

BAB III

ANALISIS DAN DESAIN SISTEM

III.1 Analisis Masalah

Analisis bertujuan untuk mengidentifikasi permasalahan-permasalahan yang ada pada sistem serta menentukan kebutuhan dari sistem yang dibangun. Layanan ambulans ini sangat dibutuhkan masyarakat terutama dalam hal mengevakuasi terjadinya bencana alam, warga masyarakat yang meninggal dunia, kecelakaan lalu lintas dan berbagi keperluan transportasi jenazah atau orang yang sedang sakit. Khusus ambulans rumah sakit keberadaannya dikhususkan hanya kepada pasien yang dirawat di rumah sakit tersebut dan tidak diperbolehkan diluar rumah sakit kecuali dalam situasi yang sangat darurat.

Seperti halnya kerja sebuah aplikasi ojek *online* yang dapat menerima pemesanan dan menemukan *driver* yang terdekat pada lokasi pemesanan, hal ini akan di implementasikan pada sebuah aplikasi yang memungkinkan setiap pengguna mendapatkan manfaat untuk mengambil tindakan dini pada sebuah kecelakaan atau pun penyakit medis lainnya.

Untuk perancangan aplikasi ini menggunakan metode *Waterfall* agar hasil yang didapat lebih maksimal.

III.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem merupakan tahap pengembangan sistem yang bertujuan untuk mengetahui kebutuhan-kebutuhan yang diperlukan oleh sistem dan penguraian terhadap sistem yang akan dibangun. Analisis kebutuhan sistem yang

digunakan dalam membuat aplikasi pemesanan ambulan berbasis *Android* adalah analisis kebutuhan masukan (*input*), analisis kebutuhan proses, analisis kebutuhan perangkat lunak (*software*), dan analisis kebutuhan perangkat keras (*hardware*).

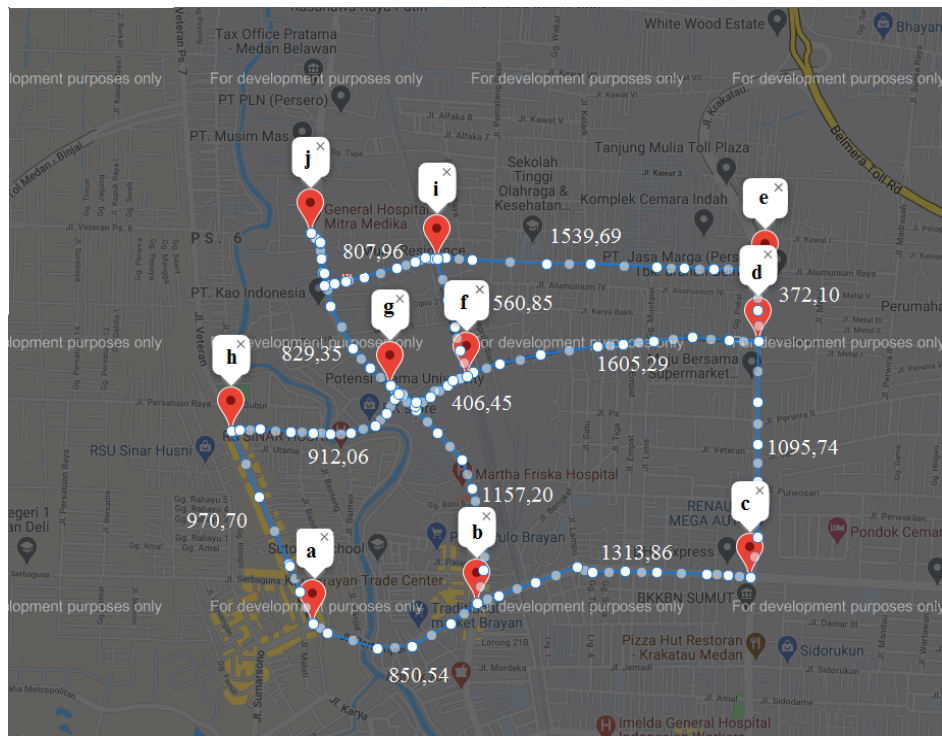
III.2.1 Analisis Penerapan Metode Algoritma Dijkstra

Menurut Marsudi (2016) *Algoritma Dijkstra* adalah sebuah algoritma untuk menentukan lintasan terpendek dalam graf tak berarah(atau graf berarah) berbobot tanpa mengenumerasi secara tepat semua lintasan yang mungkin. algoritma ini didasarkan pada sebuah metode yang dikenal sebagai pemrograman dinamik. algoritma dijkstra menentukan lintasan terpendek diantara pasangan-pasangan titik dalam suatu graf.

Berikut merupakan beberapa notasi yang digunakan dalam Algoritma Dijkstra (Dharam, 2015), yaitu:

- a. Grafik G
- b. Satu set simpul atau titik V (Vertex)
- c. Seperangkat sisi E (dimana setiap sisi memiliki berat non-negatif)
- d. Titik awal $s \in V$

Perlu diperhatikan bahwa simbol “elemen”, \in , menunjukkan bahwa elemen di sisi kiri simbol terkandung dalam sisi lain simbol. Misalnya, $s \in V$ menjelaskan bahwa s adalah elemen dari V. Berarti, dalam hal ini s adalah titik yang terdapat di dalam grafik.



Gambar III.1. Contoh Soal Penerapan Algoritma Dijkstra

Pada perhitungan tahap inialisasi dengan algoritma Dijkstra, simpul awal diberi inisial dengan “1” dan simpul lain dengan inisial “0”, kemudian setiap node yang sudah dipilih diinisialkan dengan “1”.

