BAB 3
ANALISIS DAN PERANCANGAN SISTEM

Tahap analisis dan perancangan merupakan tahap sistematis untuk menyesuaikan kegunaan dan tujuan pada aplikasi. Tahap awal pada tahap analisis dimulai dari analisis masalah, menganalisis sistem dengan menganalisis aplikasi sejenis, analisis fungsional dan non-fungsional. Sedangkan untuk tahap perancangan dimulai dengan melakukan perancangan sistem yang mencakup perancangan antar muka dan perancangan struktur menu yang nantinya akan digunakan untuk diterapkan pada aplikasi.

3.1. Analisis Sistem

Analisis sistem merupakan penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponenya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kebutuhan serta hambatan yang terjadi. Menentukan kebutuhan yang sesuai dengan kebutuhan-kebutuhan yang diharapkan, untuk melakukan analisis sistem dilakukan beberapa analisis yaitu :

a. Analisis Masalah
b. Analisis Sistem yang Berjalan
c. Analisis Teknologi
d. Analisis Aturan Bisnis
e. Analisis Arsitektur Sistem
f. Analisis Kebutuhan Non-Fungsional
g. Analisis Pengguna *Mobile*
h. Analisis Kebutuhan Perangkat Lunak
i. Analisis Kebutuhan Perangkat Keras
j. Analisis Kebutuhan Fungsional

3.1.1. Analisis Masalah

Analisis masalah adalah tahap penjabaran masalah yang ada sebelum aplikasi ini dibangun dan bertujuan untuk membantu pembangunan sistem permintaan layanan ambulan di rumah sakit Al-Islam Bandung. Berikut adalah penjabaran dari masalah-masalah yang ada antara lain sebagai berikut :

1. Dirumah sakit Al-Islam Bandung ketika pasien akan melakukan permintaan layanan ambulan, pasien terlebih dahulu menghubungi pihak rumah sakit melalui telepon dan menginformasikan alamat pasien. hal ini mengakibatkan kurang jelasnya informasi penjemputan yang membuat *Driver* ambulan kesulitan dalam mencari alamat pasien.
2. Informasi alamat yang diberikan pasien kepada pihak rumah sakit melalui telepon, sering kali membuat pihak rumah sakit kesulitan menentukan tarif permintaan layanan ambulan, karena informasi yang diberikan seringkali tidak lengkap.
3. Pihak rumah sakit kesulitan dalam hal melakukan pelayanan karena pihak rumah sakit sulit menentukan apakah dijemput dengan didampingi tenaga medis atau tanpa tenaga medis dan apa saja alat dan obat yang harus dibawa saat pertolongan pertama.

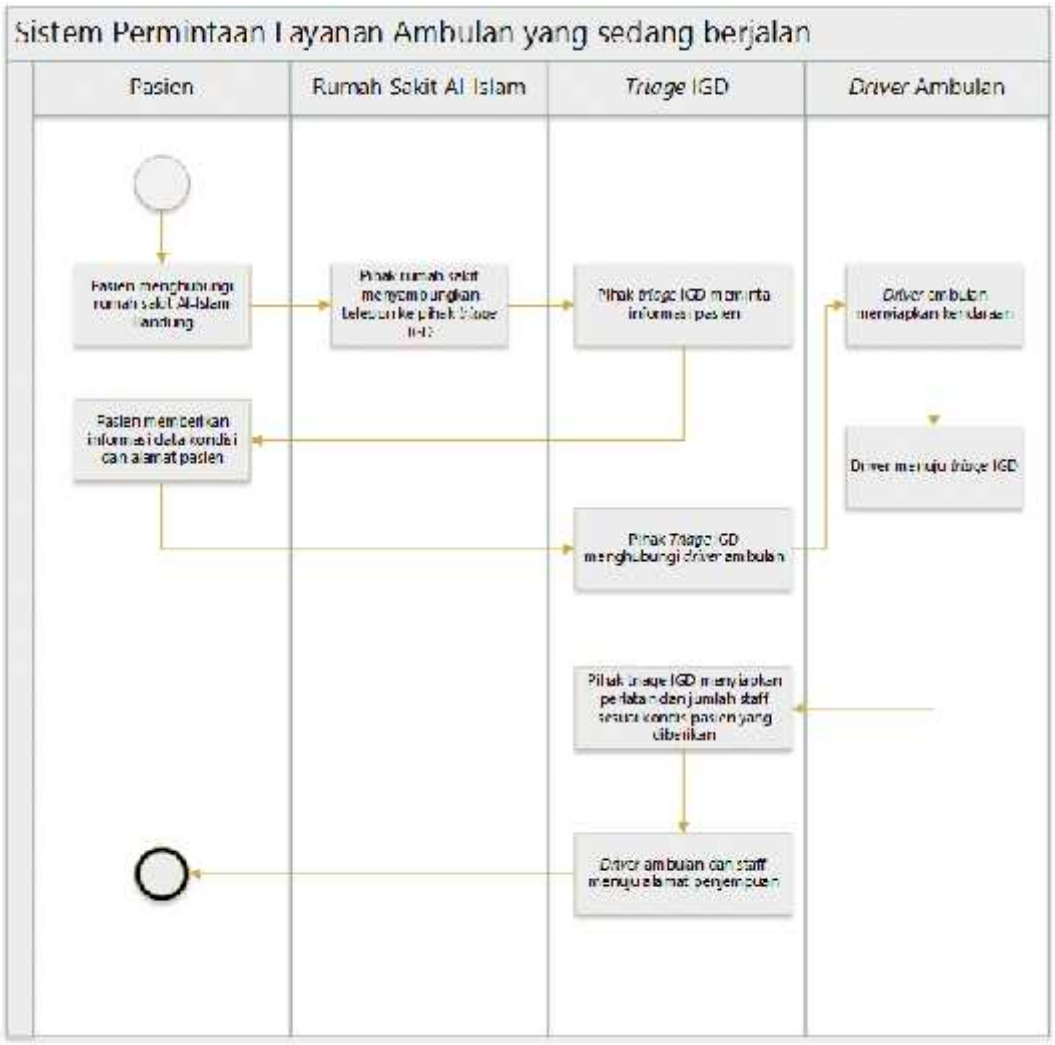
3.1.2. Analisis Sistem yang Berjalan

Analisis sistem yang sedang berjalan dilakukan dengan metode wawancara dengan Dr. Ahmad Iffa Maududy selaku Kainst gawat darurat dan bapa Ruslan salah satu *Driver* Ambulan di Rumah Sakit Al Islam Bandung, hal itu dilakukan untuk mendapatkan gambaran lengkap tentang sistem yang sedang berjalan saat ini yang kemudian hasilnya akan digambarkan kedalam bentuk *activity diagram* sesuai dengan masalah yang terjadi.

Skenario dari prosedur permintaan layanan ambulan di rumah sakit Al-Islam Bandung yaitu :

1. Pasien menghubungi pihak rumah sakit Al-Islam Bandung untuk permintaan layanan ambulan.
2. Pihak rumah sakit menyambungkan telepon tersebut kepihak *Triage* IGD.
3. Pihak *Triage* IGD meminta informasi kondisi dan alamat pasien.
4. Pasien memberikan informasi kondisi dan alamat pasien.
5. Pihak *Triage* IGD menghubungi *Driver* ambulan.
6. *Driver* ambulan menyiapkan kendaraan.
7. Pihak *Triage* IGD menyiapkan peralatan dan jumlah staff sesuai kondisi pasien yang diberikan.
8. *Driver* ambulan dan staff rumah sakit Al-Islam menuju alamat penjemputan pasien.

Activity Diagram Sistem Permintaan Layanan Ambulan dapat dilihat pada Gambar 3.1



Gambar 3.1 *Activity Diagram* Sistem Permintaan Layanan Ambulan Rumah Sakit Al-Islam Bandung

3.1.3. Analisis Teknologi

3.1.3.1. Analisis Penentuan Lokasi Dengan GPS

Pada penelitian ini metode yang digunakan untuk menentukan posisi suatu lokasi adalah metode *Cell-ID* (*Cell Identification*) yaitu penentuan posisi didasarkan pada daerah geografis yang tercakup oleh sebuah cell berhubungan dengan daerah cakupan dari sinyal radio. Ketika sebuah handphone terhubung secara aktif dengan sebuah base station, berarti handphone tersebut diasumsikan berada dalam cell dari base station tersebut. Untuk mengukur

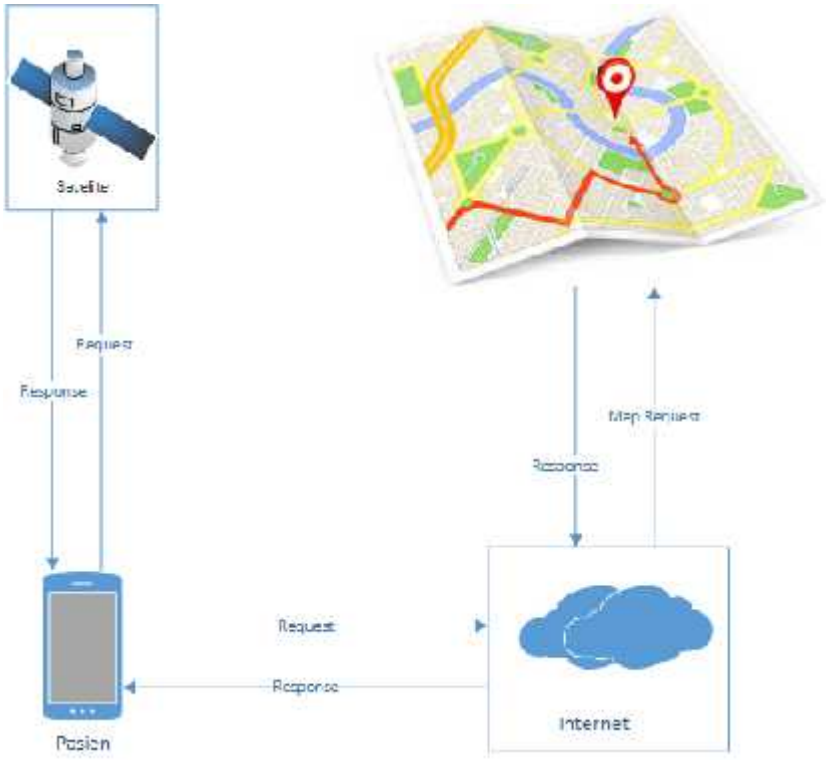
jarak dan arah handset dari base station tidak dapat diketahui dengan pasti. Oleh karena itu, untuk lebih meningkatkan lagi akurasi hasil pencarian, metode *Cell ID* ini dapat dikombinasikan dengan metode *Timing Advanced* (TA), dengan menggunakan TA ini, metode Cell ID akan ditambahkan sebuah fungsionalitas untuk menghitung *Round Trip Time* (RTT), yaitu waktu transmisi sebuah frame (dari base station ke handphone) dan waktu penerimaan sebuah frame (dari handphone ke base station). Dengan tambahan metode ini, jarak antara handphone dan base station dapat ditentukan dengan keakuratan 50 m.

Berikut ini salah satu rumus yang sering digunakan untuk menentukan jarak lokasi pada metode *CELL ID* dengan mengabaikan bentuk geografis baik bukit maupun lembah.

Jarak = Arc Cos (Cos(Lat1) * Cos(Lat2) * Cos(Long2-Long1) + Sin(Lat1) * Sin(Lat 2)) * R

Keterangan

Lat1	=	Nilai Latitude dari Lokasi Awal
Long1	=	Nilai Longitude dari Lokasi Awal
Lat2	=	Nilai Latitude dari Lokasi Tujuan
Long2	=	Nilai Longide dari Lokasi Tujuan
R	=	Jari-jari rata-rata Bumi (6378.2 KM)
Catatan	:	Nilai Lat dan Long dalam satuan radian.



Gambar 3.2 Arsitektur Teknologi GPS yang digunakan

3.1.3.2. Analisis Google Cloud Messaging

Google Cloud Messaging (GCM) merupakan proyek yang dikembangkan oleh Google Inc. GCM sendiri berfungsi untuk menyampaikan informasi berupa pesan(*notification message*) ke perangkat *mobile* Android. GCM memberikan kemudahan dalam menyampaikan pesan secara gratis tidak terikat besarnya suatu pesan. GCM servis akan mengatur setiap pesan yang dikirimkan agar sesuai dengan perangkat *mobile* tujuan pesan (*receiver*).

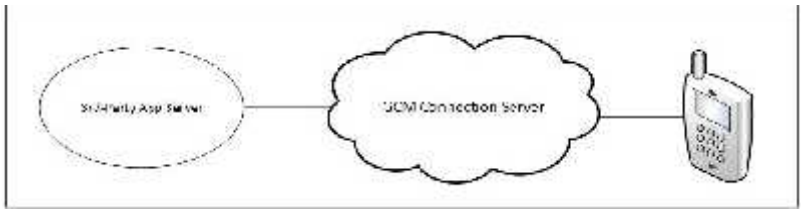
Beberapa karakteristik GCM :

a. Mengijinkan *3rd-party application servers* untuk mengirim ke Aplikasi android yang dituju.
b. Dengan menggunakan GCM, pengguna dapat mengirimkan *upstream messages*.

c. Aplikasi Android dan perangkat Android tidak perlu dalam keadaan *running* untuk bisa menerima pesan. Sistem akan melakukan *wake up* aplikasi android ketika pesan masuk dengan cara *intent broadcast*, selama terdapat *permission* pada aplikasi.
d. Dalam penggunaannya tidak diperlukan *user interface* khusus untuk menangani pesan yang diterima. GCM mengirimkan *raw message* ke aplikasi android beserta dengan kontrol notifikasinya.
e. Perangkat android hanya memerlukan minimal platform Android 2.2 (bisa jalan pada versi diatasnya) dan terdapat Google Play Store.
f. Untuk perangkat android diatas 3.0, ini mengharuskan pengguna untuk mengatur akun *Google* mereka pada perangkat *mobile* mereka. Akun Google bukan persyaratan pada perangkat yang menjalankan Android 4.0.4 atau lebih tinggi.

Terdapat 2 kunci utama atau konsep pada GCM, yaitu: *component* dan *credentials*. *Component* adalah entitas utama yang memegang peranan utama dalam GCM.*Credentials* adalah ID dan *token* dari setiap perangkat *mobile* Android yang digunakan GCM sebagai autentikasi dan alamat pengiriman pesan.

Dalam implementasi *Google Cloud Messaging* (GCM) terdapat beberapa bagian yang terlibat yaitu : *Google-provided connection server, 3rd-party app server* dan *Client App*. Arsitektur GCM dapat dilihat pada Gambar 3.3.



Gambar 3.3 Arsitektur GCM

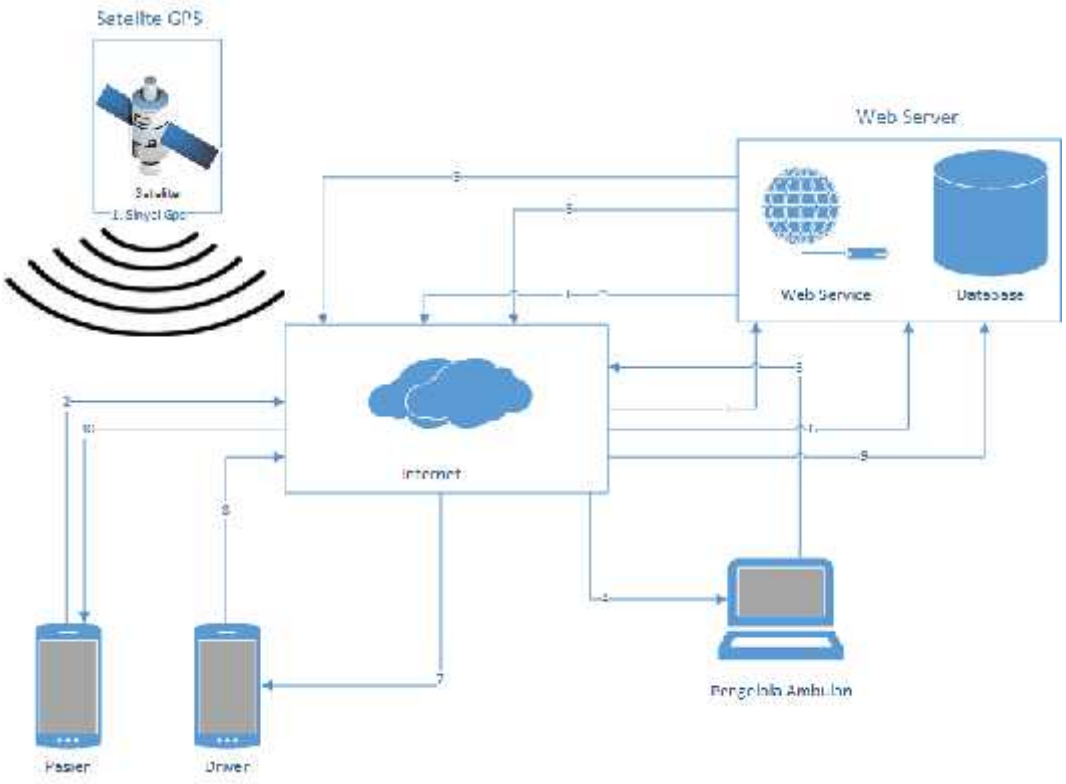
3.1.4. Analisis Aturan Bisnis

Analisis aturan bisnis yang dilakukan yaitu dengan menjelaskan ketentuan- ketentuan yang terjadi pada proses bisnis yang ada. Analisis aturan bisnis yang diterapkan dalam proses bisnis di aplikasi yang akan dibangun adalah sebagai berikut :

a. Hanya pengguna *mobile* berbasis android yang dapat menggunakan aplikasi ini dan dapat *mendownload* di Playstore.
b. Pasien dapat melakukan permintaan layanan ambulan melalui *smartphone* berbasis android
c. *Driver* ambulan mendapat *notification* ketika ada permintaan layanan ambulan
d. *Driver* ambulan dapat mencari dan mengatuhi lokasi pasien menggunakan GPS yang terpasang di *smartphone*.
e. *Driver* ambulan bisa melihat jalur yang harus dilewati, jarak, dan waktu tempuh yang ditampilkan dalam Google Maps.
f. Admin di rumah sakit Al-Islam Bandung memiliki tugas mengelola data petugas, data *Driver* ambulan ,data hak akses, data pengguna aplikasi permintaan layanan ambulan, dan data permintaan layanan ambulan.
g. Admin *Triage* IGD memilliki tugas lain yaitu mengelola data pasien.
h. Pasien dapat melihat informasi permintaan layanan ambulan, data *Driver* yang menjemput, dan data tarif permintaan layanan ambulan.

3.1.5. Analisis Arsitektur Sistem

Tahap analisis arsitektur sistem adalah tahapan untuk mendapatkan gambaran umum sistem yang akan dibangun. Baik itu gambaran sistem *Backend* admin maupun gambaran sistem *Frontend* (pengguna *mobile*). Gambaran arsitektur sistem dapat dilihat pada Gambar 3.4



Gambar 3.4 Arsitektur Sistem

Berikut penjelasan dari gambaran arsitektur sistem Permintaan Layanan Ambulan Berbasis Android :

1. Perangkat Smartphone yang mempunyai GPS menerima sinyal informasi dari satelit–satelit GPS, kemudian smartphone yang mempunyai GPS menentukan posisinya dengan menggunakan perhitungan.
2. Pasien yang akan melakukan permintaan layanan ambulan, mengirimkan data kondisi pasien, longitude dan latitude *Smartphone user* ke *web service* dengan bahasa JSON secara otomatis melalui bantuan internet.
3. *Web service* menerima data kondisi pasien dan lokasi dari smartphone pasien dengan bantuan JSON lalu *web service* mengirimkan data dari database.

4. Pengelola ambulan menerima dan melihat data kondisi pasien dari permintaan layanan ambulan yang masuk.
5. Pengelola ambulan merespon permintaan layanan yang masuk dan mengirimkan data melalui internet
6. Internet meneruskan data tersebut dengan mengakses *webservice* yang berada dalam *webserver*. *Webservice* mengirim hasil dari permintaan melalui internet.
7. *Driver* yang menggunakan Perangkat mobile menerima data yang telah diminta sebelumnya dalam bentuk JSON.
8. *Driver* mengirimkan data *longitude* dan *latitude Smartphone driver* ke *web service* dengan bahasa JSON secara otomatis melalui bantuan internet.
9. *Web service* menerima data lokasi dari smartphone *driver* dengan bantuan JSON lalu *web servise* mengirimkan data dari database ke *google cloud messaging*.
10. Internet meneruskan data tersebut dengan mengakses *webservice* yang berada dalam *webserver*. *Webservice* mengirim hasil dari permintaan melalui internet ke perangkat pasien.

3.1.6. Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional menggambarkan kebutuhan pendukung sistem yang diperlukan untuk menjalankan perangkat lunak yang dibangun. Analisis kebutuhan non fungsional sistem permintaan layanan ambulan berbasis android ini terdiri dari analisis kebutuhan pengguna, analisis kebutuhan perangkat lunak, dan analisis kebutuhan perangkat keras.

3.1.6.1. Analisis Kebutuhan Pengguna

Analisis kebutuhan pengguna merupakan proses analisis yang menekankan bahwa siapa saja pengguna yang terlibat dalam penggunaan sistem. Karakateristik pengguna yang sudah ada yaitu :

Tabel 3.1 Kebutuhan Pengguna

No	Pengguna	Kemampuan yang sudah ada	Kemampuan yang diperlukan
1.	Admin Ambulan	Mengerti komputer dan internet, mengerti microsoft office	Mengerti komputer dan internet, mengerti microsoft office, mengerti mengelola web dan database
2.	Pengguna Smartphone	Mengerti penggunaan smartphone	Mengerti penggunaan smartphone

Berdasarkan data Tabel 3.1 bahwa pengguna smartphone sudah memenuhi namun admin pengelola ambulan belum memenuhi kriteria yang diperlukan oleh karena itu diperlukannya pelatihan untuk mengelola web dan database untuk mengelola data pada aplikasi yang akan dibangun. Sedangkan kebutuhan pengguna untuk mendukung sistem yang akan dibangun dapat dilihat pada Tabel 3.2.

Tabel 3.2 Kebutuhan Pengguna Pada Sistem Yang Akan Dibangun

No	**Pengguna**	**Kemampuan**	**Tanggung Jawab**
1	Admin Ambulan	Mengerti cara menggunakan *smartphone*, komputer, internet, mengelola web dan database.	Mengelola data layanan ambulan, *Driver*, user.
2	*Driver* Ambulan	Mengerti cara menggunakan *smartphone*	Mengelola dan menjaga data *pribadi Driver* ambulan
3	*Pasien*	Mengerti cara menggunakan *smartphone*	Menjaga data pribadi *pasien*

Kebutuhan pengguna yang diperlukan adalah Admin pengelola ambulan yang merupakan pengelola data kendaraan dan data user yang diharuskan mengerti operasi standar seperti menambah data, mengubah data, menghapus data dan diharuskan untuk menjaga data sistem permintan layanan ambulan.

3.1.6.2. Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak merupakan proses analisis yang menekankan kepada pemanfaatan perangkat lunak atau *software*. Spesifikasi perangkat lunak yang sudah ada dan yang dibutuhkan sistem dapat dilihat pada tabel 3.3.

Tabel 3.3 Spesifikasi kebutuhan perangkat lunak

No	Jenis Perangkat Lunak	Yang sudah ada	Dibutuhkan
1	Sistem Operasi	Windows 10	Windows 7
2	Web Browser	Mozilla Firefox	Mozilla Firefox
3	Database Server	Mysql	Mysql

Berdasarkan tabel 3.3 maka spesifikasi kebutuhan perangkat lunak untuk sistem yang akan dibangun sudah memenuhi kriteria yang diperlukan dapat mendukung untuk pengoperasian sistem.

3.1.6.3. Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras merupukan proses analisis yang menekankan kepada pemanfaatan perangkat keras atau *hardware*. Kebutuhan perangkat keras yang sudah ada dan yang dibutuhkan sistem yang akan dibangun dapat dilihat pada tabel 3.4.

Tabel 3.4 Spesifikasi kebutuhan perangkat keras

No.	Jenis Perangkat Keras	Yang sudah ada	Dibutuhkan
1.	Prosesor	2.00GHz	1.80GHz
2.	RAM	4GB	2 GB
3.	Harddisk	500GB	10 GB

4.	Monitor	17 inch	14 inch

Berdasarkan tabel 3.4 maka spesifikasi kebutuhan perangkat keras untuk sistem yang akan dibangun sudah memenuhi kriteria yang diperlukan dapat mendukung untuk pengoperasian sistem.

3.1.7. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sistem yang akan dibangun sehingga sistem dapat berjalan dengan baik. Analisis dimodelkan dengan menggunakan *Unifed Modeling Language* (UML). Tahap-tahap pemodelan dalam analisis menggunakan UML antara lain *use case diagram, use case scenario, activity diagram, sequence diagram,* dan *class diagram.*

3.1.7.1. Analisis Kebutuhan Fungsional Pada *Back End*

Spesifikasi kebutuhan fungsional pada pembangunan sistem permintaan layanan ambulan berbasis android pada back end dapat dilihat pada tabel 3.5.

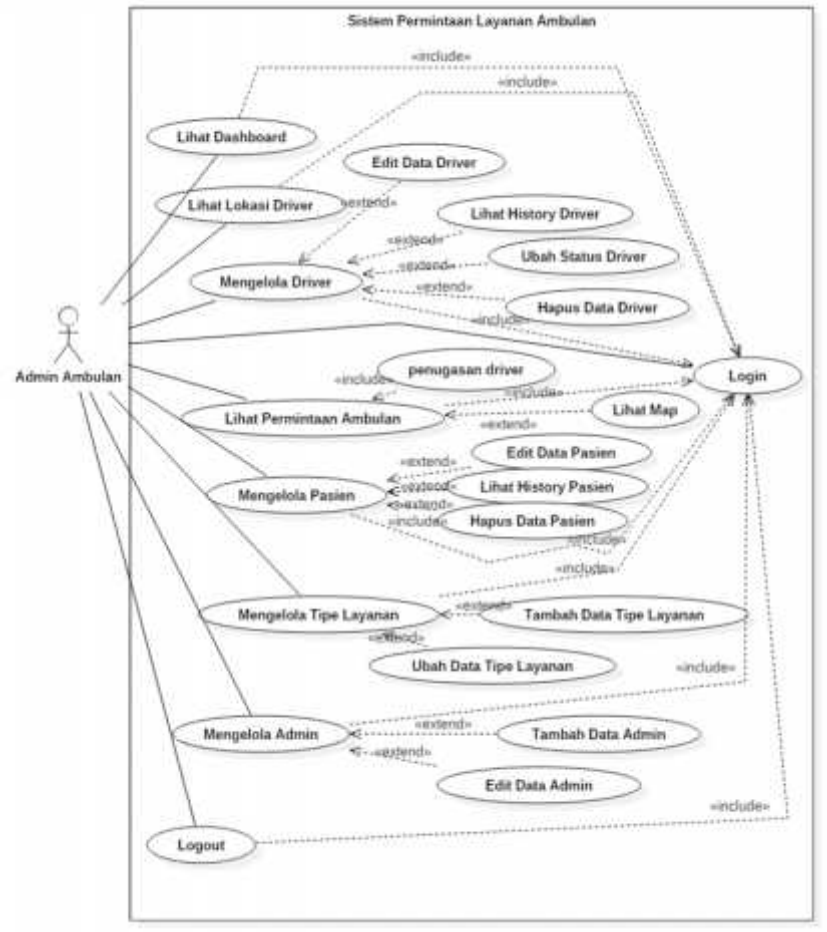
Tabel 3.5 Spesifikasi Kebutuhan Fungsional Pada Back End

Nomor	Spesifikasi Kebutuhan Fungsional
SKPL-F-001	Admin login
SKPL-F-002	Admin dapat melihat semua informasi sistem atau melihat dashboard
SKPL-F-003	Admin dapat mengetahui lokasi *Driver*
SKPL-F-004	Admin dapat mengelola data *Driver*
SKPL-F-005	Admin dapat mengedit data *Driver*
SKPL-F-006	Admin dapat melihat data *history Driver*
SKPL-F-007	Admin dapat mengubah status *Driver*
SKPL-F-008	Admin dapat menghapus data *Driver*
SKPL-F-009	Admin dapat melihat data permintaan layanan ambulan
SKPL-F-010	Admin dapat melihat data permintaan layanan ambulan melalui map
SKPL-F-011	Admin dapat mengelola data pasien

SKPL-F-012	Admin dapat mengedit data pasien
SKPL-F-013	Admin dapat melihat data *history* pasien
SKPL-F-014	Admin dapat menghapus data pasien
SKPL-F-015	Admin dapat mengelola data tipe layanan
SKPL-F-016	Admin dapat menambah data tipe layanan
SKPL-F-017	Admin dapat mengedit data tipe layanan
SKPL-F-018	Admin dapat mengelola data admin
SKPL-F-019	Admin dapat menambah data admin
SKPL-F-020	Admin dapat mengedit data admin
SKPL-F-021	Admin dapat melakukan penugasan permintaan
SKPL-F-022	Admin logout

3.1.7.1.1. *Use Case Back End*

Use case diagram merupakan konstruksi untuk mendeskripsikan hubungan yang terjadi antara aktor dengan aktivitas yang terdapat pada sistem. *Use case* menjelaskan proses apa saja yang ada dalam sistem dan bagaimana hubungannya dengan aktor. Dari hasil analisis maka *use case diagram* pada sistem ini dapat dilihat pada Gambar 3.5.



Gambar 3.5 *Use Case Diagram Back End*

3.1.7.1.1.1 Deskripsi Aktor *Back End*

Berdasarkan use case diagram pada gambar 3.5, deskripsi aktor dapat dilihat pada tabel 3.6.

Tabel 3.6 Deskripsi Aktor

No	Aktor	Deskripsi
1	Admin Ambulan	Merupakan aktor yang akan mengelola data yang berkaitan dengan sistem permintaan layanan ambulan. Dimana admin di rumah sakit Al-Islam Bandung memiliki hak akses untuk menambah, mengubah data yang terdapat di *database server*.

3.1.7.1.1.2 Deskripsi *Use Case Diagram Back End*

Deskripsi *use case diagram* berfungsi untuk menjelaskan proses yang terdapat pada setiap *use case*. Deskripsi *use case* dijelaskan pada tabel Tabel 3.7.

