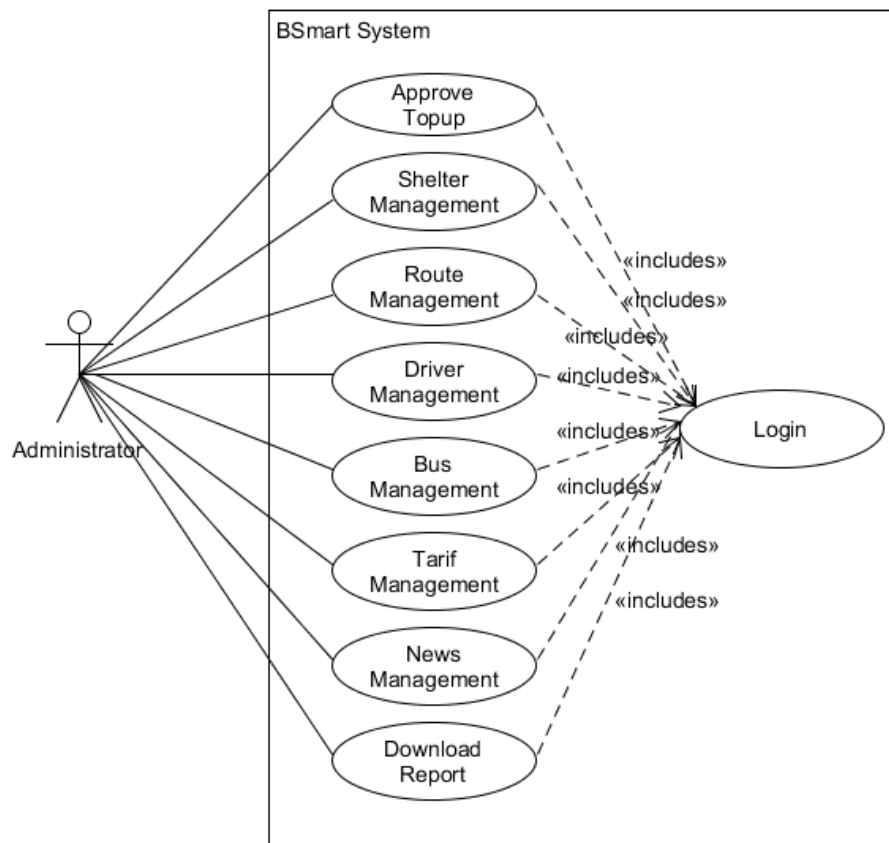
Format Data Pengiriman							
Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
Id_bus	Latitude	longitude	speed	angle_x	angle_y	route	Date

3.4.2 Perangkat Lunak Orientasi Objek

Dalam perancangan perangkat lunak, juga digunakan pendekatan objek untuk membangun sistem aplikasi *website*, menggunakan *framework* PHP (Laravel). Perancangan algoritma pada aplikasi bertujuan untuk memodelkan tahapan kerja setiap elemen sistem. Aplikasi yang dirancang menggunakan pemrograman pendekatan objek (OOP), sehingga dimodelkan dalam bentuk diagram UML sebagai berikut :

3.4.2.1 Use Case Diagram dan Skenario

Dibawah ini terdapat beberapa use case yang dibuat untuk memodelkan perilaku sistem dan aktor. Dalam sistem aplikasi, terdapat lima aktor yang memiliki hak akses yang berbeda-beda, digambarkan dalam diagram use case berikut :



Gambar III-9 Use Case Administrator

Aktor administrator secara umum memiliki hak akses untuk mengelola seluruh informasi yang tersimpan di *database*. Berikut ini skenario dari model diagram use case untuk aktor administrator :

1. Nama *use case* : Login
 Deskripsi : Menjelaskan proses login pada sistem website
 Aktor : Administrator

Tabel III-12 Skenario Use Case Login Aktor Administrator

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Melakukan <i>input</i> email dan password pada form login	
2. Klik tombol Login	3. Validasi akun email dan

Aksi Aktor	Reaksi Sistem
	password di <i>database</i>
	4. Data akun tersedia pada tabel <i>database</i> .
	5. Muncul pesan pemberitahuan “Login Berhasil”
	6. Masuk ke halaman <i>dashboard</i> (halaman utama <i>website</i>)
Skenario Alternatif	
1. Melakukan <i>input</i> email dan password pada form login	
2. Klik tombol Login	3. Validasi akun email dan password di <i>database</i> .
	4. Data akun tidak tersedia (karena salah <i>input</i> atau tidak terdaftar)
	5. Muncul pesan pemberitahuan “Login Gagal”
6. Melakukan <i>input</i> email dan password pada form login	
	7. Validasi akun email dan password di <i>database</i>
	8. Data akun tersedia pada tabel <i>database</i> .
	9. Muncul pesan pemberitahuan “Login Berhasil”
	10. Masuk ke halaman <i>dashboard</i>

2. Nama *use case* : Approve Topup

Deskripsi : Menjelaskan proses pengelolaan pengajuan topup saldo, yang terfokus pada proses menyetujui / menolak pengajuan pengisian topup.

Aktor : Administrator

Tabel III-13 Skenario Use Case Kelola Topup Aktor Administrator

Aksi Aktor	Reaksi Sistem
Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Topup Report	
	3. Menampilkan data user yang mengajukan pengisian topup
4. Memilih tombol detail topup untuk melihat data topup yang status request "Unapproved"	
	5. Menampilkan data detail bukti pengajuan dan pembayaran topup saldo
6. Memilih tombol "Approved" untuk menyetujui permintaan pengisian topup	
	7. Muncul notifikasi "Approve Sukses"
	8. Kembali ke data detail topup
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Topup Report	
	3. Menampilkan data user yang mengajukan pengisian topup
4. Memilih tombol detail topup untuk melihat data topup yang status request "Unapproved"	
	5. Menampilkan data detail bukti pengajuan dan pembayaran topup saldo

Aksi Aktor	Reaksi Sistem
6. Memilih tombol “Approved” untuk menyetujui permintaan pengisian topup	
	7. Muncul notifikasi “Approve Gagal, data tidak ditemukan “
	8. Kembali ke data detail topup

3. Nama *use case* : Shelter Management
 Deskripsi : Management data shelter
 Aktor : Administrator

Tabel III-14 Skenario Use Case CRUD Shelter Aktor Administrator

Aksi Aktor	Reaksi Sistem
Tambah Shelter Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Shelter	
	3. Menampilkan daftar data shelter yang telah tersimpan di <i>database</i>
4. Mengisi data pada form, dan memilih tombol Add Shelter	
	5. Muncul notifikasi “Shelter Data Sukses Disimpan”
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Setting Shelter	
	3. Menampilkan daftar data shelter yang telah tersimpan di <i>database</i>
4. Data form tidak diisi semua, dan	

Aksi Aktor	Reaksi Sistem
memilih tombol Add Shelter	
	5. Muncul notifikasi “Data harus diinput semua”
	6. Kembali ke form pengisian
Edit Shelter Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Shelter	
	3. Menampilkan daftar data shelter yang telah tersimpan di <i>database</i>
4. Memilih link Edit	
	5. Muncul form ubah data shelter.
6. Mengisi form, dan pilih tombol Save	
	7. Muncul notifikasi “Data Berhasil disimpan”
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Setting Shelter	
	3. Menampilkan daftar data shelter yang telah tersimpan di <i>database</i>
4. Memilih link Edit	
	5. Muncul form ubah data shelter
6. Form tidak diisi seluruhnya	
	7. Notifikasi “Data gagal disimpan”
	8. Kembali ke form input
Delete Shelter Skenario Normal	
	1. Menampilkan menu-menu

Aksi Aktor	Reaksi Sistem
2. Memilih menu Setting Shelter	
	3. Menampilkan daftar data shelter yang telah tersimpan di <i>database</i>
4. Memilih link Delete	
	5. Muncul pesan konfirmasi penghapusan data.
6. Memilih tombol “Yes / Delete”	
	7. Muncul notifikasi “Data Berhasil dihapus”

4. Nama *use case* : Route Management
 Deskripsi : Management data route
 Aktor : Administrator

Tabel III-15 Skenario Use Case CRUD Rute Aktor Administrator

Aksi Aktor	Reaksi Sistem
Tambah Rute Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Route	
	3. Menampilkan daftar data ruter yang telah tersimpan di <i>database</i>
4. Memilih tombol Add Route	
	5. Menampilkan form <i>input</i> data Rute
6. Memilih tombol Save	
	7. Muncul notifikasi “Route Data Sukses Disimpan”
	8. Menampilkan daftar data Rute

Aksi Aktor	Reaksi Sistem
	yang tersimpan di <i>database</i>
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Setting Route	
	3. Menampilkan daftar data ruter yang telah tersimpan di <i>database</i>
4. Memilih tombol Add Route	
	5. Menampilkan form <i>input</i> data Rute
6. Terdapat data yang kosong, memilih tombol Save	
	7. Muncul notifikasi “Route Data Gagal Disimpan”
	8. Menampilkan daftar data Rute yang tersimpan di <i>database</i>
Edit Rute Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Route	
	3. Menampilkan daftar data route yang telah tersimpan di <i>database</i>
4. Memilih link Edit	
	5. Muncul form ubah data
6. Pengisian form, dan pilih tombol “Save”	
	7. Muncul notifikasi “Data Berhasil disimpan”
Skenario Alternatif	

Aksi Aktor	Reaksi Sistem
	1. Menampilkan menu-menu
2. Memilih menu Setting Route	
	3. Menampilkan daftar data route yang telah tersimpan di <i>database</i>
4. Memilih link Edit	
	5. Muncul form ubah data
6. Form tidak diisi seluruhnya, dan pilih tombol "Save"	
	7. Muncul notifikasi "Data Gagal disimpan"
	8. Kembali ke form input
Hapus Rute Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Route	
	3. Menampilkan daftar data route yang telah tersimpan di <i>database</i>
4. Memilih link Delete	
	5. Muncul pesan konfirmasi penghapusan data
6. Pilih "Yes / Delete"	
	7. Muncul notifikasi "Data Berhasil dihapus"

5. Nama *use case* : Driver Management
 Deskripsi : Management data sopir bus
 Aktor : Administrator

Tabel III-16 Skenario Use Case CRUD data Driver Aktor Administrator

Aksi Aktor	Reaksi Sistem
------------	---------------

Aksi Aktor	Reaksi Sistem
Tambah Driver Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Driver	
	3. Menampilkan daftar data driver yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Pengisian form, dan memilih tombol “Add Driver”	
	5. Muncul notifikasi “Driver Data Sukses Disimpan”
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Setting Driver	
	3. Menampilkan daftar data driver yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Pengisian form tidak diisi seluruhnya, dan memilih tombol “Add Driver”	
	5. Muncul notifikasi “Driver Data Gagal Disimpan”
	6. Menampilkan kembali form input
Edit Driver Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Driver	
	3. Menampilkan daftar data driver yang telah tersimpan di

Aksi Aktor	Reaksi Sistem
	<i>database</i> , dan form penambahan data
4. Memilih link “Edit”	
	5. Muncul form ubah data
6. Mengisi data, dan memilih tombol “Save”	
	7. Muncul notifikasi “Data berhasil disimpan”
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Setting Driver	
	3. Menampilkan daftar data driver yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Memilih link “Edit”	
	5. Muncul form ubah data
6. Data tidak diisi semua, dan memilih tombol “Save”	
	7. Muncul notifikasi “Data gagal disimpan”
	8. Kembali ke form input
Hapus Driver Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Driver	
	3. Menampilkan daftar data driver yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Memilih link “Delete”	

Aksi Aktor	Reaksi Sistem
	5. Muncul pesan notifikasi konfirmasi
6. Pilih “Yes”	
	7. Muncul notifikasi “Data berhasil dihapus”

6. Nama *use case* : Bus Management
 Deskripsi : Management data bus
 Aktor : Administrator

Tabel III-17 Skenario Use Case CRUD data bus aktor Administrator

Aksi Aktor	Reaksi Sistem
Tambah Bus Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Bus	
	3. Menampilkan daftar data Bus yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Mengisi form data dan pilih tombol “Add Bus”	
	5. Muncul notifikasi “Data berhasil disimpan”
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Setting Bus	
	3. Menampilkan daftar data Bus yang telah tersimpan di <i>database</i> , dan form penambahan data

Aksi Aktor	Reaksi Sistem
4. Mengisi form data tidak diisi seluruhnya, dan pilih tombol “Add Bus”	
	5. Muncul notifikasi “Data gagal disimpan”
	6. Kembali ke form input
Edit Bus Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Bus	
	3. Menampilkan daftar data Bus yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Memilih link “Edit”	
	5. Muncul form edit data bus
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Setting Bus	
	3. Menampilkan daftar data Bus yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Memilih link “Edit”	
	5. Muncul form edit data bus, data tidak diisi semua, lalu pilih “Simpan”
	6. Muncul notifikasi “Gagal simpan data”
	7. Kembali ke form input
Hapus Bus Skenario Normal	

Aksi Aktor	Reaksi Sistem
	1. Menampilkan menu-menu
2. Memilih menu Setting Bus	
	3. Menampilkan daftar data Bus yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Memilih link “Delete”	
	5. Muncul pesan konfirmasi
6. Pilih “Yes”	
	7. Muncul pesan “Data berhasil dihapus”

7. Nama *use case* : News Management
 Deskripsi : Management data News
 Aktor : Administrator

Tabel III-18 Skenario Use Case CRUD data berita aktor Administrator

Aksi Aktor	Reaksi Sistem
Tambah Berita Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting News	
	3. Menampilkan daftar data Berita yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Mengisi form berita, dan pilih tombol “Add News”	
	5. Muncul pesan “Data berhasil disimpan”
Skenario Alternatif	

Aksi Aktor	Reaksi Sistem
	1. Menampilkan menu-menu
2. Memilih menu Setting News	
	3. Menampilkan daftar data Berita yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Form isian tidak diisi seluruhnya, dan pilih tombol “Add News”	
	5. Muncul pesan “Data gagal disimpan”
	6. Kembali ke form input
Edit Berita Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting News	
	3. Menampilkan daftar data Berita yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Pilih link “Edit News”	
	5. Muncul form edit berita
6. Isi form berita, dan pilih tombol “Save News”	
	7. Muncul pesan “Data berhasil disimpan”
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Setting News	
	3. Menampilkan daftar data Berita

Aksi Aktor	Reaksi Sistem
	yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Pilih link “Edit News”	
	5. Muncul form edit berita
6. Isi form berita tapi tidak seluruhnya, dan pilih tombol “Save News”	
	7. Muncul pesan “Data gagal disimpan”
	8. Kembali ke form input
Hapus Berita Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting News	
	3. Menampilkan daftar data Berita yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Pilih link “Delete News”	
	5. Muncul pesan konfirmasi hapus data
6. Pilih “Yes”	
	7. Muncul pesan “Data berhasil dihapus”

8. Nama *use case* : Tarif Management
 Deskripsi : Management data tarif
 Aktor : Administrator

Tabel III-19 Skenario Use Case tambah, hapus data tarif aktor Administrator

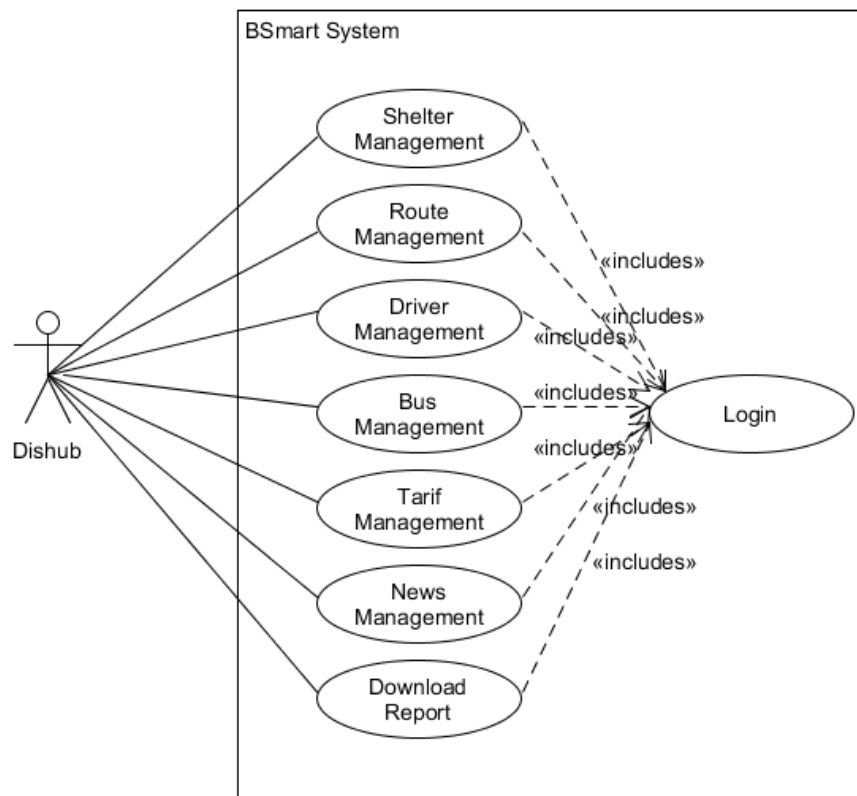
Aksi Aktor	Reaksi Sistem
Tambah Tarif Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Tarif	
	3. Menampilkan daftar data Tarif yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Mengisi Form, pilih tombol "Save"	
	5. Muncul pesan "Data berhasil disimpan"
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Setting Tarif	
	3. Menampilkan daftar data Tarif yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Mengisi Form tapi tidak seluruhnya, pilih tombol "Save"	
	5. Muncul pesan "Data berhasil disimpan"
	6. Kembali ke form input
Hapus Tarif Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Tarif	
	3. Menampilkan daftar data Tarif yang telah tersimpan di <i>database</i> , dan form penambahan data