Titik awal = A (Simpang Zipur)

Titik tujuan = J(RS. Mitra Medika)

Tabel III.1. Penyelesaian Algoritma Literasi 1

Literasi Ke-1										
Node	A	B	C	D	E	F	G	H	I	J
Status	1	0	0	0	0	0	0	0	0	0
Bobot	-	850	-	-	-	-	-	970	-	-
Predecessor	A	A	-	-	-	-	-	A	-	-

Titik terendah yang terhubung pada titik A yaitu titik B

Tabel III.2. Penyelesaian Algoritma Literasi 2

Literasi Ke-2										
Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	0	0	0	0	0	0	0	0
Bobot	-	850	2164	-	-	-	2007	970	-	-
predecessor	A	A	B	-	-	-	B	A	-	-

Titik yang telah dilewati yaitu A ke B

Titik terendah yang terhubung ke titik B yaitu H

Tabel III.3. Penyelesaian Algoritma Literasi 3

Literasi Ke-3										
Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	0	0	0	0	0	1	0	0
Bobot	-	850	2164	-	-	-	1882	970	-	-
predecessor	A	A	B	-	-	-	H	A	-	-

Titik yang telah dilewati yaitu A ke B dan A ke H

Pada titik A - B - G = 2007 dan A-H-G = 1882, karena A - H - G lebih rendah maka

pada predecessor titik G di ubah menjadi B

Titik terendah yang terhubung ketitik H dan yang terhubung ketitik B yaitu G

Tabel III.4. Penyelesaian Algoritma Literasi 4

Literasi Ke-4										
Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	0	0	0	0	1	1	0	0
Bobot	-	850	2164	-	-	2289	2007	970	-	2712,1

Predecessor	A	A	B	-	-	G	H	A	-	G
--------------------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

Titik yang telah dilewati yaitu A - B dan A - H - G

Titik terendah yang terhubung ketitik G dan yang terhubung ketitik B yaitu C

Tabel III.5. Penyelesaian Algoritma Literasi 5

Literasi Ke-5										
Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	1	0	0	0	1	1	0	0
Bobot	-	850	2164	3260	-	2289	2007	970	-	2712
Predecessor	A	A	B	C	-	G	H	A	-	G

Titik yang telah dilewati yaitu A - B - C dan A - H - G

Titik terendah yang terhubung ketitik G dan yang terhubung ketitik C yaitu F

Tabel III.6. Penyelesaian Algoritma Literasi 6

Literasi Ke-6										
Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	1	0	0	1	1	1	0	0
Bobot	-	850	2164	3260	-	2289	2007	970	2849	2712
Predecessor	A	A	B	C	-	G	H	A	-	G

Titik yang telah dilewati yaitu A - B - C - D dan A - H - G

Titik terendah yang terhubung ketitik G dan yang terhubung ketitik D yaitu J

Karena titik J yang terendah maka rute terdekat menuju J yaitu A - H - G - J

III.2.2 Analisis Kebutuhan Masukan

Analisis kebutuhan masukan (*input*) yang dilakukan dalam perancangan aplikasi adalah sebagai berikut:

a. Data Pengguna

Data pengguna ini berisikan data pribadi pengguna Aplikasi, seperti KTP, nama pengguna, dan *password* yang akan digunakan untuk *Login*.

b. Data Rumah Sakit

Data Rumah Sakit ini berisi nama Rumah Sakit tersebut serta alamat Rumah Sakit dan no telepon Rumah Sakit.

c. Data Rute

Data rute ini berisi rute-rute antara node Rumah Sakit ke node lain. Dan dalam setiap rute memiliki beberapa node.

III.2.3 Analisis Kebutuhan Proses

Analisis kebutuhan proses dalam pembuatan aplikasi pemesanan ambulans berbasis android adalah sebagai berikut:

a. Proses *Login*

b. Proses pemilihan menu

c. Proses pemesanan ambulans

d. Proses penginputan data

III.2.4 Analisis Kebutuhan *Hardware* dan *Software*

Dalam perancangan aplikasi pemesanan ambulans secara *online* berbasis android dibutuhkan beberapa *hardware* dan *software*, antara lain sebagai berikut :

1. *Personal Komputer* (Laptop)

Hardware yang digunakan sebagai berikut :

- *Processor* : Core i3
- *Memory* : 6 Gb
- *Harddisk* : 500 Gb

Software yang digunakan sebagai berikut :

- Sistem Operasi : Windows 10
- Android IDE : Eclipse IDE

2. *Mobile Device* (Android)

Hardware yang digunakan sebagai berikut :

- *Processor* : 1,5Ghz Octa-core
- *Memory* : 4 Gb
- *Internal Device* : 64 Gb

Software yang digunakan sebagai berikut :

- Sistem Operasi : Windows 10

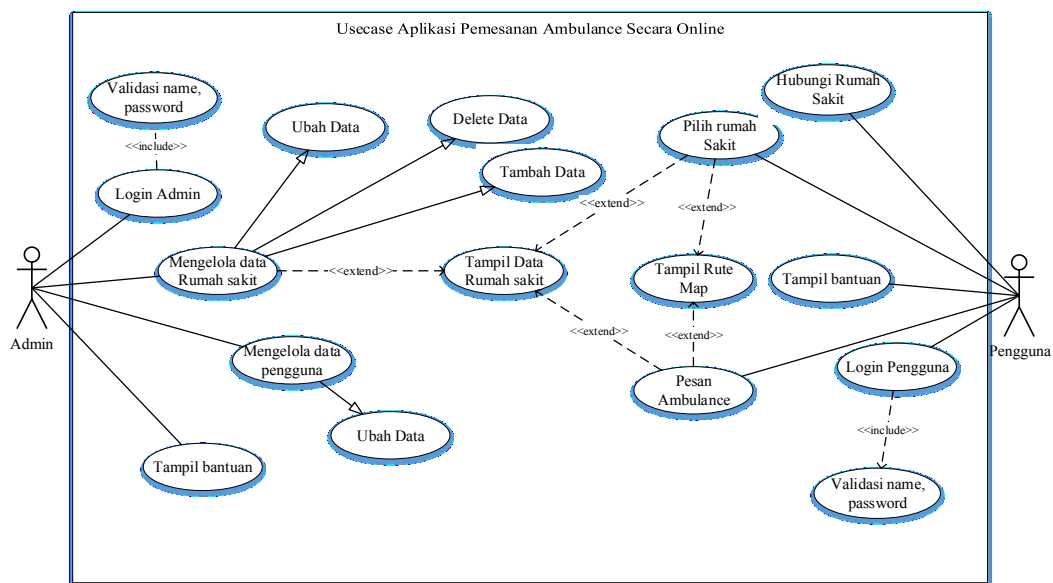
III.3 Perancangan Sistem

Perancangan sistem merupakan suatu proses menggambarkan suatu sistem untuk memenuhi kebutuhan yang telah dijabarkan pada tahap analisis, dengan menggambarkan sketsa atau alur sistem yang akan dibangun dari tahap sebelumnya yaitu analisis yang telah direncanakan menjadi satu kesatuan rancangan khusus yang nantinya dapat dimengerti. Perancangan sistem juga dapat menjelaskan bagian yang terdapat pada sistem secara terperinci, sehingga dalam pembuatannya harus jelas, lengkap, dan mudah dimengerti agar dapat memenuhi kebutuhan sistem dan pengguna lebih mudah untuk menggunakan sistem tersebut, dalam aplikasi ini perancangan yang digunakan adalah perancangan dengan *Activity diagram*, *use*

case diagram, sequence diagram, class diagram perancangan dilakukan setelah tahap analisa selesai.