Tabel 3.7 Deskripsi *Use Case Diagram*

No	Use Case	Deskripsi
1	Login	Fungsional ini digunakan oleh Admin pengelola ambulan untuk login ke halaman utama.
2	Lihat dashboard	Fungsional ini digunakan oleh Admin untuk mengetahui semua informasi yang berkaitan dengan sitem permintaaan layanan ambulan.
3	Lihat Lokasi *Driver*	Fungsional ini digunakan oleh Admin untuk mengetahui lokasi keberadaan semua *Driver* ambulan.
4	Mengelola data *Driver*	Fungsional ini digunakan oleh Admin pengelola ambulan untuk mengelola data *Driver* ambulan.
5	Edit data *Driver*	Fungsional ini digunakan oleh Admin pengelola ambulan untuk mengubah data *Driver* ambulan.
6	Lihat *history Driver*	Fungsional ini digunakan oleh Admin pengelola ambulan untuk melihat data *history Driver* ambulan.
7	Ubah Status *Driver*	Fungsional ini digunakan oleh Admin pengelola ambulan untuk meng *Approve Driver ambulan yang baru mendaftar*.
8	Hapus data *Driver*	Fungsional ini digunakan oleh Admin pengelola ambulan untuk *menghapus data Driver ambulan.*
9	Lihat permintaan ambulan	Fungsional ini digunakan oleh Admin pengelola ambulan untuk melihat semua data permintaan layanan ambulan yang masuk.
10	Lihat map	Fungsional ini digunakan oleh Admin pengelola

		ambulan untuk melihat data permintaan melalui map.
11	Mengelola data pasien	Fungsional ini digunakan oleh Admin pengelola ambulan untuk mengelola data *user*.
12	Edit data pasien	Fungsional ini digunakan oleh Admin pengelola ambulan untuk mengubah *data user.*
13	Lihat *History* pasien	Fungsional ini digunakan oleh Admin pengelola ambulan untuk melihat data *history* permintaan ambulan oleh *user*.
14	Hapus data pasien	Fungsional ini digunakan oleh Admin pengelola ambulan untuk *menghapus data user.*
15	Mengelola Tipe layanan	Fungsional ini digunakan oleh Admin pengelola ambulan untuk mengelola data tipe layanan.
16	Tambah data tipe layanan	Fungsional ini digunakan oleh Admin pengelola ambulan untuk menambah data tipe layanan.
17	Edit data tipe layanan	Fungsional ini digunakan oleh Admin pengelola ambulan untuk meng*ubah data tipe layanan.*
18	Mengelola Admin	Fungsional ini digunakan oleh Admin pengelola ambulan untuk meng*elola data admin.*
19	Tambah data admin	Fungsional ini digunakan oleh Admin pengelola ambulan untuk menambah *data admin pengelola ambulan.*
20	Edit data admin	Fungsional ini digunakan oleh Admin pengelola ambulan untuk mengubah *data admin pengelola ambulan.*
22	Penugasan driver	Fungsional ini digunakan oleh Admin pengelola ambulan untuk menugaskan driver saat permintaan layanan masuk
21	Logout	Fungsional ini digunakan oleh Admin pengelola ambulan untuk keluar dari halaman utama.

3.1.7.1.2. *Use Case Scenario Back End*

Berdasarkan *use case diagram* pada gambar 3.5, maka *use case scenario back end* untuk aplikasi sistem permintaan layanan ambulan di rumah sakit Al-Islam Bandung adalah sebagai berikut :

1. *Use Case Scenario Login*

Use case scenario login dapat dilihat pada tabel 3.8.

Tabel 3.8 *Use Case Scenario Login*

Nama Use Case	*Login*	
Tujuan	Admin pengelola ambulan berhasil masuk ke halaman utama	
Kondisi Awal	Menampilkan form login	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk login ke halaman utama.	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil masuk ke halaman utama	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal masuk ke halaman utama	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol login	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan form login
	2.	Admin pengelola ambulan mengisi form login
	3.	Admin pengelola ambulan menekan tombol login
	4.	Sistem melakukkan validasi form
	5.	Admin berhasil masuk ke halaman utama
Skenario Alternatif	Langkah	Aksi
	4.1	Field Form tidak terisi penuh

	4.2	Sistem menampilkan pesan field kosong
	4.3	Sistem menampilkan pesan email atau password salah

2. *Use Case Scenario* Lihat *Dashboard*

Use case scenario lihat *dashboard* dapat dilihat pada tabel 3.9.

Tabel 3.9 Use Case Scenario* Lihat *Dashboard

Nama Use Case	Lihat *Dashboard*	
Tujuan	Admin pengelola ambulan melihat semua informasi sitem permintaan layanan ambulan.	
Kondisi Awal	Admin pengelola ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh Admin untuk mengetahui semua informasi yang berkaitan dengan sitem permintaaan layanan ambulan.	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil melihat semua informasi sistem permintaan layanan ambulan	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal melihat semua informasi sistem permintaan layanan ambulan	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol Dashboard.	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan semua informasi tentang sistem permintaan layanan ambulan

3. *Use Case Scenario* Lihat Lokasi *Driver*

Use case scenario lihat lokasi *Driver* dapat dilihat pada tabel 3.10.

Tabel 3.10 Use Case Scenario* Lihat Lokasi *Driver

Nama Use Case	Lihat Lokasi *Driver*	
Tujuan	Admin pengelola ambulan berhasil melihat lokasi *Driver*	
Kondisi Awal	Admin pengelola ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk melihat posisi *Driver* ambulan	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil melihat posisi *Driver*	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal melihat posisi *Driver*	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol lihat lokasi *Driver*	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan pesan konfirmasi
	2.	Admin pengelola ambulan menekan tombol ya
	3.	Lokasi *Driver* berhasil diketahui
Skenario Alternatif	Langkah	Aksi
	2.1	Admin pengelola ambulan menekan tombol tidak

4. *Use Case Scenario* Mengelola Data *Driver*

Use case scenario mengelola data *Driver* dapat dilihat pada tabel 3.11.

Tabel 3.11 *Use Case Scenario* Mengelola Data *Driver*

Nama Use Case	Mengelola Data *Driver*	
Tujuan	Admin pengelola ambulan mampu mengelola data *Driver* ambulan.	
Kondisi Awal	Admin pengelola ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk mengelola data *Driver* ambulan.	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil mengelola data *Driver* ambulan	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal mengelola data *Driver* ambulan	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol mengelola data *Driver* ambulan	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan halaman utama.
	2.	Admin pengelola ambulan menekan tombol mengelola data *Driver* ambulan
	3.	Sistem melakukkan validasi *pasien*
	4.	Admin berhasil masuk ke halaman kelola data *Driver*
Skenario Alternatif	Langkah	Aksi
	4.1	Sitem menampilkan pesan gagal memasuki halaman kelola data *Driver*
	4.2	Sistem kembali ke halaman login

5. *Use Case Scenario* Edit Data *Driver*

Use case scenario edit data *Driver* dapat dilihat pada tabel 3.12.

Tabel 3.12 *Use Case Scenario* Edit Data *Driver*

Nama Use Case	Edit Data *Driver*	
Tujuan	Admin pengelola ambulan berhasil mengubah data *Driver* ambulan	
Kondisi Awal	Admin pengelola ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk mengubah data *Driver* ambulan	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil mengubah data *Driver* ambulan	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal mengubah data *Driver* ambulan	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol ubah data *Driver* ambulan	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan form ubah *Driver* ambulan
	2.	Admin pengelola ambulan mengisi form ubah *Driver* ambulan
	3.	Admin pengelola ambulan menekan tombol ubah
	4.	Sistem melakukkan validasi form
	5.	Data *Driver* ambulan berhasil diubah
Skenario Alternatif	Langkah	Aksi
	2.1	Field Form tidak terisi penuh
	2.2	Sistem menampilkan pesan field kosong

	3.1	Admin pengelola ambulan menekan tombol batal
	3.2	Sistem menutup form ubah data *Driver* ambulan

6. *Use Case Scenario* Lihat *History Driver*

Use case scenario history dapat dilihat pada tabel 3.13.

Tabel 3.13 *Use Case Scenario* Lihat *History Driver*

Nama Use Case	Lihat *History Driver*	
Tujuan	*Admin* pengelola ambulan berhasil melihat data *history Driver* ambulan	
Kondisi Awal	Admin pengelola ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk melihat data *history Driver* ambulan.	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil melihat data *history Driver* ambulan	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal melihat data *history Driver* ambulan	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol lihat *history*	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan halaman kelola *Driver*
	2.	Admin pengelola ambulan memilih dan menekan tombol lihat *history Driver*
	3.	Data *History Driver* ambulan berhasil

		dilihat
Skenario Alternatif	Langkah	Aksi
	-	-

7. *Use Case Scenario* Ubah Status *Driver*

Use case scenario ubah status *Driver* dapat dilihat pada tabel 3.14.

Tabel 3.14 *Use Case Scenario* Ubah Status *Driver*

Nama Use Case	Ubah Status *Driver*	
Tujuan	Admin pengelola ambulan berhasil mengubah status *Driver* ambulan	
Kondisi Awal	Admin pengelola ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk meng Approve *Driver* ambulan yang baru mendaftar.	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil mengubah status *Driver* ambulan	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal mengubah status *Driver* ambulan	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol approve *Driver* ambulan	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan data *Driver* ambulan
	2.	Admin pengelola ambulan menekan tombol Approve
	3.	Sistem melakukkan validasi
	4.	Status *Driver* ambulan berhasil diubah
Skenario Alternatif	Langkah	Aksi

	2.1	Admin pengelola ambulan menekan tombol decline
	3.2	Sistem menutup halaman kelola *Driver* ambulan

8. Use Case Scenario Hapus Data *Driver*

Use case scenario hapus data *Driver* dapat dilihat pada tabel 3.15.

Tabel 3.15 *Use Case Scenario* hapus Data *Driver*

Nama Use Case	Hapus Data *Driver*	
Tujuan	Admin pengelola ambulan berhasil menghapus data *Driver* ambulan	
Kondisi Awal	Admin pengelola ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk menghapus data *Driver* ambulan	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil menghapus data *Driver* ambulan	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal menghapus data *Driver* ambulan	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol hapus data *Driver* ambulan	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan pesan konfirmasi penghapusan
	2.	Admin pengelola ambulan menekan tombol ya
	3.	Data *Driver* ambulan berhasil dihapus
Skenario Alternatif	Langkah	Aksi
	2.1	Admin pengelola ambulan menekan tombol tidak

9. *Use Case Scenario* Lihat Permintaan Ambulan

Use case scenario lihat permintaan ambulan dapat dilihat pada tabel 3.16.

Tabel 3.16 *Use Case Scenario* Lihat Permintaan Ambulan

Nama Use Case	Lihat Permintaan Ambulan	
Tujuan	Admin pengelola ambulan berhasil melihat informasi permintaan ambulan	
Kondisi Awal	Admin pengelola ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk melihat semua data permintaan layanan ambulan yang masuk.	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil melihat permintaan layanan ambulan	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal melihat permintaan layanan ambulan	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol permintaan ambulan	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan halaman utama
	2.	Admin pengelola ambulan menekan tombol permitaan ambulan
	3.	Daftar permintaan layanan ambulan berhasil ditampilkan
Skenario Alternatif	Langkah	Aksi
	2.1	Admin pengelola ambulan menekan tombol tidak

10. *Use Case Scenario* Lihat Map

Use case scenario lihat map dapat dilihat pada tabel 3.17.

Tabel 3.17 *Use Case Scenario* Lihat Map

Nama Use Case	Lihat Map	
Tujuan	Admin pengelola ambulan berhasil melihat informasi permintaan layanan ambulan melalui map	
Kondisi Awal	Admin pengelola ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk melihat data permintaan melalui map.	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil melihat informasi permintaan layanan ambulan melalui map	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal melihat informasi permintaan layanan ambulan melalui map	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol lihat map	
Skenario Utama	Langkah	Aksi
	1.	Admin pengelola ambulan menekan tombol lihat map
	2.	Informasi permintaan layanan ambulan berhasil dilihat melalui map
Skenario Alternatif	Langkah	Aksi
	2.1	Informasi permintaan layanan ambulan gagal dimuat.

11. *Use Case Scenario* Mengelola Data Pasien

Use case scenario mengelola data pasien dapat dilihat pada tabel 3.18.

Tabel 3.18 *Use Case Scenario* Mengelola Data Pasien

Nama Use Case	*Mengelola Data* pasien	
Tujuan	Admin pengelola ambulan mampu mengelola data *pasien.*	
Kondisi Awal	Admin pengelola ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk mengelola data *pasien.*	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil mengelola data *pasien.*	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal mengelola data *pasien.*	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol mengelola data *pasien.*	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan halaman utama.
	2.	Admin pengelola ambulan menekan tombol mengelola data *pasien*
	3.	Sistem melakukkan validasi *pasien*
	4.	Admin berhasil masuk ke halaman kelola data *pasien*
Skenario Alternatif	Langkah	Aksi
	4.1	Sitem menampilkan pesan gagal memasuki halaman kelola data *pasien*
	4.2	Sistem kembali ke halaman login

12. *Use Case Scenario* Edit Data *Pasien*

Use case scenario edit data *Pasien* dapat dilihat pada tabel 3.19.

Tabel 3.19 *Use Case Scenario* Edit Data *Pasien*

Nama Use Case	Edit Data *Pasien*	
Tujuan	Admin pengelola ambulan berhasil mengubah data *pasien*	
Kondisi Awal	Admin pengelola ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk mengubah *data pasien.*	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil mengubah data *pasien*	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal mengubah data *pasien*	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol ubah data *pasien*	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan pesan konfirmasi pengubahan data *pasien*
	2.	*Pasien* atau *Driver* ambulan menekan tombol ya
	3.	Data *pasien* berhasil diubah
Skenario Alternatif	Langkah	Aksi
	2.1	*Pasien* atau *Driver* ambulan menekan tombol tidak

13. *Use Case Scenario* Lihat *History Pasien*

Use case scenario lihat *history pasien* dapat dilihat pada tabel 3.20.

Tabel 3.20 *Use Case Scenario* Lihat *History Pasien*

Nama Use Case	Lihat *History Pasien*	
Tujuan	*Admin* pengelola ambulan berhasil melihat data *history pasien*	
Kondisi Awal	Admin pengelola ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk melihat data *history pasien.*	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil melihat data *history pasien.*	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal melihat data *history pasien*	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol lihat *history*	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan halaman kelola *pasien*
	2.	Admin pengelola ambulan memilih dan menekan tombol lihat *history pasien*
	3.	Data *History pasien* berhasil dilihat
Skenario Alternatif	Langkah	Aksi
	-	-

14. *Use Case Scenario* Hapus Data *Pasien*

Use case scenario hapus data *pasien* dapat dilihat pada tabel 3.21.

Tabel 3.21 *Use Case Scenario* hapus Data *Driver*

Nama Use Case	Hapus Data *Driver*	
Tujuan	Admin pengelola ambulan berhasil menghapus data *pasien*	
Kondisi Awal	Admin pengelola ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk menghapus data *pasien*	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil menghapus data *pasien*	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal menghapus data *pasien*	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol hapus data *pasien*	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan pesan konfirmasi penghapusan
	2.	Admin pengelola ambulan menekan tombol ya
	3.	Data *pasien* berhasil dihapus
Skenario Alternatif	Langkah	Aksi
	2.1	Admin pengelola ambulan menekan tombol tidak

15. *Use Case Scenario* Mengelola Tipe Layanan

Use case scenario mengelola tipe layanan dapat dilihat pada tabel 3.22.

Tabel 3.22 *Use Case Scenario* Mengelola Tipe Layanan

Nama Use Case	Mengelola Tipe Layanan	
Tujuan	Admin pengelola ambulan mampu mengelola data tipe layanan ambulan.	
Kondisi Awal	Admin pengelola ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk mengelola data tipe layanan.	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil mengelola data tipe layanan ambulan	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal mengelola data tipe layanan ambulan	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol tipe layanan	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan halaman utama.
	2.	Admin pengelola ambulan menekan tombol tipe layanan
	3.	Sistem melakukkan validasi *pasien*
	4.	Admin berhasil masuk ke halaman tipe layanan
Skenario Alternatif	Langkah	Aksi
	4.1	Sitem menampilkan pesan gagal memasuki halaman tipe layanan
	4.2	Sistem kembali ke halaman login

16. *Use Case Scenario* Tambah Data Tipe Layanan

Use case scenario tambah data tipe layanan dapat dilihat pada tabel 3.23.

Tabel 3.23 *Use Case Scenario* Tambah Data Tipe Layanan

Nama Use Case	Tambah Data Tipe Layanan	
Tujuan	Admin pengelola ambulan berhasil menambahkan tipe layanan	
Kondisi Awal	Admin pengelola ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk menambah data tipe layanan.	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil menambahkan tipe layanan	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal menambahkan tipe layanan	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol tambah tipe layanan	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan form tambah tipe layanan
	2.	Admin pengelola ambulan mengisi form tambah tipe layanan
	3.	Admin pengelola ambulan menekan tombol tambah
	4.	Sistem melakukkan validasi form
	5.	Data tipe layanan berhasil ditambah
Skenario Alternatif	Langkah	Aksi
	2.1	Field Form tidak terisi penuh
	2.2	Sistem menampilkan pesan field kosong
	3.1	Admin pengelola ambulan menekan tombol batal

	3.2	Sistem menutup form tambah tipe layanan

17. *Use Case Scenario* Edit Data Tipe Layanan

Use case scenario edit data tipe layanan dapat dilihat pada tabel 3.24.

Tabel 3.24 *Use Case Scenario* Edit Data Tipe Layanan

Nama Use Case	Edit Data Tipe Layanan	
Tujuan	Admin pengelola ambulan berhasil mengubah data tipe layanan	
Kondisi Awal	Admin pengelola ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk mengubah data tipe layanan	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil mengubah data tipe layanan	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal mengubah data tipe layanan	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol ubah data tipe layanan	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan form ubah tipe layanan
	2.	Admin pengelola ambulan mengisi form ubah tipe layanan
	3.	Admin pengelola ambulan menekan tombol ubah
	4.	Sistem melakukkan validasi form
	5.	Data tipe layanan berhasil diubah
Skenario Alternatif	Langkah	Aksi

	2.1	Field Form tidak terisi penuh
	2.2	Sistem menampilkan pesan field kosong
	3.1	Admin pengelola ambulan menekan tombol batal
	3.2	Sistem menutup form ubah data tipe layanan

18. *Use Case Scenario* Mengelola Data Admin

Use case scenario Admin dapat dilihat pada tabel 3.25.

Tabel 3.25 *Use Case Scenario* Admin

Nama Use Case	Mengelola Data Admin	
Tujuan	Admin pengelola ambulan mampu mengelola data Admin ambulan.	
Kondisi Awal	Admin pengelola ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk mengelola data Admin.	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil mengelola data Admin ambulan	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal mengelola data Admin ambulan	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol Admin	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan halaman utama.
	2.	Admin pengelola ambulan menekan tombol Admin
	3.	Sistem melakukkan validasi *pasien*
	4.	Admin berhasil masuk ke halaman Admin

Skenario Alternatif	Langkah	Aksi
	4.1	Sitem menampilkan pesan gagal memasuki halaman Admin
	4.2	Sistem kembali ke halaman login

19. *Use Case Scenario* Tambah Data Admin

Use case scenario tambah data admin dapat dilihat pada tabel 3.26.

Tabel 3.26 *Use Case Scenario* Tambah Data Admin

Nama Use Case	Tambah Data Admin	
Tujuan	Admin pengelola ambulan berhasil menambahkan Admin	
Kondisi Awal	Admin pengelola ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk menambah data Admin.	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil menambahkan Admin	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal menambahkan Admin	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol tambah Admin	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan form tambah Admin
	2.	Admin pengelola ambulan mengisi form tambah Admin
	3.	Admin pengelola ambulan menekan tombol tambah
	4.	Sistem melakukkan validasi form

	5.	Data Admin berhasil ditambah
Skenario Alternatif	Langkah	Aksi
	2.1	Field Form tidak terisi penuh
	2.2	Sistem menampilkan pesan field kosong
	3.1	Admin pengelola ambulan menekan tombol batal
	3.2	Sistem menutup form tambah Admin

20. *Use Case Scenario* Edit Data Admin

Use case scenario edit data admin dapat dilihat pada tabel 3.27.

Tabel 3.27 *Use Case Scenario* Edit Data Admin

Nama Use Case	Edit Data Admin	
Tujuan	Admin pengelola ambulan berhasil mengubah data Admin	
Kondisi Awal	Admin pengelola ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk mengubah data Admin	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil mengubah data Admin	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal mengubah data Admin	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol ubah data Admin	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan form ubah Admin
	2.	Admin pengelola ambulan mengisi

		form ubah Admin
	3.	Admin pengelola ambulan menekan tombol ubah
	4.	Sistem melakukkan validasi form
	5.	Data Admin berhasil diubah
Skenario Alternatif	Langkah	Aksi
	2.1	Field Form tidak terisi penuh
	2.2	Sistem menampilkan pesan field kosong
	3.1	Admin pengelola ambulan menekan tombol batal
	3.2	Sistem menutup form ubah data Admin

21. *Use Case Scenario Logout*

Use case scenario logout dapat dilihat pada tabel 3.28.

Tabel 3.28 *Use Case Scenario Logout*

Nama Use Case	*Logout*	
Tujuan	Admin pengelola ambulan berhasil keluar dari sistem	
Kondisi Awal	Menampilkan halaman utaman	
Deskripsi	Fungsional ini digunakan oleh Admin pengelola ambulan untuk keluar dari sistem	
Kondisi Akhir (Jika Berhasil)	Admin pengelola ambulan berhasil masuk keluar dari sistem	
Kondisi Akhir (Jika Gagal)	Admin pengelola ambulan gagal keluar dari sistem	
Aktor	Admin pengelola ambulan	
Trigger	Admin pengelola ambulan menekan tombol logout	
Skenario Utama	Langkah	Aksi
	1.	Admin pengelola ambulan menekan tombol logout

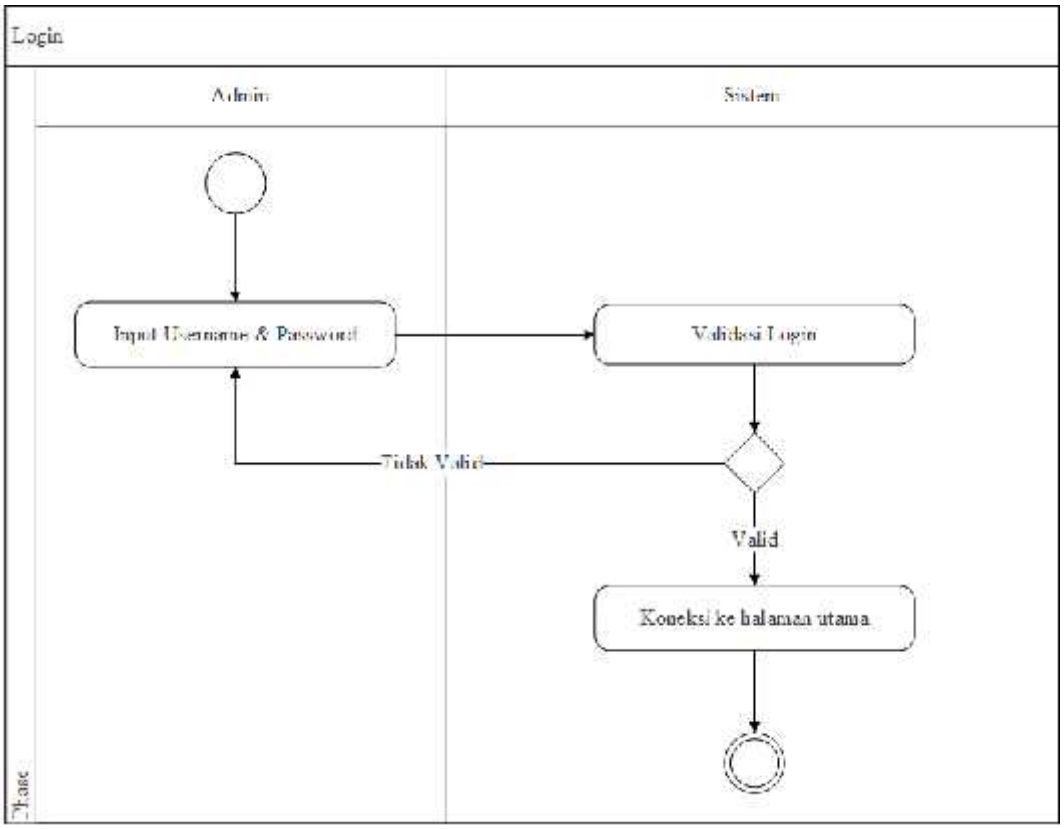
	2.	Admin berhasil keluar dari sistem
Skenario Alternatif	Langkah	Aksi
	-	-

3.1.7.1.3. *Activity Diagram Back End*

Activity diagram menggambarkan alur aktivitas pada sistem yang sedang dibangun. *Activity diagram* dari pembangunan sistem permintaan layanan ambulan di rumah sakit Al-Islam Bandung ini dibuat berdasarkan beberapa *use case* yang dapat dilihat pada gambar sebagai berikut :

1. *Activity Diagram Login*

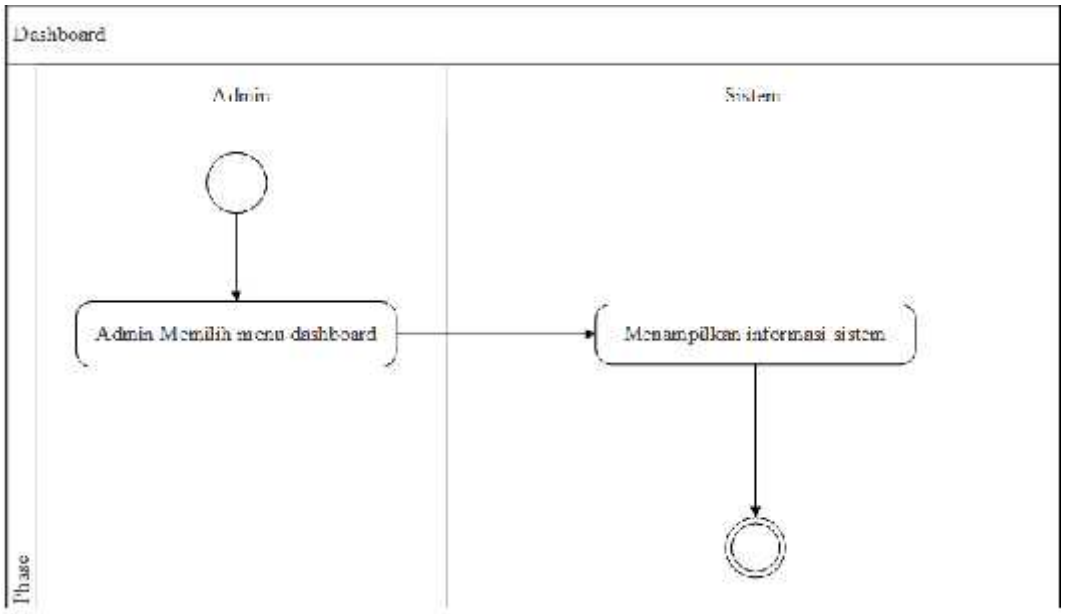
Activity diagram login dapat dilihat pada gambar 3.6.



Gambar 3.6 *Activity Diagram Login* Admin

2. *Activity Diagram Dashboard*

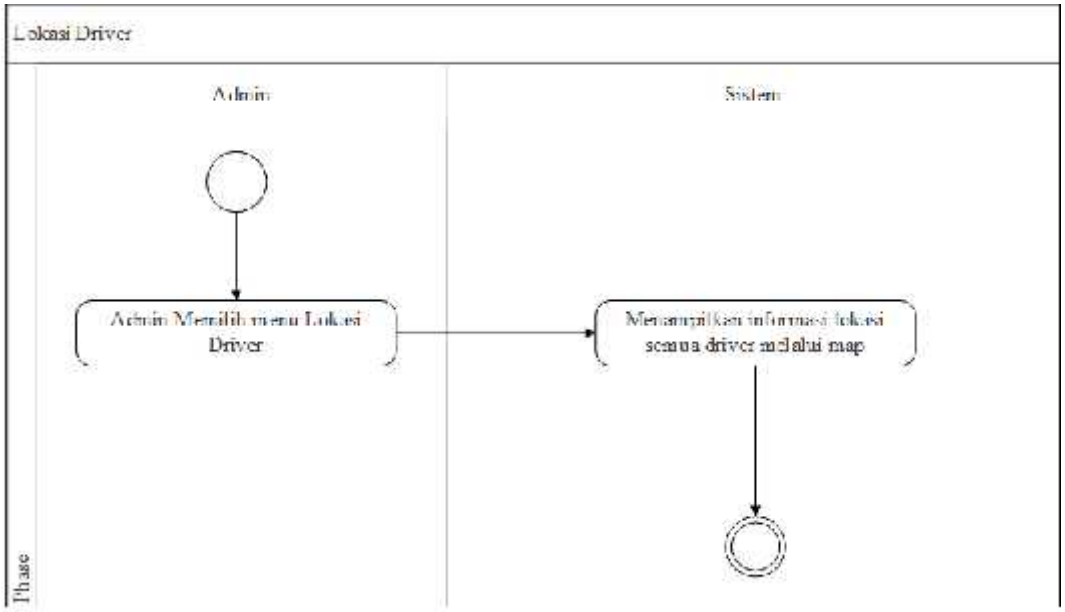
Activity diagram dashboard dapat dilihat pada gambar 3.7.