Aksi Aktor	Reaksi Sistem
4. Pilih link "Delete"	
	5. Muncul pesan konfirmasi hapus data
6. Pilih "Yes"	
	7. Muncul pesan "Data berhasil dihapus"

9. Nama *use case* : Download Report
 Deskripsi : Menjelaskan proses mengunduh report
 Aktor : Administrator

Tabel III-20 Skenario Use Case mengunduh data aktor Administrator

Aksi Aktor	Reaksi Sistem
Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Report	
	3. Menampilkan data shelter yang tersipan di <i>database</i>
4. Pilih tombol "Download PDF"	
	5. Mengunduh report



Gambar III-10 Use Case Dishub

Pada Gambar III-9 di atas, merupakan model UML *use case* untuk aktor Dinas Perhubungan, secara umum memiliki enam buah *action* yang terfokus untuk mengelola laporan, monitoring, dan informasi seputar bus. Berikut ini skenario *use case user* dinas perhubungan :

10. Nama *use case* : Login
 Deskripsi : Menjelaskan proses login pada sistem website
 Aktor : Dishub

Tabel III-21 Skenario Use Case Login Aktor Dishub

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Melakukan <i>input</i> email dan password pada form login	
2. Klik tombol Login	3. Validasi akun email dan

Aksi Aktor	Reaksi Sistem
	password di <i>database</i>
	4. Data akun tersedia pada tabel <i>database</i> .
	5. Muncul pesan pemberitahuan “Login Berhasil”
	6. Masuk ke halaman <i>dashboard</i> (halaman utama <i>website</i>)
Skenario Alternatif	
1. Melakukan <i>input</i> email dan password pada form login	
2. Klik tombol Login	3. Validasi akun email dan password di <i>database</i> .
	4. Data akun tidak tersedia (karena salah <i>input</i> atau tidak terdaftar)
	5. Muncul pesan pemberitahuan “Login Gagal”
6. Melakukan <i>input</i> email dan password pada form login	
	7. Validasi akun email dan password di <i>database</i>
	8. Data akun tersedia pada tabel <i>database</i> .
	9. Muncul pesan pemberitahuan “Login Berhasil”
	10. Masuk ke halaman <i>dashboard</i> (halaman utama <i>website</i>)

11. Nama *use case* : Shelter Management

Deskripsi : Management Shelter

Aktor : Dishub

Tabel III-22 Skenario Use Case Tambah, Edit, Hapus Shelter Aktor Dishub

Aksi Aktor	Reaksi Sistem
Tambah Shelter Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Shelter	
	3. Menampilkan daftar data shelter yang telah tersimpan di <i>database</i>
4. Mengisi data pada form, dan memilih tombol Add Shelter	
	5. Muncul notifikasi “Shelter Data Sukses Disimpan”
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Setting Shelter	
	3. Menampilkan daftar data shelter yang telah tersimpan di <i>database</i>
4. Data form tidak diisi semua, dan memilih tombol Add Shelter	
	5. Muncul notifikasi “Data harus diinput semua”
	6. Kembali ke form pengisian
Edit Shelter Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Shelter	
	3. Menampilkan daftar data shelter yang telah tersimpan di <i>database</i>
4. Memilih link Edit	

Aksi Aktor	Reaksi Sistem
	5. Muncul form ubah data shelter.
6. Mengisi form, dan pilih tombol Save	
	7. Muncul notifikasi “Data Berhasil disimpan”
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Setting Shelter	
	3. Menampilkan daftar data shelter yang telah tersimpan di <i>database</i>
4. Memilih link Edit	
	5. Muncul form ubah data shelter
6. Form tidak diisi seluruhnya	
	7. Notifikasi “Data gagal disimpan”
	8. Kembali ke form input
Delete Shelter Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Shelter	
	3. Menampilkan daftar data shelter yang telah tersimpan di <i>database</i>
4. Memilih link Delete	
	5. Muncul pesan konfirmasi penghapusan data.
6. Memilih tombol “Yes / Delete”	
	7. Muncul notifikasi “Data Berhasil dihapus”

12. Nama *use case* : Route Management
 Deskripsi : Menjelaskan proses tambah, edit, hapus data rute
 Aktor : Dishub

Tabel III-23 Skenario Use Case Tambah, Edit, Hapus Rute Aktor Dishub

Aksi Aktor	Reaksi Sistem
Tambah Rute Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Route	
	3. Menampilkan daftar data ruter yang telah tersimpan di <i>database</i>
4. Memilih tombol Add Route	
	5. Menampilkan form <i>input</i> data Rute
6. Memilih tombol Save	
	7. Muncul notifikasi “Route Data Sukses Disimpan”
	8. Menampilkan daftar data Rute yang tersimpan di <i>database</i>
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Setting Route	
	3. Menampilkan daftar data ruter yang telah tersimpan di <i>database</i>
4. Memilih tombol Add Route	
	5. Menampilkan form <i>input</i> data Rute
6. Terdapat data yang kososng, memilih tombol Save	

Aksi Aktor	Reaksi Sistem
	7. Muncul notifikasi “Route Data Gagal Disimpan”
	8. Menampilkan daftar data Rute yang tersimpan di <i>database</i>
Edit Rute Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Route	
	3. Menampilkan daftar data route yang telah tersimpan di <i>database</i>
4. Memilih link Edit	
	5. Muncul form ubah data
6. Pengisian form, dan pilih tombol “Save”	
	7. Muncul notifikasi “Data Berhasil disimpan”
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Setting Route	
	3. Menampilkan daftar data route yang telah tersimpan di <i>database</i>
4. Memilih link Edit	
	5. Muncul form ubah data
6. Form tidak diisi seluruhnya, dan pilih tombol “Save”	
	7. Muncul notifikasi “Data Gagal disimpan”
	8. Kembali ke form input
Hapus Rute Skenario Normal	

Aksi Aktor	Reaksi Sistem
	1. Menampilkan menu-menu
2. Memilih menu Setting Route	
	3. Menampilkan daftar data route yang telah tersimpan di <i>database</i>
4. Memilih link Delete	
	5. Muncul pesan konfirmasi penghapusan data
6. Pilih "Yes / Delete"	
	7. Muncul notifikasi "Data Berhasil dihapus"

13. Nama *use case* : Driver Management

Deskripsi : Management data Driver

Aktor : Dishub

Tabel III-24 Skenario Use Case tambah, edit, hapus data Driver Aktor Dishub

Aksi Aktor	Reaksi Sistem
Tambah Driver Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Driver	
	3. Menampilkan daftar data driver yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Pengisian form, dan memilih tombol "Add Driver"	
	5. Muncul notifikasi "Driver Data Sukses Disimpan"
Skenario Alternatif	
	1. Menampilkan menu-menu

Aksi Aktor	Reaksi Sistem
2. Memilih menu Setting Driver	
	3. Menampilkan daftar data driver yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Pengisian form tidak diisi seluruhnya, dan memilih tombol “Add Driver”	
	5. Muncul notifikasi “Driver Data Gagal Disimpan”
	6. Menampilkan kembali form input
Edit Driver Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Driver	
	3. Menampilkan daftar data driver yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Memilih link “Edit”	
	5. Muncul form ubah data
6. Mengisi data, dan memilih tombol “Save”	
	7. Muncul notifikasi “Data berhasil disimpan”
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Setting Driver	
	3. Menampilkan daftar data driver yang telah tersimpan di

Aksi Aktor	Reaksi Sistem
	<i>database</i> , dan form penambahan data
4. Memilih link “Edit”	
	5. Muncul form ubah data
6. Data tidak diisi semua, dan memilih tombol “Save”	
	7. Muncul notifikasi “Data gagal disimpan”
	8. Kembali ke form input
Hapus Driver Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Driver	
	3. Menampilkan daftar data driver yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Memilih link “Delete”	
	5. Muncul pesan notifikasi konfirmasi
6. Pilih “Yes”	
	7. Muncul notifikasi “Data berhasil dihapus”

14. Nama *use case* : Bus Management

Deskripsi : Menjelaskan proses tambah, ubah, edit data Bus

Aktor : Dishub

Tabel III-25 Skenario Use Case tambah, ubah, edit data bus aktor Dishub

Aksi Aktor	Reaksi Sistem
Tambah Bus Skenario Normal	

Aksi Aktor	Reaksi Sistem
	1. Menampilkan menu-menu
2. Memilih menu Setting Bus	
	3. Menampilkan daftar data Bus yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Mengisi form data dan pilih tombol “Add Bus”	
	5. Muncul notifikasi “Data berhasil disimpan”
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Setting Bus	
	3. Menampilkan daftar data Bus yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Mengisi form data tidak diisi seluruhnya, dan pilih tombol “Add Bus”	
	5. Muncul notifikasi “Data gagal disimpan”
	6. Kembali ke form input
Edit Bus Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Bus	
	3. Menampilkan daftar data Bus yang telah tersimpan di <i>database</i> , dan form penambahan data

Aksi Aktor	Reaksi Sistem
4. Memilih link “Edit”	
	5. Muncul form edit data bus
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Setting Bus	
	3. Menampilkan daftar data Bus yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Memilih link “Edit”	
	5. Muncul form edit data bus, data tidak diisi semua, lalu pilih “Simpan”
	6. Muncul notifikasi “Gagal simpan data”
	7. Kembali ke form input
Hapus Bus Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Bus	
	3. Menampilkan daftar data Bus yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Memilih link “Delete”	
	5. Muncul pesan konfirmasi
6. Pilih “Yes”	
	7. Muncul pesan “Data berhasil dihapus”

15. Nama *use case* : News Management
 Deskripsi : Management data berita
 Aktor : Dishub

Tabel III-26 Skenario Use Case tambah, ubah, edit data berita aktor Dishub

Aksi Aktor	Reaksi Sistem
Tambah Berita Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting News	
	3. Menampilkan daftar data Berita yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Mengisi form berita, dan pilih tombol "Add News"	
	5. Muncul pesan "Data berhasil disimpan"
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Setting News	
	3. Menampilkan daftar data Berita yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Form isian tidak diisi seluruhnya, dan pilih tombol "Add News"	
	5. Muncul pesan "Data gagal disimpan"

Aksi Aktor	Reaksi Sistem
	6. Kembali ke form input
Edit Berita Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting News	
	3. Menampilkan daftar data Berita yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Pilih link “Edit News”	
	5. Muncul form edit berita
6. Isi form berita, dan pilih tombol “Save News”	
	7. Muncul pesan “Data berhasil disimpan”
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Setting News	
	3. Menampilkan daftar data Berita yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Pilih link “Edit News”	
	5. Muncul form edit berita
6. Isi form berita tapi tidak seluruhnya, dan pilih tombol “Save News”	
	7. Muncul pesan “Data gagal disimpan”
	8. Kembali ke form input
Hapus Berita Skenario Normal	

Aksi Aktor	Reaksi Sistem
	1. Menampilkan menu-menu
2. Memilih menu Setting News	
	3. Menampilkan daftar data Berita yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Pilih link “Delete News”	
	5. Muncul pesan konfirmasi hapus data
6. Pilih “Yes”	
	7. Muncul pesan “Data berhasil dihapus”

16. Nama *use case* : Tarif Management
 Deskripsi : Management data Tarif
 Aktor : Dishub

Tabel III-27 Skenario Use Case tambah, hapus data tarif aktor Dishub

Aksi Aktor	Reaksi Sistem
Tambah Tarif Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Tarif	
	3. Menampilkan daftar data Tarif yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Mengisi Form, pilih tombol “Save”	
	5. Muncul pesan “Data berhasil disimpan”

Aksi Aktor	Reaksi Sistem
Skenario Alternatif	
	1. Menampilkan menu-menu
2. Memilih menu Setting Tarif	
	3. Menampilkan daftar data Tarif yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Mengisi Form tapi tidak seluruhnya, pilih tombol "Save"	
	5. Muncul pesan "Data berhasil disimpan"
	6. Kembali ke form input
Hapus Tarif Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Setting Tarif	
	3. Menampilkan daftar data Tarif yang telah tersimpan di <i>database</i> , dan form penambahan data
4. Pilih link "Delete"	
	5. Muncul pesan konfirmasi hapus data
6. Pilih "Yes"	
	7. Muncul pesan "Data berhasil dihapus"

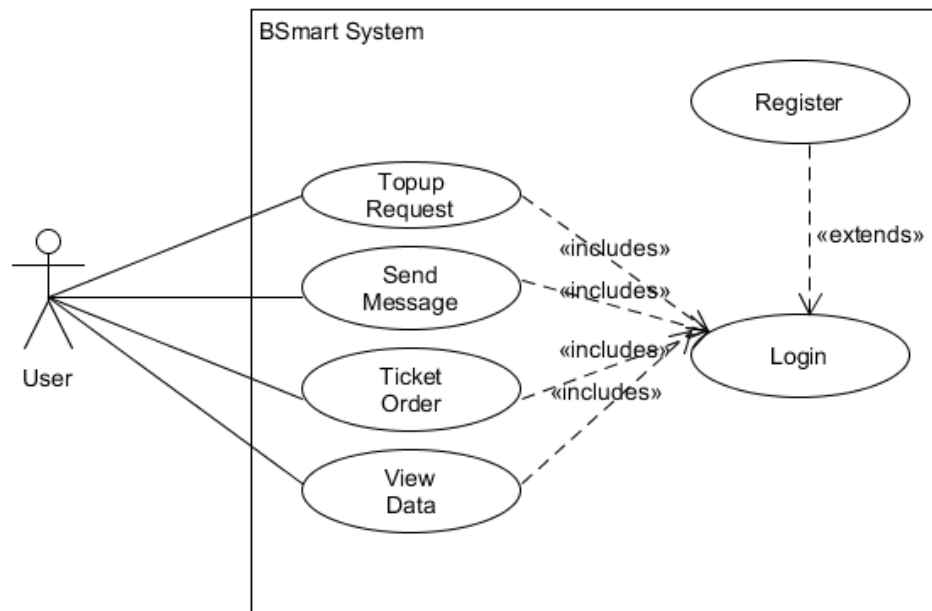
17. Nama *use case* : Download Report

Deskripsi : Menjelaskan proses mengunduh report

Aktor : Dishub

Tabel III-28 Skenario Use Case mengunduh data aktor Dishub

Aksi Aktor	Reaksi Sistem
Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Report	
	3. Menampilkan data shelter yang tersipan di <i>database</i>
4. Pilih tombol "Download PDF"	
	5. Mengunduh report



Gambar III-11 Use Case User

Pada Gambar III-11 di atas, merupakan model *use case* untuk *user*, beberapa menu untuk *user* diberikan pembatasan hak akses hanya bisa melihat (*read only*). Interaksi *input* yang dapat dilakukan oleh *user* diantaranya, menu pemesanan tiket *online*, pengajuan *topup* saldo, serta mengirim pesan ke administrator. Berikut ini skenario *use case user* :

1. Nama *use case* : Login
 Deskripsi : Menjelaskan proses Login
 Aktor : User

Tabel III-29 Skenario Use Case login aktor User

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Melakukan <i>input</i> email dan password pada form login	
2. Klik tombol Login	3. Validasi akun email dan password di <i>database</i>
	4. Data akun tersedia pada tabel <i>database</i> .
	5. Muncul pesan pemberitahuan “Login Berhasil”
	6. Masuk ke halaman <i>dashboard</i> (halaman utama <i>website</i>)
Skenario Alternatif	
7. Melakukan <i>input</i> email dan password pada form login	
8. Klik tombol Login	9. Validasi akun email dan password di <i>database</i> .
	10. Data akun tidak tersedia (karena salah <i>input</i> atau tidak terdaftar)
	11. Muncul pesan pemberitahuan “Login Gagal”
12. Melakukan <i>input</i> email dan password pada form login	
	13. Validasi akun email dan password di <i>database</i>
	14. Data akun tersedia pada tabel <i>database</i> .