III.3.1 Use Case Diagram

Use Case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem inForm asi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem inForm asi yang akan dibuat.



Gambar III.2 Use case Diagram

III.3.2 Class Diagram

Menurut Rosa A. S dan M. Shalahuddin, (2018). Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
- Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak atau *programmer* dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

- Kelas main

Kelas yang memiliki fungsi awal di eksekusi ketika sistem di jalankan.

- Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

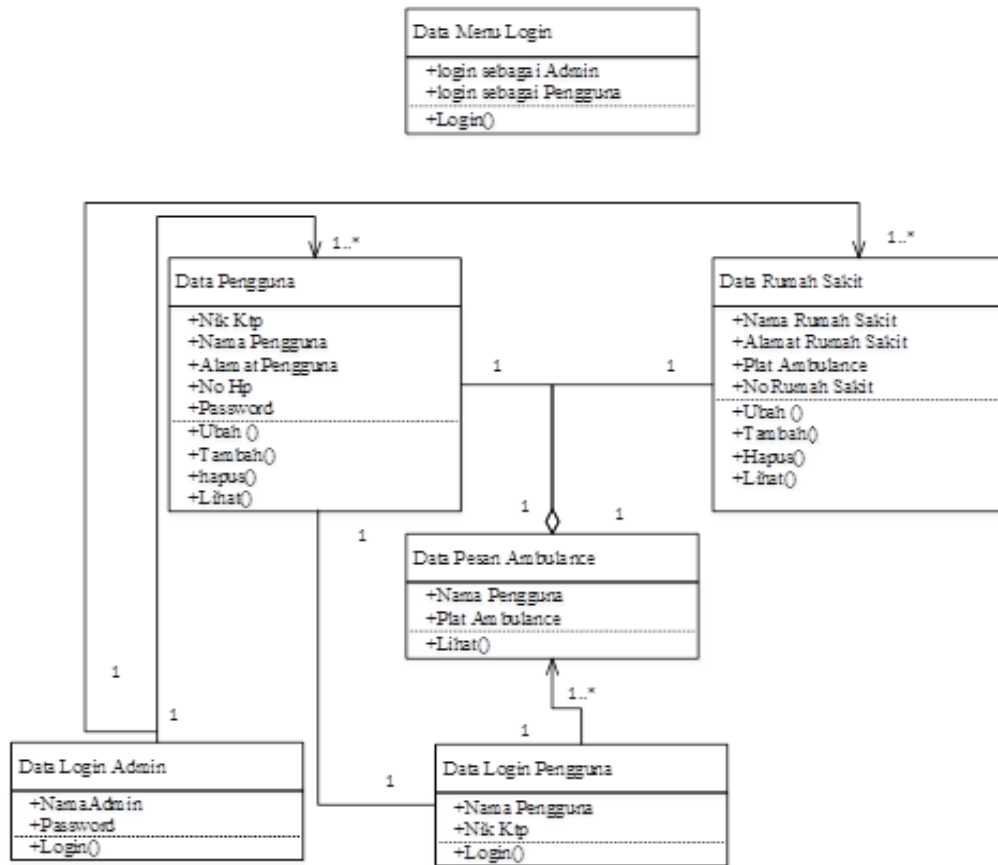
- Kelas yang diambil dari pendefinisian *use case* (*controller*)

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

- Kelas yang diambil dari pendefinisian data (*model*)

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Berikut ini *class* diagram aplikasi pemesanan ambulan :



Gambar III.3 Class Diagram Aplikasi Ambulan

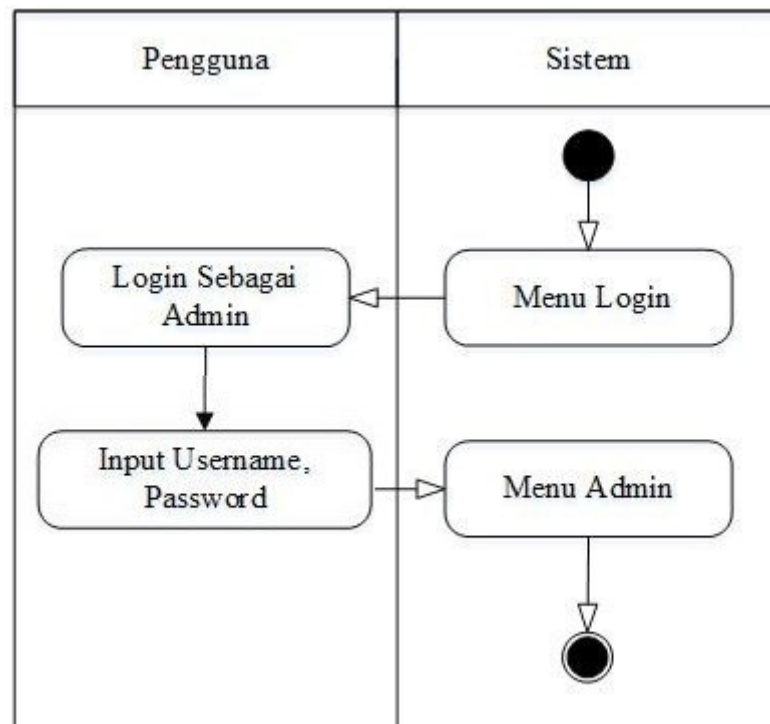
III.3.3 Activity Diagram

Menurut Rosa A. S dan M. Shalahuddin. (2018) Diagram aktivitas atau *Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Dalam aplikasi pemesanan ambulan ini terdapat 3 aktor yang aktif dalam menjalankan aplikasi antara lain : pengguna, supir, dan admin. Adapun setiap aktor

memiliki beberapa *Activity* yang dapat dilakukan. Berikut adalah detail dari setiap *Activity* diagram yang dilakukan oleh setiap aktor:

