

BAB III

ANALISIS DAN DESAIN SISTEM

III.1. Analisis Masalah

Tahap analisis sistem yang berjalan ini bertujuan untuk mencari informasi mengenai sistem yang ada guna mendapatkan bahan evaluasi untuk pengembangan pada sistem yang akan dirancang, evaluasi pada sistem yang ada adalah :

1. Masyarakat masih sulit untuk menemukan informasi dan lokasi apotek di Kota Medan.
2. Belum ada sistem informasi pencarian lokasi apotek di Kota Medan dengan memanfaatkan sistem informasi geografis dengan metode *haversine formula*.
3. Minimnya sistem informasi geografis yang menggunakan aplikasi *MapTiles*, karena aplikasi tersebut masih tergolong baru.

III.1.1 Strategi Penyelesaian Masalah

Adapun strategi pemecahan masalah yang dianjurkan oleh penulis untuk memperbaiki kesalahan sistem yang sedang berjalan adalah membangun sistem informasi geografis lokasi apotek di Kota Medan yang bertujuan untuk memudahkan masyarakat untuk mencari informasi lokasi apotek di Kota Medan dengan cepat akurat yang dapat di akses oleh siapa saja, dimana saja, kapan saja tanpa mengenal jarak dan waktu termasuk juga pada sales PT. Parit Padang global untuk menyalurkan obat-obatannya ke apotek-apotek tersebut.

Adapun beberapa tabel data apotek yang bekerja sama dengan PT. Parit Padang Global untuk memudahkan sales mencari lokasi terdekat yang ada di Kota Medan antara lain :

Tabel III.1 PT.PARIT PADANG GLOBAL



No	Nama Apotek	Alamat Apotek
1	Apotek Anugrah	Jalan Sei, Rahayu
2	Apotek Guardian	Jalan Gagak Hitam No.28, Medan Sunggal
3	Apotek Kimia Farma	Jalan Setia Budi No.102, Tanjung Rejo
4	Apotek Pandawa	Jalan Pancasila
5	Apotek Utama Farma	Jalan Rajawali

Apoteker 2022

Strategi dalam membangun sistem informasi geografis lokasi apotek di Kota Medan, penulis melakukan analisis kebutuhan atau pengumpulan data untuk keperluan pembangunan sistem, seperti mengumpulkan informasi yang dibutuhkan dan juga beberapa sample foto lokasi. Pada sistem ini juga penulis melakukan desain sistem dengan membangun sistem menggunakan bahasa pemrograman *PHP* dan *MySQL* sebagai *database*-nya juga *dreamweaver* sebagai aplikasinya. Setelah tahap-tahap pembangunan diatas penulis lakukan, selanjutnya penulis melakukan pengujian pada sistem yang telah dibangun, sehingga dapat diketahui kesalahan dari sistem tersebut sebelum di terapkan.

III.2. Penerapan Metode *Haversine Formula*

Haversine formula adalah persamaan yang penting dalam navigasi, formulasi rumus yang digunakan untuk menghitung jarak lingkaran yang jauh antara dua titik pada permukaan bola bumi berdasarkan bujur dan lintang. Penggunaan rumus ini mengasumsikan pengabaian efek ellipsoidal, juga pengabaian ketinggian bukit dan kedalaman lembah di permukaan bumi.

$$\Delta\text{lat} = \text{lat2} - \text{lat1}$$

$$\Delta\text{long} = \text{long2} - \text{long1}$$

$$a = \sin^2(\Delta\text{lat}/2) + \cos(\text{lat1}) \cdot \cos(\text{lat2}) \cdot \sin^2(\Delta\text{long}/2)$$

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

$$d = R \cdot c$$

Dimana:

Δlong : besaran perubahan longitude

Δlat : besaran perubahan latitude

R : jari-jari bumi sebesar 6371,1(km)

c : kalkulasi perpotongan sumbu

d : jarak (km)

1 derajat = 0.017453293

III.2.1. Studi Kasus

Penerapan metode *Haversine Formula*

Koordinat Rute

Titik Awal yaitu koordinat titik dimana lokasi user saat itu, sebagai contoh disini Jalan Jawa adalah sebagai titik awal, sementara titik tujuan yaitu koordinat dimana lokasi apotek dinaungi oleh PT. Parit Padang Global, sebagai contoh disini titik tujuan berada pada Jalan Krakatau Ujung. Berikut adalah perhitungan Haversine dengan contoh lokasi awal dan tujuan di atas.