Aksi Aktor	Reaksi Sistem
	15. Muncul pesan pemberitahuan "Login Berhasil"
	16. Masuk ke halaman <i>dashboard</i>

2. Nama *use case* : Register
 Deskripsi : Menjelaskan proses registrasi User
 Aktor : User

Tabel III-30 Skenario Use Case register aktor User

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih link "Register"	
2. Melakukan pengisian form register akun, pilih tombol "Register"	
	3. Sistem mengarahkan ke halaman beranda awal website, tanda register berhasil.
Skenario Alternatif	
1. Memilih link "Register"	
2. Melakukan pengisian form register akun, data tidak diisi keseluruhan, lalu pilih tombol "Register"	
	3. Muncul notifikasi kesalahan "Data harus diisi seluruhnya"
	4. Kembali ke form register

3. Nama *use case* : Pengajuan Topup
 Deskripsi : Menjelaskan proses pengajuan User

Aktor : User

Tabel III-31 Skenario Use Case pengajuan topup aktor User

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu “Topup”	
	2. Muncul form pengajuan data topup
3. Pengisian data topup, dan melampirkan gambar bukti pembayaran. Lalu pilih tombol “Topup”	
	4. Muncul pesan notifikasi “Pengajuan topup berhasil”
Skenario Alternatif	
1. Memilih menu “Topup”	
	2. Muncul form pengajuan data topup
3. Data topup tidak diisi keseluruhan, dan melampirkan gambar bukti pembayaran. Lalu pilih tombol “Topup”	
	4. Muncul pesan notifikasi “Pengajuan topup gagal, data harus diisi seluruhnya”
	5. Kembali ke form

4. Nama *use case* : Send Message

Deskripsi : Menjelaskan proses kirim pesan User

Aktor : User

Tabel III-32 Skenario Use Case send message aktor User

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu “Inbox”	
	2. Muncul form inbox
3. Pengisian data pesan yang ditujukan kepada Administrator, lalu pilih “Send Message”	
	4. Muncul pesan notifikasi “Pesan berhasil dikirim”

5. Nama *use case* : View Data
 Deskripsi : Menjelaskan proses melihat data Shelter
 Aktor : User

Tabel III-33 Skenario Use Case melihat data shelter aktor User

Aksi Aktor	Reaksi Sistem
Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Information	
	3. Menampilkan data yang tersimpan di <i>database</i>

6. Nama *use case* : Ticket Order
 Deskripsi : Menjelaskan proses pemesanan tiket online
 Aktor : User

Tabel III-34 Skenario Use Case pemesanan tiket online aktor User

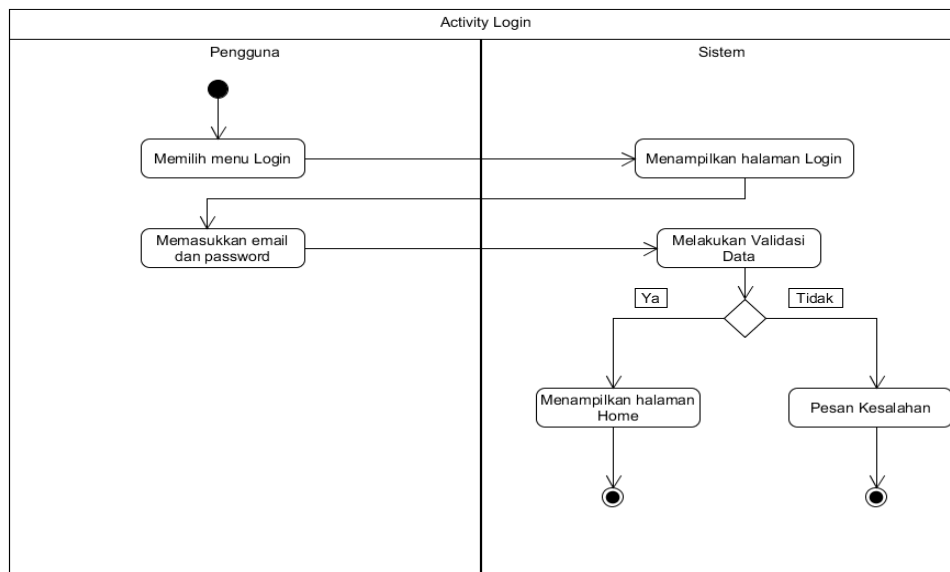
Aksi Aktor	Reaksi Sistem
Skenario Normal	
	1. Menampilkan menu-menu
2. Memilih menu Ticket	

Aksi Aktor	Reaksi Sistem
Reservation	
	3. Menampilkan seluruh shelter
4. Pilih salah satu shelter untuk memulai perjalanan	
	5. Sistem akan mencari bus dalam radius 1.5 km
	6. Data bus dengan radius 1.5 km ditampilkan

3.4.2.2 Activity Diagram

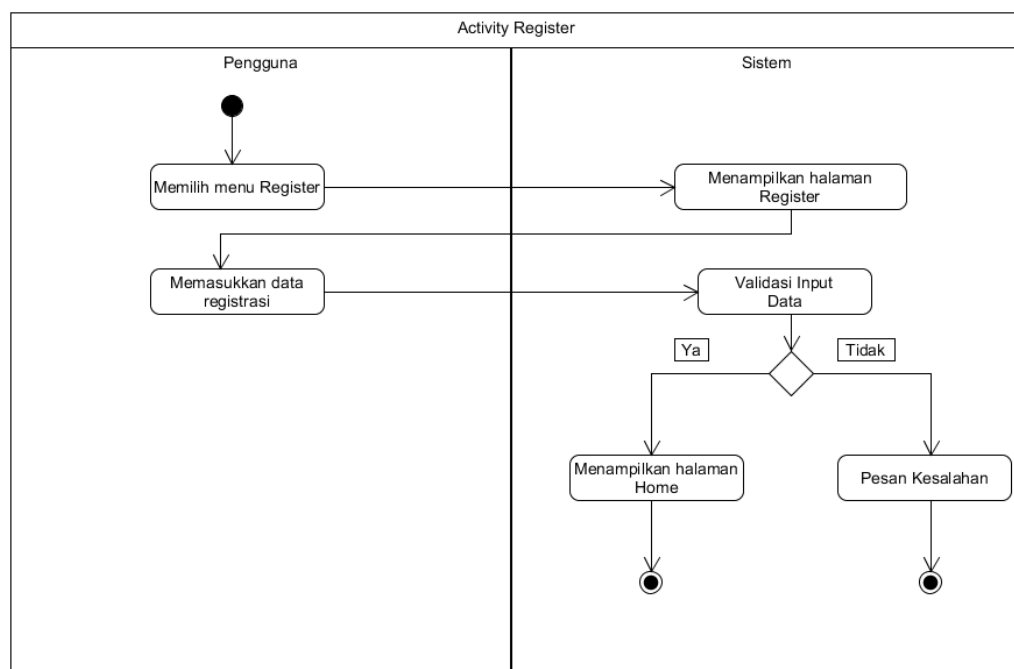
Activity diagram digunakan untuk memodelkan aliran kerja atau *workflow* dari urutan aktifitas suatu proses yang mengacu pada *usecase* diagram. Activity diagram adalah representasi grafis dari *workflow* dari kegiatan dan tindakan bertahap dengan dukungan untuk pilihan, iterasi dan concurrency. Dalam *Unified Modeling Language*, diagram aktivitas dimaksudkan untuk model kedua proses komputasi dan organisasi (yaitu *workflow*). Activity diagram menunjukkan aliran keseluruhan kontrol. Selain itu, activity diagram juga dapat memudahkan para pengembang aplikasi untuk memahami urutan kerja sistem keseluruhan secara detail, dengan demikian perancangan sistem dapat berjalan dengan sebagaimana mestinya. Berikut ini model activity diagram dari setiap use case :