Gambar 3.7 *Activity Diagram Dashboard*

3. *Activity Diagram* Lokasi *Driver*

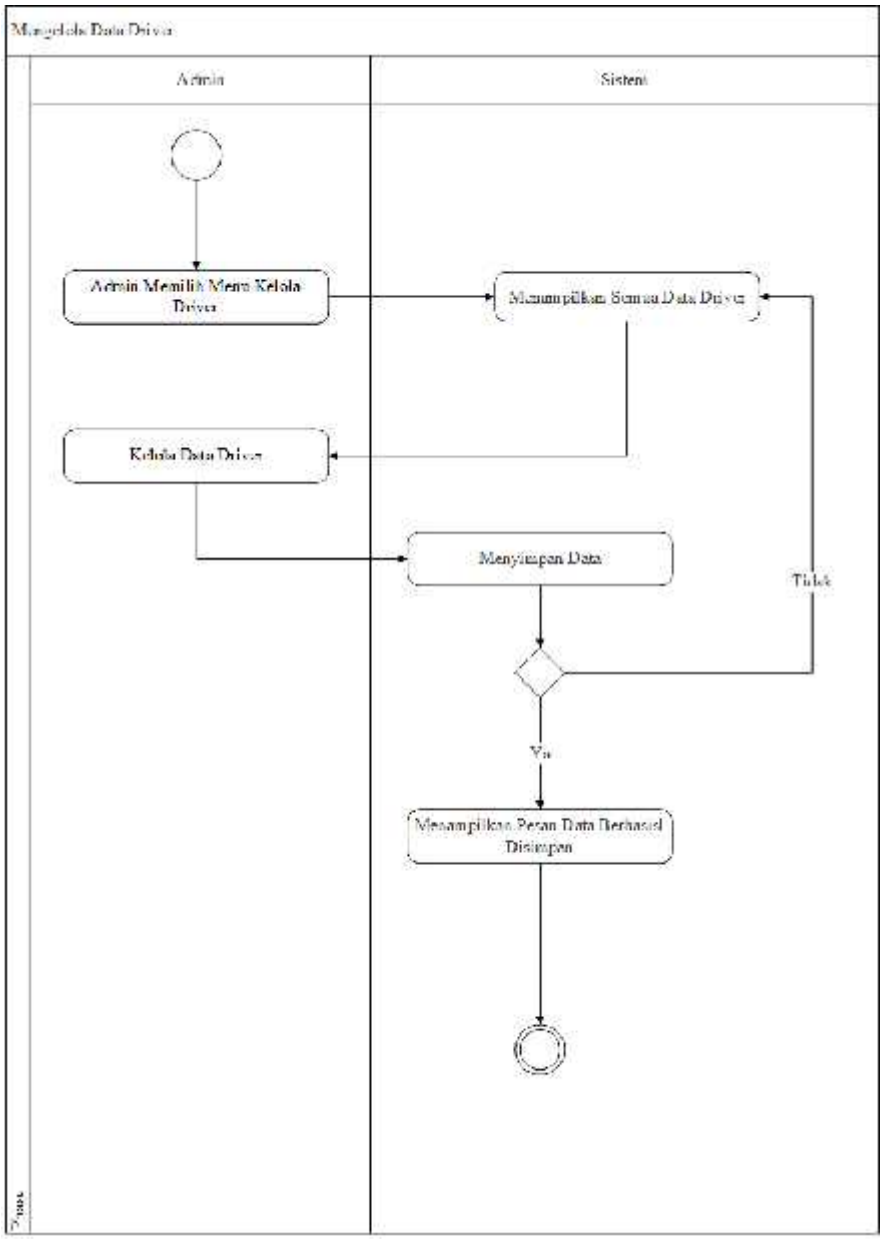
Activity diagram lokasi Driver dapat dilihat pada gambar 3.8.



Gambar 3.8 *Activity Diagram Dashboard*

4. *Activity Diagram* Mengelola Data *Driver*

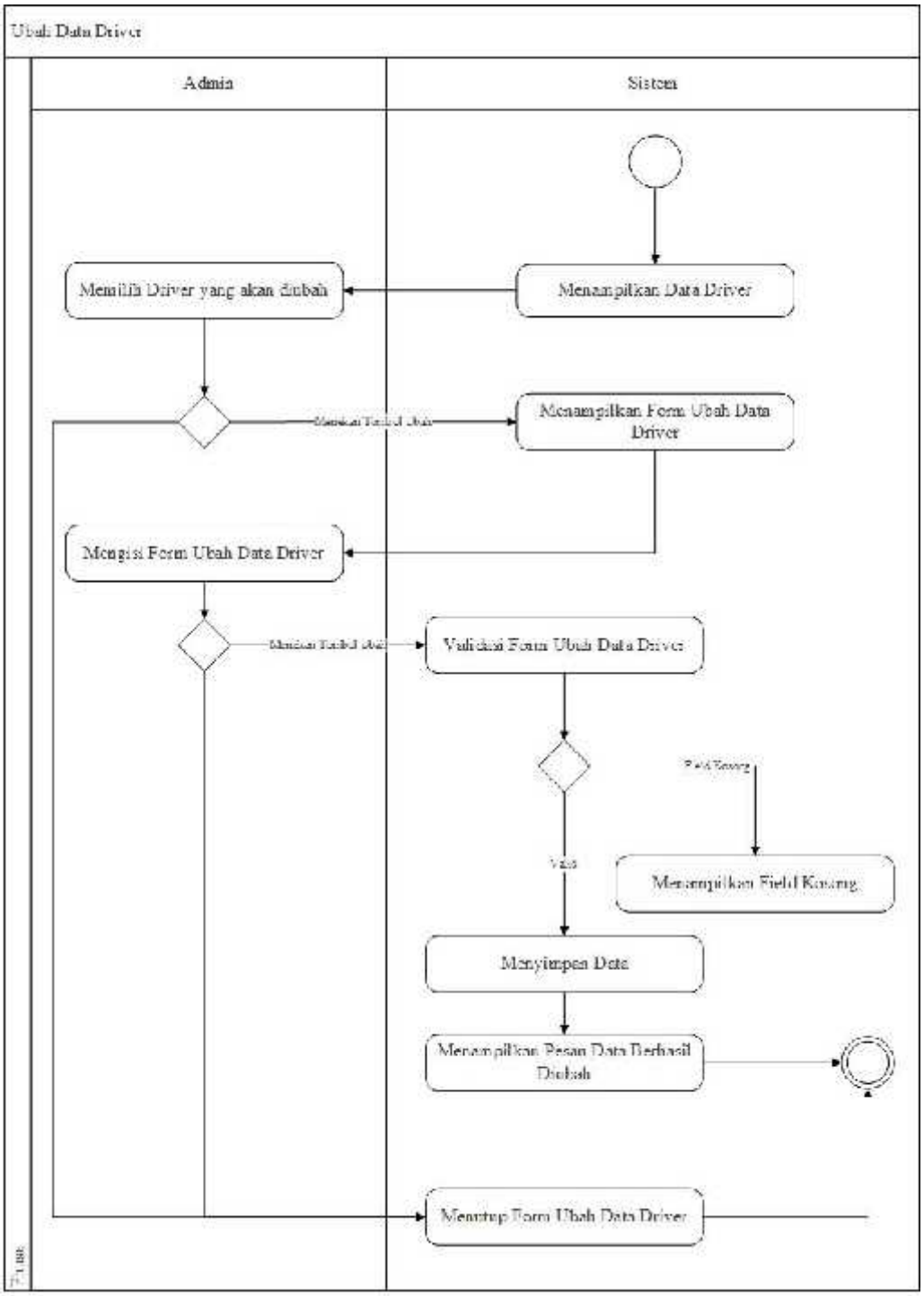
Activity diagram mengelola data *Driver* dapat dilihat pada gambar 3.9.



Gambar 3.9 *Activity Diagram* Mengelola Data *Driver*

5. *Activity Diagram* Ubah Data *Driver*

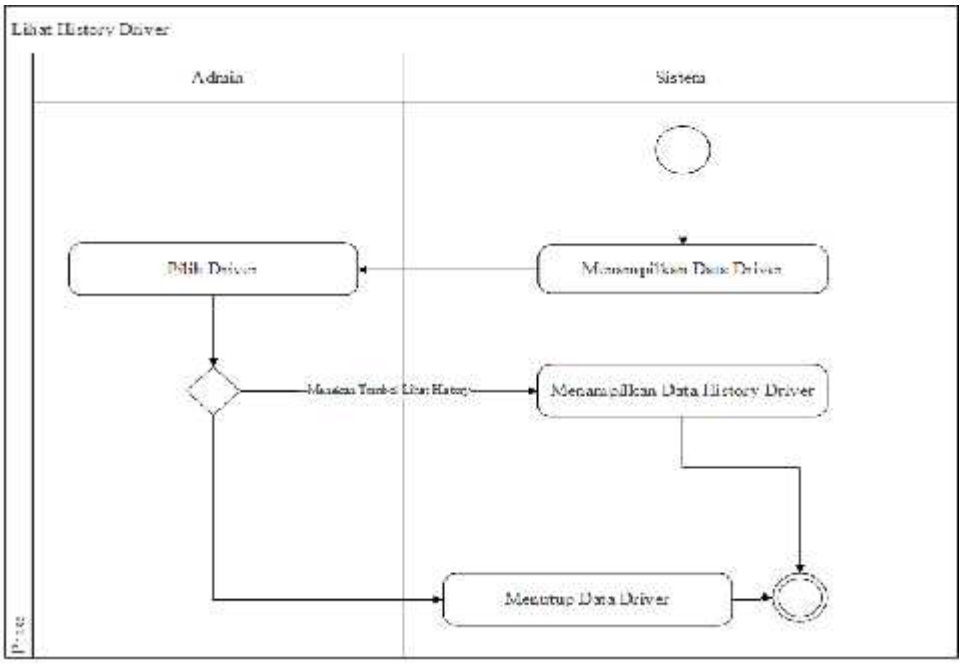
Activity diagram ubah data *Driver* dapat dilihat pada gambar 3.10.



Gambar 3.10 *Activity Diagram* Ubah Data *Driver*

6. *Activity Diagram* Lihat *History Driver*

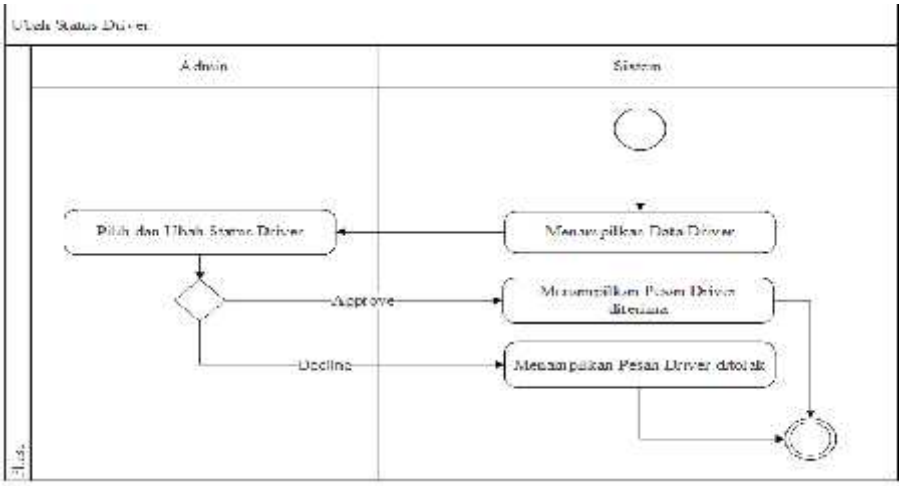
Activity diagram lihat *history Driver* dapat dilihat pada gambar 3.11.



Gambar 3.11 *Activity Diagram* Lihat *History Driver*

7. *Activity Diagram* Ubah Status *Driver*

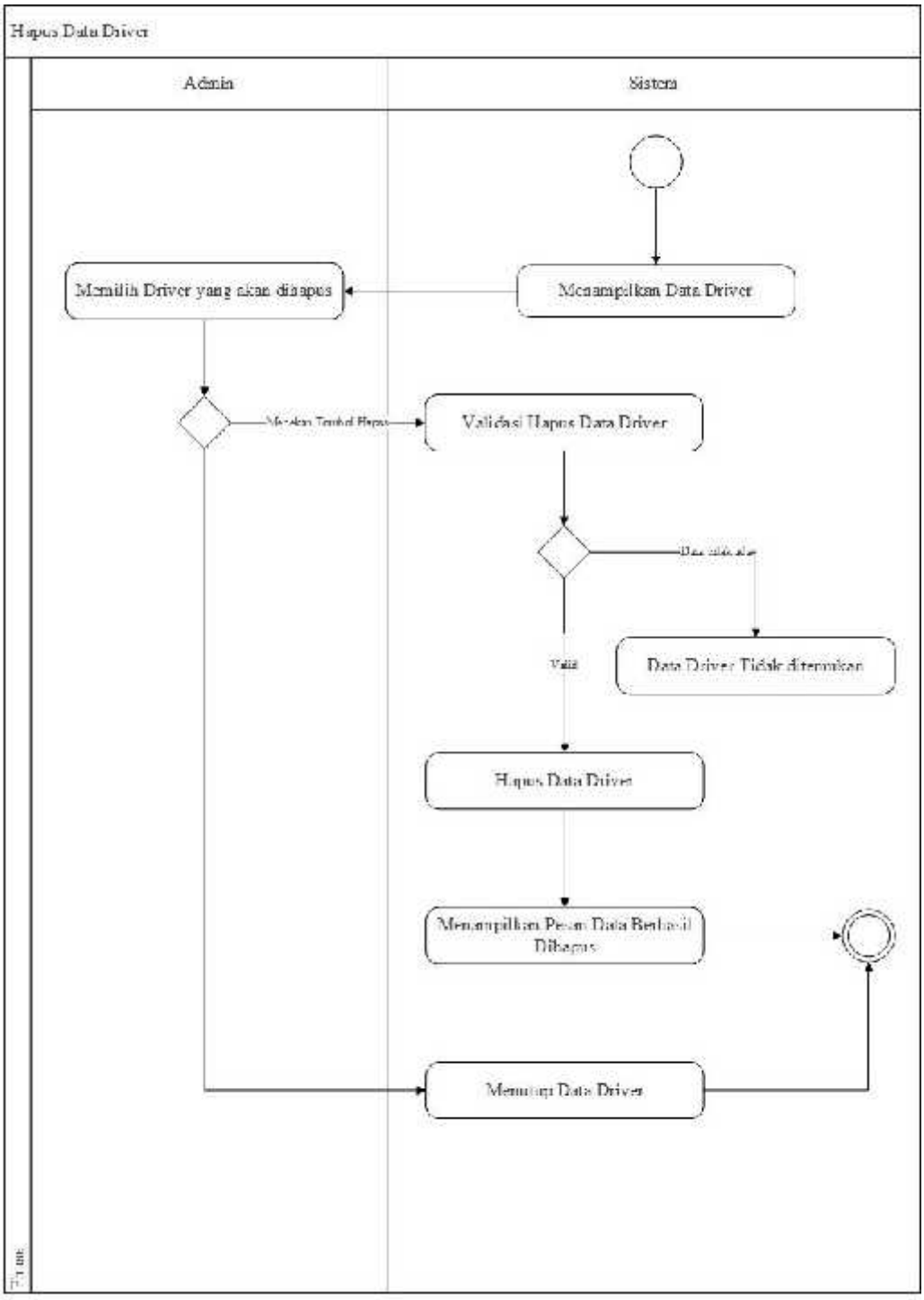
Activity diagram ubah status *Driver* dapat dilihat pada gambar 3.12.



Gambar 3.12 *Activity Diagram* Ubah Status *Driver*

8. *Activity Diagram* Hapus Data *Driver*

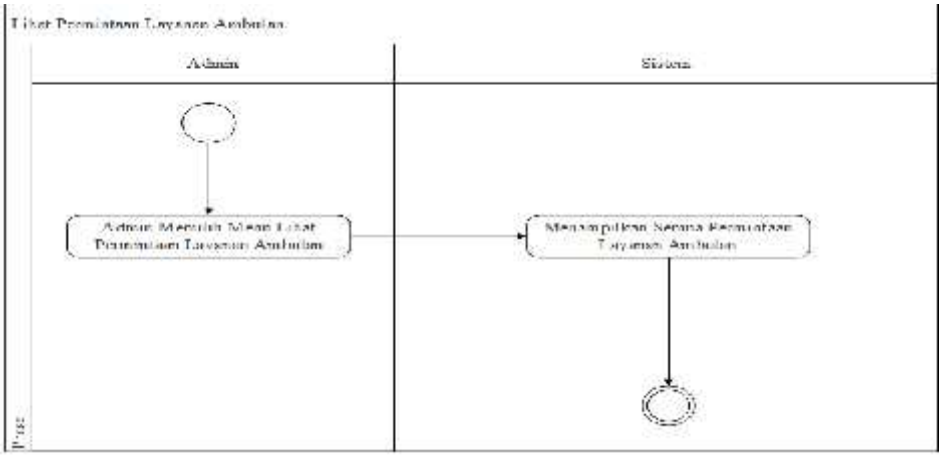
Activity diagram hapus data *Driver* dapat dilihat pada gambar 3.8.



Gambar 3.13 *Activity Diagram* Hapus Data *Driver*

9. *Activity Diagram* Lihat Permintaan Ambulan

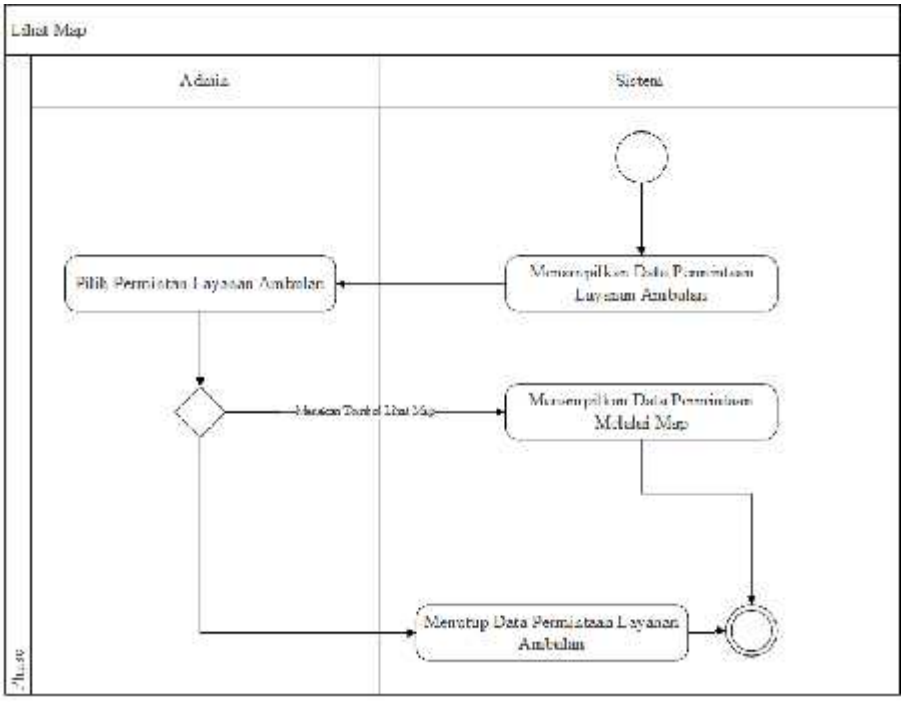
Activity diagram lihat permintaan ambulan dapat dilihat pada gambar 3.14.



Gambar 3.14 *Activity Diagram* Lihat Permintaan Ambulan

10. *Activity Diagram* Lihat Map

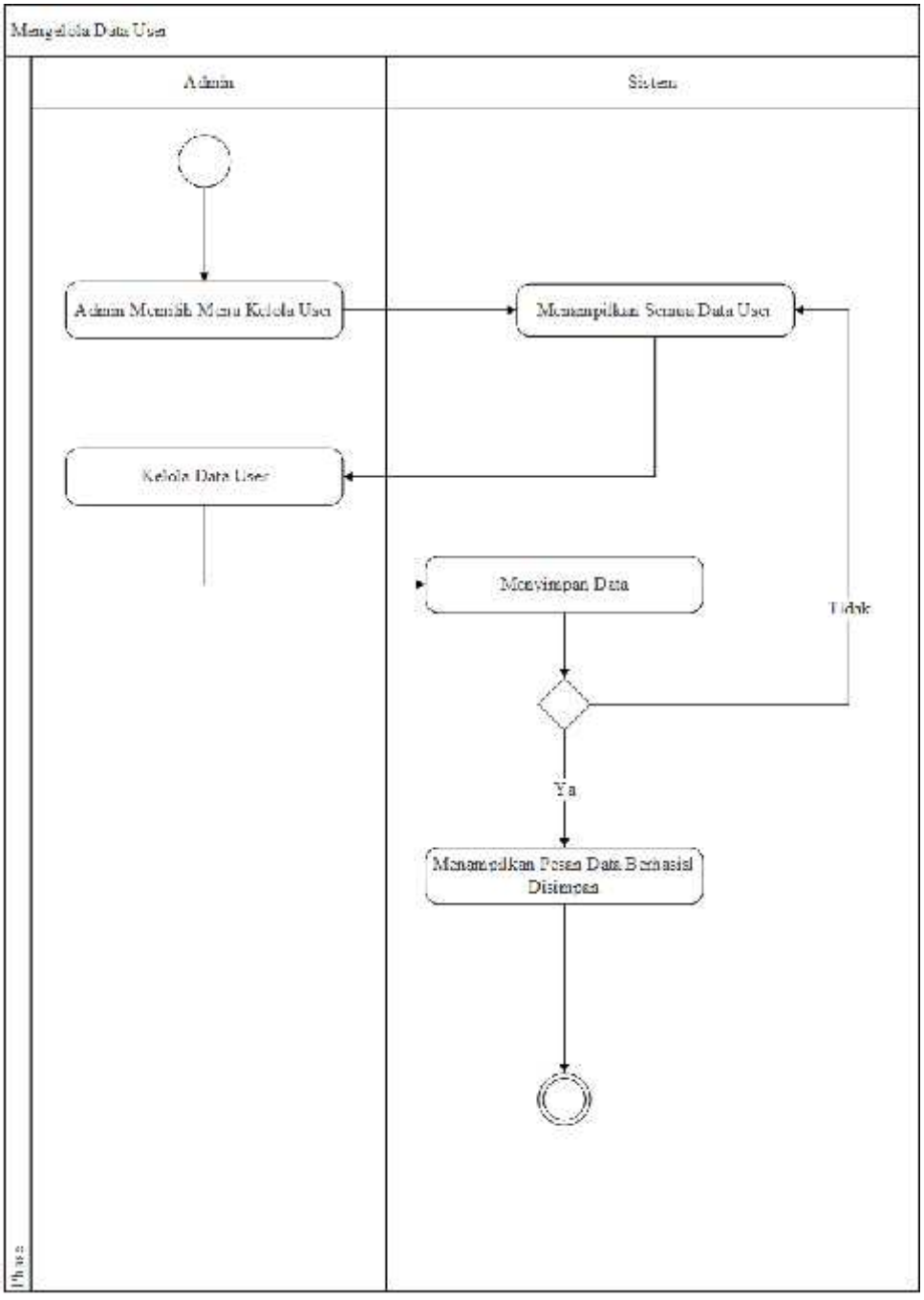
Activity diagram lihat map dapat dilihat pada gambar 3.15.



Gambar 3.15 *Activity Diagram* Lihat Map

11. *Activity Diagram* Mengelola Data *Pasien*

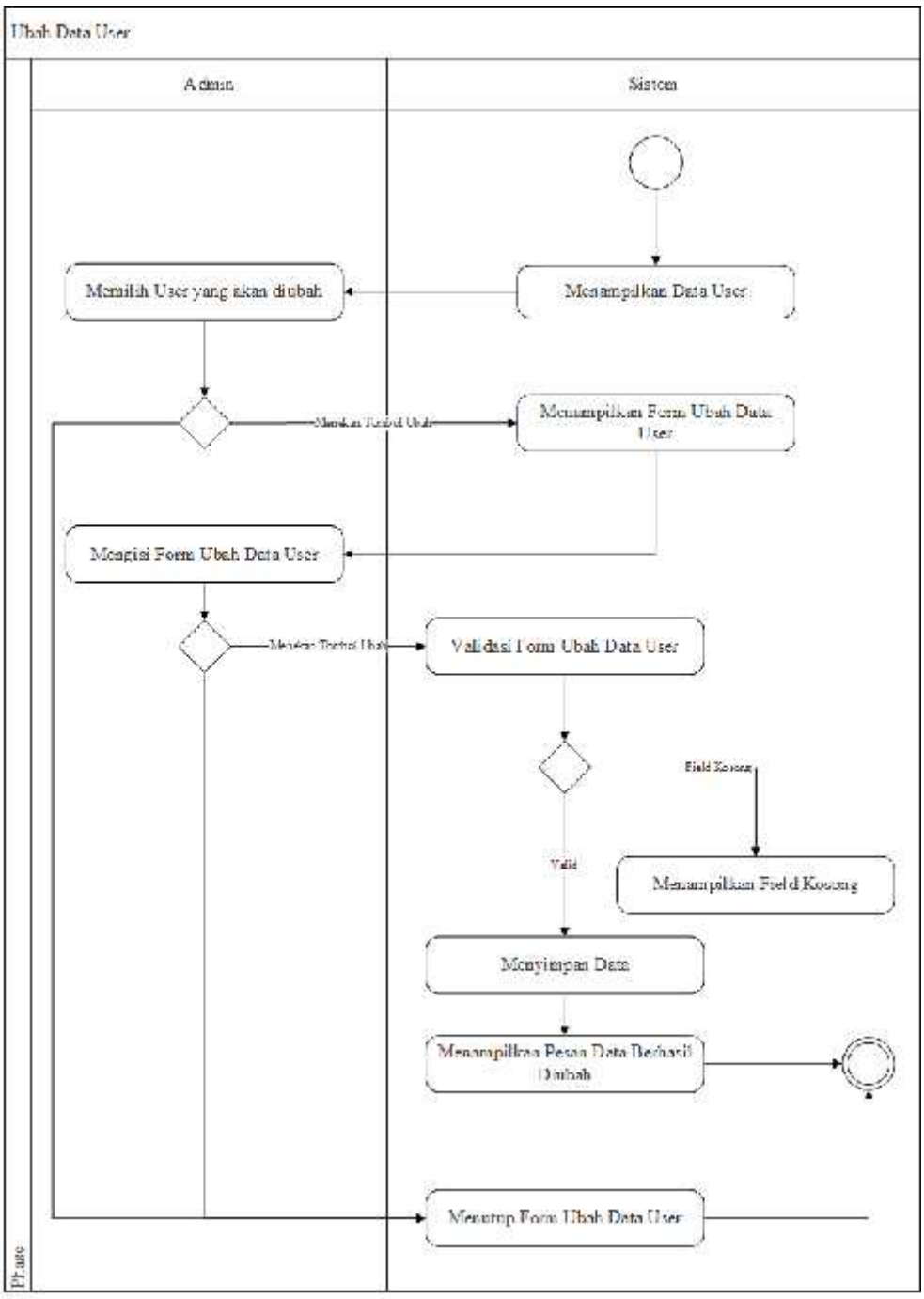
Activity diagram mengelola data *pasien* dapat dilihat pada gambar 3.16.



Gambar 3.16 *Activity Diagram* Mengelola Data *Pasien*

12. *Activity Diagram* Ubah Data *Pasien*

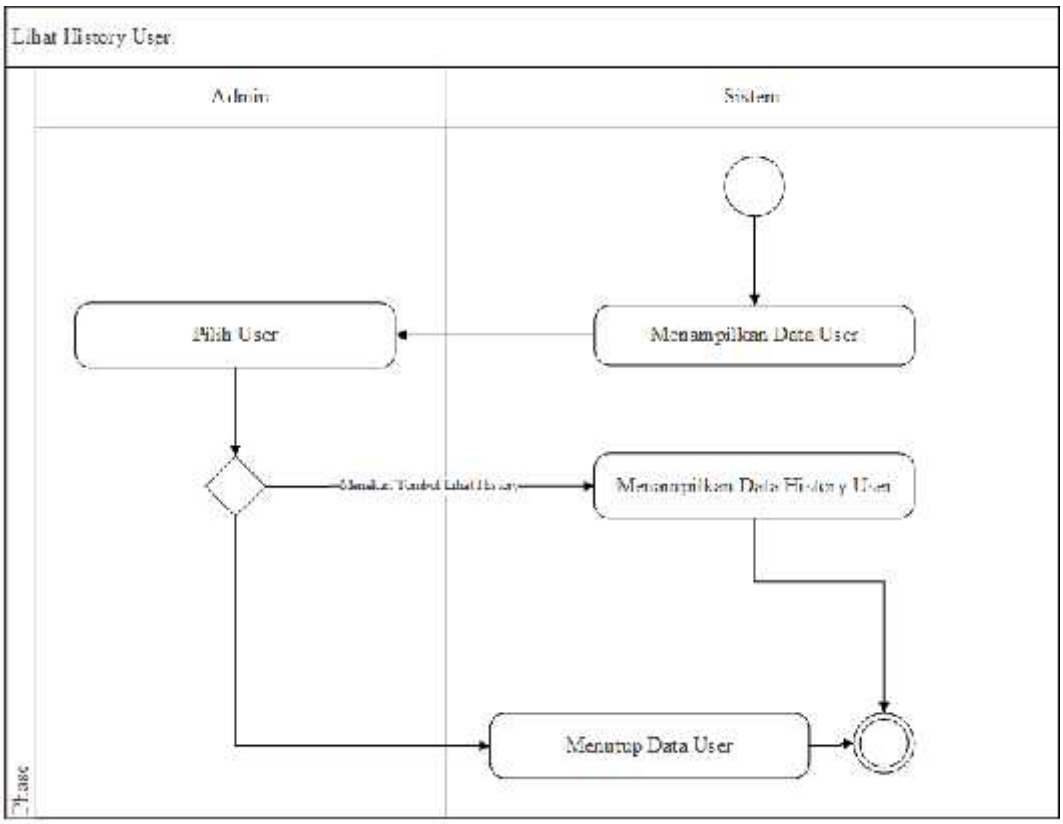
Activity diagram ubah data *pasien* dapat dilihat pada gambar 3.17.