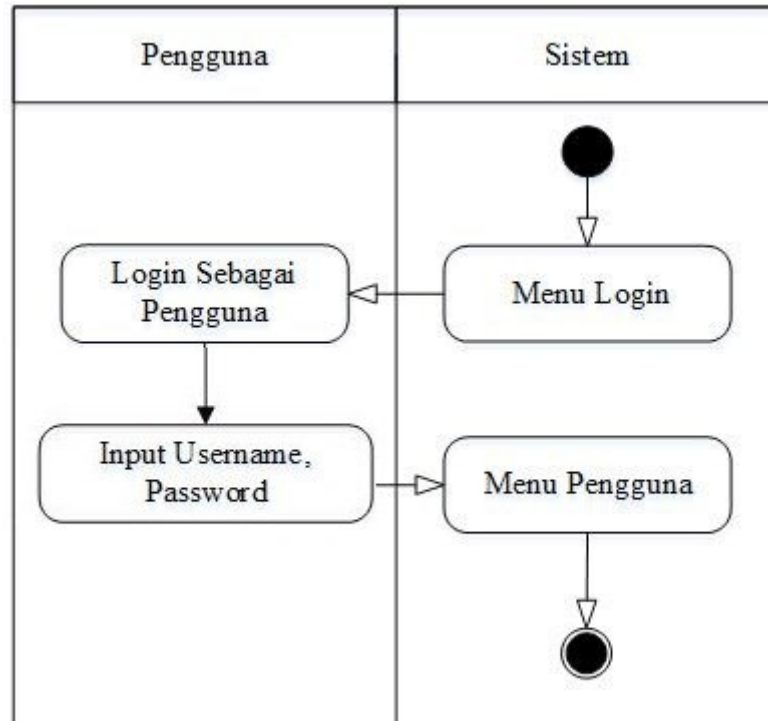
a. *Activity Login Admin*



Gambar III.4 Activity Login Admin

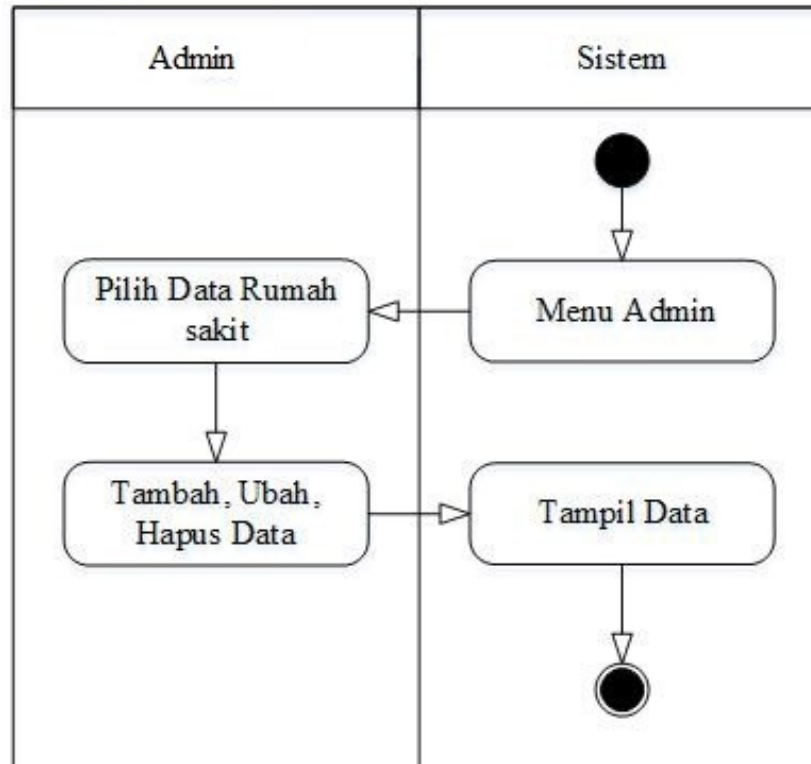
Pada gambar III.3 menjelaskan bagaimana proses antara aktor dan sistem yang terjadi saat admin melakukan proses *Login*. Adapun proses dimulai dari sistem yang menampilkan *interface* pemilihan *Login*, setelah memilih, aktor akan menginput *username* dan *password* untuk melanjutkan proses *Login* menuju menu utama.

b. *Activity Login Pengguna*



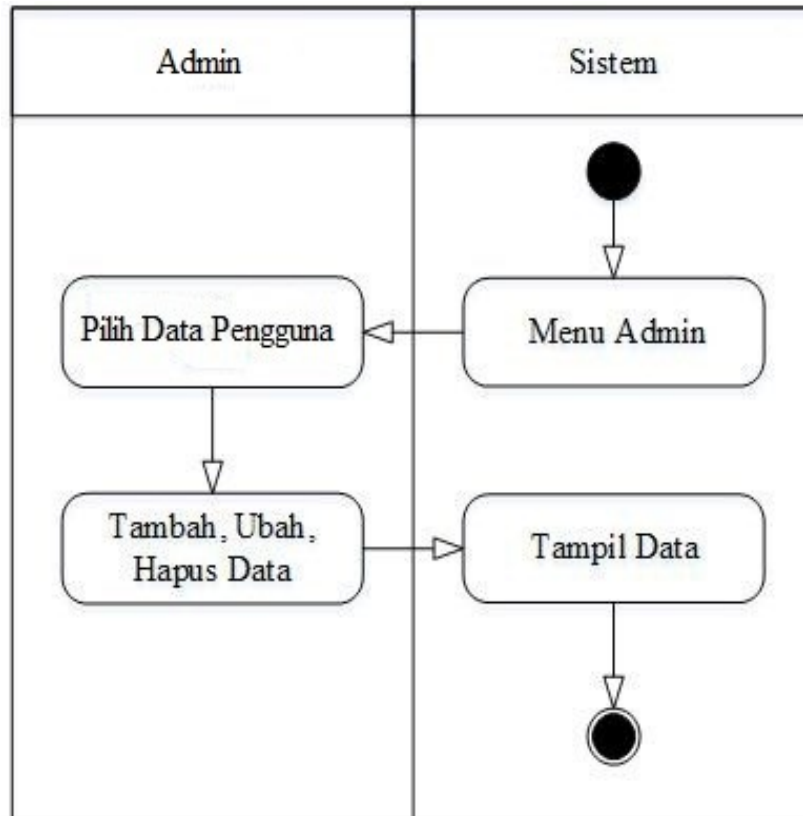
Gambar III.5 Activity Login Pengguna

Pada gambar diatas, sama halnya dengan Gambar III.3, perbedaannya di gambar III.4 pemilihan pada menu *Login*, aktor memilih sebagai pengguna dan diteruskan masuk ke menu pengguna.