Titik Awal (Lat1= 3.63767, Long1 = 98.6664, dlat = lat2-lat1(0.052907))

Titik Tujuan (Lat2= 3.59263, Long2 = 98.6633, dlong = long2-long1(0.0031))

Konversi ke dalam bentuk radian

$$\text{lat1} = 0.017453293 * 3.63767 = 0.063489319 \text{ radian}$$

$$\text{lat2} = 0.017453293 * 3.59263 = 0.062703224 \text{ radian}$$

$$\text{long1} = 0.017453293 * 98.6664 = 1.722053541 \text{ radian}$$

$$\text{long2} = 0.017453293 * 98.6633 = 1.721999483 \text{ radian}$$

$$\text{dlat} = 0.017453293 * (3.59263 - 3.63767)$$

$$= 0.017453293 * (-0.04504) = 0.0007860$$

$$\text{dlong} = 0.017453293 * (98.6633 - 98.6664)$$

$$= 0.017453293 * (-0.0031) = 0.000054105$$

Rumus Haversine :

$$a = \sin^2(\text{dlat}/2) + \cos(\text{lat1}) \cdot \cos(\text{lat2}) \cdot \sin^2(\text{dlong}/2)$$

$$c = 2 * \arctan2(\sqrt{a}, \sqrt{1-a}) = 2 * \arcsin(\sqrt{a})$$

$$d = R * c$$

Penjabarannya :

$$A = \sin^2(\text{dlat}/2) + \cos(\text{lat1}) \cdot \cos(\text{lat2}) \cdot \sin^2(\text{dlong}/2)$$

$$= \sin(0.0007860/2)^2 + \cos(0.063489319) * \cos(0.062703224) * \sin(0.000054105/2)^2$$

$$= 0.000393926$$

$$C = 2 * \arctan2(\sqrt{a}, \sqrt{1-a}) = 2 * \arcsin(\sqrt{a}) = 2 * \arcsin(\sqrt{0.000393926})$$

$$= 0.000787852$$

$$D = R * c$$

$$= 6371.1 * (0.000787852)$$

$$= \text{Jarak antara titik awal dengan tujuan adalah} = 5.019405092 \text{ Km}$$

III.3. Desain Sistem

Desain sistem secara global menggunakan bahasa pemodelan UML yang terdiri dari *Usecase Diagram*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram*.

III.3.1. Normalisasi Data

Salah satu topik yang cukup kompleks dalam dunia manajemen *database* adalah proses untuk menormalisasi tabel-tabel dalam *database relasional*. Menghindari kemungkinan kehilangan data secara tidak disengaja/tidak diketahui. Alasan utama dari normalisasi *database* minimal sampai dengan bentuk normal ketiga adalah menghilangkan kemungkinan adanya “*insertion anomalies*”, “*deletion anomalies*”, dan “*update anomalies*”. Tipe-tipe kesalahan tersebut sangat mungkin terjadi pada *database* yang tidak normal.

1. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

Tabel III.2. Tabel Tidak Normal

Kode Tempat	Nama	Alamat	Deskripsi
T001	Lokasi A	Jl. Pelabuhan Raya	-
T002	Lokasi B	Jl. Nauli	-

2. Bentuk normal tahap pertama (1st Normal Form)

Sebuah table disebut 1NF jika :

- Tidak ada baris yang duplikat dalam tabel tersebut.
- Masing-masing *cell* bernilai tunggal

Catatan: Permintaan yang menyatakan tidak ada baris yang duplikat dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci, meskipun kunci tersebut dibuat dari kombinasi lebih dari satu kolom atau bahkan kunci tersebut merupakan kombinasi dari semua kolom.