Gambar 3.17 *Activity Diagram* Ubah Data *Pasien*

13. *Activity Diagram* Lihat *History Pasien*

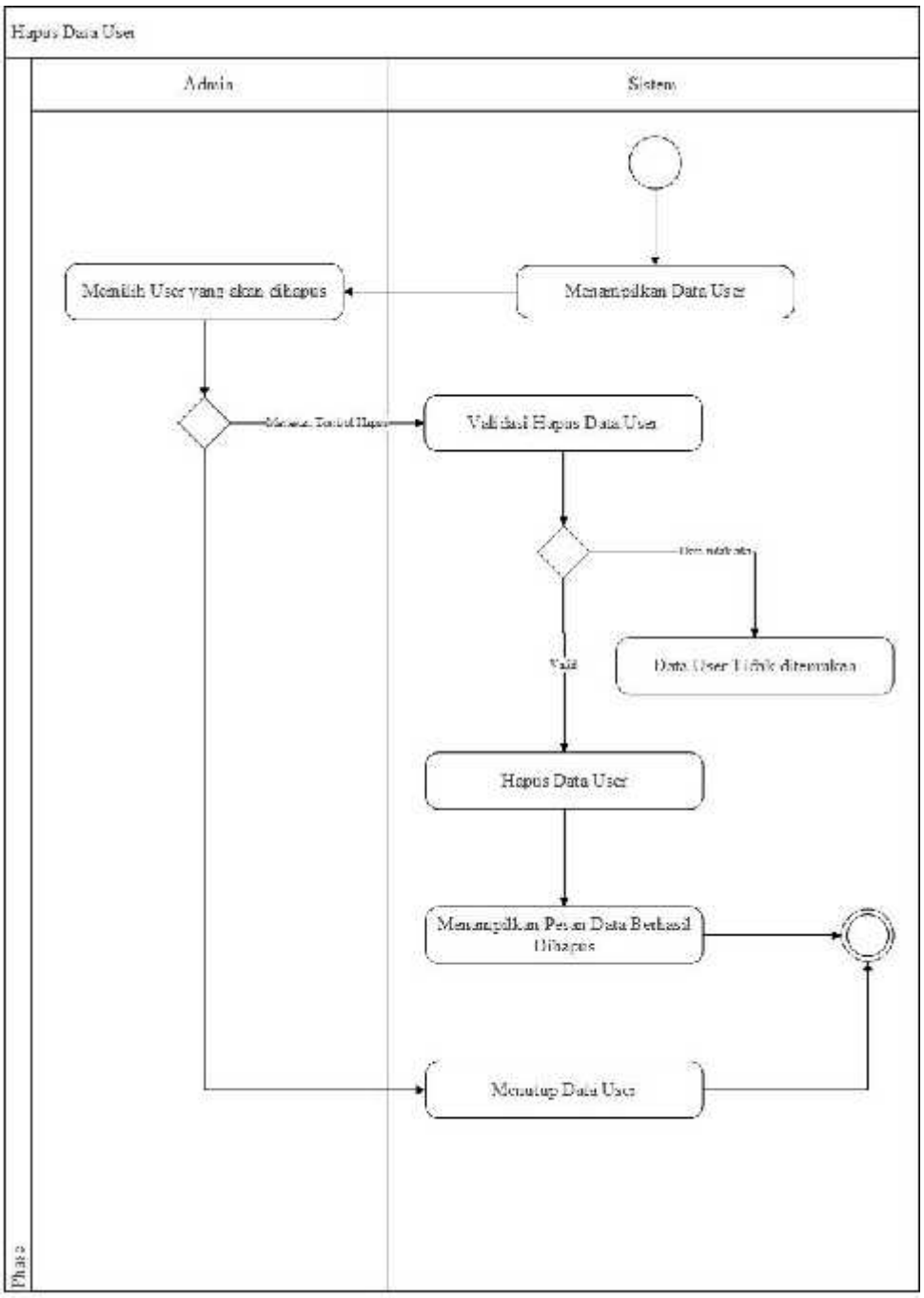
Activity diagram lihat *history pasien* dapat dilihat pada gambar 3.18.



Gambar 3.18 *Activity Diagram* Lihat *History Pasien*

14. *Activity Diagram* Hapus Data *Pasien*

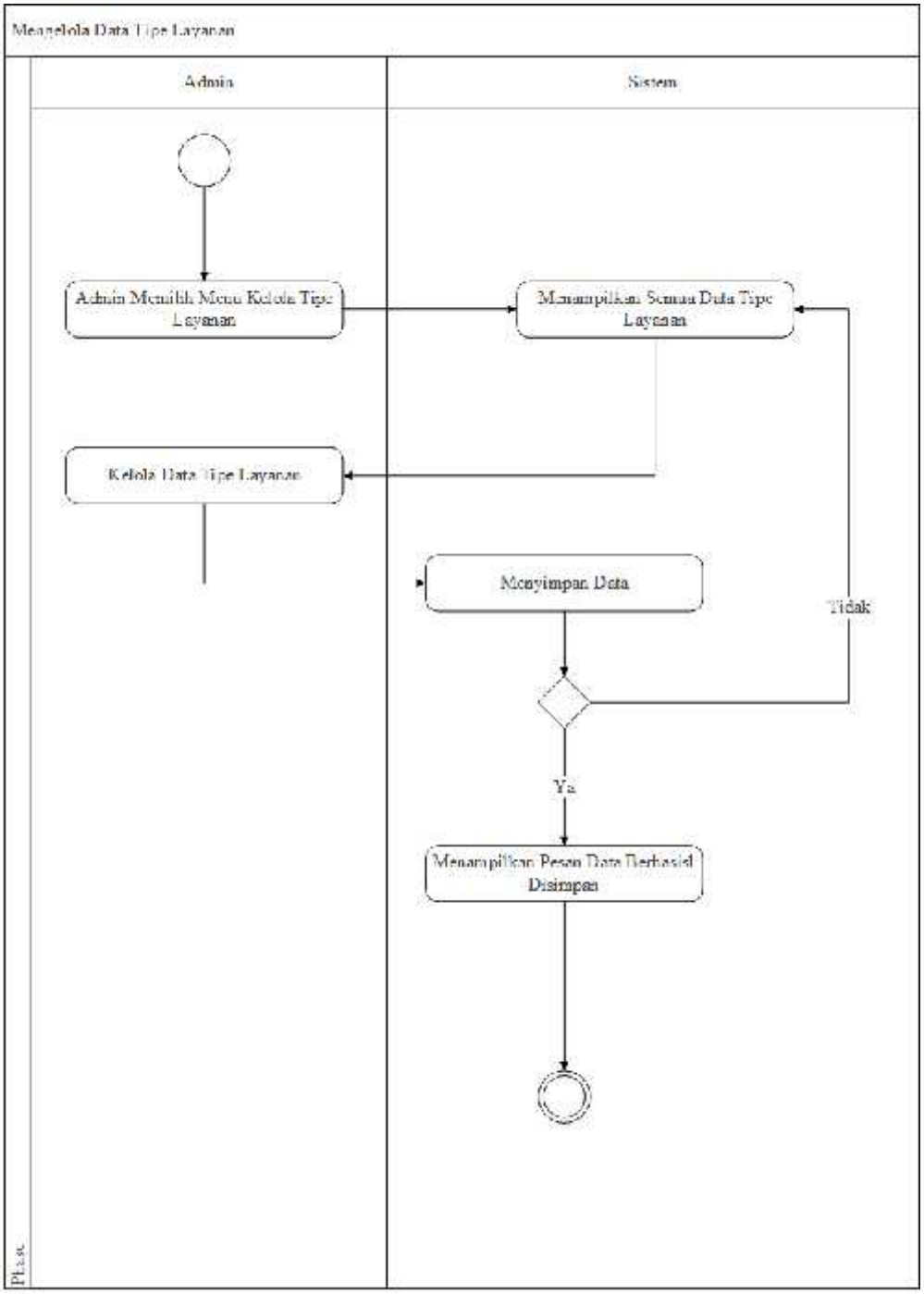
Activity diagram hapus data *pasien* dapat dilihat pada gambar 3.19.



Gambar 3.19 *Activity Diagram* Hapus Data *Pasien*

15. *Activity Diagram* Mengelola Tipe Layanan

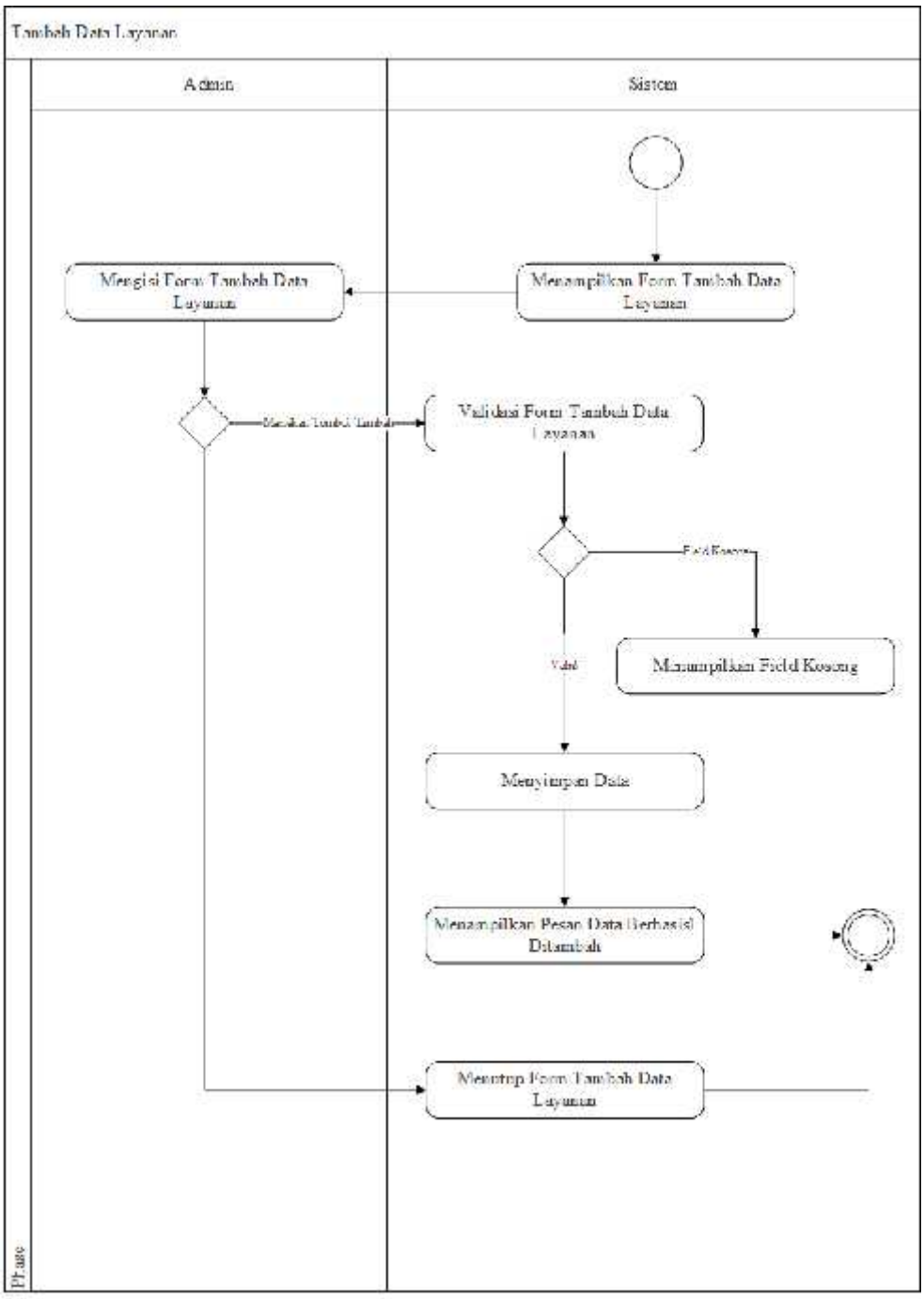
Activity diagram mengelola tipe layanan dapat dilihat pada gambar 3.20.



Gambar 3.20 *Activity Diagram* Mengelola Tipe Layanan

16. *Activity Diagram* Tambah Data Layanan

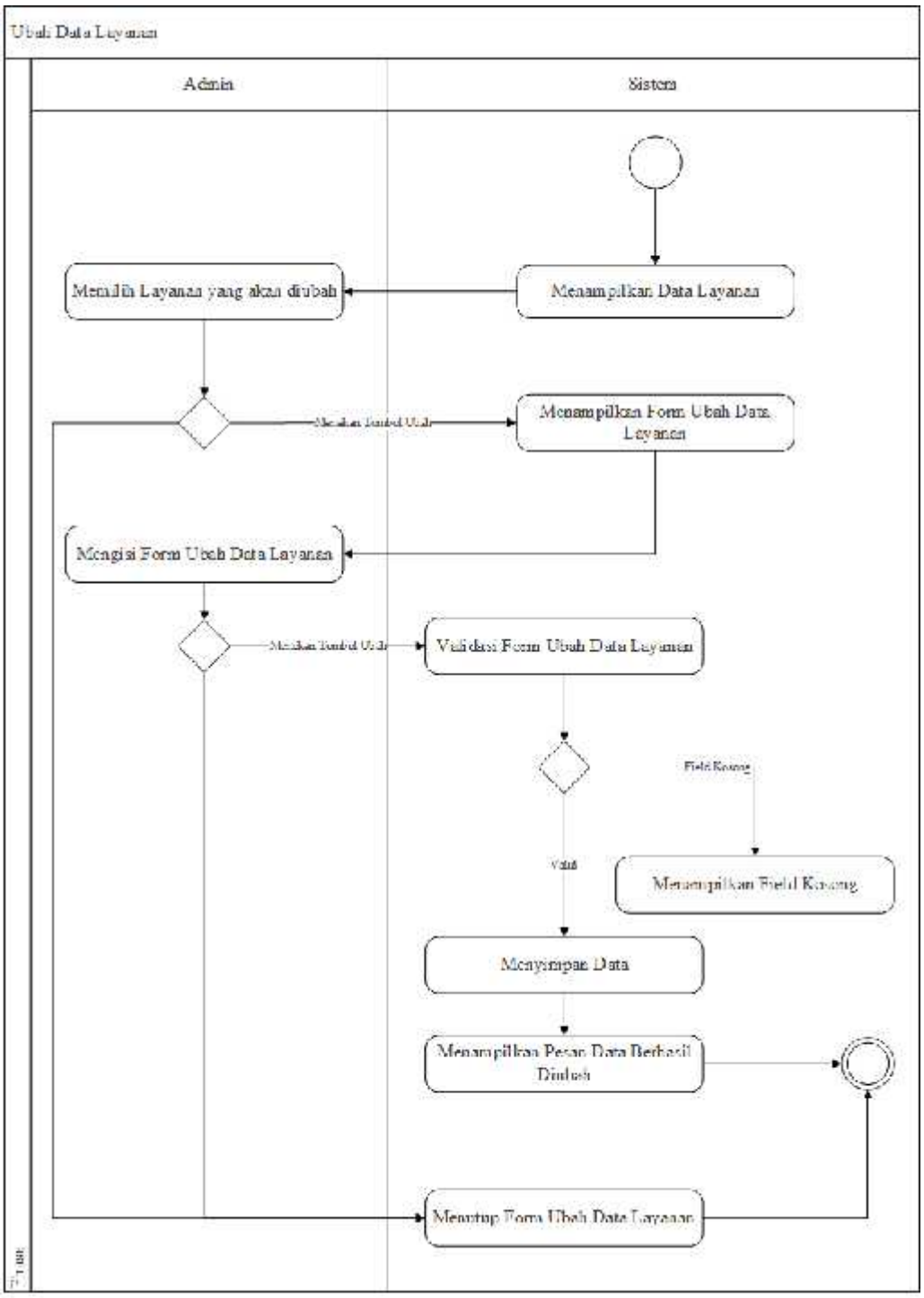
Activity diagram tambah data layanan dapat dilihat pada gambar 3.21.



Gambar 3.21 *Activity Diagram* Tambah Data Layanan

17. *Activity Diagram* Ubah Layanan

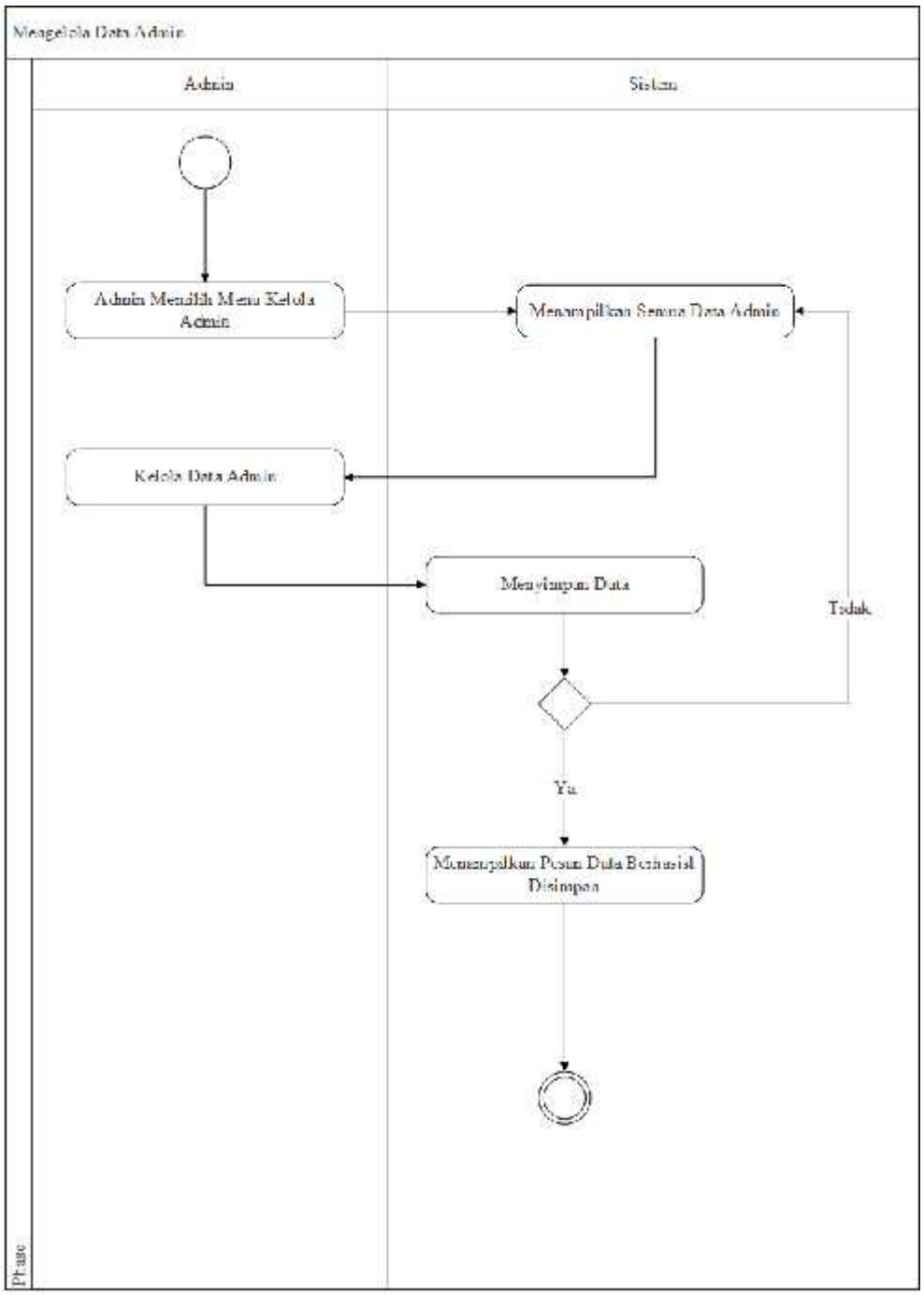
Activity diagram ubah layanan dapat dilihat pada gambar 3.22.



Gambar 3.22 *Activity Diagram* Ubah Layanan

18. *Activity Diagram* Mengelola Admin

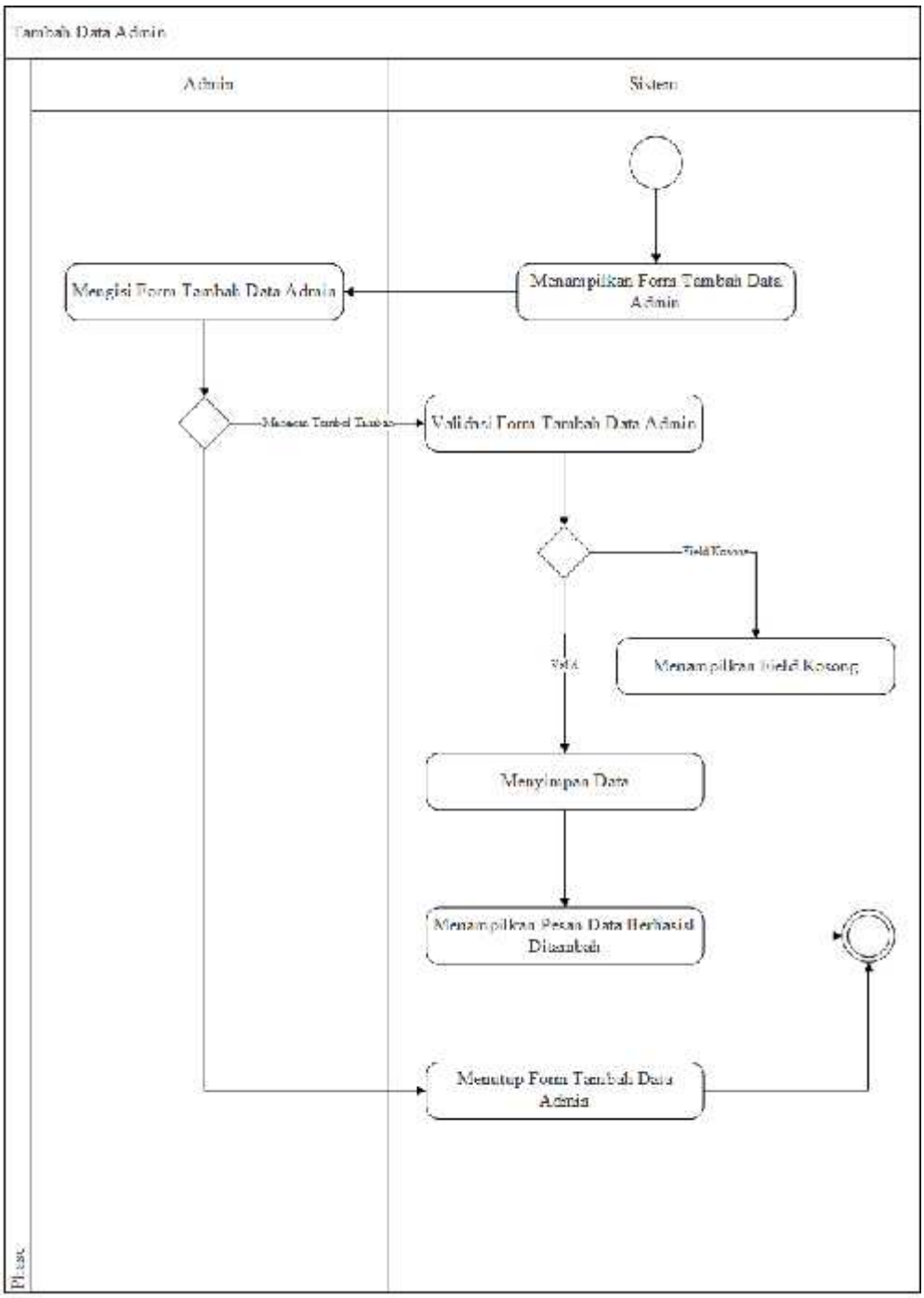
Activity diagram mengelola admin dapat dilihat pada gambar 3.23.



Gambar 3.23 *Activity Diagram* Mengelola Admin

19. *Activity Diagram* Tambah Data Admin

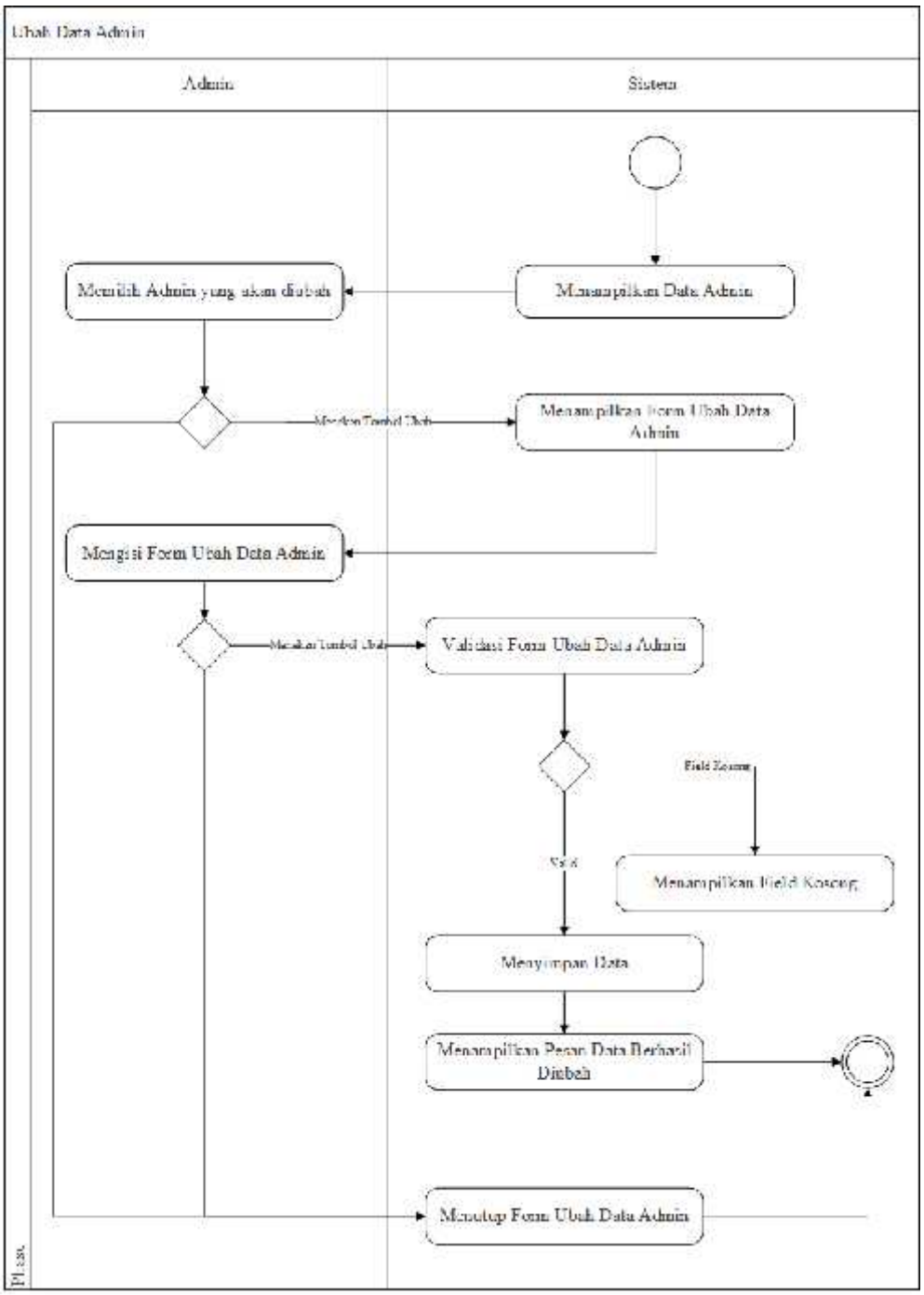
Activity diagram tambah data admin dapat dilihat pada gambar 3.24.



Gambar 3.24 *Activity Diagram* Tambah Data Admin

20. *Activity Diagram* Ubah Data Admin

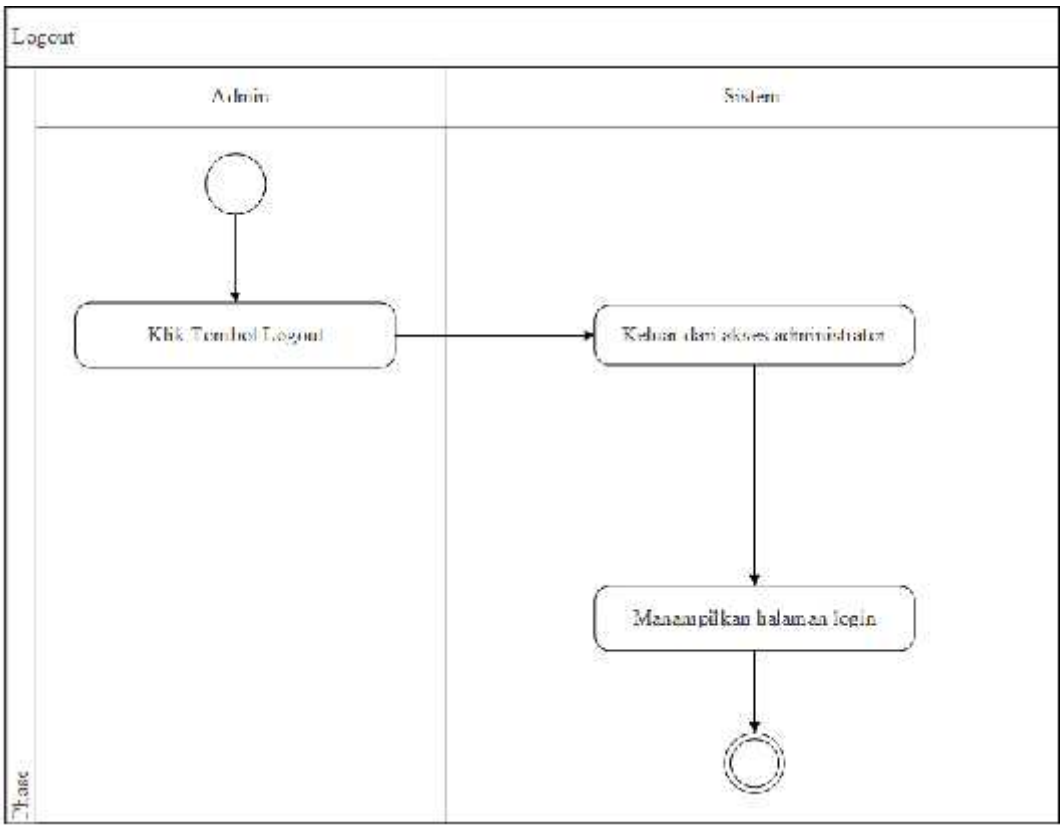
Activity diagram ubah data admin dapat dilihat pada gambar 3.25.