c. *Activity* Pengolahan Data Rumah Sakit

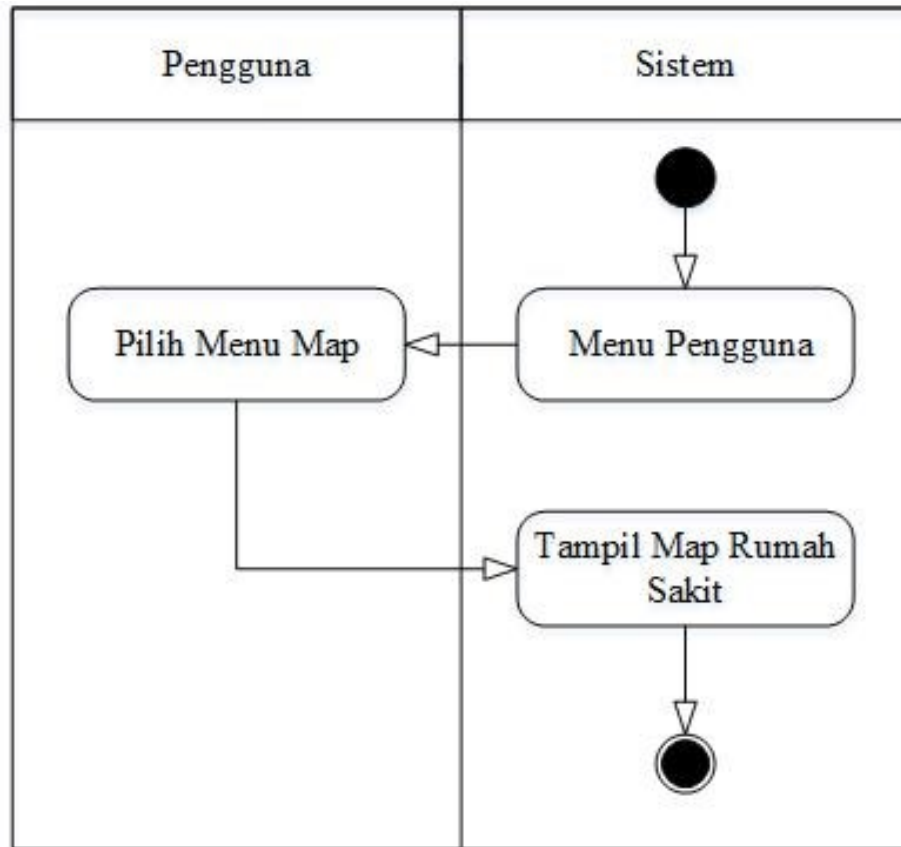
Gambar III.6 *Activity* Pengelolaan Data Rumah Sakit

Pada gambar III.5 menjelaskan salah satu menu admin yaitu bagaimana proses pengelolaan data rumah sakit yang dapat dilakukan. Dari menambah, mengubah dan menghapus data tersebut.

d. *Activity* Pengelolaan Data Pengguna

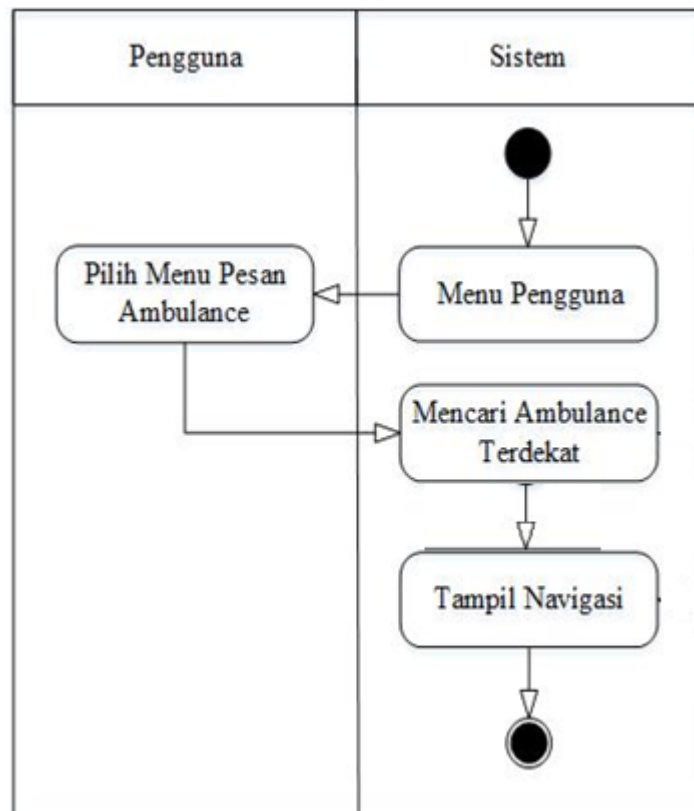
Gambar III.7 *Activity* Pengelolaan Data Pengguna

Pada gambar diatas sama halnya cara kerja sistem dan aktor dengan gambar III.6, hanya saja pada *Activity* pengelolaan data pengguna ini aktor atau admin melakukan pengelolaan terhadap data pengguna aplikasi pemesanan ambulan ini.

e. *Activity Menu Map Untuk Pengguna*

Gambar III.8 *Activity Menu Map Untuk Pengguna*

Pada *Activity Menu Map Untuk Pengguna* ini, yang berperan sebagai aktor adalah pengguna. Menu *map* ini memang hanya ada di pilihan menu pengguna. Dimana setelah memilih menu ini, maka sistem akan menampilkan lokasi-lokasi rumah sakit terdekat dengan titik pengguna aplikasi ambulan ini.