Tabel III.3. Tabel Lokasi Normal 1NF

Kode Tempat	Nama Apotek	Alamat	Deskripsi	Lon	Lat
T001	Lokasi A	Jl. Pelabuhan Raya	-	10.983.985 .717.978	3.995.340.828.79 7.300
T002	Lokasi B	Jl. Medan Nauli	-	10.983.985 .756.823	3.995.340.825.28 0.360

3. Bentuk normal tahap kedua (2nd normal form)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam *primary key* memiliki ketergantungan fungsional pada *primary key* secara utuh.

Tabel III.4. Tabel Admin 2NF

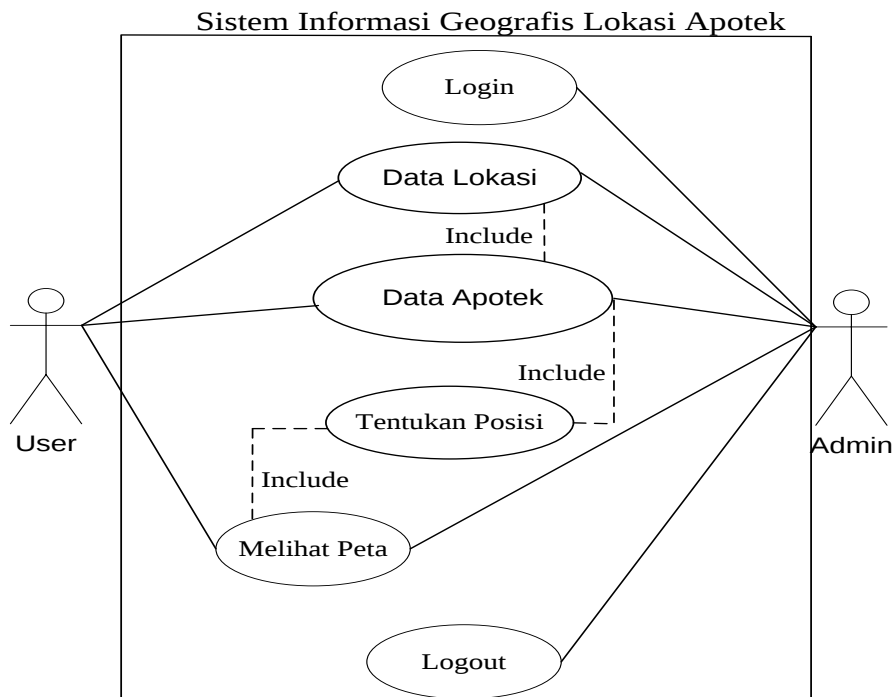
<u>id_administrator</u>	<u>Nama</u>	<u>Username</u>	<u>password</u>
1	Luky	Admin	Admin

Tabel III.5. Tabel Apotek 2NF

<u>Idapotek</u>	<u>Nama</u>	<u>Deskripsi</u>	<u>Foto</u>
1	Apotek A	-	http://localhost/apotek/gambar/xx....
2	Apotek B	-	http://localhost/apotek/gambar/xx....

III.3.2. Usecase Diagram

Usecase diagram merupakan model diagram UML (*Unified Modelling Language*) yang digunakan untuk menggambarkan *requirement* fungsional yang diharapkan dari sebuah sistem, proses sistem yang akan dirancang digambarkan dengan *usecase* diagram yang terdapat pada Gambar III.1 :