Gambar 3.25 *Activity Diagram* Ubah Data Admin

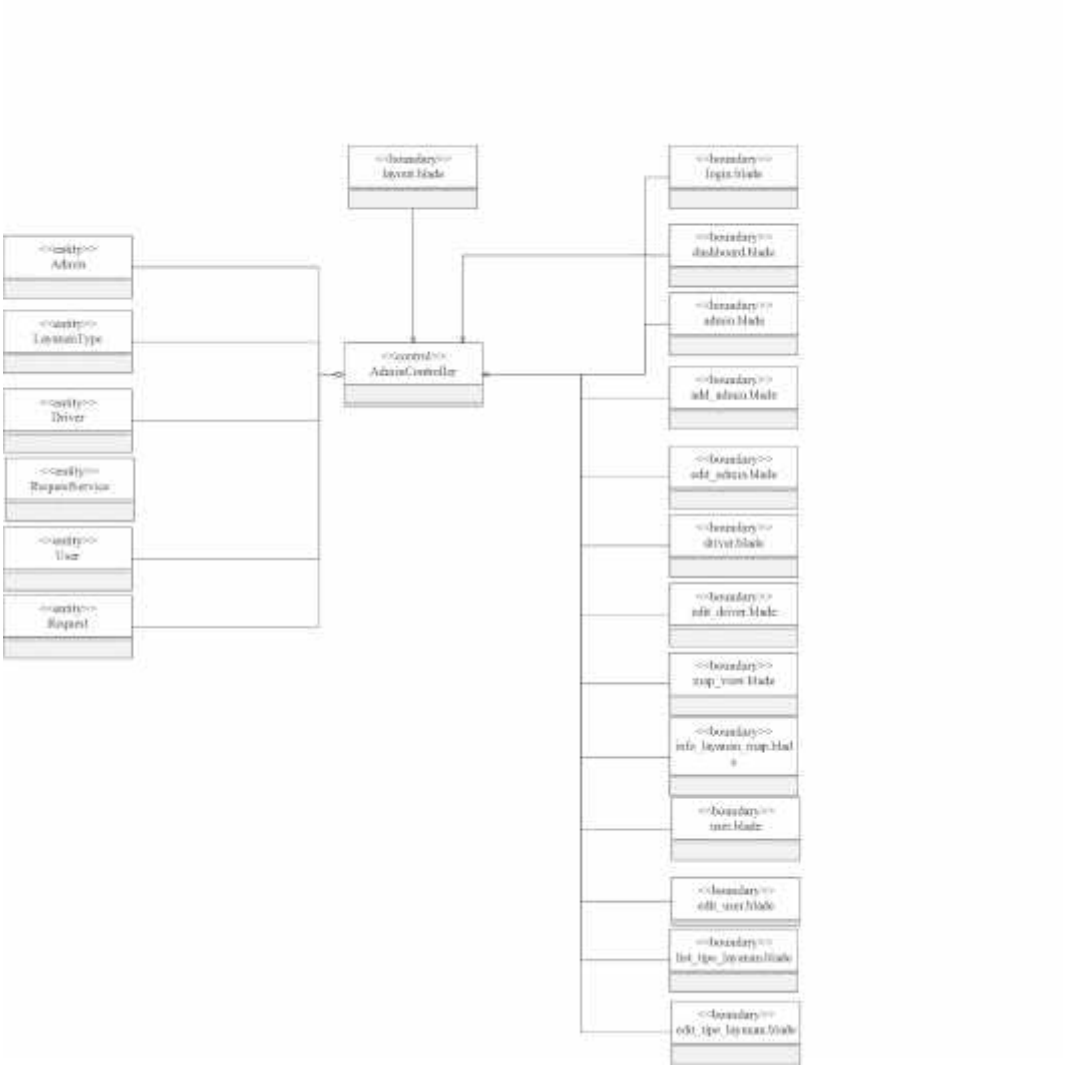
21. *Activity Diagram Logout*

Activity diagram logout dapat dilihat pada gambar 3.26.



Gambar 3.26 *Activity Diagram Logout*

3.1.7.1.4. *Class Diagram Back End*



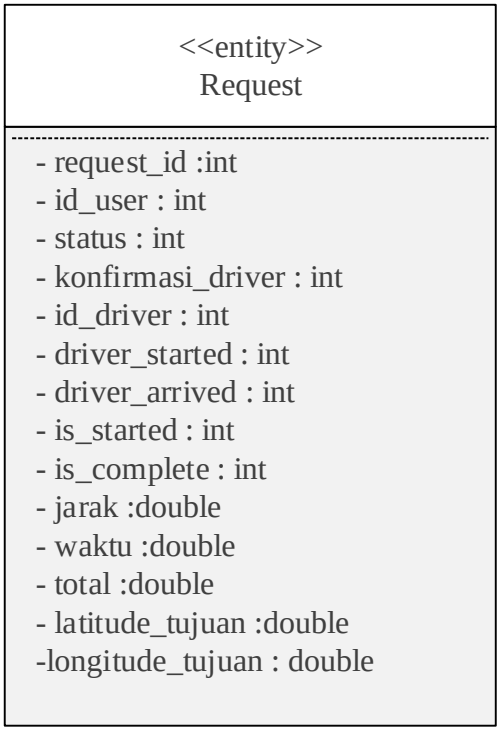
Gambar 3.27 *Class Diagram Backend*

Deskripsi Class Diagram

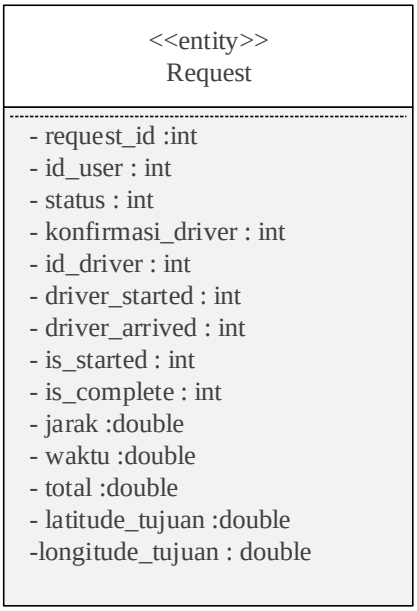
1) AdminController



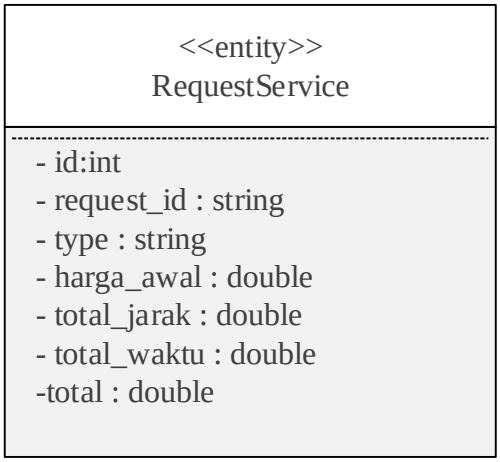
2) Request



3) User



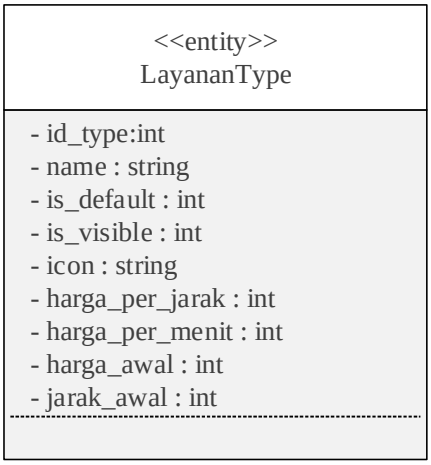
4) RequestService



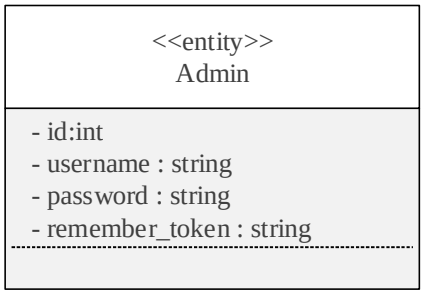
5) Driver



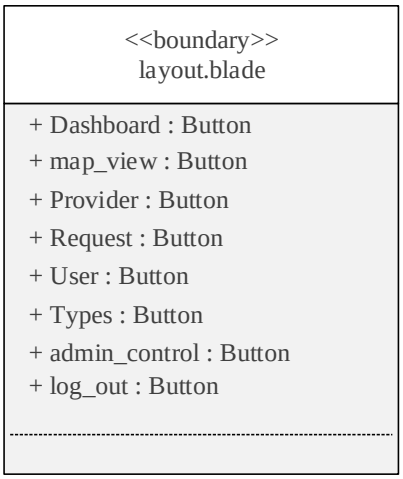
6) LayananType



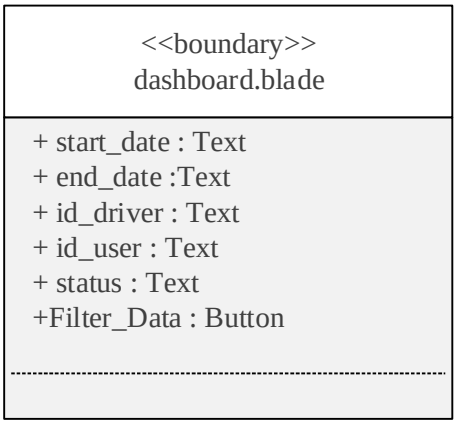
7) Admin



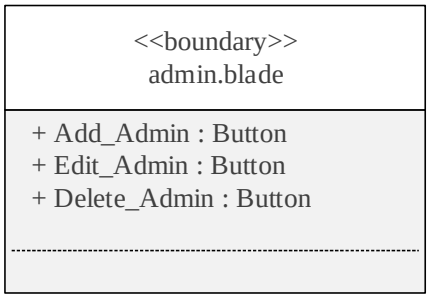
8) Layout.blade



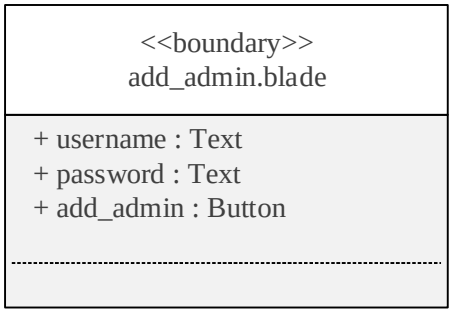
9) Dashboard.blade



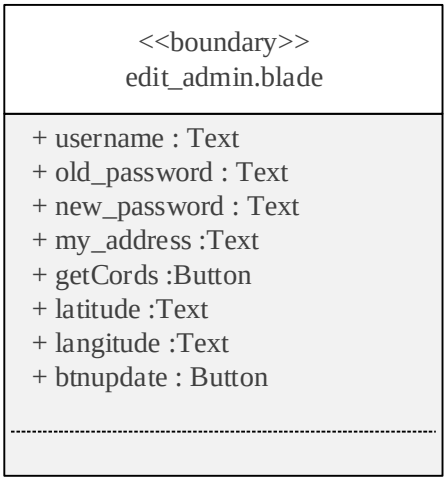
10) Admin.blade



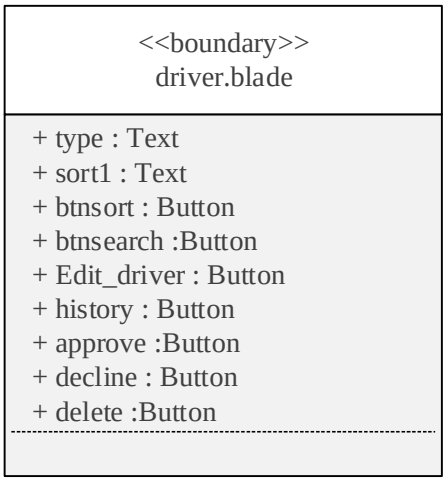
11) Add_admin.blade



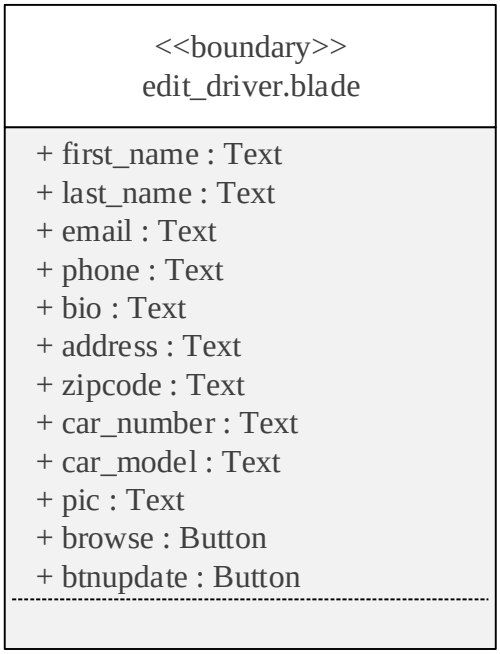
12) Edit_admin.blade



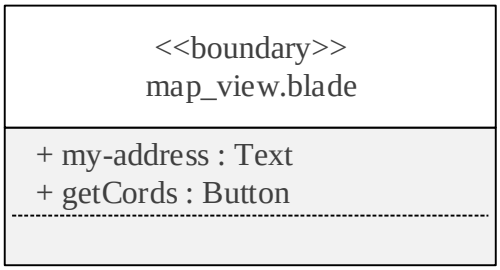
13) Driver.blade



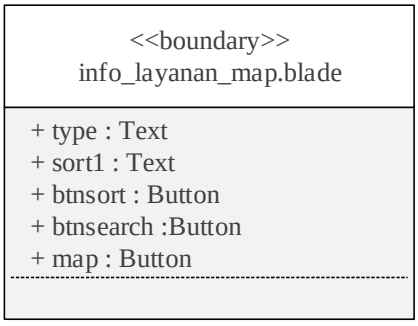
14) Edit_driver.blade



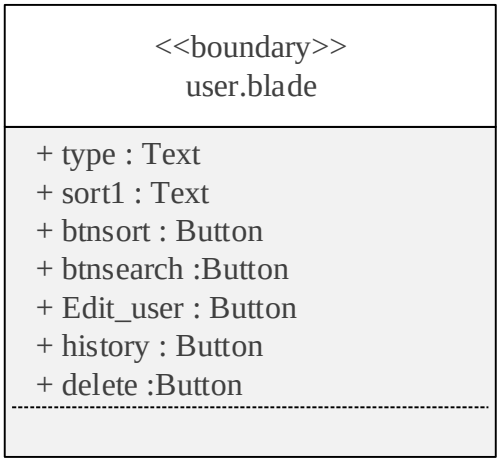
15) Map_view.blade



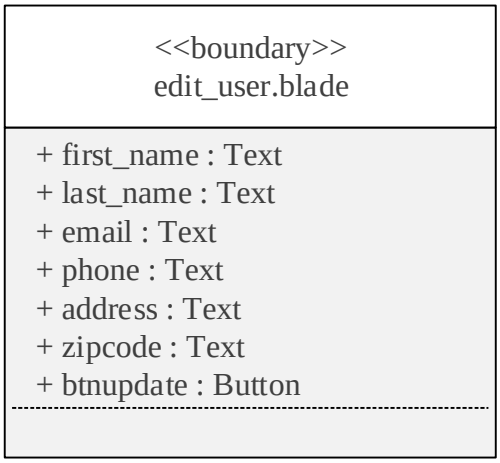
16) Info_layanan.blade



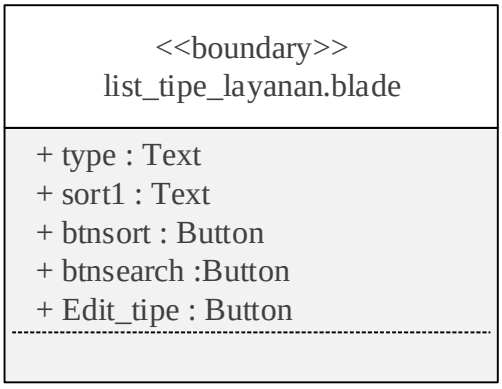
17) User.blade



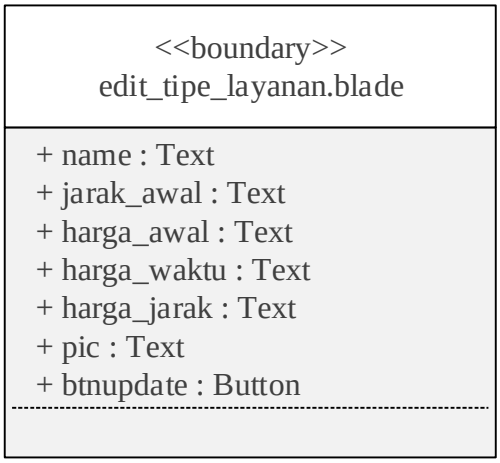
18) Edit_user.blade



19) List_tipe_layanan.blade



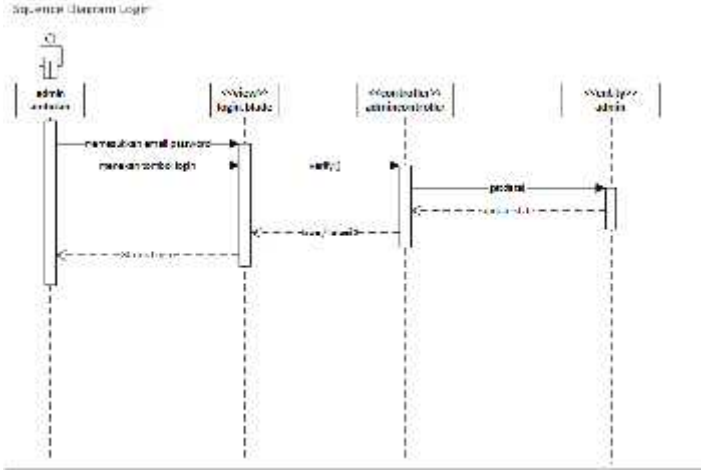
20) Edit_tipe_layanan.blade



3.1.7.1.5. *Squence Diagram Back End*

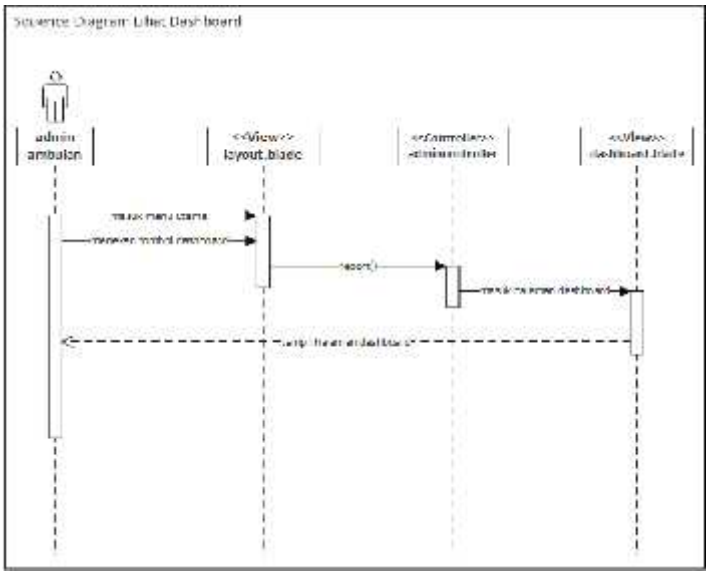
Sequence diagram digunakan untuk menggambarkan arus pekerjaan, pesan yang disampaikan dan bagaimana elemen-elemen di dalamnya bekerja sama dari waktu ke waktu untuk mencapai hasil.

1. Sequence Diagram Login



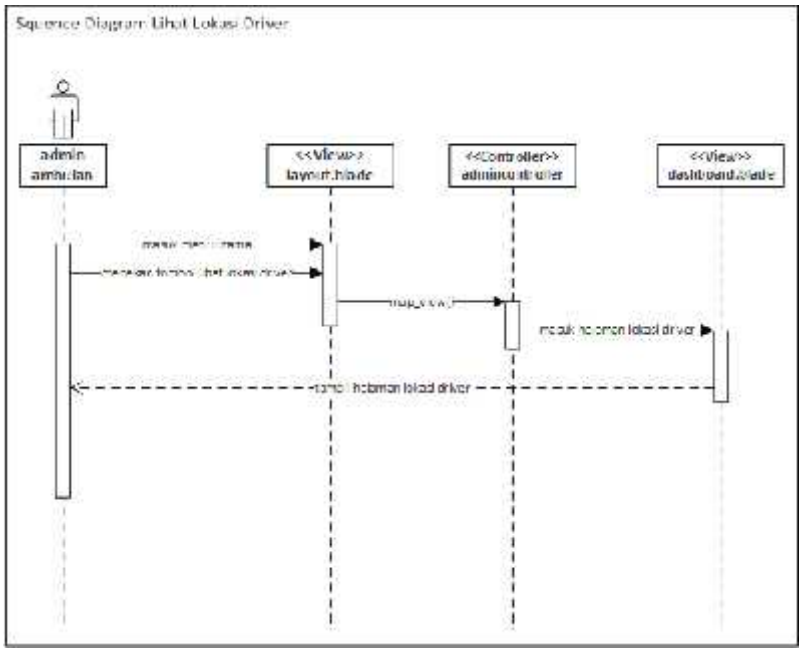
Gambar 3.28 *Squence Diagram Login*

2. Squence Diagram Lihat Dashboard



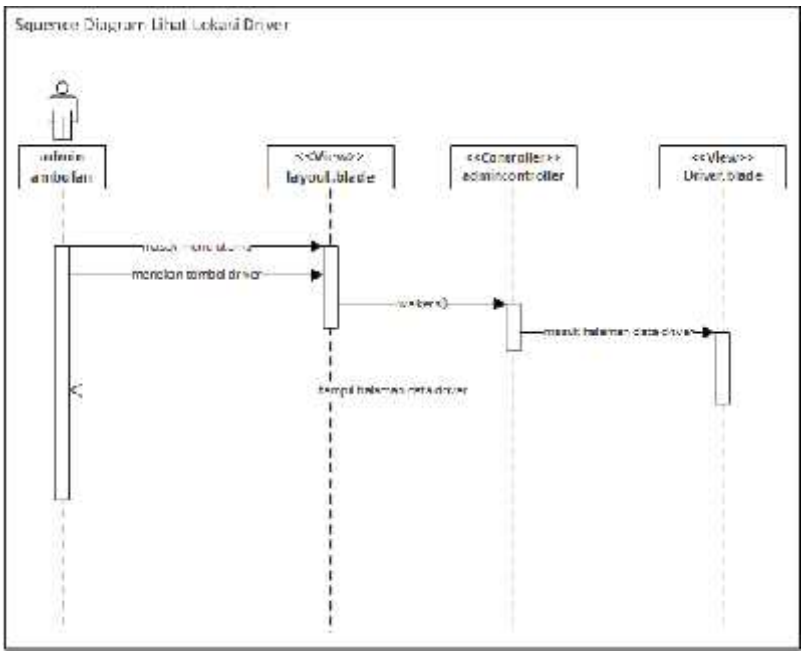
Gambar 3.29 *Squence Diagram* Lihat Dashboard

3. Squence Diagram Lihat Lokasi Driver



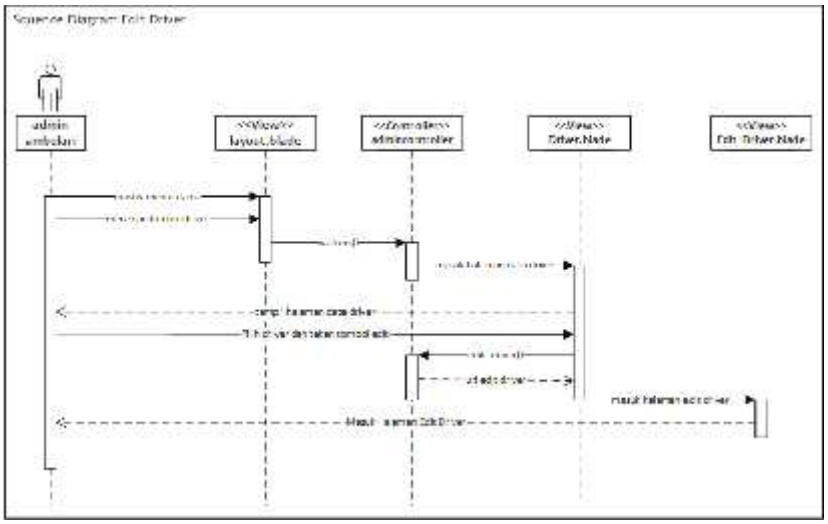
Gambar 3.30 *Squence Diagram* Lihat Lokasi Driver

4. Squence Diagram Mengelola Driver



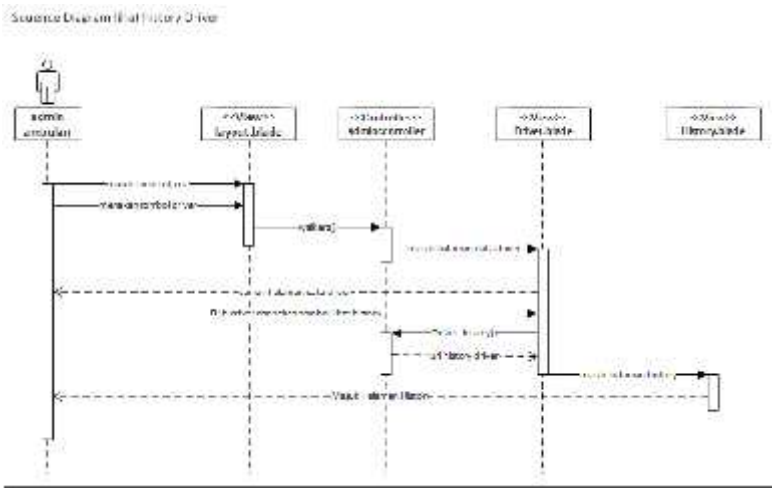
Gambar 3.31 *Squence Diagram* Mengelola Driver

5. Squence Diagram Edit Data Driver



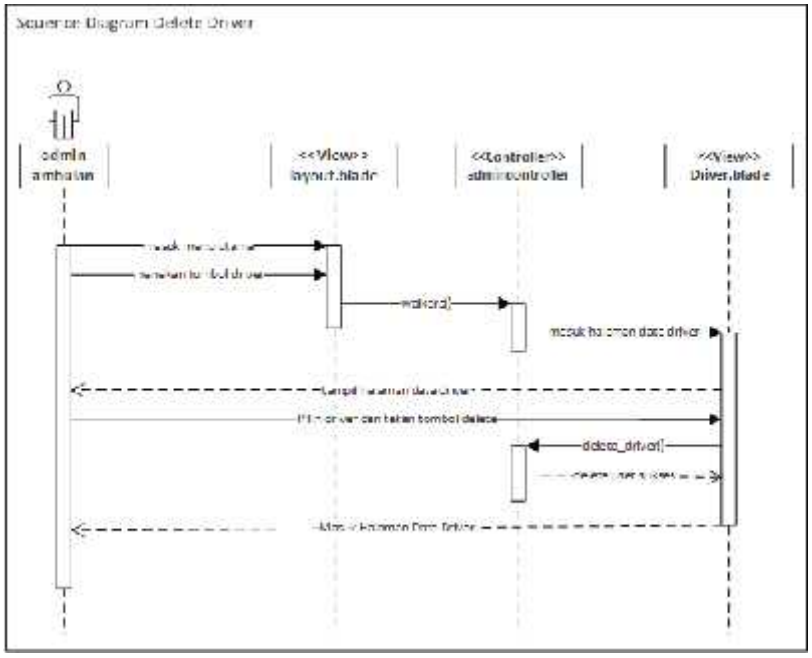
Gambar 3.32 *Squence Diagram* Edit Data Driver

6. Squence Diagram Lihat History Driver



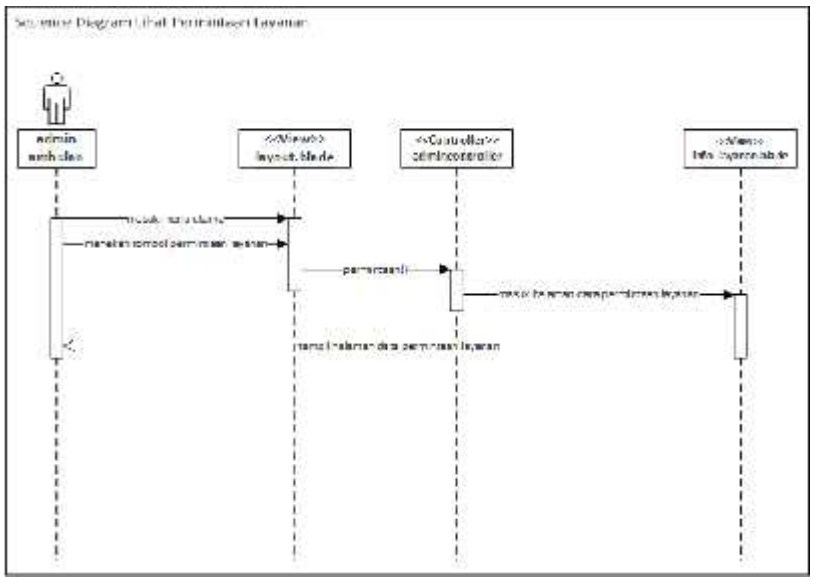
Gambar 3.33 *Squence Diagram* Lihat History Driver

7. Squence Diagram Delete Data Driver



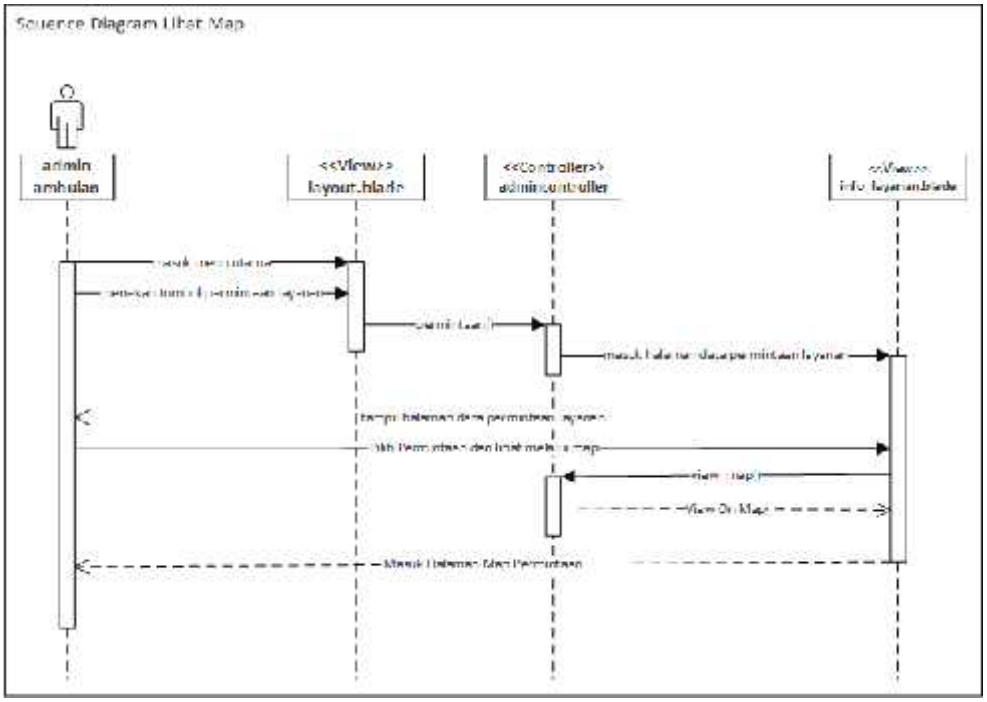
Gambar 3.34 *Squence Diagram Delete* Data Driver