f. *Activity* Pemesan Ambulan

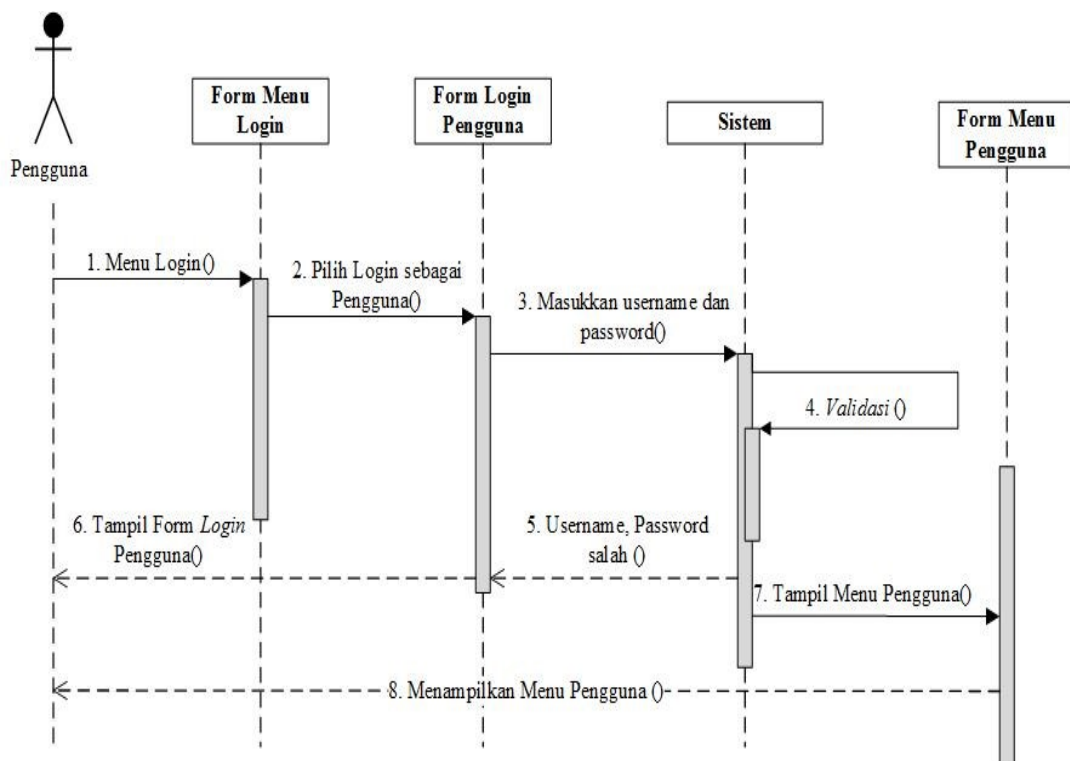
Gambar III.9 *Activity* Pemesanan Ambulan

Pada *Activity* Pemesanan Ambulan ini adalah menu utama pada aplikasi pemesanan ambulance, dan menu utama ini hanya dapat digunakan oleh pengguna dalam mencari Rumah Sakit terdekat.

II.3.4 Sequence Diagram

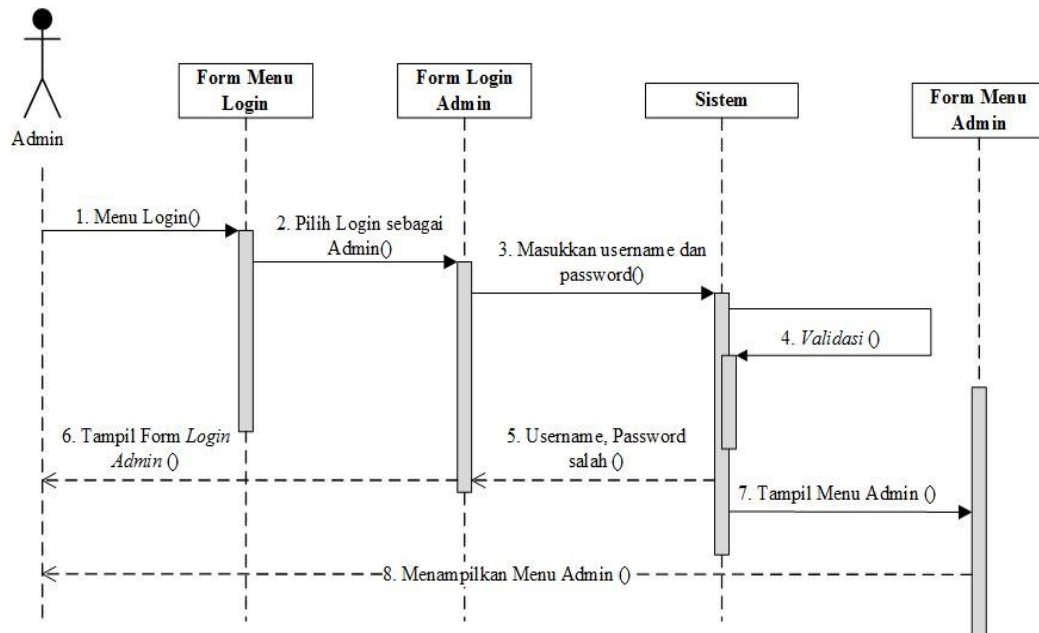
Menurut Rosa A. S dan M. Shalahuddin. (2018) Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesanyang dikirimkan dan diterima antar objek. Oleh karena itu, untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Berikut ini beberapa sekuen diagram yang berjalan di aplikasi yang peneliti buat.

1. Sequence Diagram Login Admin



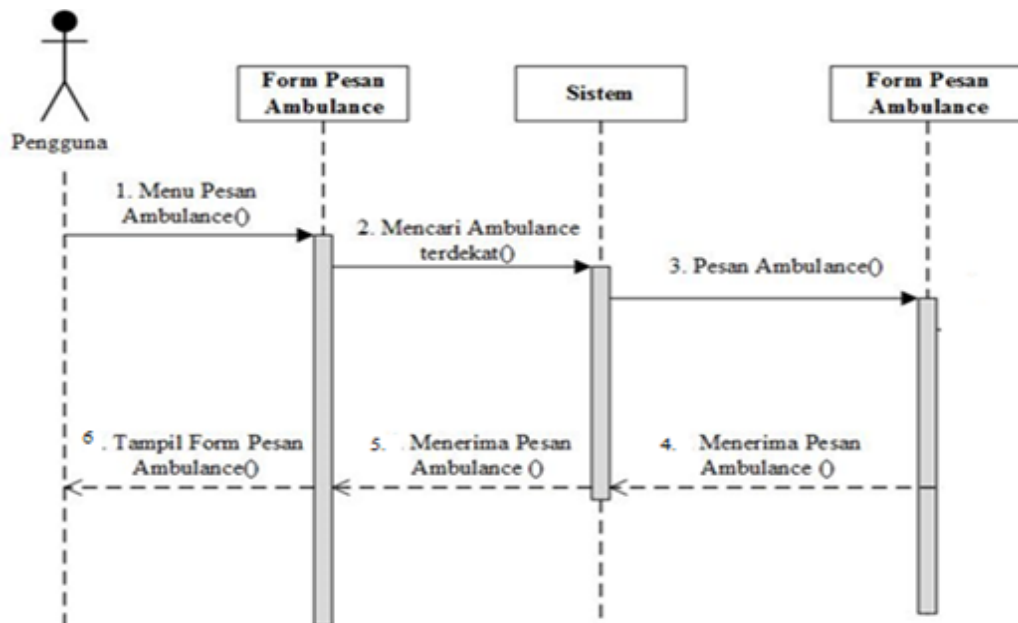
Gambar III.10 Sequence Diagram Login Admin