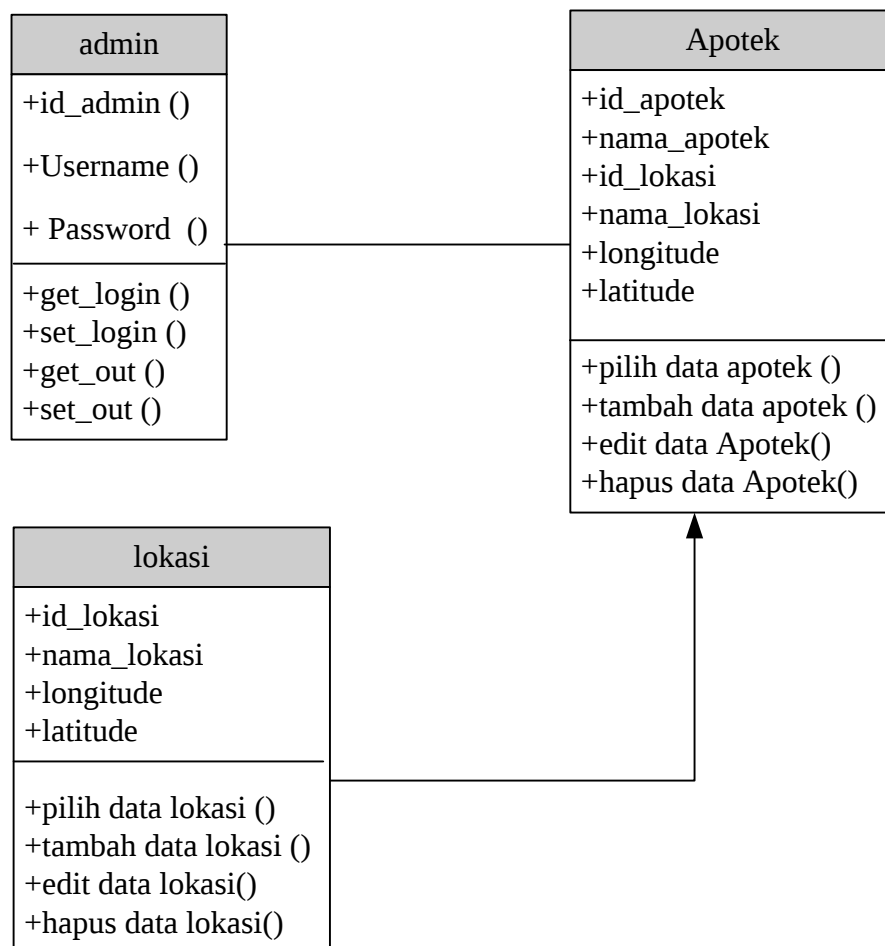
Gambar III.1.Usecase Diagram Sistem

Dari *usecase* diagram yang didapat keterangan bahwa pada sistem yang akan dibangun nantinya terdapat dua aktor yaitu *user* dan *admin*, aktor *user* disini adalah masyarakat dan juga sales yang mengantarkan obat-obatan dari PT. Parit Padang Global ke apotek-apotek yang menjalin kerja sama yang terletak di Kota Medan. Aktor *user* hanya dapat mengakses *case* data lokasi, data apotek dan juga melihat peta, pada *case* tersebut aktor *user* tidak dapat memanipulasi data melainkan hanya melihat *case* tersebut saja.

Sedangkan pada aktor *admin* dapat mengakses keseluruhan *case* yang ada pada sistem yaitu *case login*, data lokasi, data apotek, tentukan posisi dan melihat peta. Aktor *admin* dalam menggunakan hak aksesnya harus melalui prosedur *case login* terlebih dahulu karena aktor *admin* dapat memanipulasi data yang ada pada sistem, sehingga data-data yang ada dalam sistem dapat selalu di *update*.

III.3.3. Class Diagram

Rancangan kelas-kelas yang akan digunakan pada sistem yang akan dirancang dapat dilihat pada gambar III.2 berikut :