1. Activity Login : pengguna memasukkan email dan password untuk mengakses *dashboard website*.



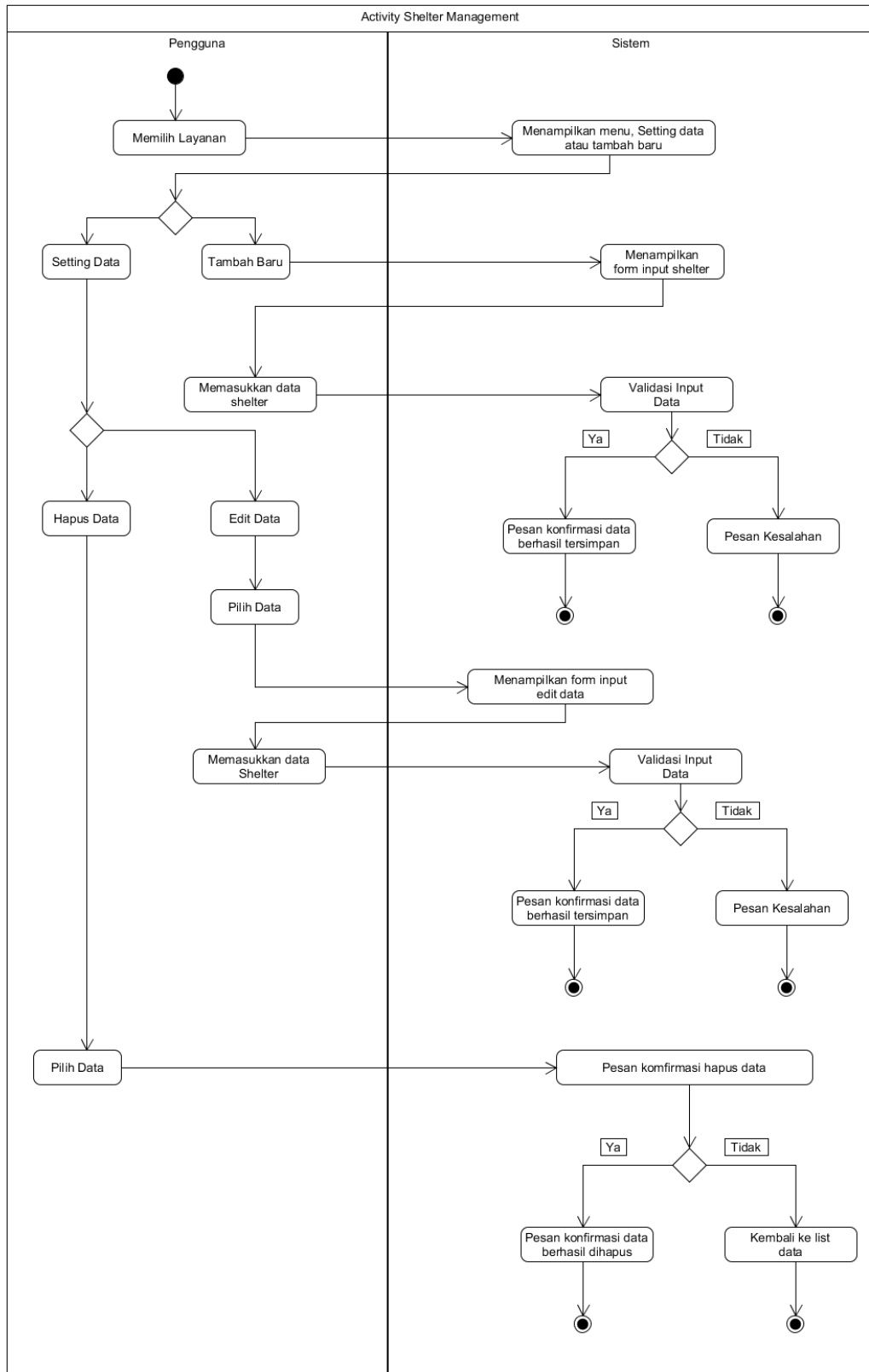
Gambar III-12 Activity Diagram Login

2. Activity Register : Bagi pengguna baru, dapat daftar melakukan registrasi akun baru Bsmart.



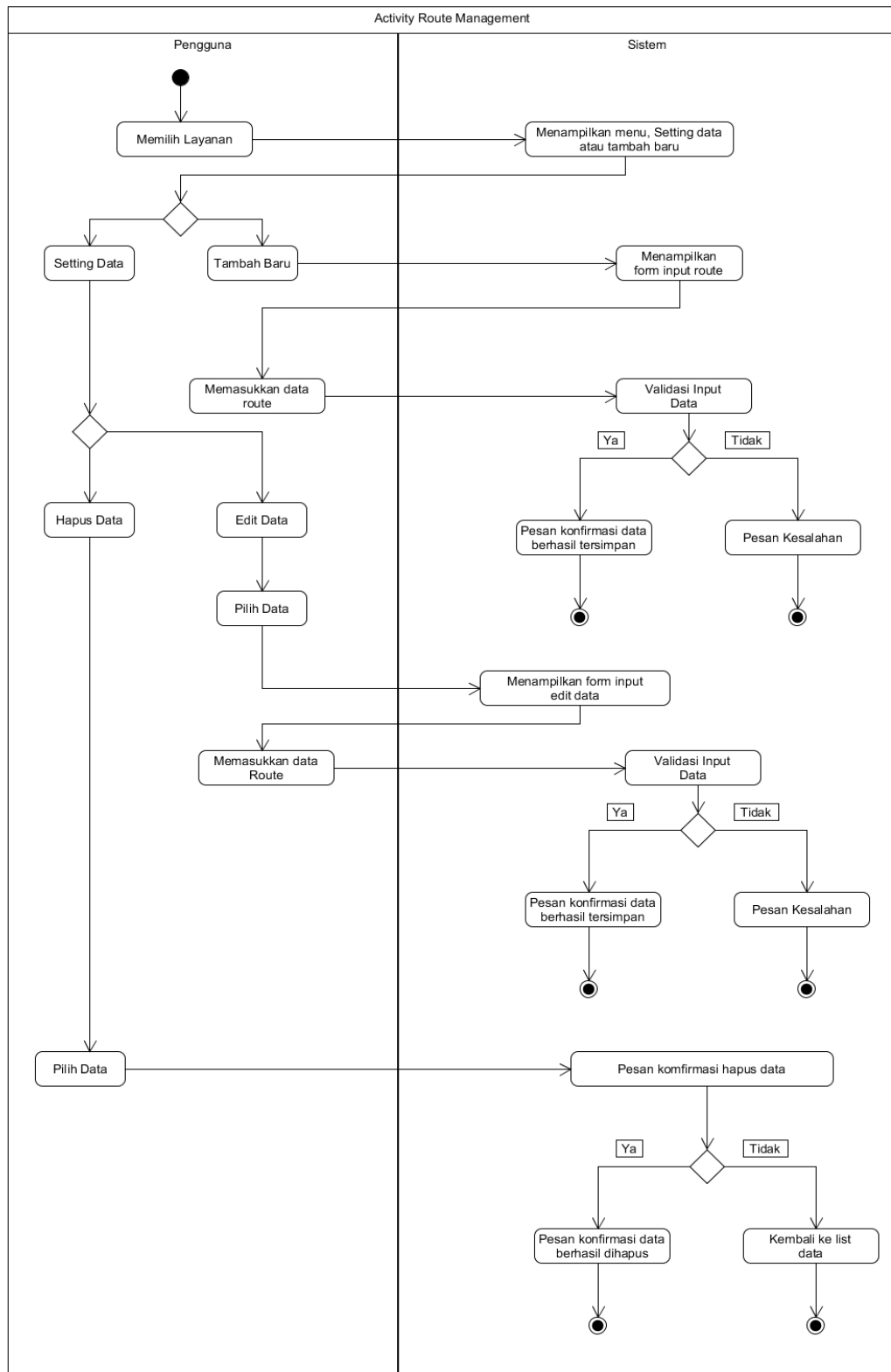
Gambar III-13 Activity Diagram Register

3. Activity Shelter Management : Melakukan penambahan, modifikasi dan menghapus data shelter.



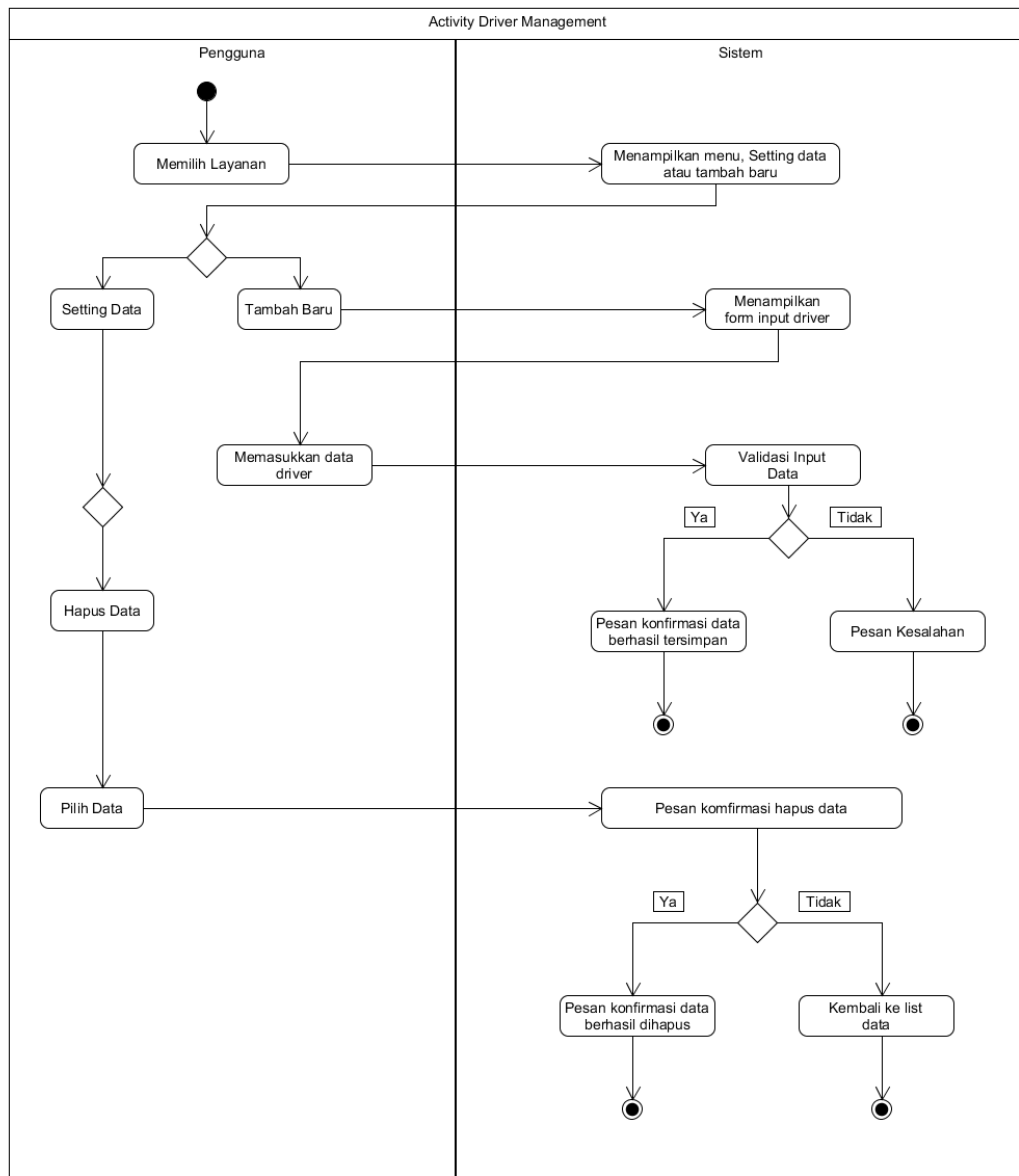
Gambar III-14 Activity Diagram Shelter Management

4. Activity Route Management : Tambah dan modifikasi data route



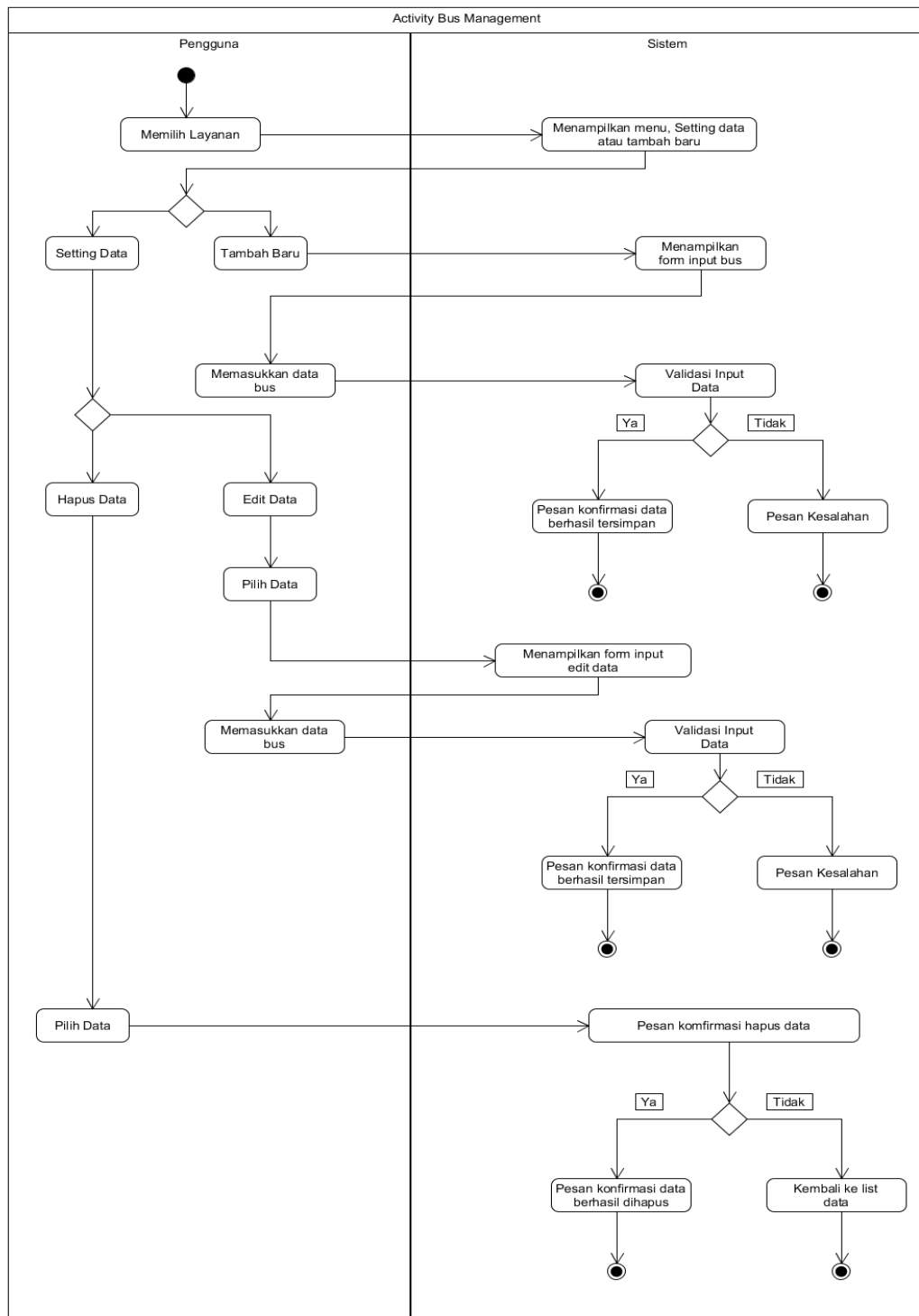
Gambar III-15 Activity Diagram Route Management

5. Activity Driver Management : Tambah dan modifikasi driver



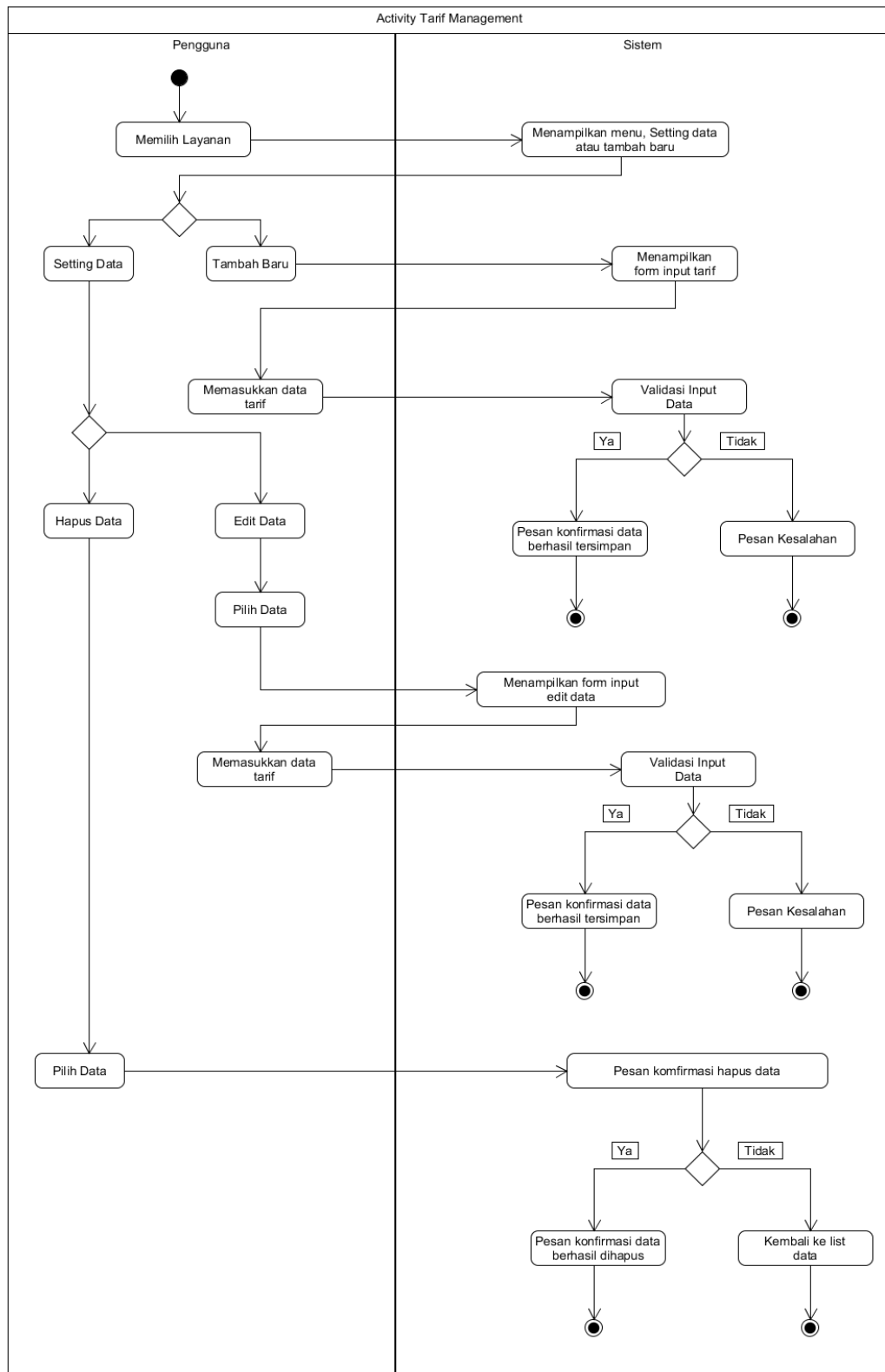
Gambar III-16 Activity Diagram Driver Management

6. Activity Bus Management : Tambah dan modifikasi data bus



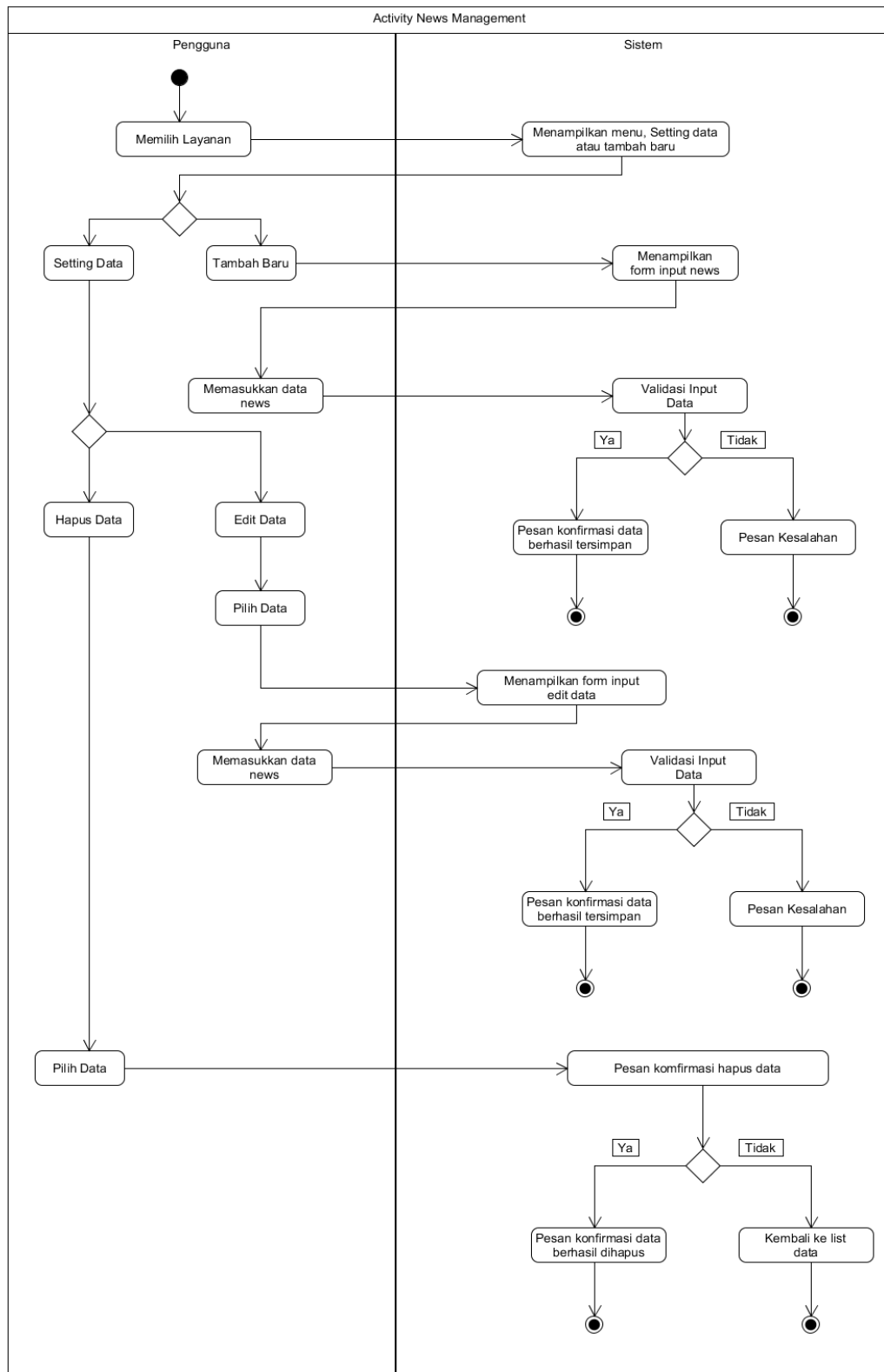
Gambar III-17 Activity Diagram Bus Management