8. Squence Diagram Lihat Permintaan Ambulan



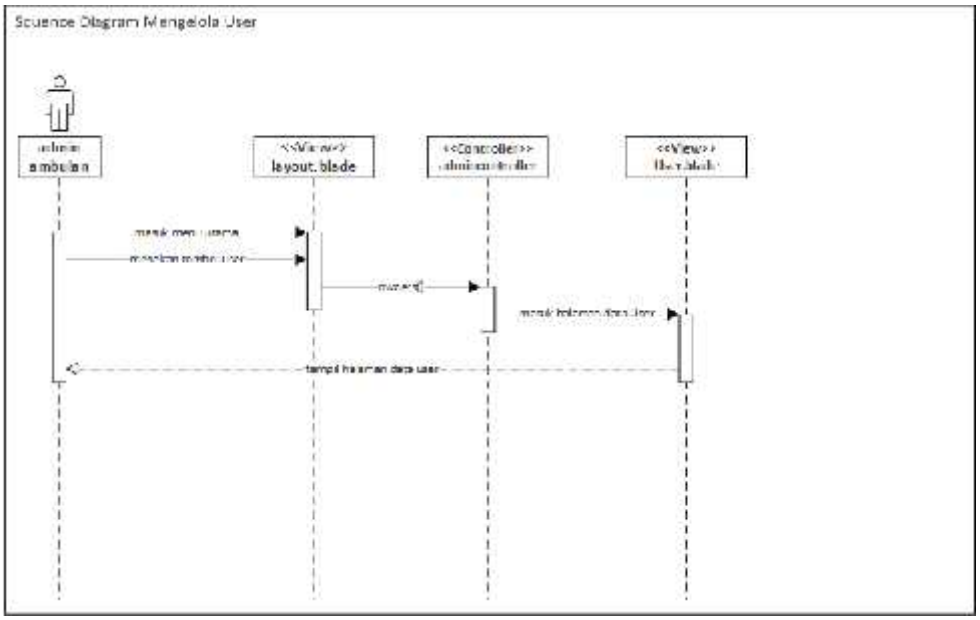
Gambar 3.35 *Squence Diagram* Lihat Permintaan Ambulan

9. Squence Diagram Lihat Map



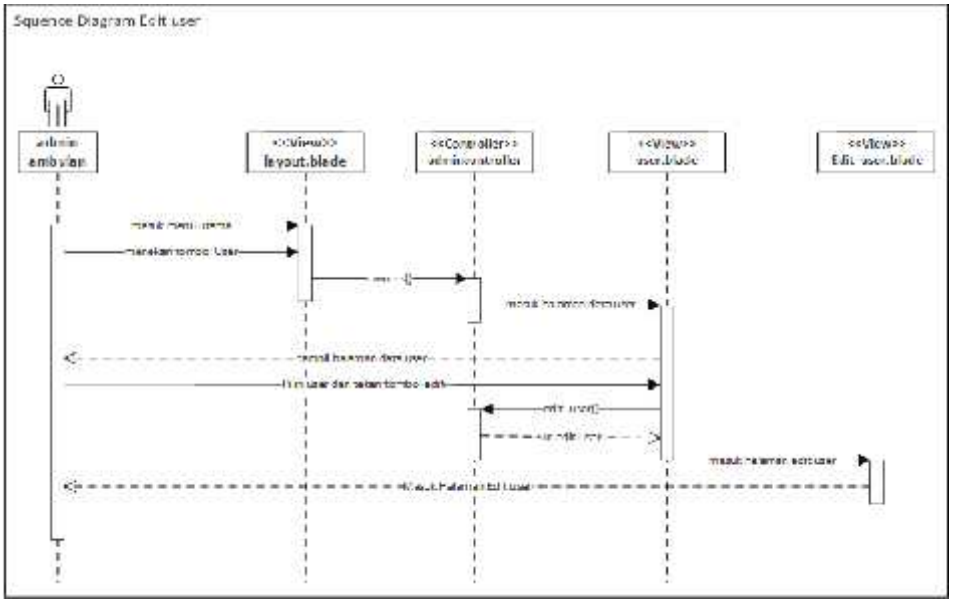
Gambar 3.36 *Squence Diagram* Lihat Map

10. Squence Diagram Mengelola Pasien



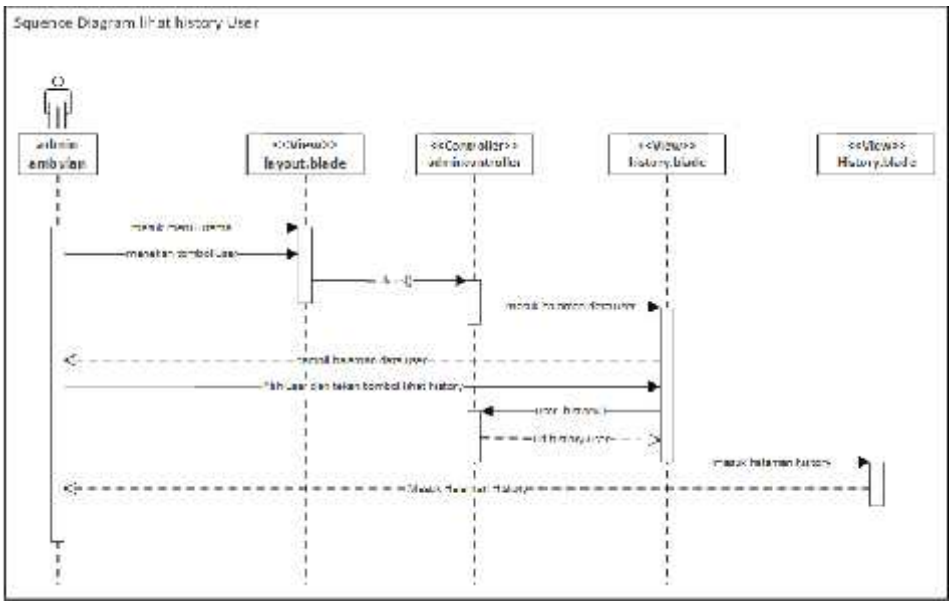
Gambar 3.37 *Squence Diagram* Mengelola Pasien

11. Squence Diagram Edit Data Pasien



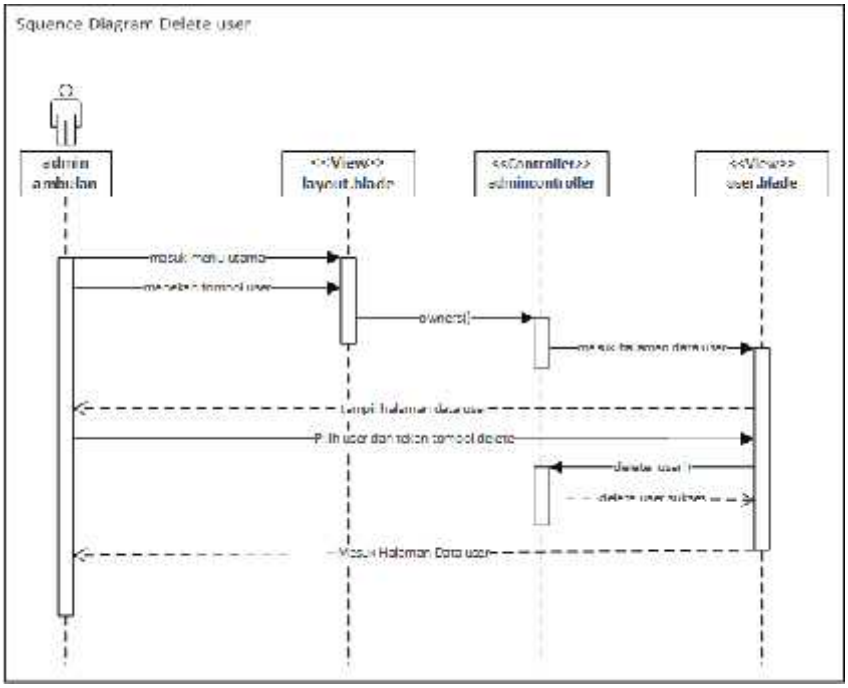
Gambar 3.38 *Squence Diagram* Edit Data Pasien

12. Squence Diagram Lihat History Pasien



Gambar 3.39 *Squence Diagram* Lihat History Pasien

13. Squence Diagram Hapus Data Pasien



Gambar 3.40 *Squence Diagram* Hapus Data Pasien

3.1.7.2.*Analisis Kebutuhan Fungsional Pada Front End*

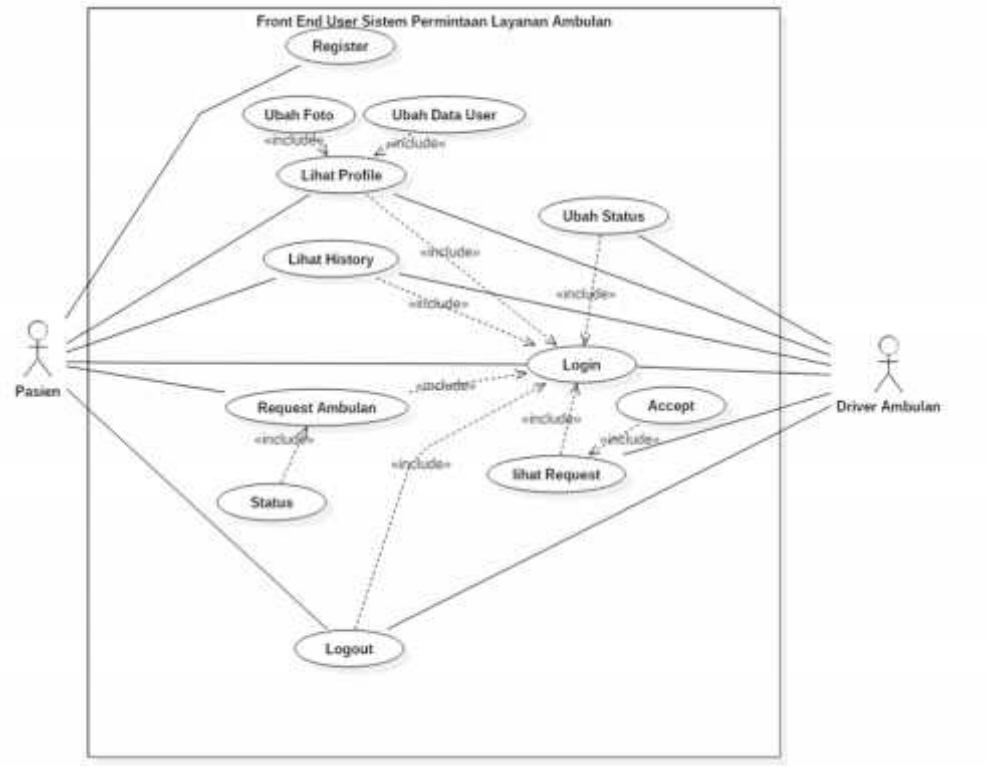
Spesifikasi kebutuhan fungsional pada pembangunan sistem permintaan layanan ambulan berbasis android pada back end dapat dilihat pada tabel 3.29.

Tabel 3.29 Spesifikasi Kebutuhan Fungsional Pada Frontend

Nomor	Spesifikasi Kebutuhan Fungsional
SKPL-F-023	Pasien /Driver login
SKPL-F-024	Pasien dapat melakukan registrasi
SKPL-F-025	Pasien / Driver dapat melihat data profile
SKPL-F-026	Pasien / Driver dapat Ubah data user
SKPL-F-027	Pasien / Driver dapat ubah foto
SKPL-F-028	Pasien / Driver dapat melihat history
SKPL-F-029	Pasien dapat melakukan pemesanan ambulan
SKPL-F-030	Driver dapat menerima *notification* dari pemesanan ambulan
SKPL-F-031	Driver dapat mengubah status ambulan
SKPL-F-032	Driver menerima pemesanan
SKPL-F-033	Pasien dapat melihat status pemesanan
SKPL-F-034	Pasien / Driver logout

3.1.7.2.1. Use Case Front End

Use case diagram merupakan konstruksi untuk mendeskripsikan hubungan yang terjadi antara aktor dengan aktivitas yang terdapat pada sistem. *Use case* menjelaskan proses apa saja yang ada dalam sistem dan bagaimana hubungannya dengan aktor. Dari hasil analisis maka *use case diagram* pada frontend ini dapat dilihat pada Gambar 3.41.



Gambar 3.41 *Use Case Diagram Frontend*

2.1.7.2.1.1 Deskripsi Aktor *Front End*

Berdasarkan use case diagram pada gambar 3.41, deskripsi aktor dapat dilihat pada tabel 3.30.

Tabel 3.30 Deskripsi Aktor

No	Aktor	Deskripsi
1	Pasien	*Pasien* Pengguna *mobile* ini merupakan pengguna aplikasi sistem permintaan layanan ambulan
2	*Driver* Ambulan	*Driver* ambulan Pengguna *mobile* ini merupakan pengguna aplikasi yang menerima permintaan layanan ambulan yang diajukan pasien.

2.1.7.2.1.2 Deskripsi *Use Case Diagram Front End*

Deskripsi *use case diagram* berfungsi untuk menjelaskan proses yang terdapat pada setiap *use case*. Deskripsi *use case diagram front end* dijelaskan pada tabel Tabel 3.31.

Tabel 3.31 Deskripsi *Use Case Diagram Frontend*

No	Use Case	Deskripsi
1	*Login*	Fungsional ini digunakan oleh *Pasien/ Driver* ambulan untuk login ke halaman utama.
2	*Register*	Fungsional ini digunakan oleh *Pasien* untuk mendaftar sebagai user baru.
3	Lihat *Profile*	Fungsional ini digunakan oleh *Pasien /Driver* untuk mengecek data *profile Pasien /Driver*.
4	Ubah foto	Fungsional ini digunakan oleh *Pasien /Driver* untuk mengubah foto *profile*.
5	Ubah data *user*	Fungsional ini digunakan oleh *Pasien /Driver* untuk mengelola data *profile.*
6	Lihat *history*	Fungsional ini digunakan oleh *Pasien /Driver* untuk melihat *history* pemesanan ambulan.
7	Ubah status	Fungsional ini digunakan oleh *Driver* untuk mengubah status ambulan.
8	Lihat Request	Fungsional ini digunakan oleh *Driver* ambulan untuk melihat data permintaan ambulan yang masuk.
9	Accept Ambulan	Fungsional ini digunakan oleh *Driver* ambulan untuk menerima pesanan layanan permintaan ambulan yang masuk.
10	Decline	Fungsional ini digunakan oleh *Driver* ambulan untuk menolak pesanan layanan permintaan ambulan yang masuk.

11	Request Ambulan	Fungsional ini digunakan oleh *Pasien* untuk melakukan pemesanan ambulan.
12	Status	Fungsional ini digunakan oleh *Pasien* untuk melihat status pemesanan ambulan.
13	*Logout*	Fungsional ini digunakan oleh *Pasien / Driver* ambulan untuk keluar dari halaman utama.

3.1.7.2.2. *Use Case Scenario Front End*

Berdasarkan *use case diagram front end* pada gambar 3.41, maka *use case scenario front end* untuk aplikasi sistem permintaan layanan ambulan di rumah sakit Al-Islam Bandung adalah sebagai berikut :

3.1.7.2.2.1. Use Case Scenario Frontend

1. *Use Case Scenario Login*

Use case scenario login dapat dilihat pada tabel 3.32.

Tabel 3.32 *Use Case Scenario Login*

Nama Use Case	*Login*	
Tujuan	*Pasien/Driver* ambulan berhasil masuk ke halaman utama	
Kondisi Awal	Menampilkan form *login*	
Deskripsi	Fungsional ini digunakan oleh *Pasien/Driver* ambulan untuk login ke halaman utama.	
Kondisi Akhir (Jika Berhasil)	*Pasien/Driver* ambulan berhasil masuk ke halaman utama	
Kondisi Akhir (Jika Gagal)	*Pasien/Driver* ambulan gagal masuk ke halaman utama	
Aktor	*Pasien/Driver* Ambulan	
Trigger	*Pasien/Driver* ambulan menekan tombol login	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan form login
	2.	*Pasien/Driver* ambulan mengisi form

		login
	3.	*Pasien/Driver* ambulan menekan tombol login
	4.	Sistem melakukkan validasi form
	5.	*Pasien/Driver* berhasil masuk ke halaman utama
Skenario Alternatif	Langkah	Aksi
	4.1	Field Form tidak terisi penuh
	4.2	Sistem menampilkan pesan field kosong
	4.3	Sistem menampilkan pesan email atau password salah

2. *Use Case Scenario Register*

Use case scenario register dapat dilihat pada tabel 3.33.

Tabel 3.33 *Use Case Scenario Register*

Nama Use Case	*Register*	
Tujuan	*Pasien/Driver* ambulan berhasil mendaftar sebagai *pasien* baru.	
Kondisi Awal	Menampilkan form *register*	
Deskripsi	Fungsional ini digunakan oleh *Pasien/Driver* untuk mendaftar sebagai pasien baru.	
Kondisi Akhir (Jika Berhasil)	*Pasien/Driver* ambulan berhasil mendaftar sebagai pasien baru.	
Kondisi Akhir (Jika Gagal)	*Pasien/Driver* ambulan gagal mendaftar sebagai pasien baru.	
Aktor	*Pasien/Driver* ambulan	
Trigger	*Pasien/Driver* ambulan menekan tombol register	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan form register

	2.	*Pasien/Driver* ambulan mengisi form register
	3.	*Pasien/Driver* ambulan menekan tombol register
	4.	Sistem melakukkan validasi form
	5.	*Pasien/Driver* berhasil menjadi pasien baru
Skenario Alternatif	Langkah	Aksi
	4.1	Field Form tidak terisi penuh
	4.2	Sistem menampilkan pesan field kosong

3. *Use Case Scenario* Lihat *Profile*

Use case scenario login dapat dilihat pada tabel 3.34.

Tabel 3.34 *Use Case Scenario* Lihat *Profile*

Nama Use Case	Lihat *Profile*	
Tujuan	*Pasien/Driver* ambulan berhasil melihat data *profile pasien*.	
Kondisi Awal	*Pasien/Driver* ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh *Pasien/Driver* untuk mengecek data *profile Pasien/Driver*.	
Kondisi Akhir (Jika Berhasil)	*Pasien/Driver* ambulan berhasil melihat data *profile pasien*.	
Kondisi Akhir (Jika Gagal)	*Pasien/Driver* ambulan gagal melihat data *profile pasien*.	
Aktor	*Pasien/Driver* Ambulan	
Trigger	*Pasien/Driver* ambulan menekan tombol *profile*	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan halaman utama
	2.	*Pasien/Driver* ambulan menekan

		tombol *profile*
Skenario Alternatif	Langkah	Aksi

4. *Use Case Scenario* Ubah Foto

Use case scenario ubah foto dapat dilihat pada tabel 3.35.

Tabel 3.35 *Use Case Scenario* Ubah Foto

Nama Use Case	Ubah Foto	
Tujuan	*Pasien/Driver* ambulan berhasil mengubah foto *profile*	
Kondisi Awal	*Pasien/Driver* ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh *Pasien/Driver* untuk mengubah foto *profile*.	
Kondisi Akhir (Jika Berhasil)	*Pasien/Driver* ambulan berhasil mengubah foto *profile*	
Kondisi Akhir (Jika Gagal)	*Pasien/Driver* ambulan gagal mengubah foto *profile*	
Aktor	*Pasien/Driver* ambulan	
Trigger	*Pasien/Driver* ambulan menekan tombol ubah foto	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan halaman utama
	2.	*Pasien/Driver* ambulan menekan tombol *profile*
	3.	*Pasien/Driver* ambulan menekan tombol ubah foto
	4.	Sistem melakukkan validasi form
	5.	*Pasien/Driver* ambulan berhasil merubah foto *profile*
Skenario Alternatif	Langkah	Aksi
	4.1	Gagal upload foto

5. *Use Case Scenario* Ubah Data *Pasien*

Use case scenario ubah data *pasien* dapat dilihat pada tabel 3.36.

Tabel 3.36 *Use Case Scenario* Ubah Data *Pasien*

Nama Use Case	Ubah Data *Pasien*	
Tujuan	*Pasien/Driver* ambulan berhasil mengubah data pribadi *pasien.*	
Kondisi Awal	*Pasien/Driver* ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh *Pasien/Driver* untuk mengelola data *profile.*	
Kondisi Akhir (Jika Berhasil)	*Pasien/Driver* ambulan berhasil mengubah data pribadi *pasien.*	
Kondisi Akhir (Jika Gagal)	*Pasien/Driver* ambulan gagal mengubah data pribadi *pasien.*	
Aktor	*Pasien/Driver* ambulan	
Trigger	*Pasien/Driver* ambulan menekan tombol ubah *profile*	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan halaman utama
	2.	*Pasien/Driver* ambulan menekan tombol profile
	3.	*Pasien/Driver* ambulan menekan tombol edit *profile*
	4.	Sistem melakukkan validasi form
	5.	*Pasien/Driver* ambulan berhasil mengubah data *profile*
Skenario Alternatif	Langkah	Aksi
	4.1	Field Form tidak terisi penuh
	4.2	Sistem menampilkan pesan field kosong

6. *Use Case Scenario* Lihat *History*

Use case scenario lihat *history* dapat dilihat pada tabel 3.37.

Tabel 3.37 *Use Case Scenario* Lihat *History*

Nama Use Case	Lihat *History*	
Tujuan	*Pasien/Driver* ambulan berhasil melihat data *history* layanan permintaan ambulan	
Kondisi Awal	*Pasien/Driver* ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh *Pasien/Driver* untuk melihat *history* pemesanan ambulan.	
Kondisi Akhir (Jika Berhasil)	*Pasien/Driver* ambulan berhasil melihat data *history* layanan permintaan ambulan	
Kondisi Akhir (Jika Gagal)	*Pasien/Driver* ambulan gagal melihat data *history* layanan permintaan ambulan	
Aktor	*Pasien/Driver* ambulan	
Trigger	*Pasien/Driver* ambulan menekan tobol *history*	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan halaman utama
	2.	*Pasien/Driver* ambulan menekan tombol *history*
	3.	*Pasien/Driver* ambulan berhasil melihat data *history*
Skenario Alternatif	Langkah	Aksi

7. *Use Case Scenario* Ubah Status

Use case scenario ubah status dapat dilihat pada tabel 3.38.

Tabel 3.38 *Use Case Scenario* Ubah Status

Nama Use Case	Ubah Status	
Tujuan	*Driver* ambulan berhasil mengubah status ambulan	
Kondisi Awal	*Driver* ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh *Driver* untuk mengubah status ambulan.	
Kondisi Akhir (Jika Berhasil)	*Driver* ambulan berhasil mengubah status ambulan	
Kondisi Akhir (Jika Gagal)	*Driver* ambulan gagal mengubah status ambulan	
Aktor	*Driver* ambulan	
Trigger	*Driver* ambulan menekan ubah status	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan halaman utama
	2.	Admin pengelola ambulan berhasil mengubah status ambulan
Skenario Alternatif	Langkah	Aksi

8. *Use Case Scenario* Lihat *Request*

Use case scenario lihat *request* dapat dilihat pada tabel 3.39.

Tabel 3.39 *Use Case Scenario* Lihat *Request*

Nama Use Case	Lihat *Request*
Tujuan	*Driver* ambulan berhasil melihat notifikasi pemesanan masuk
Kondisi Awal	*Driver* ambulan dalam keadaan login
Deskripsi	Fungsional ini digunakan oleh *Driver* ambulan untuk melihat data permintaan ambulan yang masuk.
Kondisi Akhir (Jika Berhasil)	*Driver* ambulan berhasil melihat notifikasi

	pemesanan masuk	
Kondisi Akhir (Jika Gagal)	*Driver* ambulan gagal melihat notifikasi pemesanan masuk	
Aktor	*Driver* ambulan	
Trigger	-	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan halaman utama
	2.	Sistem menerima pesanan masuk
	3.	*Driver* ambulan berhasil mendapatkan notikasi pesanan
Skenario Alternatif	Langkah	Aksi

9. *Use Case Scenario Accept* Ambulan

Use case scenario accept ambulan dapat dilihat pada tabel 3.40.

Tabel 3.40 *Use Case Scenario Accept* Ambulan

Nama Use Case	*Accept* Ambulan	
Tujuan	*Driver* ambulan berhasil menerima pemesanan ambulan yang masuk	
Kondisi Awal	*Driver* ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh *Driver* ambulan untuk menerima pesanan layanan permintaan ambulan yang masuk.	
Kondisi Akhir (Jika Berhasil)	*Driver* ambulan berhasil menerima pemesanan ambulan yang masuk	
Kondisi Akhir (Jika Gagal)	*Driver* ambulan gagal menerima pemesanan ambulan yang masuk	
Aktor	*Driver* ambulan	
Trigger	*Driver* ambulan menekan tombol *Accept*	
Skenario Utama	Langkah	Aksi

	1.	Sistem menampilkan halaman utama
	2.	Sistem menerima pesanan ambulan
	3.	*Driver* ambulan menekan tombol *Accept*
	4.	Sistem menerima pesan dari *Driver*
	5.	*Driver* ambulan berhasil menerima pemesanan ambulan yang masuk
Skenario Alternatif	Langkah	Aksi

10. *Use Case Scenario Decline* Ambulan

Use case scenario decline ambulan dapat dilihat pada tabel 3.41.

Tabel 3.41 *Use Case Scenario Decline* Ambulan

Nama Use Case	*Decline* Ambulan	
Tujuan	*Driver* ambulan berhasil menolak permintaan ambulan yang masuk	
Kondisi Awal	*Driver* ambulan dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh *Driver* ambulan untuk menolak pesanan layanan permintaan ambulan yang masuk.	
Kondisi Akhir (Jika Berhasil)	*Driver* ambulan berhasil menolak permintaan ambulan yang masuk	
Kondisi Akhir (Jika Gagal)	*Driver* ambulan gagal menolak permintaan ambulan yang masuk	
Aktor	*Driver* ambulan	
Trigger	*Driver* ambulan menekan tombol *Decline*	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan halaman utama
	2.	Sistem menerima pesanan ambulan
	3.	*Driver* ambulan menekan tombol

		Decline
	4.	Sistem menerima pesan dari *Driver*
	5.	*Driver* ambulan berhasil menolak pemesanan ambulan yang masuk

11. *Use Case Scenario Request* Ambulan

Use case scenario request ambulan dapat dilihat pada tabel 3.42.

Tabel 3.42 *Use Case Scenario Request* Ambulan

Nama Use Case	*Request* Ambulan	
Tujuan	*Pasien* berhasil memesan ambulan	
Kondisi Awal	*Pasien* dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh *Pasien* untuk memesan ambulan.	
Kondisi Akhir (Jika Berhasil)	*Pasien* berhasil memesan ambulan	
Kondisi Akhir (Jika Gagal)	*Pasien* gagal memesan ambulan	
Aktor	*Pasien/* Pasien	
Trigger	*Pasien/* Pasien menekan tombol pesan	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan halaman utama
	2.	*Pasien* menekan tombol pesan
	3.	*Pasien* berhasil memesan ambulan
Skenario Alternatif	Langkah	Aksi

12. *Use Case Scenario* Lihat Status Pemesanan

Use case scenario lihat status pemesanan dapat dilihat pada tabel 3.43.

Tabel 3.43 *Use Case Scenario* Lihat Status Pemesanan

Nama Use Case	Lihat Status Pemesanan	
Tujuan	*Pasien* berhasil melihat status pemesanan ambulan	
Kondisi Awal	*Pasien* dalam keadaan login	
Deskripsi	Fungsional ini digunakan oleh *Pasien* untuk melihat data status permintaan layanan ambulan..	
Kondisi Akhir (Jika Berhasil)	*Pasien* berhasil melihat status pemesanan ambulan	
Kondisi Akhir (Jika Gagal)	*Pasien* gagal melihat status pemesanan ambulan	
Aktor	*Pasien/* Pasien	
Trigger	*Pasien/* Pasien menekan tombol *mybooking*	
Skenario Utama	Langkah	Aksi
	1.	Sistem menampilkan halaman utama
	2.	*Pasien/* Pasien menekan tombol *mybooking*
	3.	*Pasien/* Pasien berhasil melihat data status pemesanan
Skenario Alternatif	Langkah	Aksi

13. *Use Case Scenario Logout*

Use case scenario logout dapat dilihat pada tabel 3.44.