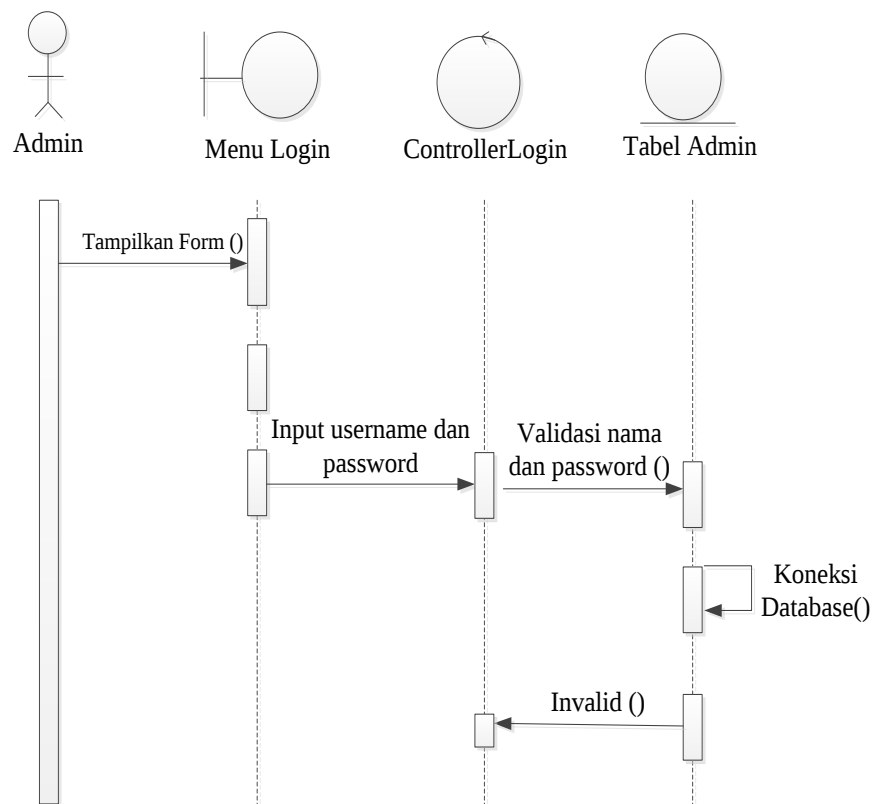
Gambar III.2. Class Diagram Sistem

III.3.4. Sequence Diagram

Rangkaian kegiatan pada setiap terjadi *event* sistem digambarkan pada *sequence diagram* berikut :

1. Sequence Diagram Data Login

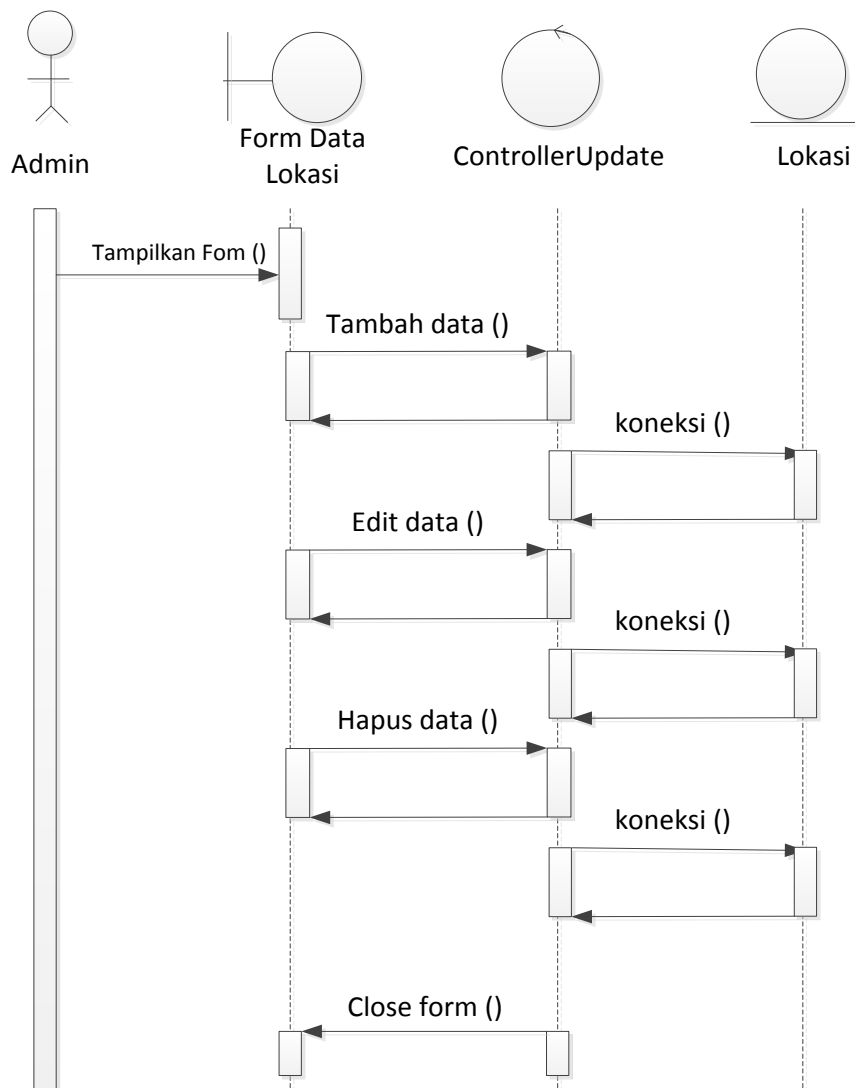
Serangkaian kerja melihat Login dapat terlihat seperti pada gambar III.3 berikut :