7. Activity Tarif Management : Tambah dan modifikasi data tarif



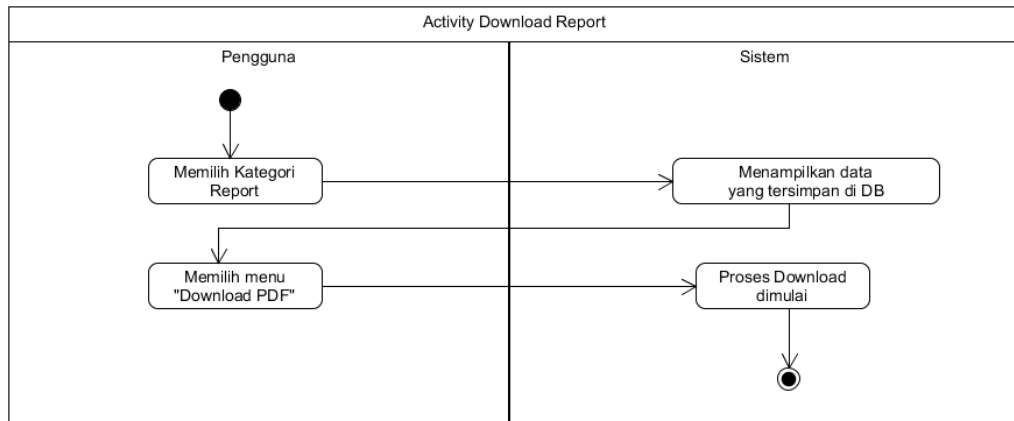
Gambar III-18 Activity Diagram Tarif Management

8. Activity News Management : Tambah dan modifikasi data berita



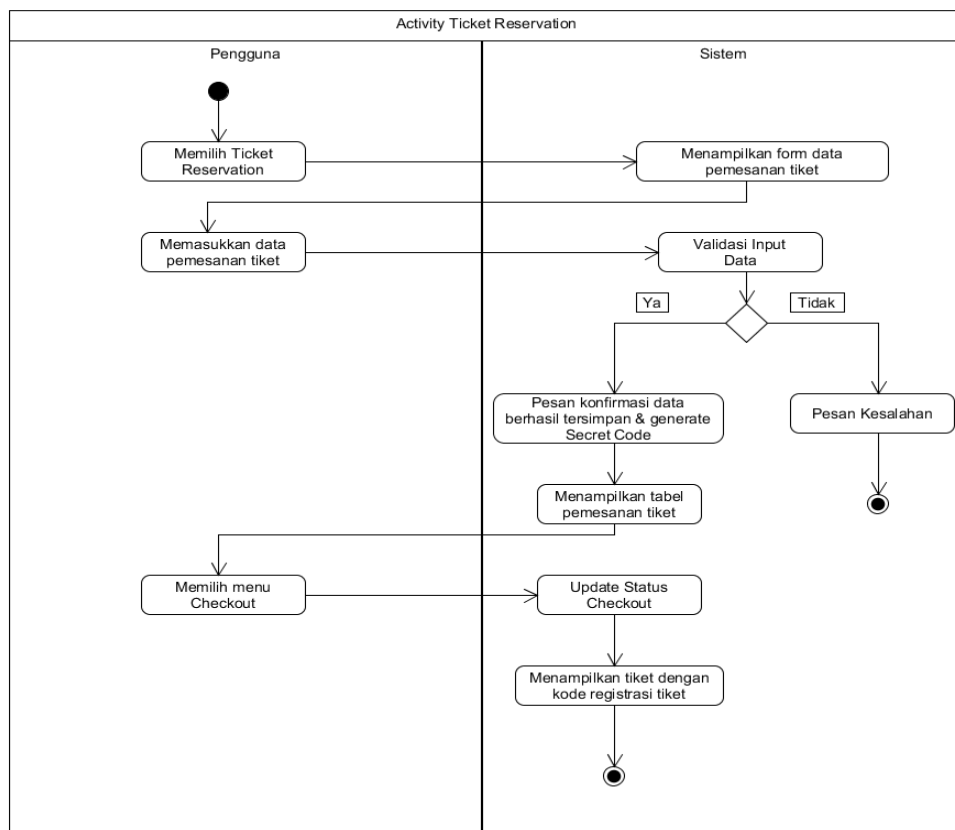
Gambar III-19 Activity Diagram News Management

9. Activity Download Report : Melakukan unduh report setiap kategori data



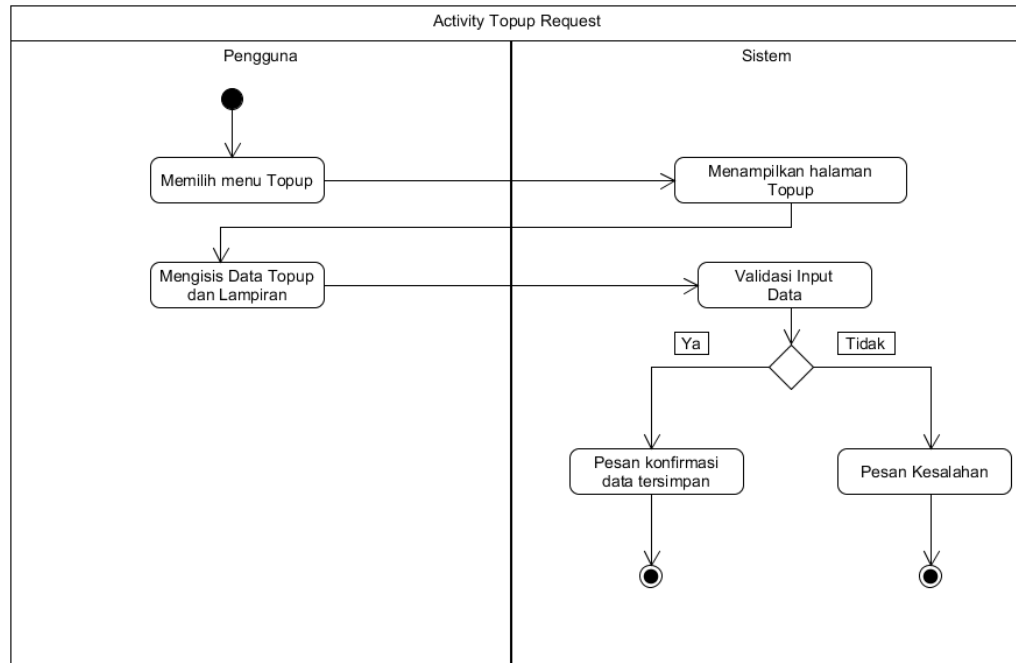
Gambar III-20 Activity Diagram Download Report

10. Activity Ticket Reservation



Gambar III-21 Activity Diagram Ticket Reservation

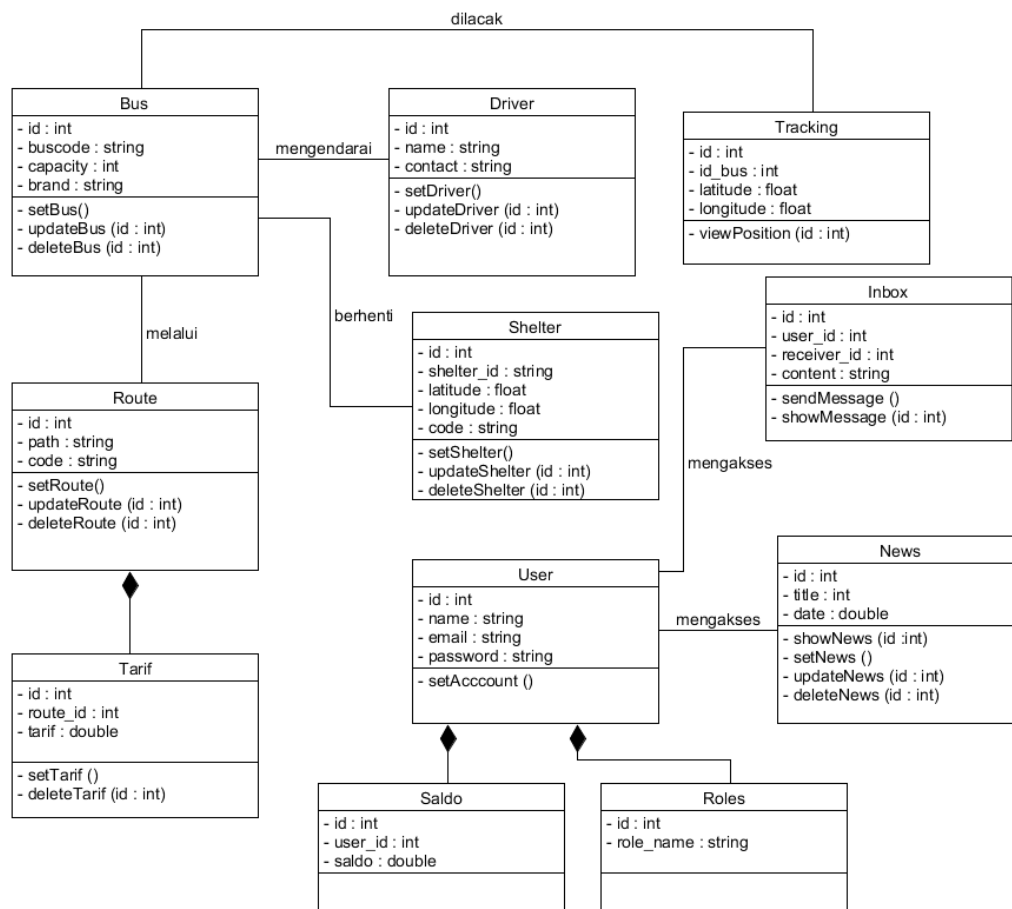
11. Activity Topup Request



Gambar III-22 Activity Diagram Topup Request

3.4.2.3 Class Diagram

Class Diagram merupakan salah satu model dari UML yang digunakan untuk menggambarkan *class* atau *blueprint* objek pada sebuah sistem perangkat lunak berorientasi objek. Selain itu, pada *class diagram* digambarkan juga interaksi antar *class* dalam sebuah konstruksi piranti lunak seperti hubungan asosiasi, agregasi, komposisi, dan inheritance. Dalam penelitian ini, terdapat beberapa *class* yang menangani setiap fungsional fitur sistem. Antar *class* saling berkaitan, berikut ini model *class diagram* sesuai perancangan sistem :

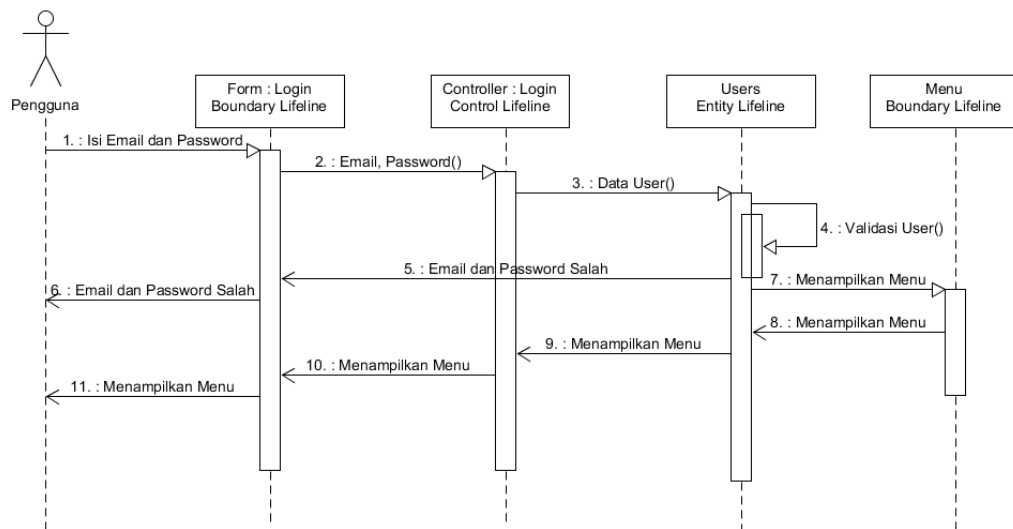


Gambar III-23 Class Diagram

3.4.2.4 Sequence Diagram

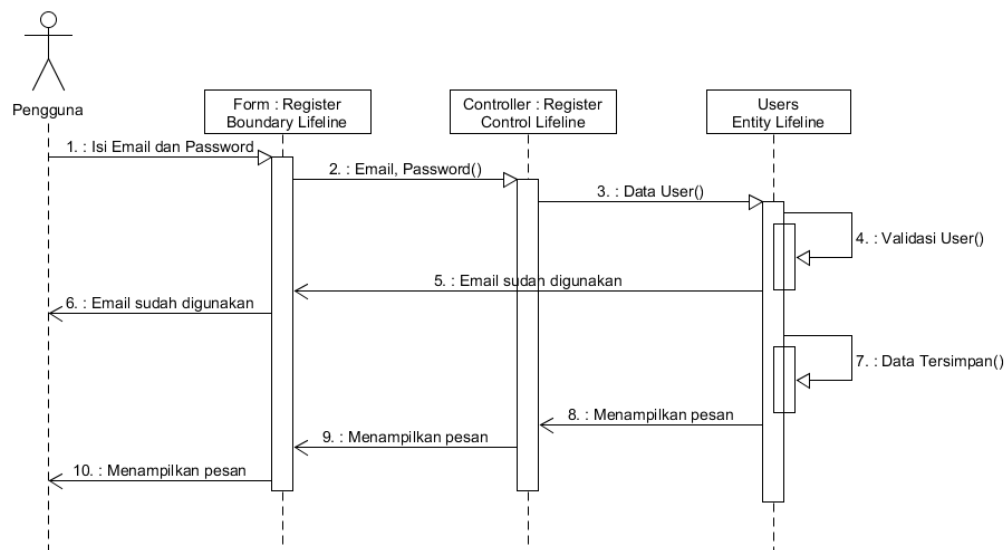
Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh objek – objek yang melakukan suatu tugas atau aksi tertentu. Berikut ini diagram sequence dalam model perancangan sistem Bsmart yang digunakan, diantaranya :

1. Sequence Login : User memberikan masukkan email dan password, sistem akan melakukan validasi ketersediaan data di tabel users, apabila data tersedia dan benar, maka akan muncul menu-menu, jika salah maka sistem akan memberikan *feedback* pesan peringatan kesalahan login.



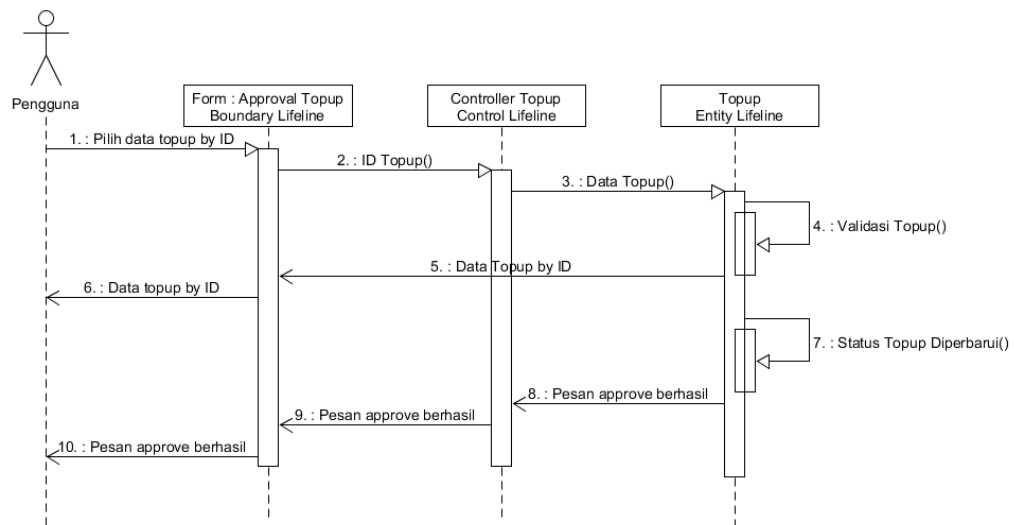
Gambar III-24 Sequence Diagram Login

2. Sequence Register : User baru melakukan registrasi melalui *form* pendaftaran, data akan tersimpan di tabel users.



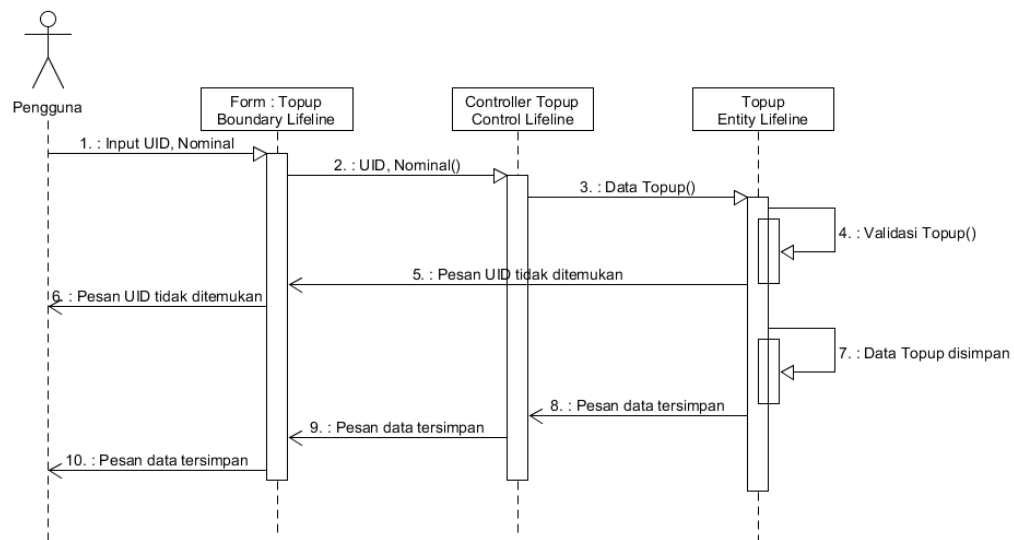
Gambar III-25 Sequence Diagram Register

3. Sequence Topup Approve : Admin memiliki kewenangan untuk melakukan persetujuan pengajuan saldo topup yang dilakukan user, data akan diambil berdasarkan id user dan id topup.