2. Sequence Diagram Login Pengguna



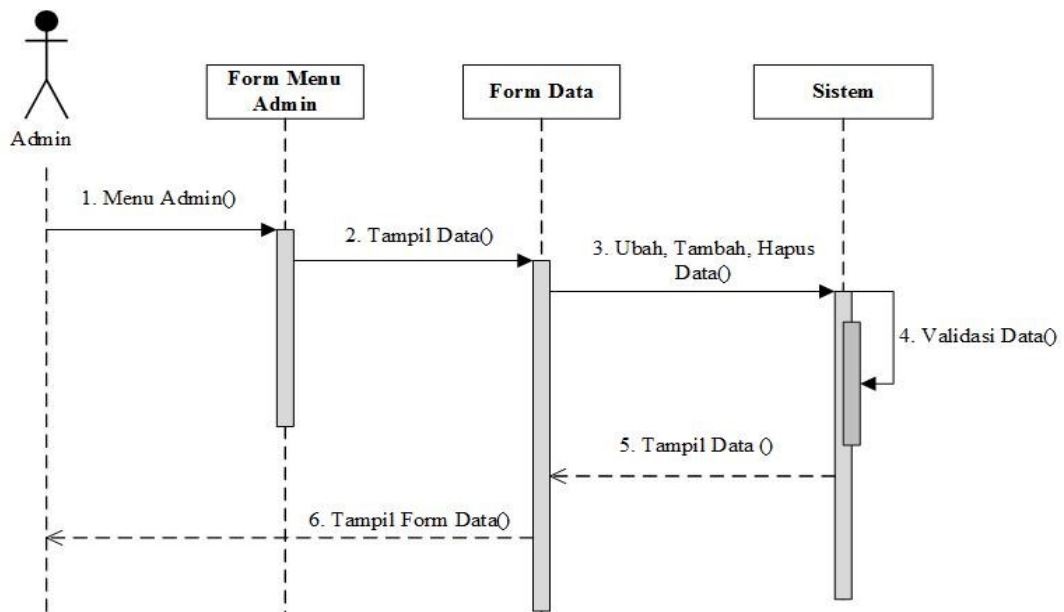
Gambar III.11 Sequence Diagram Login Pengguna

3. Sequence Diagram Pemesanan Ambulan



Gambar III.12 Sequence Diagram Pemesanan Ambulan

4. *Sequence Diagram Admin*



Gambar III.13 *Sequence Diagram Admin*

III.4 Perancangan Antar Muka (*Mockup Interface*)

Perancangan antarmuka (*mockup*) sistem bertujuan untuk memberikan gambaran terkait sistem yang akan dibangun dan terdiri dari beberapa bagian *mockup* yaitu:

The image displays two wireframe diagrams for a login menu, both titled "Menu Login".

The left diagram shows a vertical container with a header "Menu Login". Below the header, there are two rounded rectangular buttons stacked vertically. The top button is labeled "Login Admin" and the bottom button is labeled "Login Pengguna".

The right diagram shows a vertical container with a header "Menu Login". Below the header, there are two input fields. The first is labeled "User Name" and the second is labeled "Password". Below the input fields, there are two rounded rectangular buttons: "Login" on the left and "Batal" on the right.

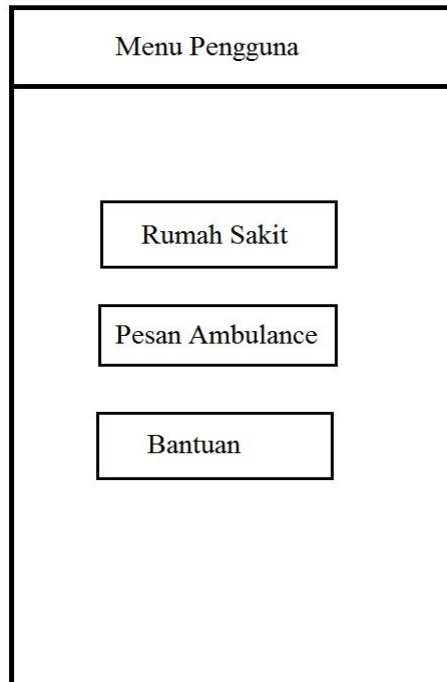
Gambar III.14 Form Login

Pada tampilan *Form Login* terdapat dua tampilan, tampilan pertama yaitu menu *Login*, pada tampilan menu *Login* terdapat tiga *button* yaitu *Login admin*, dan *Login pengguna*. Pada tampilan kedua terdapat dua kolom dan dua *button*, yaitu *username* dan *password*, *button Login* dan *batal*. Ketika *user* menekan salah satu *button* pada menu *Login*, *user* akan masuk ke tampilan selanjutnya yaitu tampilan yang berisi *Form username* dan *password* kemudian isi *username* dan *password* pada kolom tersebut lalu tekan salah satu *button Login* atau *batal* untuk melanjutkan ke *Form* selanjutnya.

The image shows three sequential screenshots of an admin interface. The first screenshot, titled 'Menu Admin', displays three buttons: 'RUMAH SAKIT', 'DATA PENGGUNA', and 'Bantuan'. The second screenshot, titled 'RUMAH SAKIT', shows a form with four input fields: 'Nama Rumah Sakit', 'Alamat Rumah Sakit', 'Plat Ambulance', and 'No Hp'. Below these fields are three buttons: 'Simpan', 'Ubah', and 'Hapus'. The third screenshot, titled 'DATA PENGGUNA', shows a form with five input fields: 'Nama Pengguna', 'Alamat', 'Nik Ktp', 'No Hp', and 'Password'. Below these fields are three buttons: 'Simpan', 'Ubah', and 'Hapus'.