Gambar III.3. Sequence Diagram Data Login

2. Sequence Diagram Data Lokasi

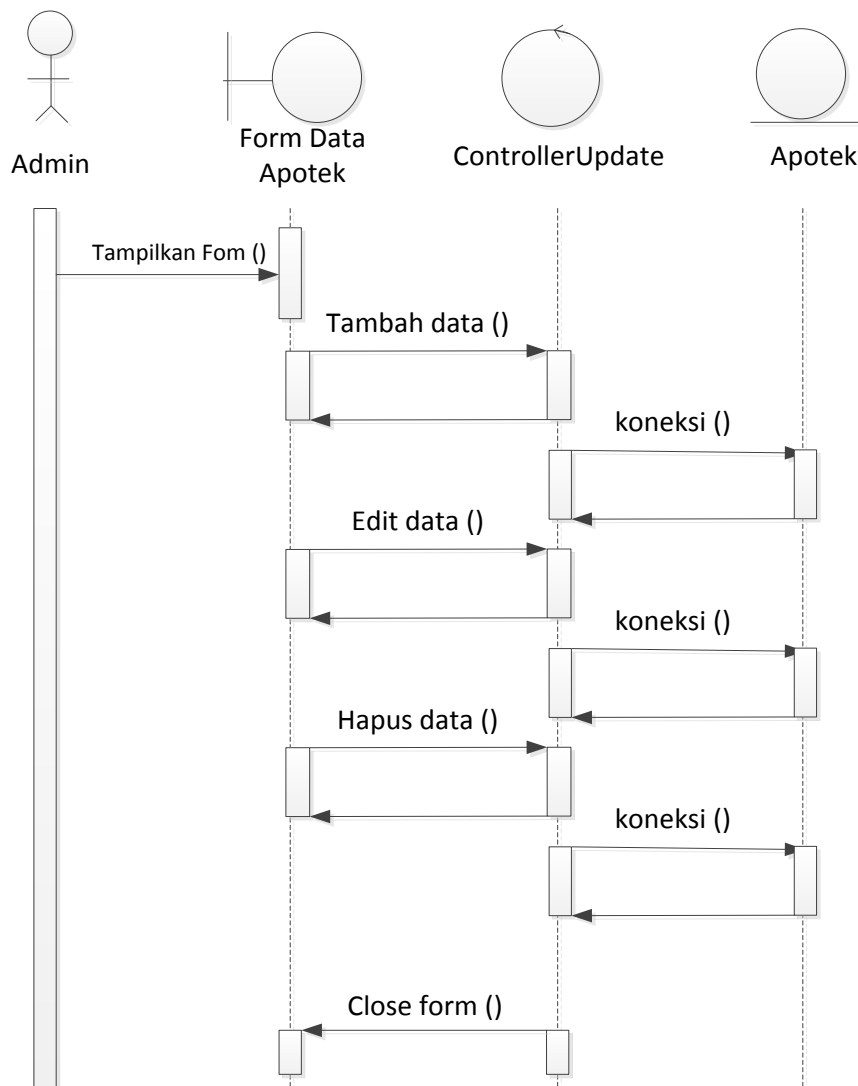
Serangkaian kerja melakukan pengolahan data Lokasi dapat terlihat seperti pada gambar III.4 berikut :