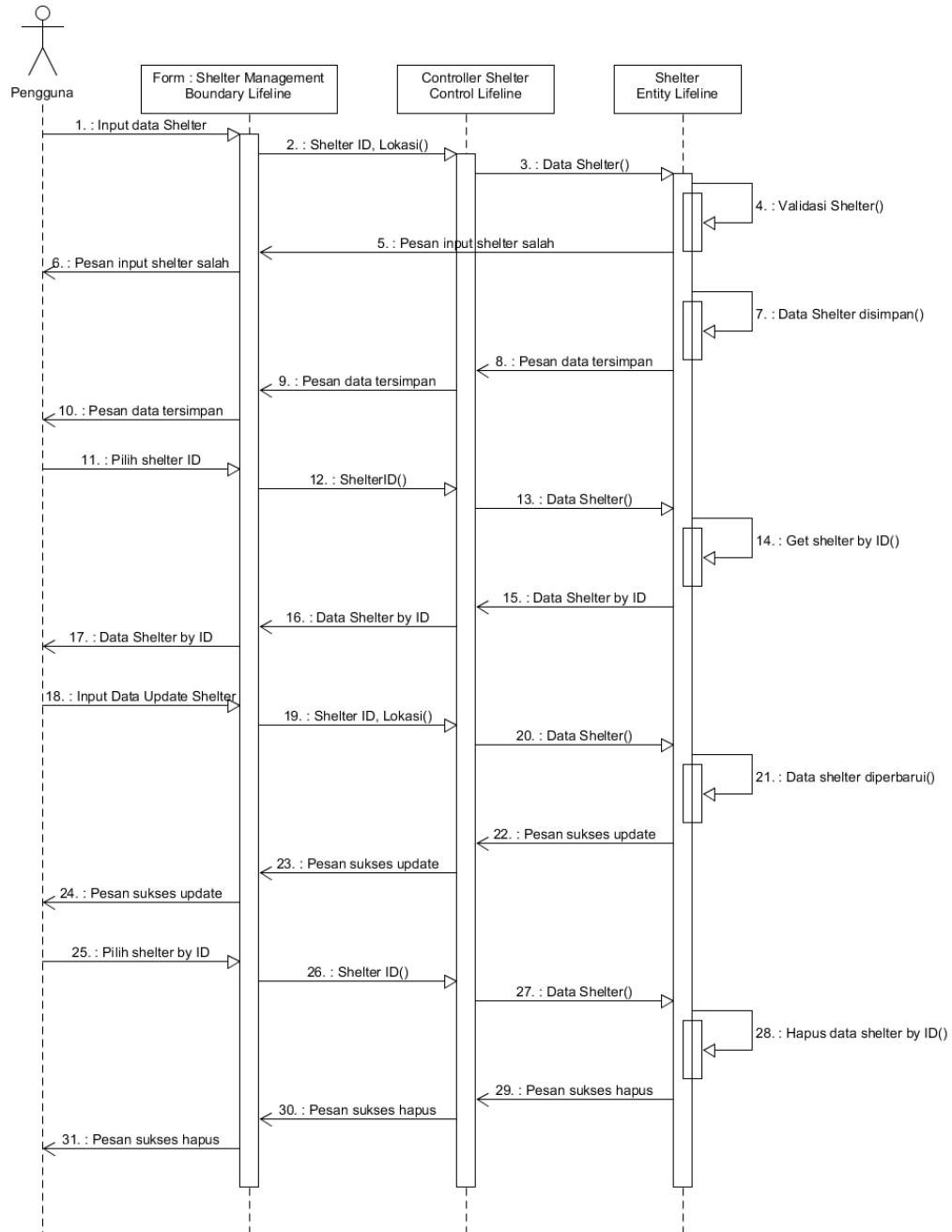
Gambar III-26 Sequence Diagram Approve Topup

4. Sequence Topup Request : User dapat melakukan pengisian saldo topup melalui *form topup request*.



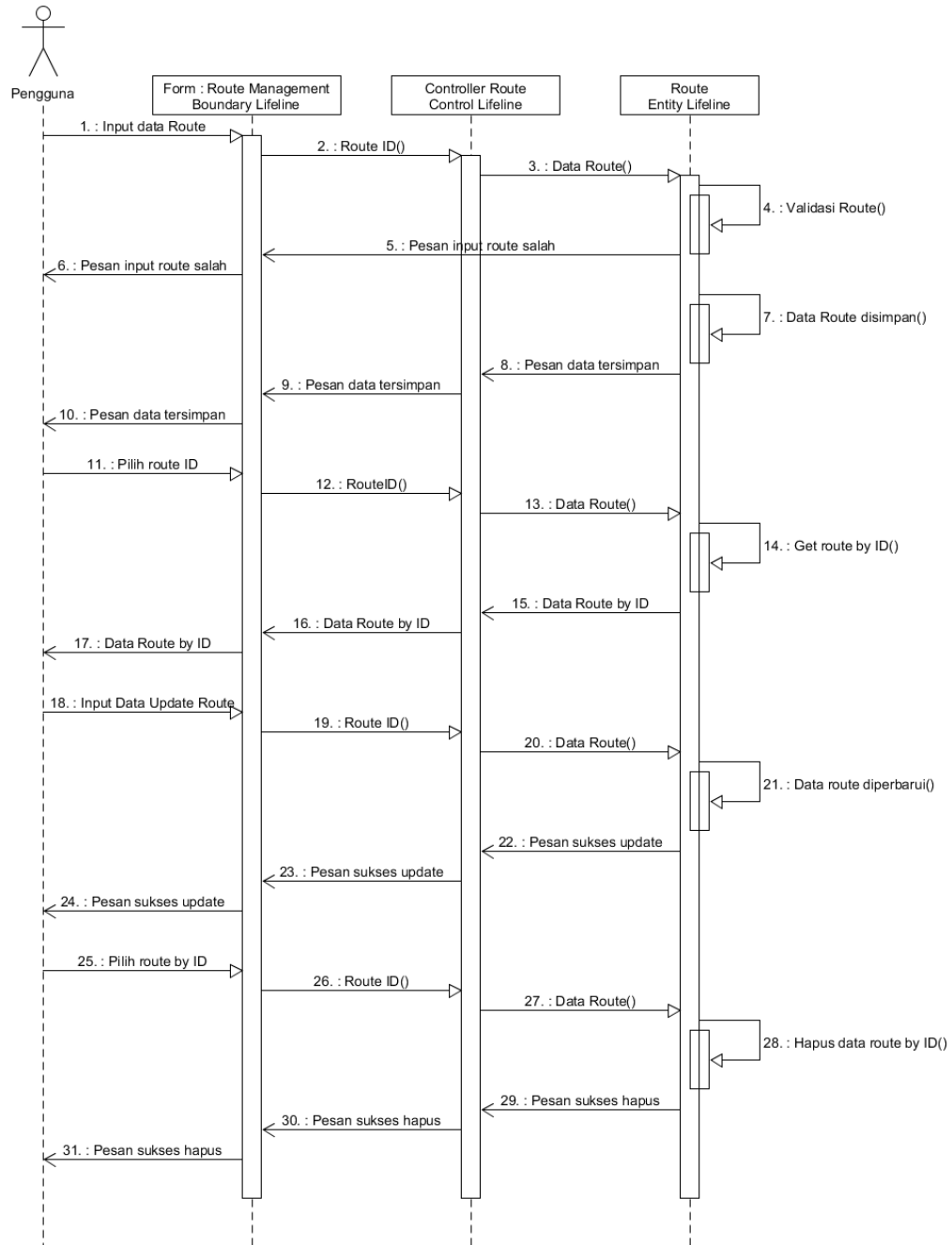
Gambar III-27 Sequence Diagram Topup Request

5. Sequence Shelter Management : Admin dapat melakukan penambahan dan modifikasi data shelter yang tersimpan pada tabel shelter.



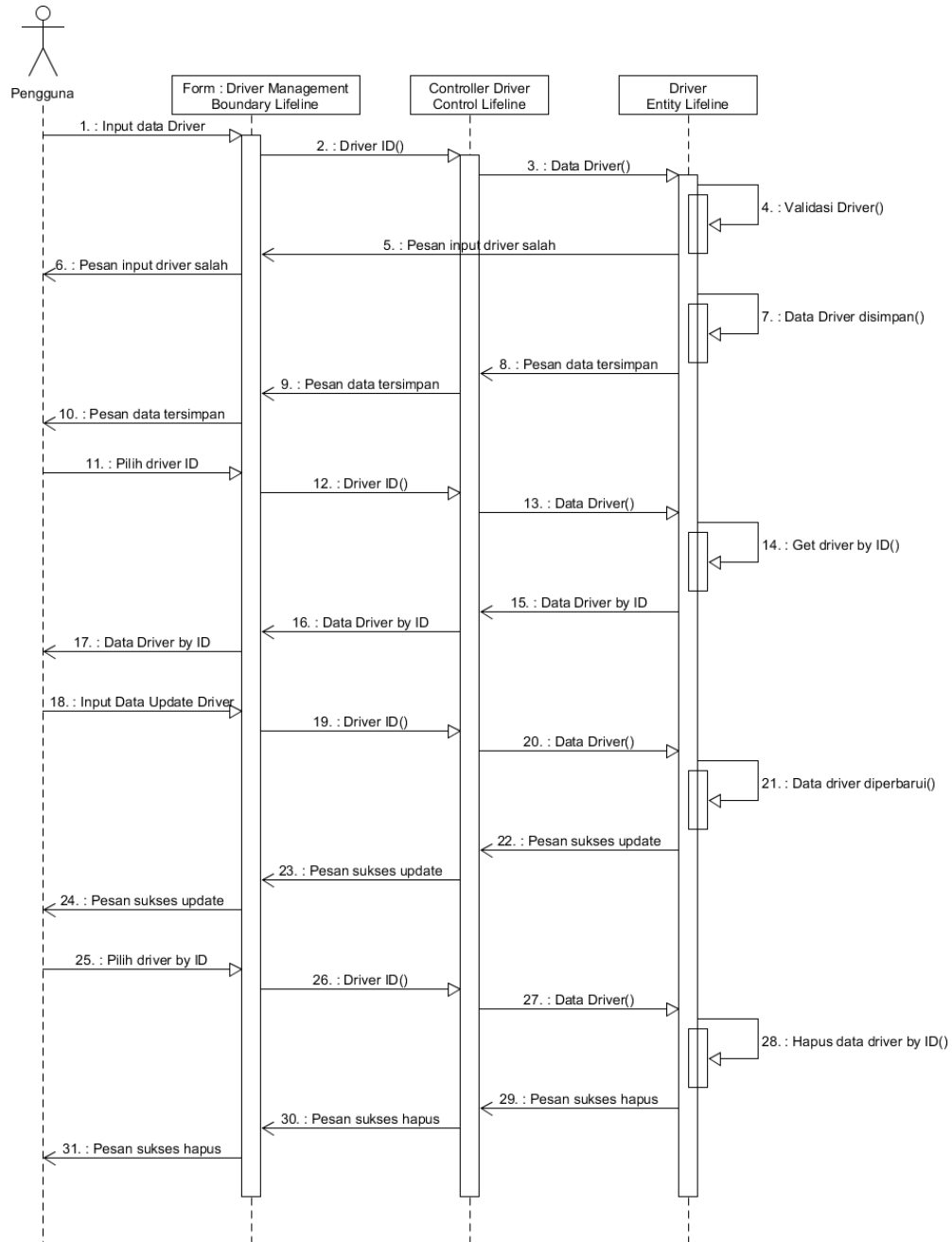
Gambar III-28 Sequence Diagram Shelter Management

6. Sequence Route Management : Admin dapat melakukan penambahan dan modifikasi data route.



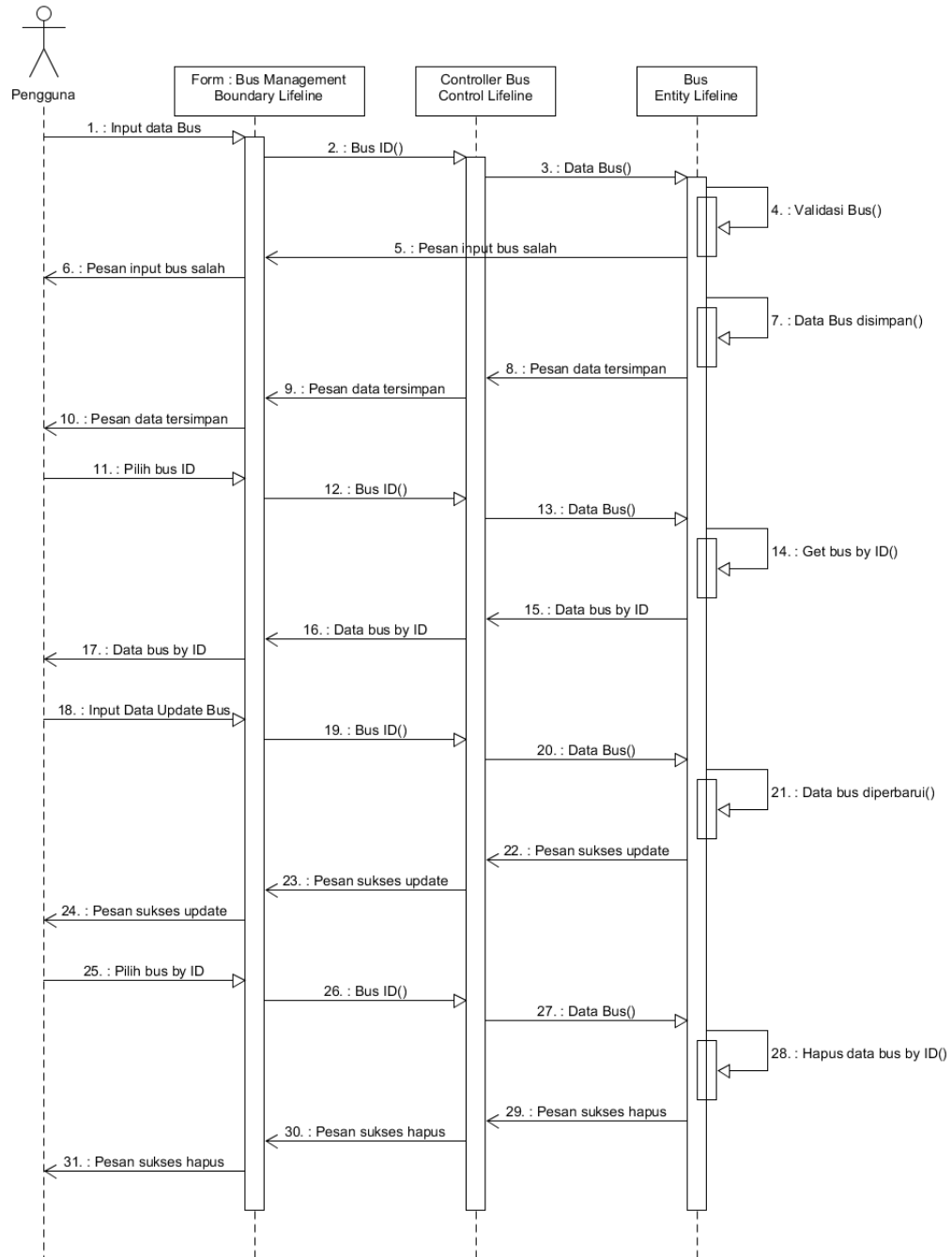
Gambar III-29 Sequence Diagram Route Management

7. Sequence Driver Management : Admin dapat melakukan penambahan dan modifikasi data driver yang tersimpan pada tabel driver.