Gambar III.15 Form Data Admin

Pada *Form* data admin terdapat tampilan menu admin yang berisikan data rumah sakit dan data pengguna. Ketika admin menekan *button* data rumah sakit maka akan masuk ke *Form* selanjutnya yang berisikan beberapa kolom seperti nama rumah sakit, alamat rumah sakit, plat ambulan, dan no telepon rumah sakit yang nantinya akan diisi untuk kelengkapan data. Di dalam *Form* ini juga terdapat tiga *button*, yaitu *button* simpan, ubah, dan hapus. *Button* simpan digunakan saat data telah diisi kemudian akan menyimpan data tersebut, *button* ubah berfungsi untuk mengubah data yang telah ada sebelumnya untuk diperbarui, dan *button* hapus digunakan untuk menghapus data yang salah atau data yang tidak digunakan lagi. Pada tampilan ini hanya admin yang dapat mengakses dan mengolah data, dikarenakan pendaftaran aplikasinya secara *offline* maka tidak sembarang orang dapat membuka *Form* data admin tersebut.




Gambar III.16 Menu Pengguna

Tampilan menu pengguna yaitu tampilan yang hanya akan muncul pada layar hp pengguna. Di dalam *Form* menu pengguna ini terdapat tiga *button*, yaitu *button* rumah sakit yang didalamnya berisikan data-data rumah sakit, *button* pesan ambulance yang ketika *user* menekan *button* tersebut maka akan langsung muncul pesanan ambulance yang letaknya dekat dengan lokasi kita berada, yang ketiga *button* bantuan, di dalam *button* bantuan terdapat *inForm* asi bagaimana sistem berjalannya aplikasi tersebut dan cara penggunaannya.

DATA RUMAH SAKIT
Tampilan Map
Lokasi Anda

Gambar III.17 *Form Menu Rumah Sakit*

Tampilan *map* rumah sakit yaitu tampilan map yang menunjukkan lokasi rumah sakit dan lokasi pengguna. Tampilan *map* akan muncul ketika pengguna menekan tombol rumah sakit yang kemudian akan muncul dimana saja letak rumah sakit yang ada di kota Medan.



Pesan Ambulance

Plat Ambulance :

Rumah Sakit :

Telp

Gambar III.18 *Form* Menu Pesan Ambulan

Pada *Form* menu pesan ambulan ini akan muncul ketika pengguna menekan tombol pesan ambulan pada *Form* sebelumnya. Di dalam *Form* pesan ambulan ini terdapat data ambulan yaitu plat no ambulan, dan nama rumah sakitnya. Di dalam *Form* ini terdapat tiga *fitur* yaitu, telpon dan *navigasi*. Ketika pengguna ingin menelepon Rumah Sakit pengguna hanya perlu menekan *button* telp maka akan otomatis menelepon Rumah Sakit. Jika pengguna menekan *navigasi* maka akan muncul tampilan *map* ambulan yang sedang menuju lokasi.

Bantuan

Gambar III.19 Form Menu Bantuan

Pada *Form* Menu Bantuan terdapat in*Form* asi bagaimana sistem berjalannya aplikasi ini dan bagaimana cara penggunaanya.

III.5 Desain Tabel

Perancangan desain tabel yang penulis buat yakni langkah selanjutnya dalam pembuatan sebuah aplikasi pemesanan ambulan. Struktur tabel pemesanan ambulan ini sendiri sebagai berikut :

1. Struktur Tabel *Login* Admin

Struktur tabel *Login* admin yakni digunakan untuk menyimpan data yang digunakan untuk *Login* seorang admin untuk masuk ke pengolahan data pengguna dan data rumah sakit. Berikut ini struktur tabel *Login* admin

:

Tabel III.7 Tabel *Login Admin*

	Nama Database	Ambulan		
	Nama Tabel	Log Admin		
No	Nama Field	Tipe Data	Boleh Kosong	Kunci
1.	<i>Id</i>	varchar(20)	Tidak	<i>Primary Key</i>
2.	Nama	varchar(20)	Tidak	-
3	<i>Password</i>	varchar(20)	Tidak	-

2. Struktur Tabel Data Rumah sakit

Struktur tabel data rumah sakit merupakan data *inForm* asi yang berkaitan dengan rumah sakit, Berikut ini struktur data rumah sakit :

Tabel III.8 Tabel Data Rumah Sakit

	Nama Database	Ambulan		
	Nama Tabel	DataRumahSakit		
No	Nama Field	Tipe Data	Boleh Kosong	Kunci
1.	Nama Rumah Sakit	varchar(20)	Tidak	<i>Primary Key</i>
2.	Alamat	varchar(20)	Tidak	-
3.	Plat Ambulan	varchar(20)	Tidak	-
4.	No Rumah Sakit	varchar(20)	Tidak	-

3. Struktur Tabel Data Pengguna

Struktur tabel data pengguna merupakan data inForm asi pengguna aplikasi pemesanan ambulance, adapun *field* nama pengguna akan dijadikan sebagai *username* untuk *Login* sebagai pengguna pada aplikasi.

Tabel III.9 Tabel Data Pengguna

	Nama Database	Ambulan		
	Nama Tabel	DataPengguna		
No	Nama Field	Tipe Data	Boleh Kosong	Kunci
1.	NIK KTP	varchar(20)	Tidak	<i>Primary Key</i>
2.	Nama Pengguna	varchar(20)	Tidak	-
3.	Alamat	varchar(20)	Tidak	-
4.	<i>Password</i>	varchar(20)	Tidak	-
5.	No Hp	varchar(20)	Tidak	-

4. Struktur Tabel Data Rute

Struktur tabel data rute merupakan data inForm asi rute jalan menuju rumah sakit dari beberapa node yang dibuat. Data ini akan digunakan untuk menampilkan rute jalan kerumah sakit yang tercepat dengan bantuan algoritma djikstra.

Tabel III.10 Tabel Data Rute

	Nama Database	Ambulan		
	Nama Tabel	Graph		
No	Nama Field	Tipe Data	Boleh Kosong	Kunci
1.	Id	integer(20)	Tidak	<i>Primary Key</i>
2.	Simpul_awal	integer (20)	Tidak	-
3.	Simpul_tujuan	integer (20)	Tidak	-
4.	Jalur	text(20)	Tidak	-
5.	Bobot	double(20)	Tidak	-