Tabel 3.44 *Use Case Scenario Logout*

Nama Use Case	*Logout*
Tujuan	*Pasien/Driver* ambulan berhasil keluar dari sistem
Kondisi Awal	Menampilkan halaman utama
Deskripsi	Fungsional ini digunakan oleh *Pasien/Driver* ambulan untuk keluar dari sistem

Kondisi Akhir (Jika Berhasil)	*Pasien/Driver* ambulan berhasil masuk keluar dari sistem	
Kondisi Akhir (Jika Gagal)	*Pasien/Driver* ambulan gagal keluar dari sistem	
Aktor	*Pasien/Driver* ambulan	
Trigger	*Pasien/Driver* ambulan menekan tombol logout	
Skenario Utama	Langkah	Aksi
	1.	*Pasien/Driver* ambulan menekan tombol logout
	2.	*Pasien/Driver* ambulan berhasil keluar dari sistem
Skenario Alternatif	Langkah	Aksi
	-	-

3.1.7.2.3. *Class Diagram Front End*

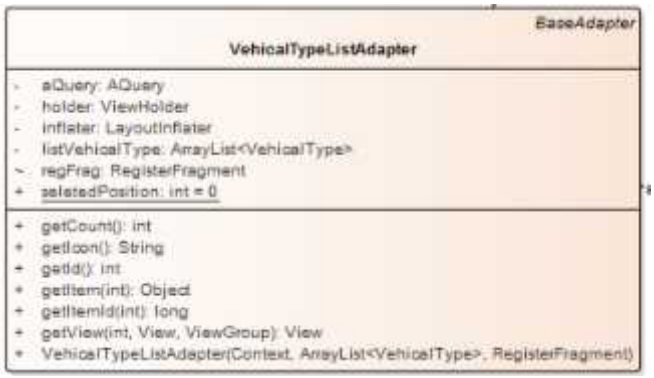
Class diagram menggambarkan struktur dan deskripsi *class, package, dan* objek serta memiliki hubungan antara satu objek dengan objek yang lainnya Class diagram dari aplikasi sistem permintaan layanan ambulan ini dibuat berdasarkan *class – class* yang terdapat di dalam program. Untuk lebih jelasnya dapat dilihat pada gambar 3.28

Deskripsi Class Frontend

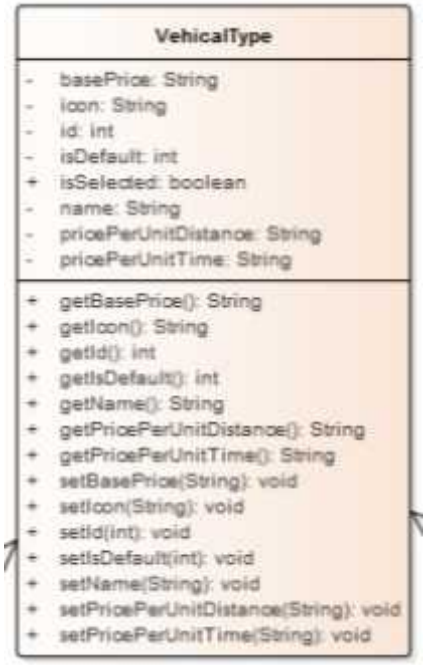
1) **HistoryAdafter**



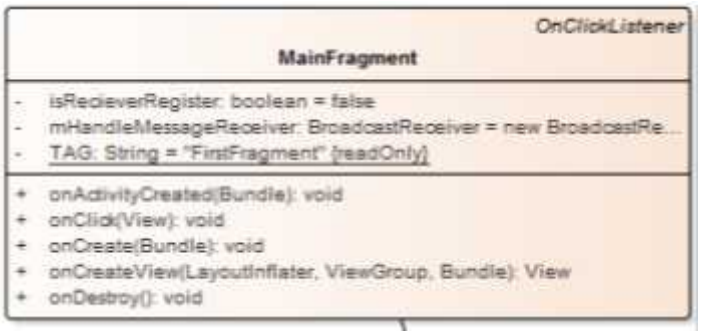
2) **VehicalTypeListAdapter**



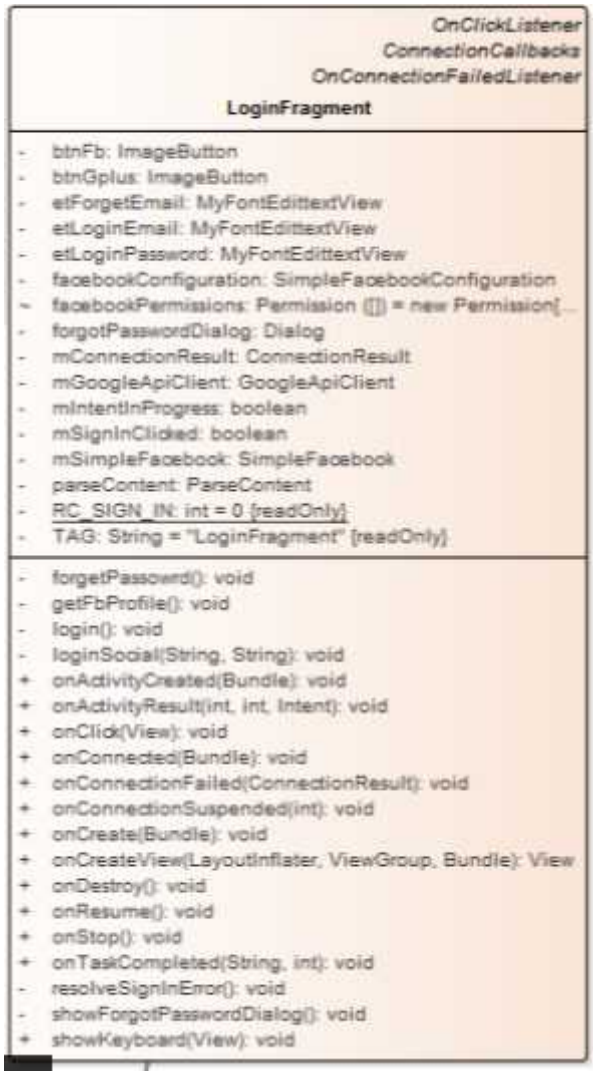
3) VehicalType



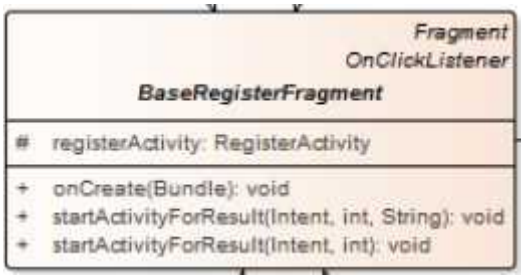
4) MainFragment



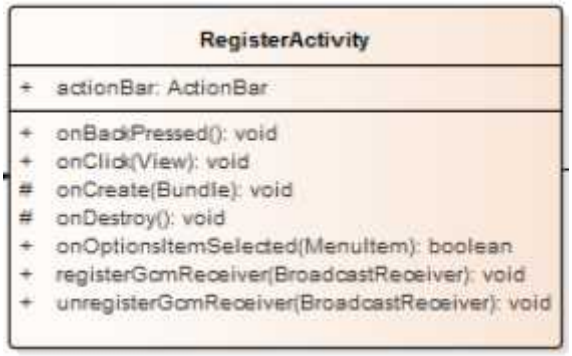
5) LoginFragment



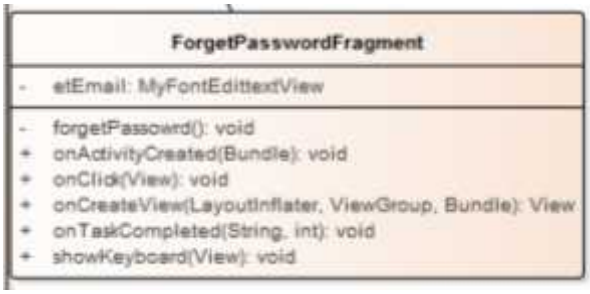
6) BaseRegisterFragment



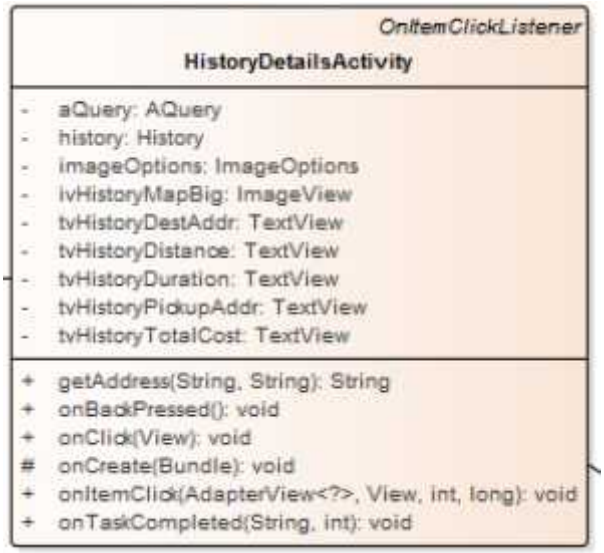
7) RegisterActivity



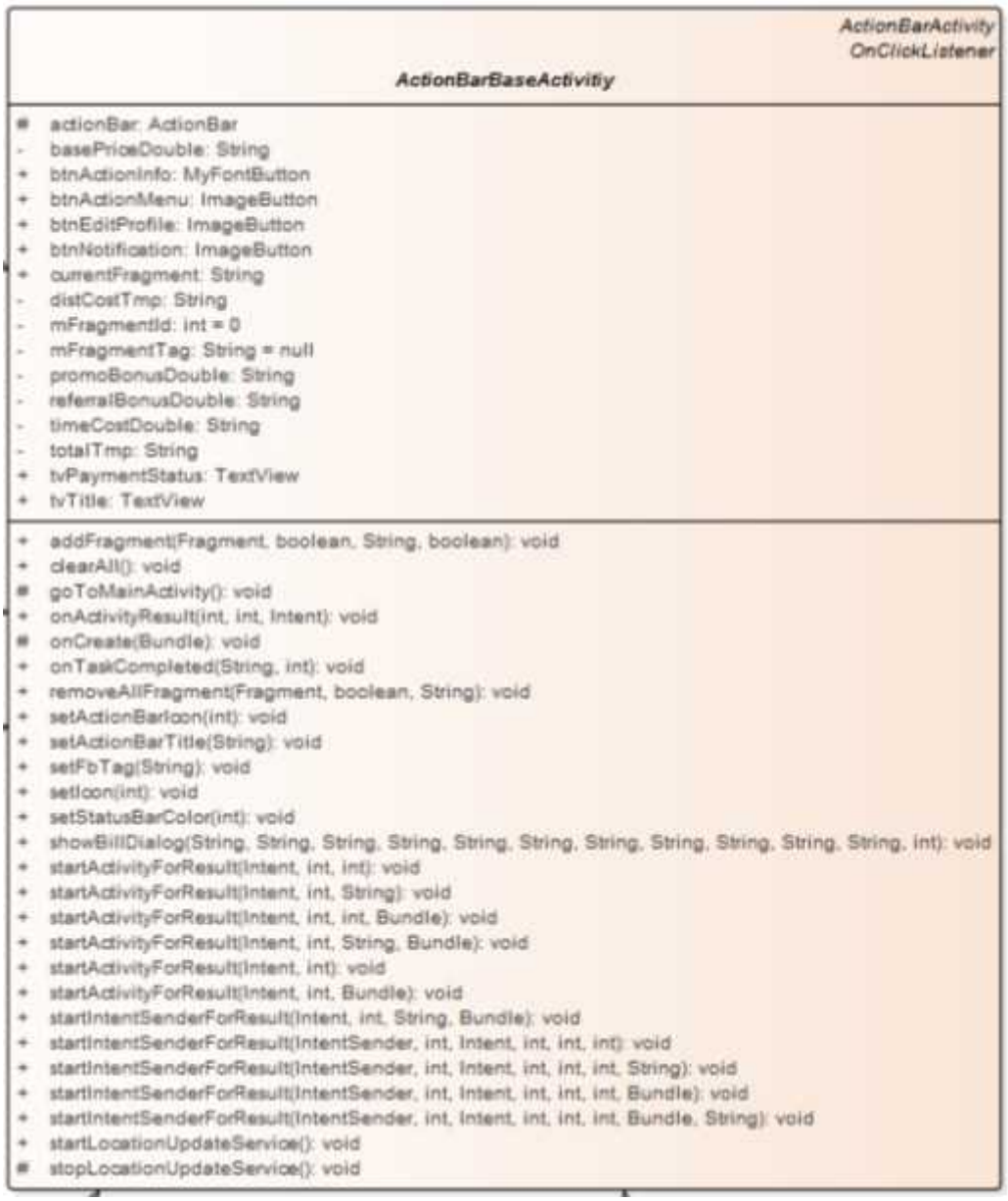
8) ForgotPasswordFragment



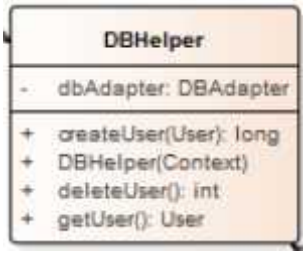
9) HistoryDetailActivity



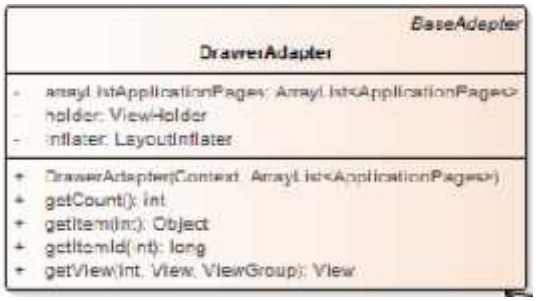
10) ActionBarBaseActivity



11) DBhelper



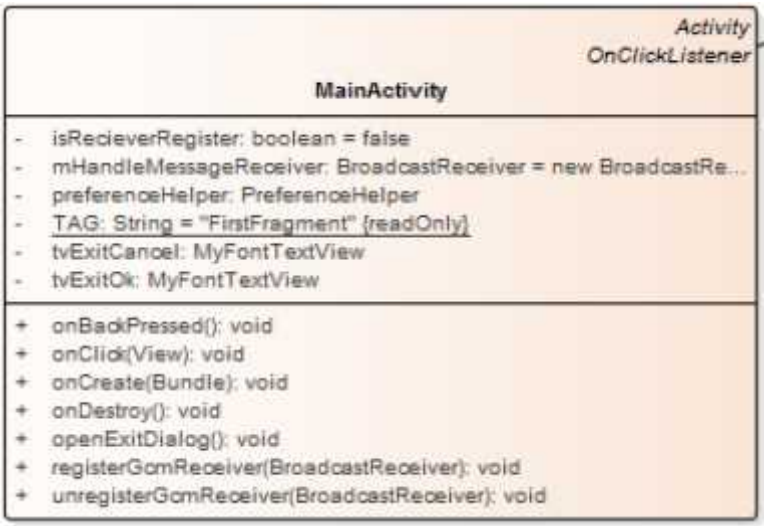
12) DrawerAdapter



13) HistoryAdapter



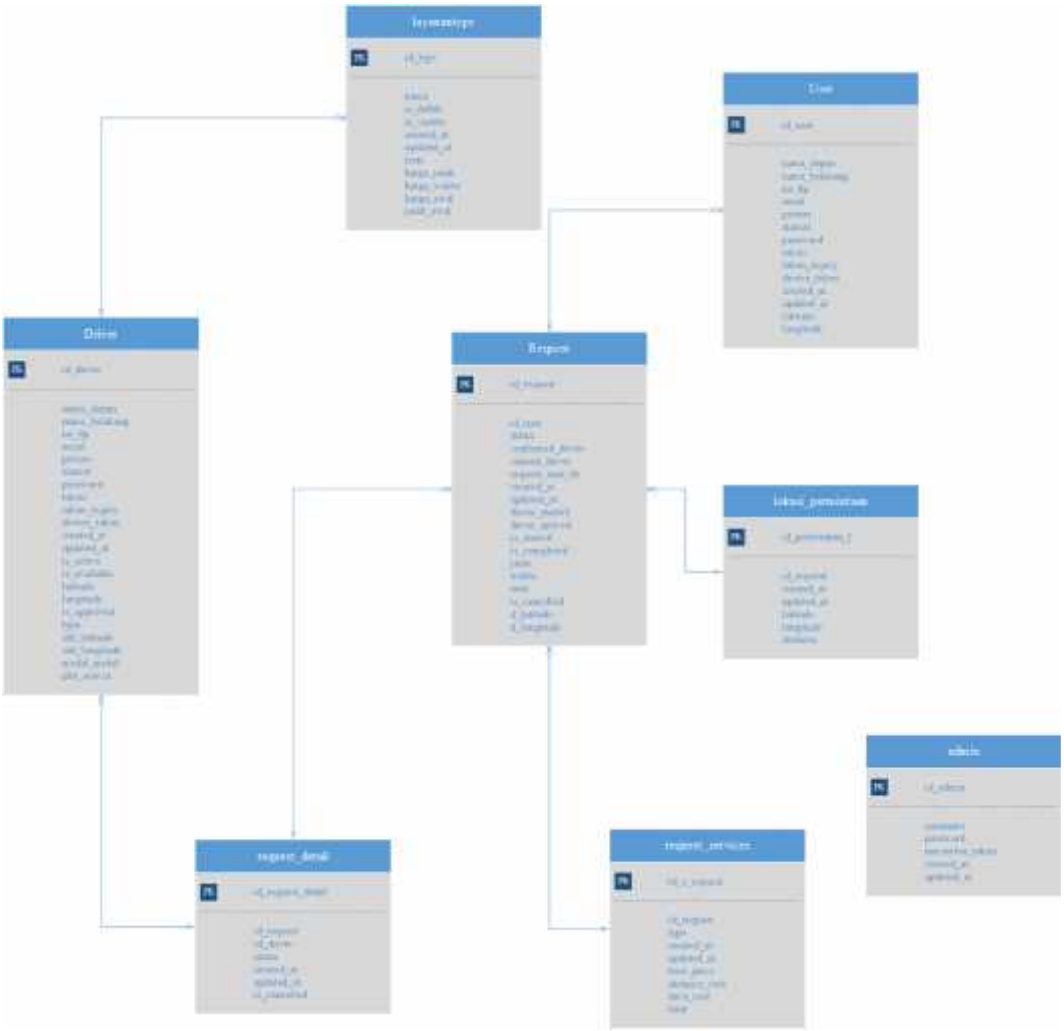
14) MainActivity



3.2. Perancangan Sistem

Adapun tahapan yang dilakukan dalam perancangan sistem ini membahas mengenai tujuan perancangan data, perancangan struktur menu perancangan antarmuka, dan perancangan jaringan simantik.

3.2.1. Perancangan data



Gambar 3.42 *Skema Relasi*

Pada penelitian ini database yang digunakan adalah MySQL maka akan ditampilkan struktur dari masing-masing tabel yang akan digunakan :

a. Admin

Tabel 3.45 Struktur *Tabel Admin*

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
Id_admin	int	10	Primary key	Not null, auto increment
username	varchar	255		Not null
password	varchar	255		Not null
Remember_token	varchar	255		Not null
Created_at	Timestamp			Not null
Updated_at	timestamp			Not null

b. Driver

Tabel 3.46 Struktur *Tabel Driver*

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
Id_driver	int	10	Primary key	Not null, auto increment
Nama_depan	varchar	255		Not null
Nama_belakang	varchar	255		Not null
No_tlp	varchar	255		Not null
email	varchar	255		Not null
password	varchar	255		
picture	varchar	255		Not null
alamat	varchar	255		
Device_token	varchar	255		Not null
token	varchar	255		
Token_expiry	varchar	255		
Created_at	Timestamp			Not null
Updated_at	timestamp			Not null
Is_active	int	11		
Is_available	int	11		
latitude	double			
longitude	double			
Is_approved	int	11		
type	int	11	FK	Not null
Old_latitude	Double			
Old_langitude	Double			
Model_mobil	varchar	255		
Plat_nomor	varchar	255		

c. User

Tabel 3.47 Struktur *Tabel User*

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
Id_user	int	10	Primary key	Not null, auto increment
Nama_depan	varchar	255		Not null
Nama_belakang	varchar	255		Not null
No_tlp	varchar	255		Not null
email	varchar	255		Not null
password	varchar	255		
picture	varchar	255		Not null
alamat	varchar	255		
Device_token	varchar	255		Not null
token	varchar	255		
Created_at	Timestamp			Not null
Updated_at	timestamp			Not null
latitude	double			
longitude	double			

d. layanantype

Tabel 3.48 Struktur *Tabel User*

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
Id_type	int	10	Primary key	Not null, auto increment
Nama	varchar	255		Not null
Is_default	Int	11		Not null
Is_visible	int	11		Not null
Created_at	Timestamp			
Updated_at	timestamp			
icon	varchar	255		
Harga_jarak	double			
Harga_waktu	double			
Harga_awal	double			
Jarak_awal	double			

e. request

Tabel 3.49 Struktur *Tabel Request*

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
Id_request	int	10	Primary key	Not null, auto increment
Id_user	varchar	255	FK	Not null
status	Int	11		Not null
Confirmed_driver	int	11		Not null
Current_driver	Int	11		

Request_start_time	datetime			
Created_at	timestamp			
Update_at	timestamp			
Driver_started	int	11		
Driver_arrived	int	11		
Is_started	int	11		
Is_completed	int	11		
jarak	float			
waktu	float			
total	float			
Is_cancelled	int	11		
D_latitude	Double			
D_longitude	double			

f. request_detail

Tabel 3.50 Struktur *Tabel Request_detail*

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
Id_request_detail	int	10	Primary key	Not null, auto increment
Id_request	Int	11	FK	Not null
Id_driver	Int	11	FK	Not null
status	int	11		Not null
Created_at	timestamp			
Update_at	timestamp			
Is_cancelled	int	11		

g. request_service

Tabel 3.51 Struktur *Tabel Request_service*

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
Id_s_request	int	10	Primary key	Not null, auto increment
Id_request	Int	11	FK	Not null
type	Int	11	FK	Not null
Created_at	timestamp			
Update_at	timestamp			
Harga_awal	float			
Harga_jarak	float			
Harga_waktu	float			
total	float			

h. lokasi_permintaan

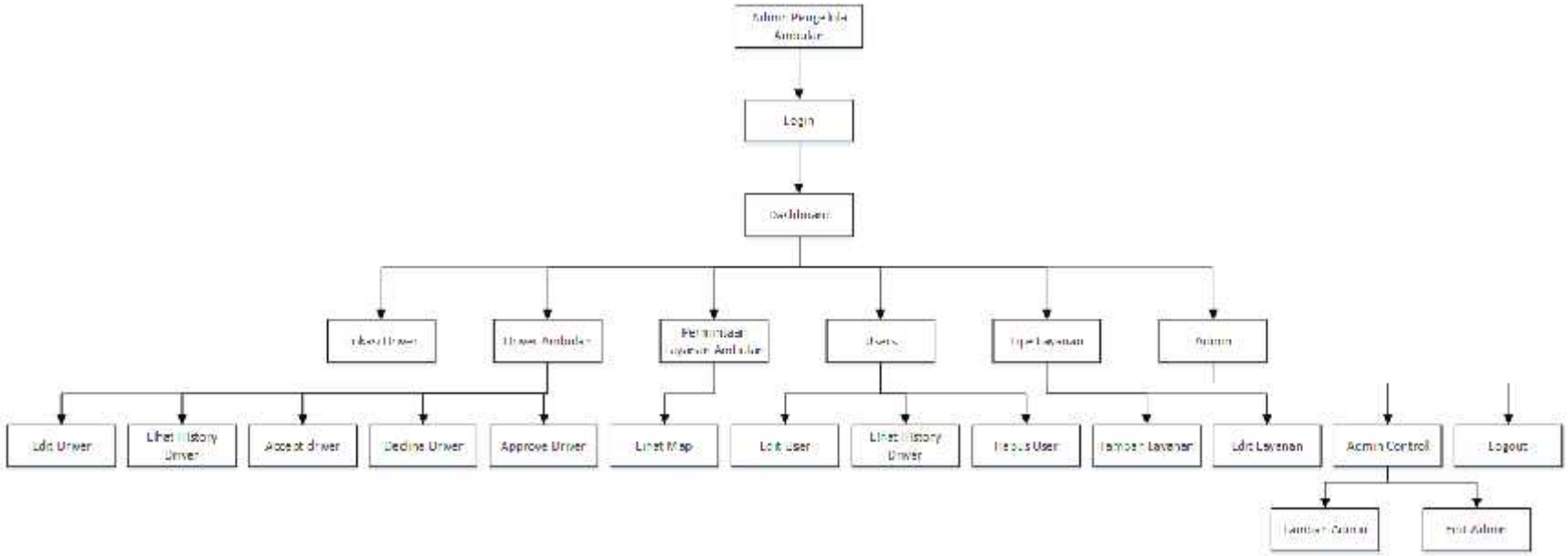
Tabel 3.52 Struktur *Tabel lokasi_permintaan*

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
Id_permintaan_l	int	10	Primary key	Not null, auto increment
Id_request	Int	11	FK	Not null
Created_at	timestamp			
Update_at	timestamp			
Latitude	double			
longitude	doble			
distance	float			

3.2.2. Perancangan Struktur Menu

1. Arsitektur Menu *Backend*

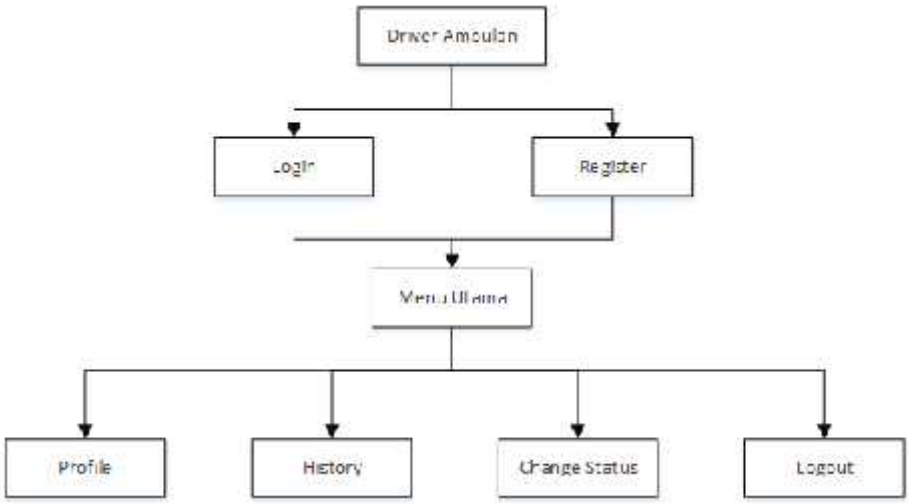
Berikut merupakan Arsitektur Menu pada backend yang dapat dilihat pada Gambar 3.289



Gambar 3.43 Arsitektur Menu *Backend*

2. Arsitektur Menu *Frontend Driver*

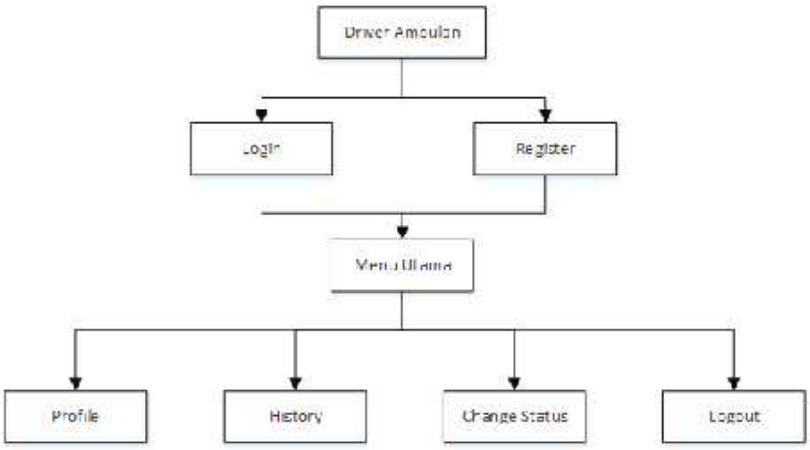
Berikut merupakan Arsitektur Menu pada *frontend driver* yang dapat dilihat pada Gambar 3.44.



Gambar 3.44 Arsitektur Menu Frontend *Driver*

3. Arsitektur Menu *Frontend User*

Berikut merupakan Arsitektur Menu pada *frontend user* yang dapat dilihat pada Gambar 3.45



Gambar 3.45 Arsitektur Menu Frontend *User*

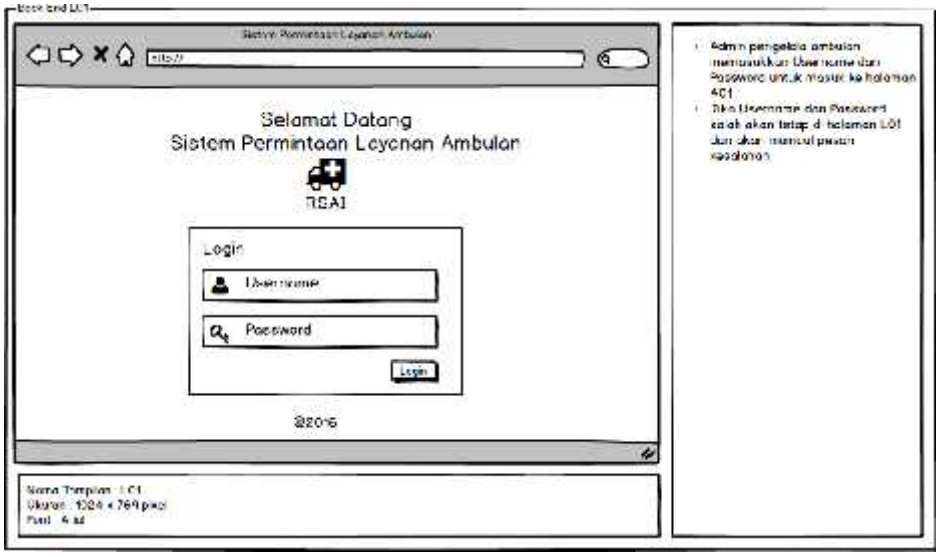
3.2.3. Perancangan Antarmuka

Tahap perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, sehingga akan mempermudah dalam mengimplementasikan aplikasi serta akan memudahkan pembuatan aplikasi. Berikut ini gambaran perancangan antarmuka yang ada dari aplikasi yang akan dibangun yaitu:

3.2.3.1 Perancangan Antarmuka *Backend*

1. Perancangan Antarmuka *Backend* Login

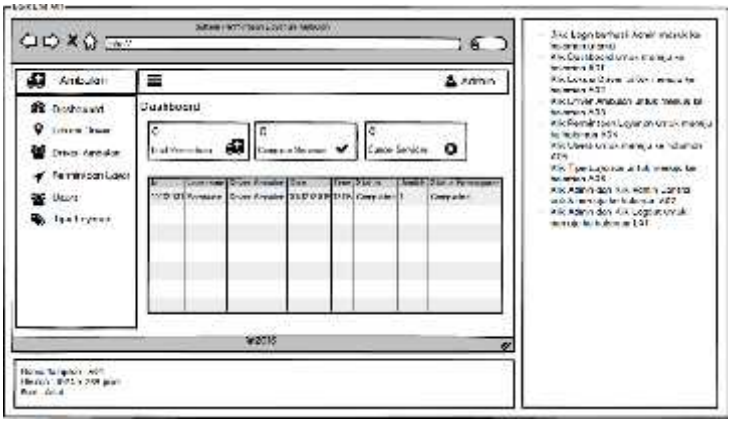
Berikut adalah Gambar 3.46 antarmuka *Backend* L01 login yang menggambarkan rancangan antarmuka halaman login.



Gambar 3.46 Antarmuka *Backend* L01 Login

2. Perancangan Antarmuka *Backend* Halaman Utama

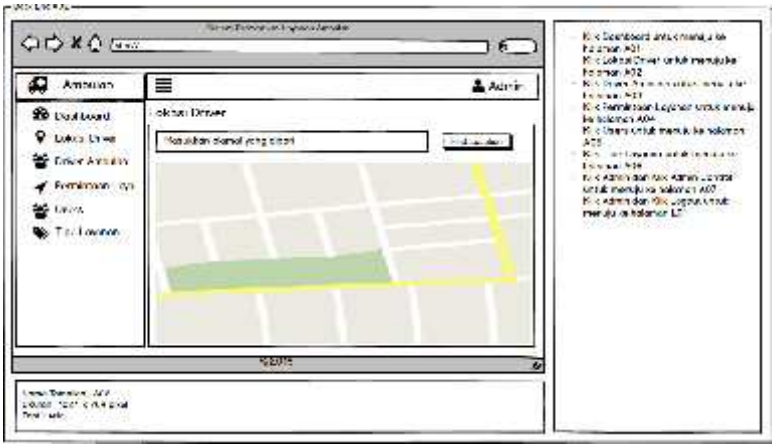
Berikut adalah Gambar 3.47 antarmuka *Backend* A01 Halaman Utama yang menggambarkan rancangan antarmuka halaman utama.