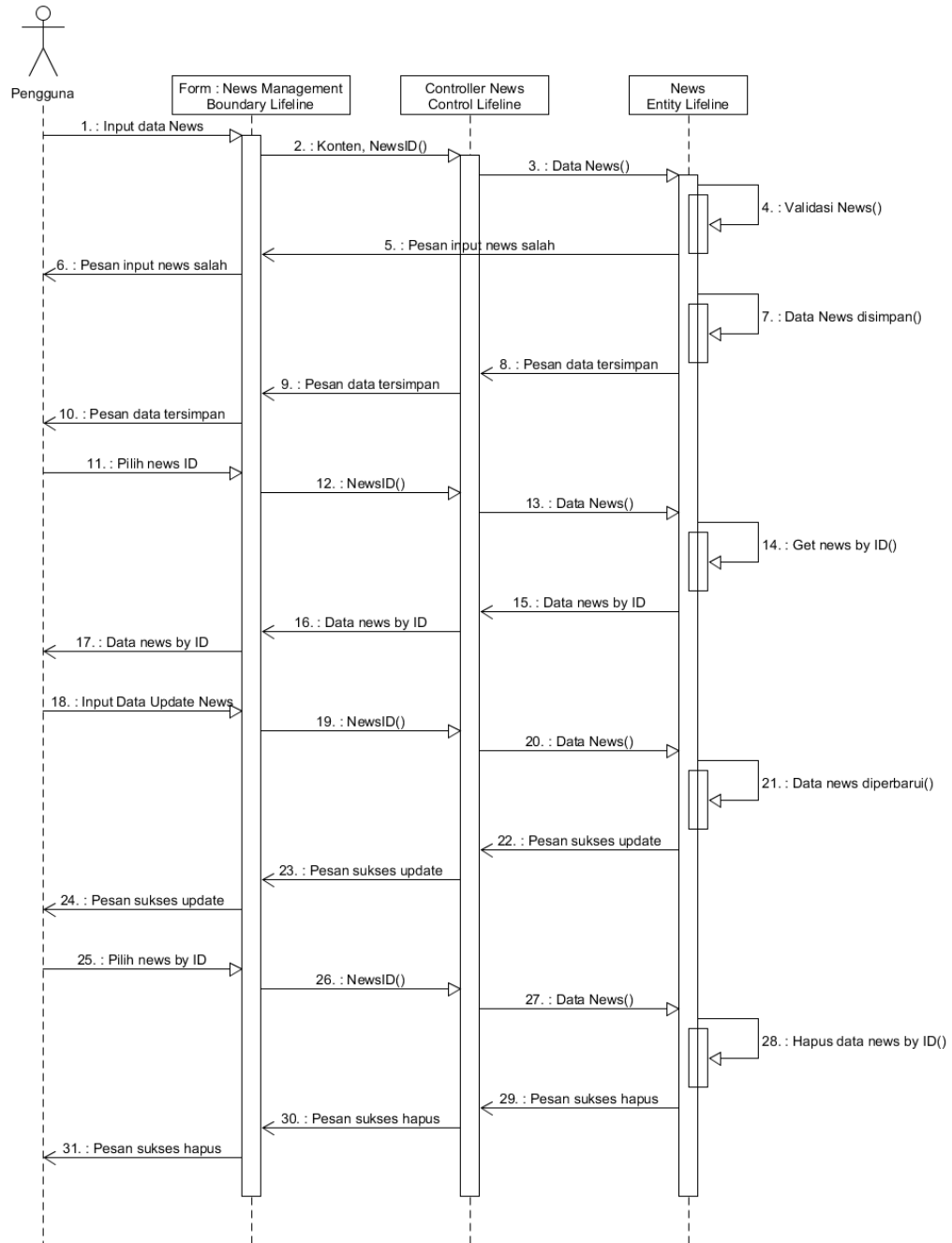
Gambar III-30 Sequence Diagram Driver Management

8. Sequence Bus Management : Admin dapat melakukan penambahan dan modifikasi data bus yang tersimpan di tabel bus.



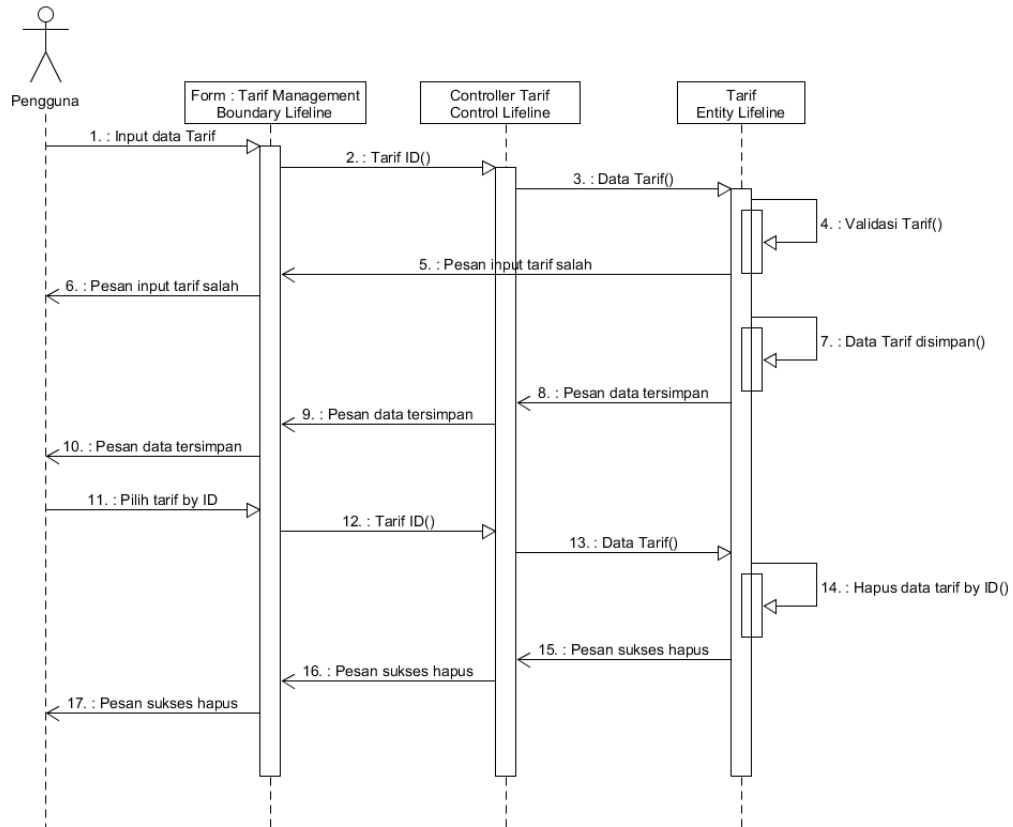
Gambar III-31 Sequence Diagram Bus Management

9. Sequence News Management : Admin dapat melakukan penambahan dan modifikasi data berita yang tersimpan di tabel news.



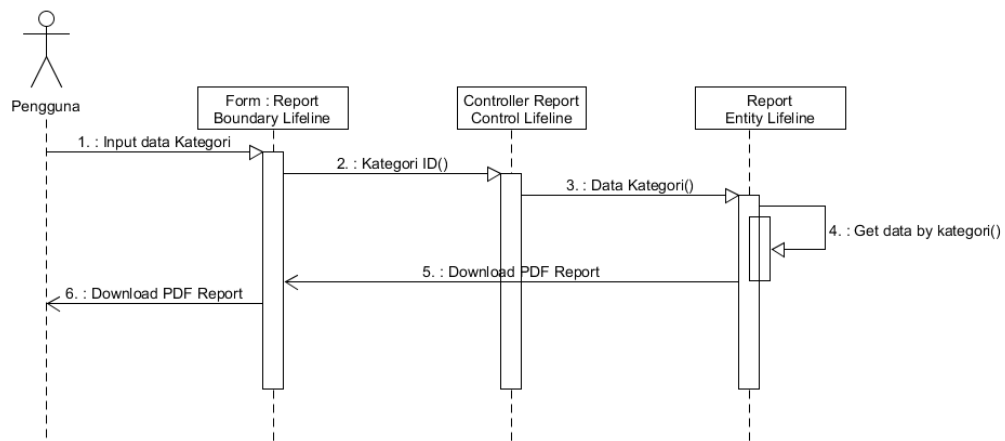
Gambar III-32 Sequence Diagram News Management

10. Sequence Tarif Management : Admin dapat melakukan penambahan dan modifikasi data tarif yang tersimpan pada tabel tarif.



Gambar III-33 Sequence Diagram Tarif Management

11. Sequence Download Report : Admin dapat mengunduh data sebagai *report* pada setiap kategori data.



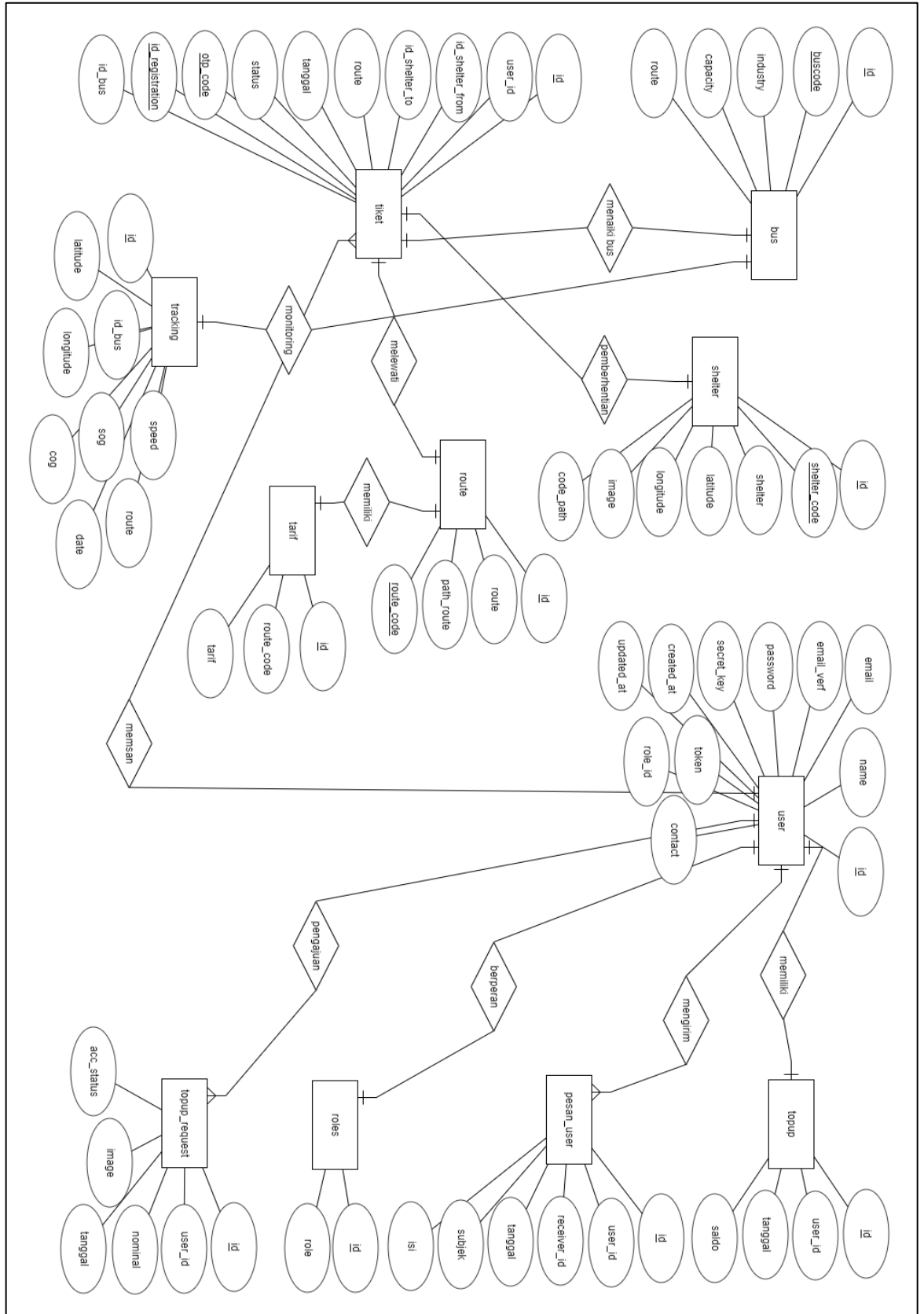
Gambar III-34 Sequence Diagram Download Report

3.4.3 Perancangan Basis Data

Dalam perancangan sistem pada penelitian ini, dibutuhkan sebuah basis data yang digunakan untuk menyimpan data. Basis data terdiri dari beberapa tabel, tabel merupakan kumpulan dari *field* dan *record*. *Field* merupakan kumpulan dari karakter yang membentuk satu arti, sedangkan *record* merupakan kumpulan *field* yang dilengkapi dengan struktur dan tipe data, biasanya dihitung dalam satuan baris (*row*).

3.4.3.1 Entity Relational Diagram (ERD)

Apabila digambarkan dalam bentuk *entity relational diagram* (ERD), desain basis data yang dirancang adalah sebagai berikut :



Gambar III-35 ERD

Pada Gambar III-35 dapat terlihat relasi dan derajat kardinalitas setiap entita. Berikut ini penjelasan mengenai diagram ERD diatas :

1. Relasi ke-1 : Bus – Tiket
 - Tabel utama : Bus
 - Tabel kedua : Tiket
 - Relasi : *one to one*
 - Atribut penghubung : id_bus (FK id_bus di Tiket)
2. Relasi ke-2 : Bus – Tracking
 - Tabel Utama : Bus
 - Tabel kedua : Tracking
 - Relasi : *one to one*
 - Atribut penghubung : id_bus (FK id_bus di Tracking)
3. Relasi ke-3 : Shelter – Tiket
 - Tabel utama : Shelter
 - Tabel kedua : Tiket
 - Relasi : *one to one*
 - Atribut penghubung : id_shelter_from, id_shelter_to (FK id_shelter_from, id_shelter_to di Tiket)
4. Relasi ke-4 : Route – Tiket
 - Tabel utama : Route
 - Tabel kedua : Tiket
 - Relasi : *one to one*
 - Atribut penghubung : route (FK route di Tiket)
5. Relasi ke-5 : Route – Tarif
 - Tabel utama : Route
 - Tabel kedua : Tarif
 - Relasi : *one to one*
 - Atribut penghubung : route_code (FK route_code di Tarif)
6. Relasi ke-6 : User – Topup
 - Tabel utama : User
 - Tabel kedua : Topup

- Relasi : *one to one*
 - Atribut penghubung : user_id (FK user_id di Topup)
7. Relasi ke-7 : User – Pesan User
- Tabel utama : User
 - Tabel kedua : Pesan User
 - Relasi : *one to one*
 - Atribut penghubung : user_id, receiver_id (FK user_id, receiver_id di Pesan User)
8. Relasi ke-8 : Roles - User
- Tabel utama : Roles
 - Tabel kedua : User
 - Relasi : *one to one*
 - Atribut penghubung : role_id (FK role_id di User)
9. Relasi ke-9 : User – Topup Request
- Tabel utama : User
 - Tabel kedua : Topup Request
 - Relasi : *one to one*
 - Atribur penghubung : user_id (FK user_id di Topup Request)
10. Relasi ke-10 : User – Tiket
- Tabel utama : User
 - Tabel kedua : Tiket
 - Relasi : *one to many*
 - Atribut penghubung : user_id (FK user_id di Tiket)