Gambar 3.47 Antarmuka *Backend* A01 Halaman Utama

3. Perancangan Antarmuka *Backend* Lokasi *Driver*

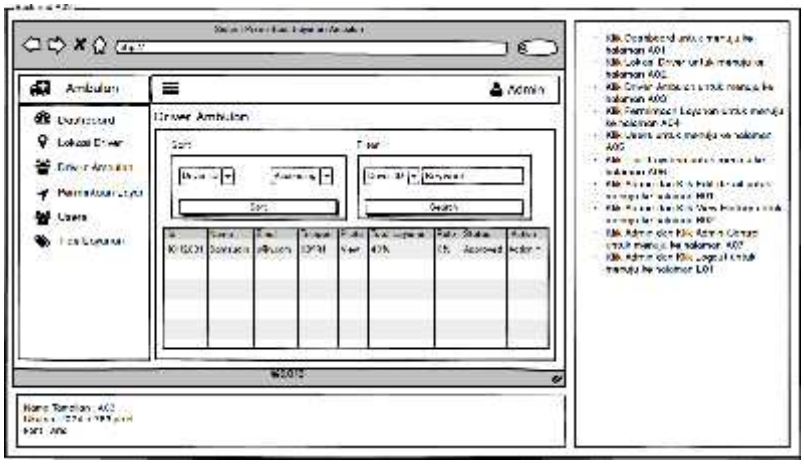
Berikut adalah Gambar 3.31 antarmuka *Backend* A02 Lokasi *Driver* yang menggambarkan rancangan antarmuka halaman lokasi *Driver*.



Gambar 3.48 Antarmuka *Backend* A02 Lokasi *Driver*

4. Perancangan Antarmuka *Backend Driver* Ambulan

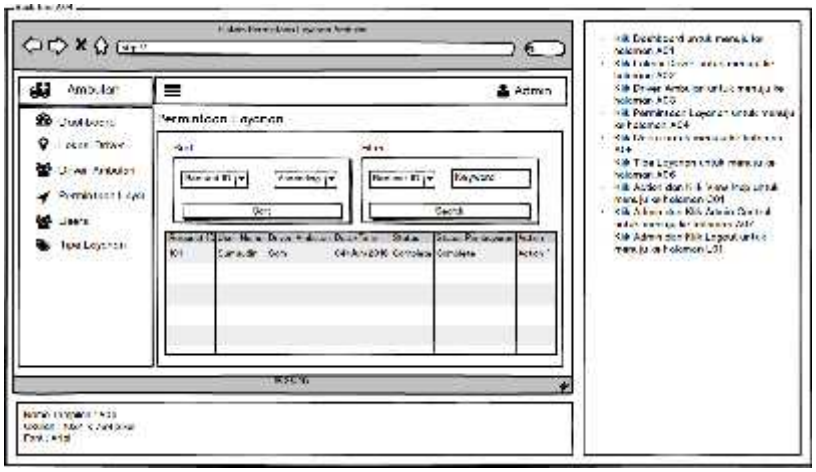
Berikut adalah Gambar 3.49 antarmuka *Backend* A03 *Driver* Ambulan yang menggambarkan rancangan antarmuka halaman *Driver* ambulan.



Gambar 3.49 Antarmuka *Backend* A03 *Driver* Ambulan

5. Perancangan Antarmuka *Backend* Permintaan Layanan

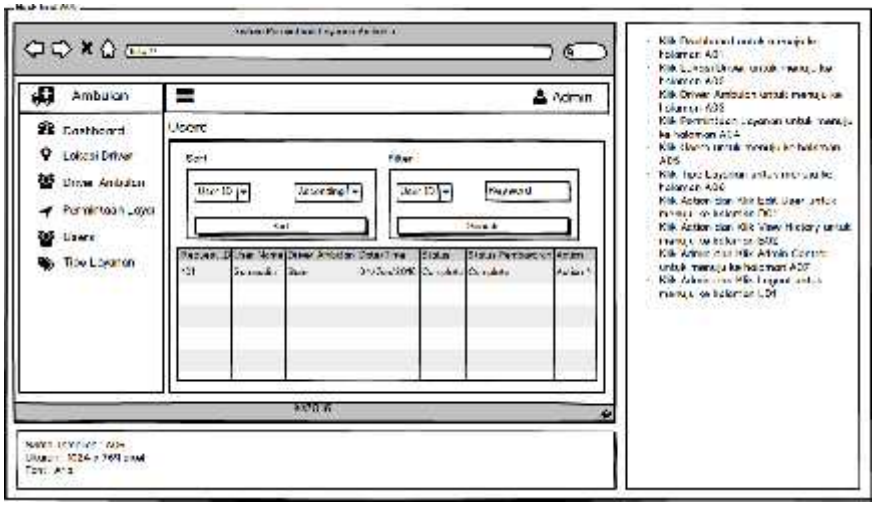
Berikut adalah Gambar 3.33 antarmuka *Backend* A03 Permintaan Layanan yang menggambarkan rancangan antarmuka halaman permintaan layanan.



Gambar 3.50 Antarmuka *Backend* A04 Permintaan Layanan

6. Perancangan Antarmuka *Backend User*

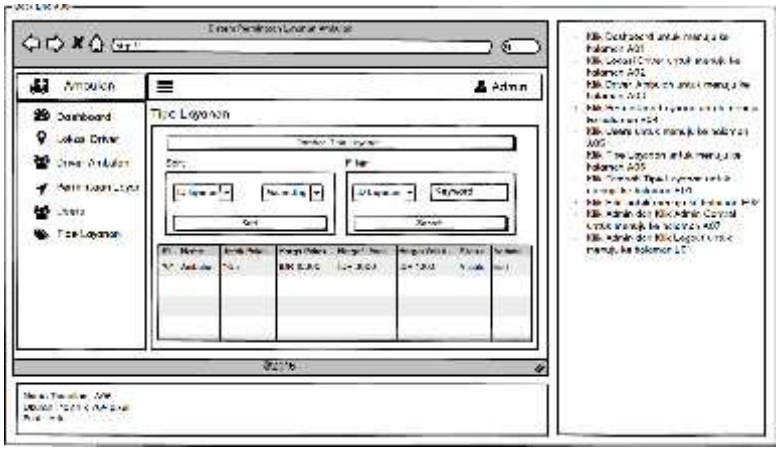
Berikut adalah Gambar 3.51 antarmuka *Backend* A05 *User* yang menggambarkan rancangan antarmuka halaman *User*.



Gambar 3.51 Antarmuka *Backend* A05 User

7. Perancangan Antarmuka *Backend* Tipe Layanan

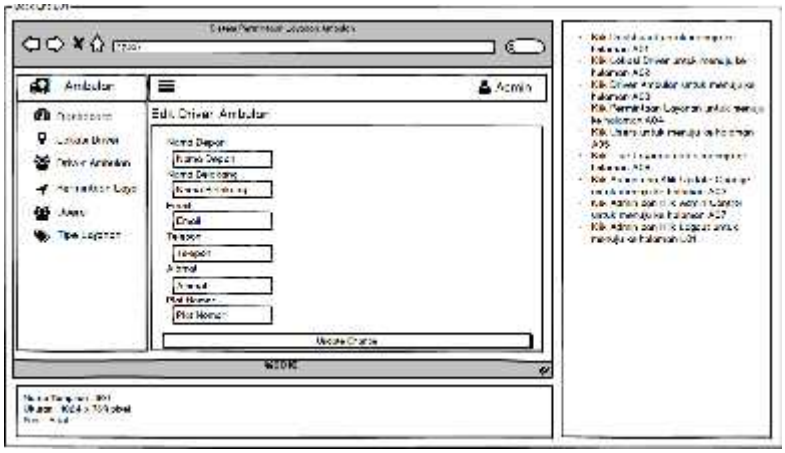
Berikut adalah Gambar 3.52 antarmuka *Backend* A06 Tipe Layanan yang menggambarkan rancangan antarmuka halaman tipe layanan.



Gambar 3.52 Antarmuka *Backend* A06 Tipe Layanan

8. Perancangan Antarmuka *Backend* Edit Detail *Driver*

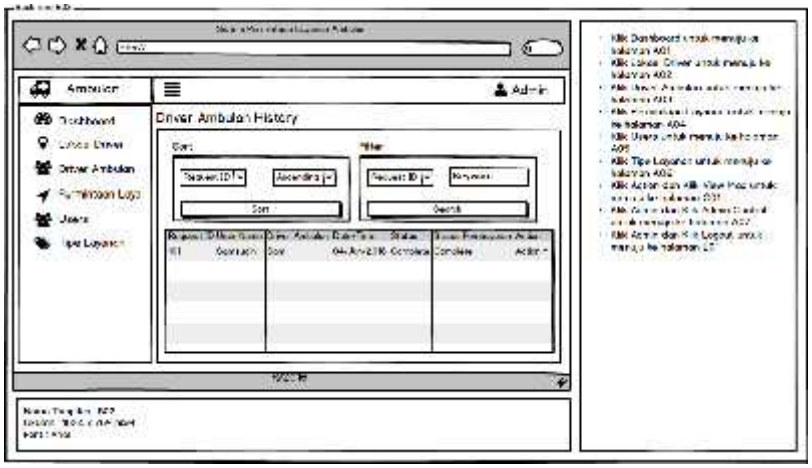
Berikut adalah Gambar 3.53 antarmuka *Backend* B01 Edit Detail *Driver* yang menggambarkan rancangan antarmuka halaman edit *Driver* ambulan.



Gambar 3.53 Antarmuka *Backend* B01 Edit Detail *Driver*

9. Perancangan Antarmuka *Backend View History* Ambulan

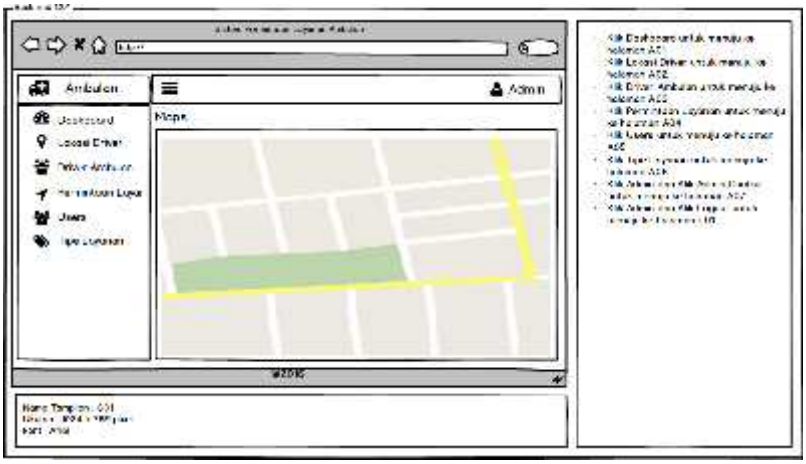
Berikut adalah Gambar 3.54 antarmuka *Backend* B02 *View History* Ambulan yang menggambarkan rancangan antarmuka halaman *History* Ambulan.



Gambar 3.54 Antarmuka *Backend* B02 *View History* Ambulan

10. Perancangan Antarmuka *Backend View* Map

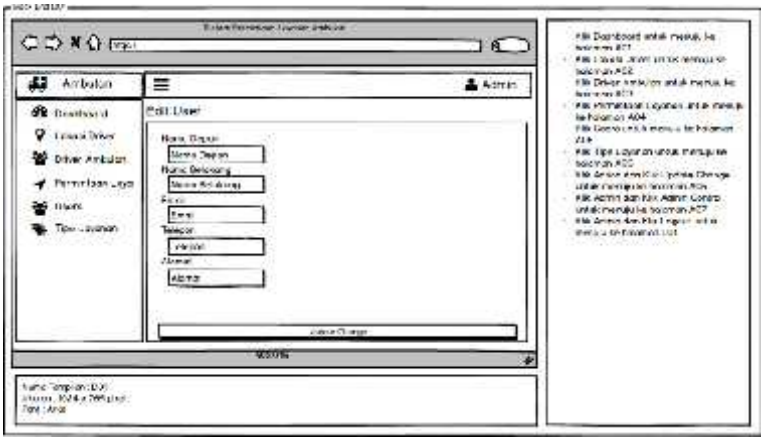
Berikut adalah Gambar 3.55 antarmuka *Backend* C01 *View* Map yang menggambarkan rancangan antarmuka halaman *View* Map.



Gambar 3.55 Antarmuka *Backend* C01 View Map

11. Perancangan Antarmuka *Backend Edit User*

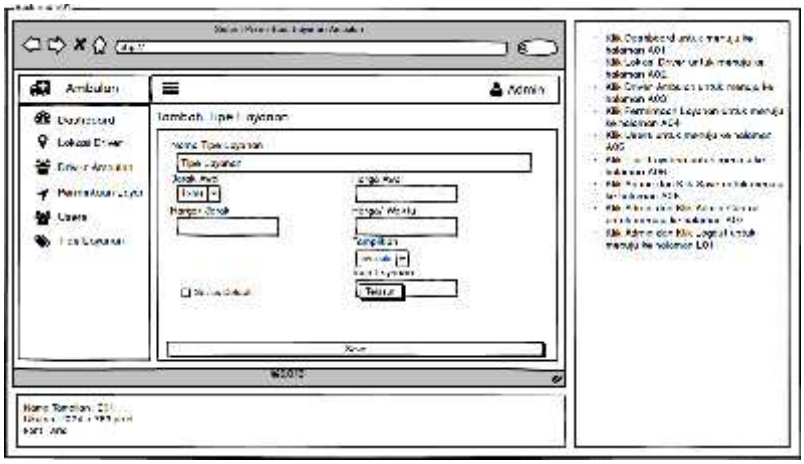
Berikut adalah Gambar 3.39 antarmuka *Backend* D01 *Edit User* yang menggambarkan rancangan antarmuka halaman *edit user*.



Gambar 3.56 Antarmuka *Backend* D01 *Edit User*

12. Perancangan Antarmuka *Backend* Tambah Tipe Layanan

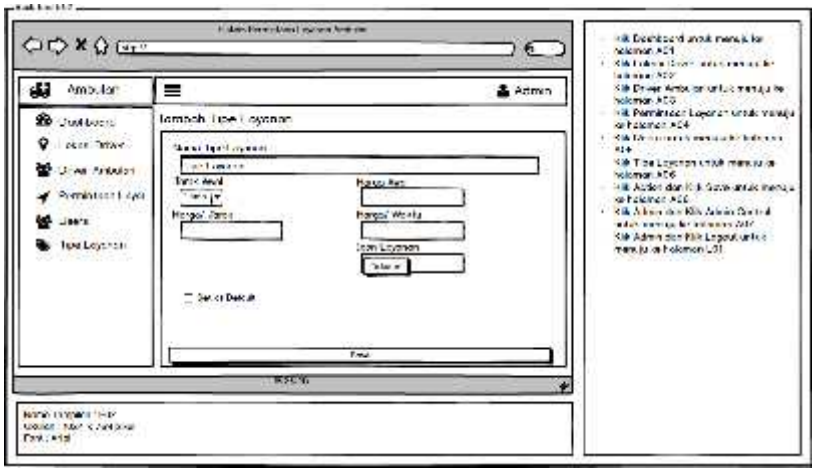
Berikut adalah Gambar 3.57 antarmuka *Backend* E01 Tambah Tipe Layanan yang menggambarkan rancangan antarmuka halaman tambah tipe layanan.



Gambar 3.57 Antarmuka *Backend* E01 Tambah Tipe Layanan

13. Perancangan Antarmuka *Backend Edit* Tipe Layanan

Berikut adalah Gambar 3.58 antarmuka *Backend* E02 *Edit* Tipe Layanan yang menggambarkan rancangan antarmuka halaman *Edit* Tipe Layanan.



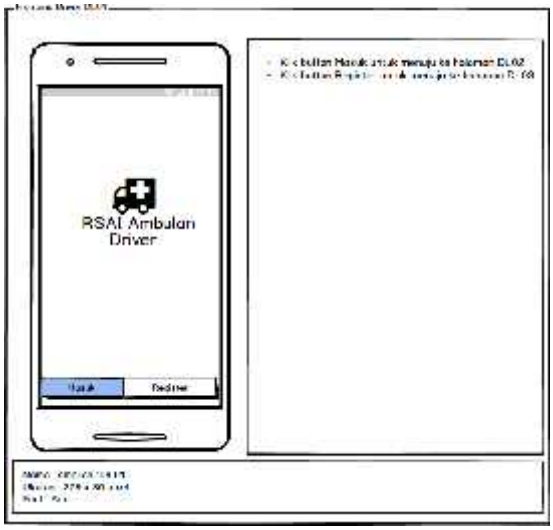
Gambar 3.58 Antarmuka *Backend* E02 Edit Tipe Layanan

3.2.3.2 Perancangan Antarmuka *Frontend*

3.2.3.2.1 Perancangan Antarmuka *Frontend Driver*

1. Perancangan Antarmuka *Frontend* Halaman Pembuka *Driver*

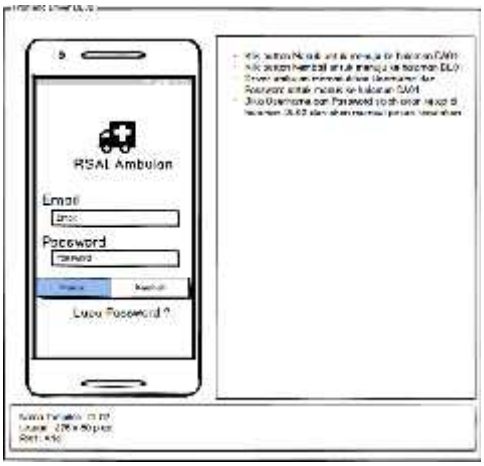
Berikut adalah Gambar 3.59 antarmuka *Frontend Driver* DL01 Halaman Pembuka *Driver* yang menggambarkan rancangan antarmuka halaman pembuka *driver*.



Gambar 3.59 Antarmuka *Frontend* DL01 Halaman Pembuka *Driver*

2. Perancangan Antarmuka *Frontend Login Driver*

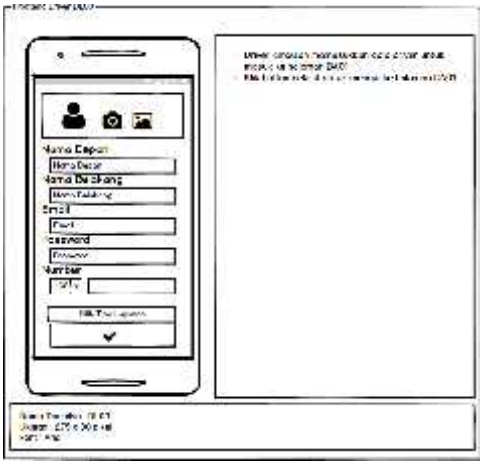
Berikut adalah Gambar 3.60 antarmuka *Frontend* DL02 *Login Driver* yang menggambarkan rancangan antarmuka halaman *login driver*.



Gambar 3.60 Antarmuka *Frontend* DL02 *Login Driver*

3. Perancangan Antarmuka *Frontend Register Driver*

Berikut adalah Gambar 3.61 antarmuka *Frontend* DL03 *Register Driver* yang menggambarkan rancangan antarmuka halaman *register driver*.



Gambar 3.61 *Frontend* DL03 *Register Driver*

4. Perancangan Antarmuka *Frontend* Halaman Utama *Driver*

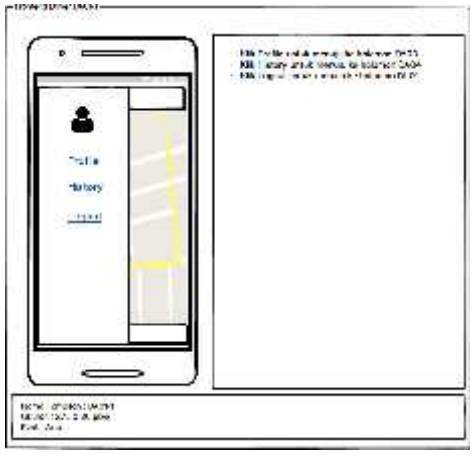
Berikut adalah Gambar 3.62 antarmuka *Frontend* DA01 Halaman Utama *Driver* yang menggambarkan rancangan antarmuka halaman utama *driver*.



Gambar 3.62 *Frontend* Halaman Utama *Driver*

5. Perancangan Antarmuka *Frontend* Menu *Driver*

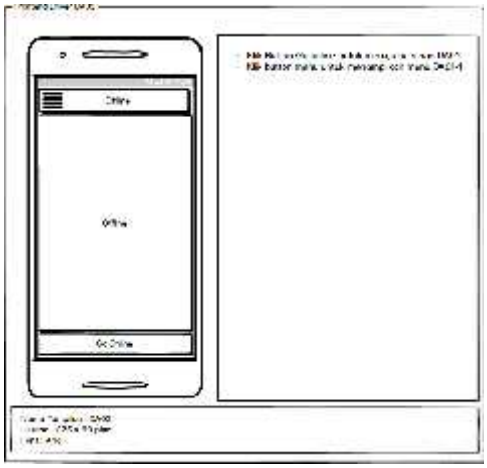
Berikut adalah Gambar 3.46 antarmuka *Frontend DA01-1* Menu *Driver* yang menggambarkan rancangan antarmuka halaman menu *driver*.



Gambar 3.63 *Frontend* DA01-1 Menu *Driver*

6. Perancangan Antarmuka *Frontend* Ubah Status *Driver*

Berikut adalah Gambar 3.64 antarmuka *Frontend DA02* Ubah Status *Driver* yang menggambarkan rancangan antarmuka halaman ubah status *driver*.



Gambar 3.64 *Frontend* DA02 Ubah Status *Driver*

7. Perancangan Antarmuka *Frontend Edit Profile Driver*

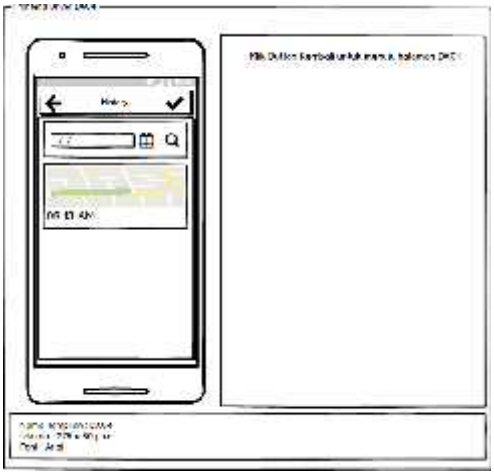
Berikut adalah Gambar 3.65 antarmuka *Frontend* DA03 *Edit Profile Driver* yang menggambarkan rancangan antarmuka halaman *edit profile driver*.



Gambar 3.65 *Frontend* DA03 *Edit Profile Driver*

8. Perancangan Antarmuka *Frontend History Driver*

Berikut adalah Gambar 3.66 antarmuka *Frontend* DA04 *History Driver* yang menggambarkan rancangan antarmuka halaman *history driver*.



Gambar 3.66 *Frontend* DA04 *History Driver*

3.2.3.2.2 Perancangan Antarmuka *Frontend* User

1. Perancangan Antarmuka *Frontend* Halaman Pembuka *User*

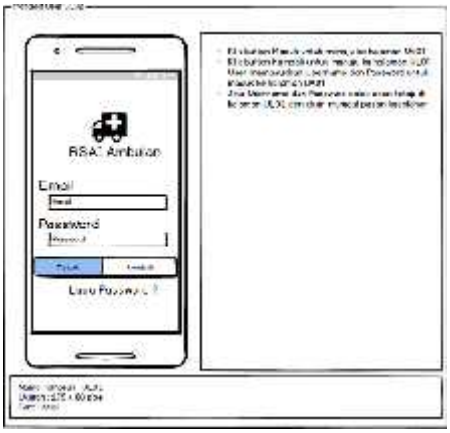
Berikut adalah Gambar 3.67 antarmuka *Frontend* UL01 Halaman Pembuka *User* yang menggambarkan rancangan antarmuka halaman pembuka *user*.



Gambar 3.67 *Frontend* UL01 Halaman Pembuka *User*

2. Perancangan Antarmuka *Frontend Login User*

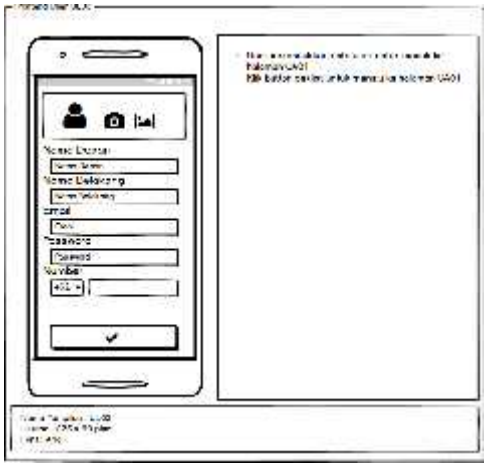
Berikut adalah Gambar 3.68 antarmuka *Frontend* UL02 *Login User* yang menggambarkan rancangan antarmuka halaman *login user*



Gambar 3.68 *Frontend* UL02 *Login User*

3. Perancangan Antarmuka *Frontend Register User*

Berikut adalah Gambar 3.69 antarmuka *Frontend* UL03 *Register User* yang menggambarkan rancangan antarmuka halaman *register user*.



Gambar 3.69 *Frontend* UL03 *Register User*

4. Perancangan Antarmuka *Frontend* Halaman Utama *User*

Berikut adalah Gambar 3.70 antarmuka *Frontend* UA01 Halaman Utama *User* yang menggambarkan rancangan antarmuka halaman utama *user*.



Gambar 3.70 *Frontend* UA01 Halaman Utama *User*

5. Perancangan Antarmuka *Frontend* Menu *User*

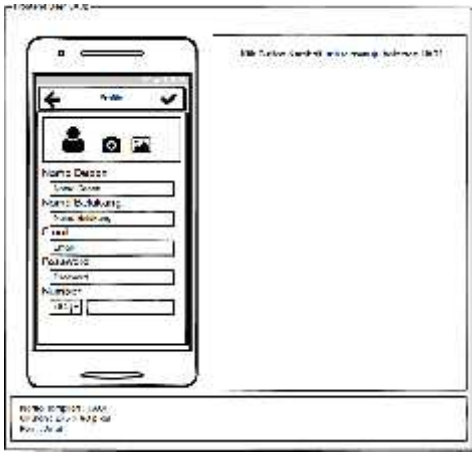
Berikut adalah Gambar 3.71 antarmuka *Frontend* UA01-1 Menu *User* yang menggambarkan rancangan antarmuka halaman menu *user*.



Gambar 3.71 *Frontend* UA01-1 Menu *User*

6. Perancangan Antarmuka *Frontend History Driver*

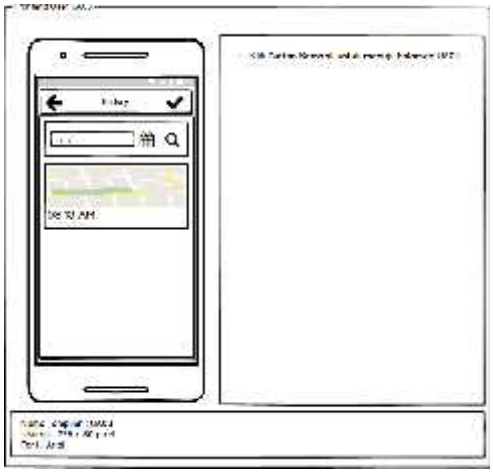
Berikut adalah Gambar 3.72 antarmuka *Frontend* DA04 *History Driver* yang menggambarkan rancangan antarmuka halaman *history driver*.



Gambar 3.72 *Frontend* DA04 *History Driver*

7. Perancangan Antarmuka *Frontend History User*

Berikut adalah Gambar 3.73 antarmuka *Frontend* UA03 *History User* yang menggambarkan rancangan antarmuka halaman *history user*.

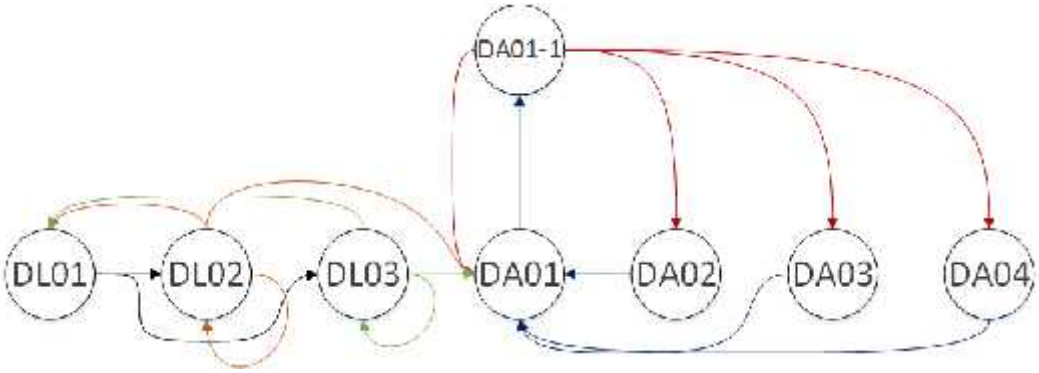


Gambar 3.73 *Frontend* UA03 *History User*

3.2.4. Jaringan Semantik

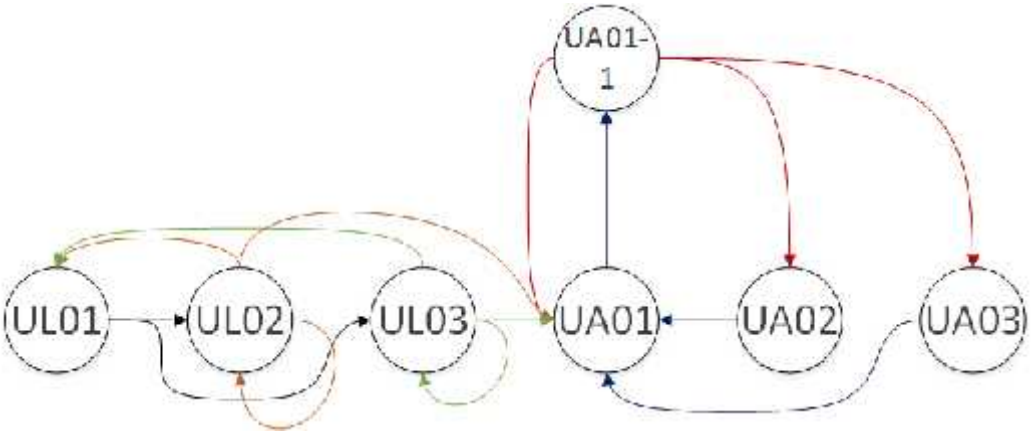
Berikut beberapa jaringan semantik yang ada pada sistem permintaan layanan ambulan ini adalah sebagai berikut:

1. Jaringan Semantik *Frontend Driver*



Gambar 3.74 Jaringan Semantik *Frontend Driver*

2. Jaringan Semantik *Frontend User*



Gambar 3.75 Jaringan Semantik *Frontend User*