3.4.3.2 Struktur Tabel

Basis data yang dirancang dalam penelitian ini terdiri dari lima belas tabel. Tabel-tabel tersebut memiliki struktur sebagai berikut :

1. Tabel bus berisi data spesifikasi bus yang dimiliki oleh dinas Damri Kota Bandung

Tabel III-35 Struktur Tabel Bus

No	Name	Type	Collation	Attributes	Null	Default	Extra
1.	Id	bigint(20)		unsigned	No	None	AI,PK
2.	bus_code	varchar(30)	latin1		No	None	
3.	Capacity	int(3)			No	None	
4.	Industry	varchar(50)	latin1		No	None	
5.	Route	varchar(30)	latin1		No	None	

2. Tabel driver berisi data diri para pengemudi bus

Tabel III-36 Struktur Tabel Driver

No	Name	Type	Collation	Attributes	Null	Default	Extra
1.	Id (PK)	bigint(20)		unsigned	No	None	AI,PK
2.	driver_id	varchar(30)	latin1		No	None	
3.	Nama	int(3)			No	None	
4.	Kontak	varchar(50)	latin1		No	None	
5.	img_profile	varchar(30)	latin1		No	None	
6.	exp_grade	int(50)			No	None	

3. Tabel log perjalanan berisi data pemesanan tiket *online* yang memiliki relasi dengan tabel users dan shelter

Tabel III-37 Struktur Tabel Log Perjalanan

No	Name	Type	Collation	Attributes	Null	Default	Extra
1.	Id (PK)	int(11)		unsigned	No	None	AI,PK
2.	user_id	bigint(20)			No	None	FK
3.	id_shl_from	int(10)			No	None	FK
4.	id_shl_to	int(10)			No	None	FK
5.	code_route	varchar(50)	latin1		No	None	
6.	Tanggal	Date			No	None	

No	Name	Type	Collation	Attributes	Null	Default	Extra
7.	Status	varchar(50)	latin1		No	0	
8.	otp_code	varchar(50)	latin1		No	0	
9.	id_reg	varchar(50)	latin1		No	0	
10.	id_bus	int(10)			No	None	FK

4. Tabel migration berisi riwayat migrasi atau perubahan database yang telah dilakukan oleh administrator.

Tabel III-38 Struktur Tabel Migrations

No	Name	Type	Collation	Attributes	Null	Default	Extra
1.	Id (PK)	int(11)		unsigned	No	None	AI,PK
2.	Migration	varchar(255)	latin1		No	None	
3.	Batch	int(11)			No	None	

5. Tabel news berisi data berita-berita yang dapat diakses oleh pengguna

Tabel III-39 Struktur Tabel News

No	Name	Type	Collation	Attributes	Null	Default	Extra
1.	Id (PK)	bigint(20)		unsigned	No	None	AI,PK
2.	judul_brt	varchar(300)	latin1		No	None	
3.	Tanggal	Date			No	None	
4.	jenis_brt	varchar(300)	latin1		No	None	
5.	Konten	Longtext	latin1		No	None	
6.	Img	varchar(100)	latin1		No	None	

6. Tabel notifikasi berisi bermacam-macam kategori pemberitahuan bagi pengguna sistem Bsmart.

Tabel III-40 Struktur Tabel Notifikasi

No	Name	Type	Collation	Attributes	Null	Default	Extra
1.	Id (PK)	int(11)		unsigned	No	None	AI,PK
2.	Tanggal	Date			No	None	
3.	Jenis	varchar(100)	latin1		No	None	
4.	Isi	mediumtext	latin1		No	None	

7. Tabel pesan berisi histori pengiriman pesan antara pengguna dengan administrator

Tabel III-41 Struktur Tabel Pesan User

No	Name	Type	Collation	Attributes	Null	Default	Extra
1.	Id (PK)	int(11)		unsigned	No	None	AI,PK
2.	user_id	bigint(20)			No	None	FK
3.	receiver_id	bigint(20)			No	None	FK
4.	Tanggal	Date			No	None	
5.	Subjek	varchar(50)	latin1		No	0	
6.	Isi	longtext	latin1		No	0	

8. Tabel roles berisi data hak akses yang terdapat pada sistem BSmart

Tabel III-42 Struktur Tabel Roles

No	Name	Type	Collation	Attributes	Null	Default	Extra
1.	Id (PK)	int(11)		unsigned	No	None	AI,PK
2.	role_name	varchar(100)	latin1		No	None	

9. Tabel route berisi data rute operasional bus Damri

Tabel III-43 Struktur Tabel Route

No	Name	Type	Collation	Attributes	Null	Default	Extra
1.	Id (PK)	int(11)		unsigned	No	None	AI,PK

2.	Route	varchar(150)	latin1		No	None	
3.	path_route	mediumtext	latin1		No	None	
4.	route_code	varchar(50)	latin1		No	None	

10. Tabel saldo berisi data saldo pengguna yang digunakan untuk pemesanan tiket *online*, berelasi dengan tabel users

Tabel III-44 Struktur Tabel Saldo

No	Name	Type	Collation	Attributes	Null	Default	Extra
1.	id (PK)	int(11)		Unsigned	No	None	AI,PK
2.	user_id	bigint(20)			No	None	FK
3.	Tanggal	Date			No	None	
4.	Saldo	int(200)			No	None	

11. Tabel shelter berisi data dan lokasi shelter

Tabel III-45 Struktur Tabel Shelter

No	Name	Type	Collation	Attributes	Null	Default	Extra
1.	id (PK)	int(11)		Unsigned	No	None	AI,PK
2.	shelter_id	varchar(10)	latin1		No	None	
3.	Shelter	varchar(200)	latin1		No	None	
4.	Latitude	float(10,6)			No	Name	
5.	Longitude	float(10,6)			No	Name	
6.	Img	varchar(200)	latin1		No	None	
7.	Code	varchar(200)	latin1		No	None	

12. Tabel tarif berisi data nominal tarif setiap rute perjalanan bus Damri

Tabel III-46 Struktur Tabel Tarif

No	Name	Type	Collation	Attributes	Null	Default	Extra
1.	Id (PK)	int(11)		unsigned	No	None	PK
2.	route_code	bigint(20)			No	None	FK
3.	Tarif	int(10)			No	None	

13. Tabel topup berisi data pengajuan penambahan saldo yang dilakukan oleh pengguna, dan menunggu verifikasi administrator. Memiliki relasi dengan tabel users.

Tabel III-47 Struktur Tabel Topup

No	Name	Type	Collation	Attributes	Null	Default	Extra
1.	id (PK)	int(11)		Unsigned	No	None	AI,PK
2.	user_id	bigint(20)			No	None	FK
3.	Tanggal	Date			No	None	
4.	Nominal	int(200)			No	None	
5.	Img	varchar(200)	latin1		No	None	
6.	acc_status	varchar(10)	latin1		No	None	

14. Tabel tracking berisi data hasil monitoring posisi bus. Berelasi dengan tabel bus.

Tabel III-48 Struktur Tabel Tracking

No	Name	Type	Collation	Attributes	Null	Default	Extra
1.	id (PK)	int(11)		Unsigned	No	None	AI,PK
2.	id_bus	bigint(20)			No	None	FK
3.	Latitude	float(10,6)			No	Name	
4.	Longitude	float(10,6)			No	Name	
5.	Speed	double(8,2)			No	None	
6.	Sog	double(8,2)			No	None	
7.	Cog	double(8,2)			No	None	
8.	Route	varchar(100)	latin1		No	None	

No	Name	Type	Collation	Attributes	Null	Default	Extra
9.	Date	Date			No	None	

15. Tabel users berisi data diri user yang digunakan untuk melakukan pemesanan tiket / login.

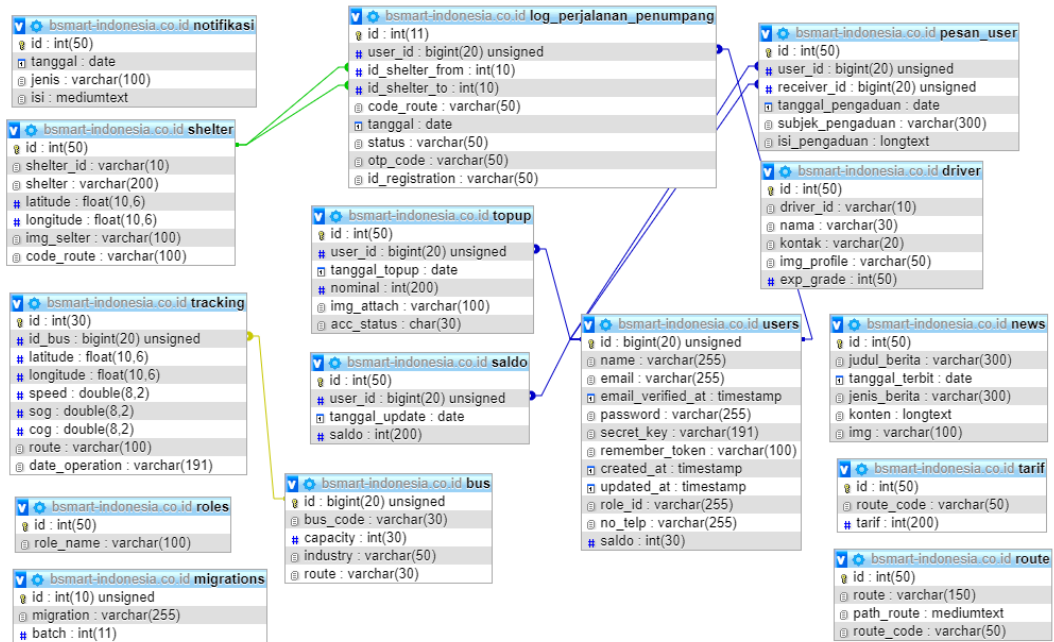
Tabel III-49 Struktur Tabel Users

No	Name	Type	Collation	Attributes	Null	Default	Extra
1.	id (PK)	int(11)		Unsigned	No	None	AI,PK
2.	Name	varchar(255)	utf8mb4		No	None	
3.	Email	varchar(255)	utf8mb4		No	None	
4.	email_verf	timestamp			Yes	Null	
5.	Password	varchar(255)	utf8mb4		No	None	
6.	secret_key	varchar(255)	utf8mb4		No	None	
7.	Token	varchar(255)	utf8mb4		Yes	Null	
8.	crt_at	timestamp			Yes	Null	
9.	verf_at	timestamp			Yes	Null	
10	role_id	varchar(255)	utf8mb4		No	None	
11.	no_telp	varchar(255)	utf8mb4		No	None	

Keterangan :

- AI : Auto Increment
- PK : Primary Key
- FK : Foreign Key

Dalam beberapa tabel, terdapat *foreign key* yang digunakan sebagai *index* relasi terhadap tabel / data lain. Relasi tabel menggambarkan keterkaitan data atau fungsional sistem penyimpanan. Adapun rancangan relasi yang telah dibuat dalam penelitian ini diantaranya dapat dilihat pada gambar dibawah ini :



Gambar III-36 Relasi Antar Tabel

3.4.4 Perancangan Antarmuka

Penelitian ini memerlukan segmen perangkat lunak yang digunakan sebagai *user interface*. Sistem perangkat lunak terbagi menjadi dua jenis, yaitu *website* dan *mobile apps*. Perancangan antarmuka diperlukan untuk memberikan gambaran *layouting* tampilan, serta kesesuaian tata letak data *view*. Berikut ini beberapa perancangan *mockup website* dan *mobile* :

1. Landing Page

2. Form Login

3. Form Register

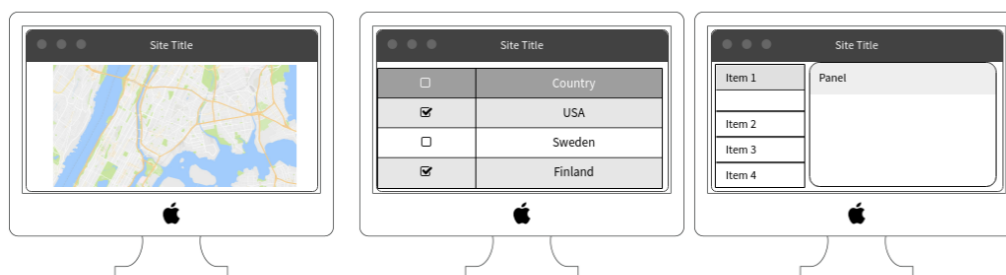


Gambar III-37 Mockup Website 1

4. Map Navigation

5. Datatable

6. Dashboard

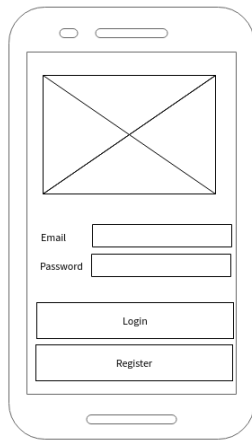


Gambar III-38 Mockup Website 2

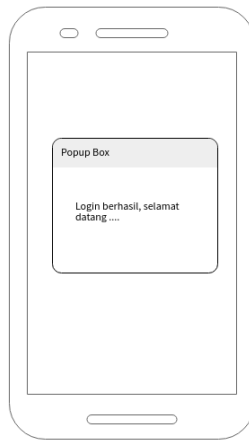
Berikut ini tampilan skenario *mockup* perancangan antarmuka aplikasi *mobile* :

1. User mengisi email dan password pada halaman login.
2. Aplikasi akan melakukan validasi data akun di *server*, akan muncul popup konfirmasi validasi akun.
3. Apabila proses login berhasil, maka akan muncul halaman utama aplikasi, berisi menu-menu.
4. Halaman report digunakan untuk melaporkan kejadian darurat dengan mekanisme pengambilan gambar, serta dikirim ke administrator melalui email.
5. Halaman emergency call berisi layanan telpon darurat dalam kondisi tertentu, dapat diakses melalui sambungan telpon atau pesan singkat (SMS).
6. Dashboard, menampilkan *dashboard website*.
7. Melihat informasi shelter di kota Bandung
8. Logout akun.

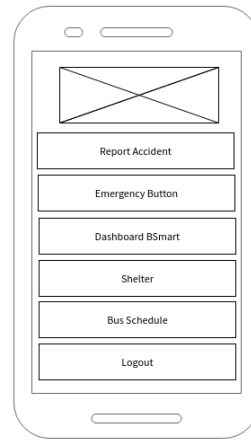
1. Form login



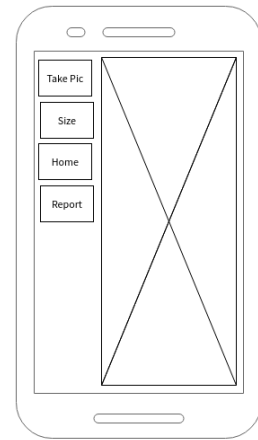
2. Popup validasi



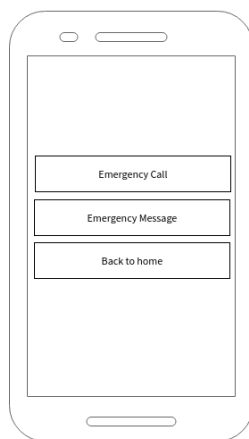
3. Menu utama



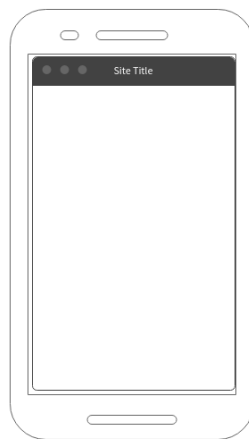
4. Panel report



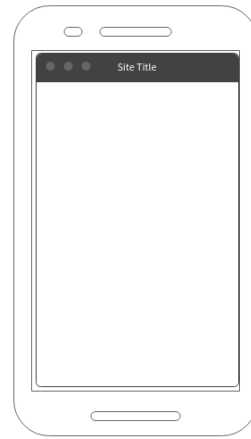
5. Emergency Call



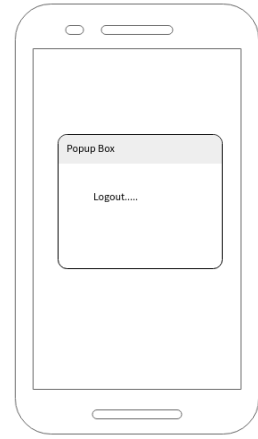
6. Dashboard



7. Information



8. Logout



Gambar III-39 Mockup Mobile App