Gambar III.4. Sequence Diagram Data Lokasi

3. Sequence Diagram Data Apotek

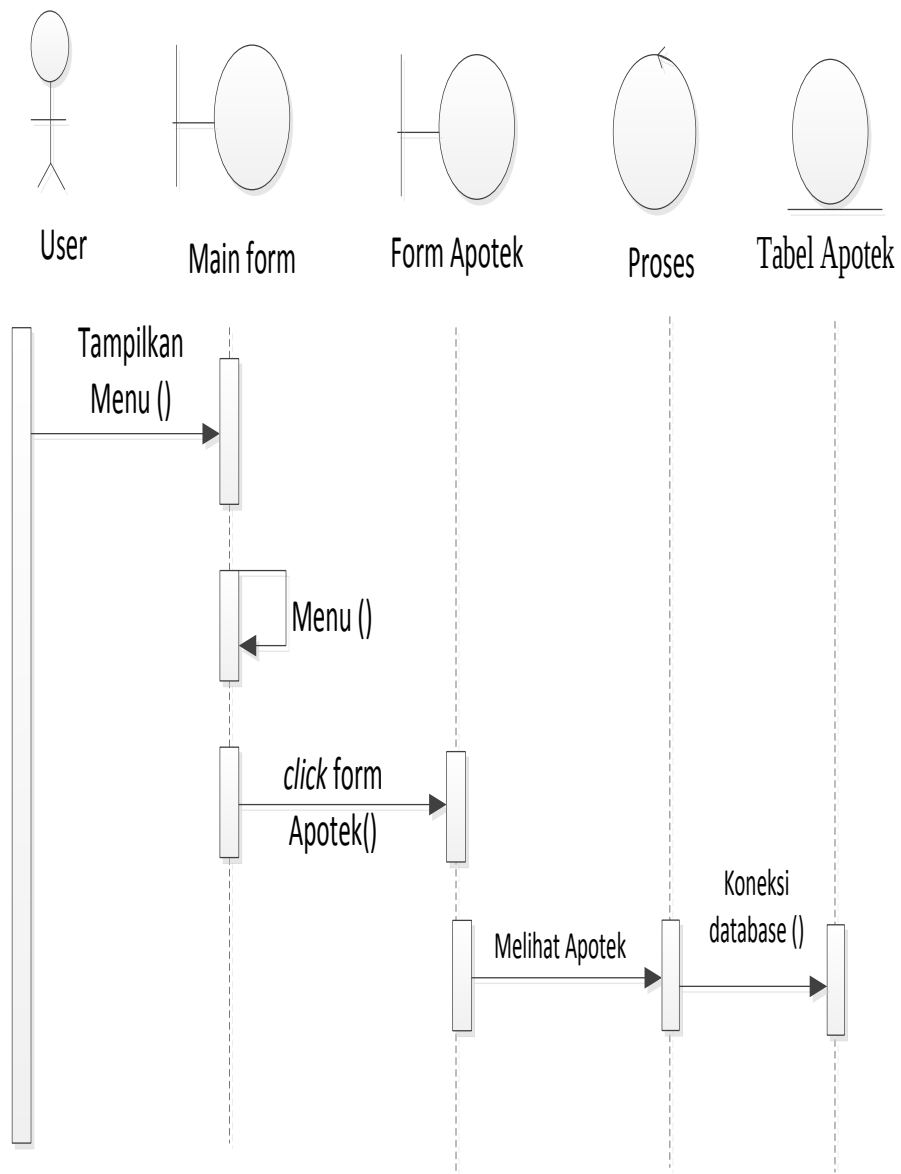
Serangkaian kerja melakukan pengolahan data Apotek dapat terlihat seperti pada gambar III.5 berikut :



Gambar III.5. Sequence Diagram Apotek

4. Sequence Diagram Lihat Data Apotek

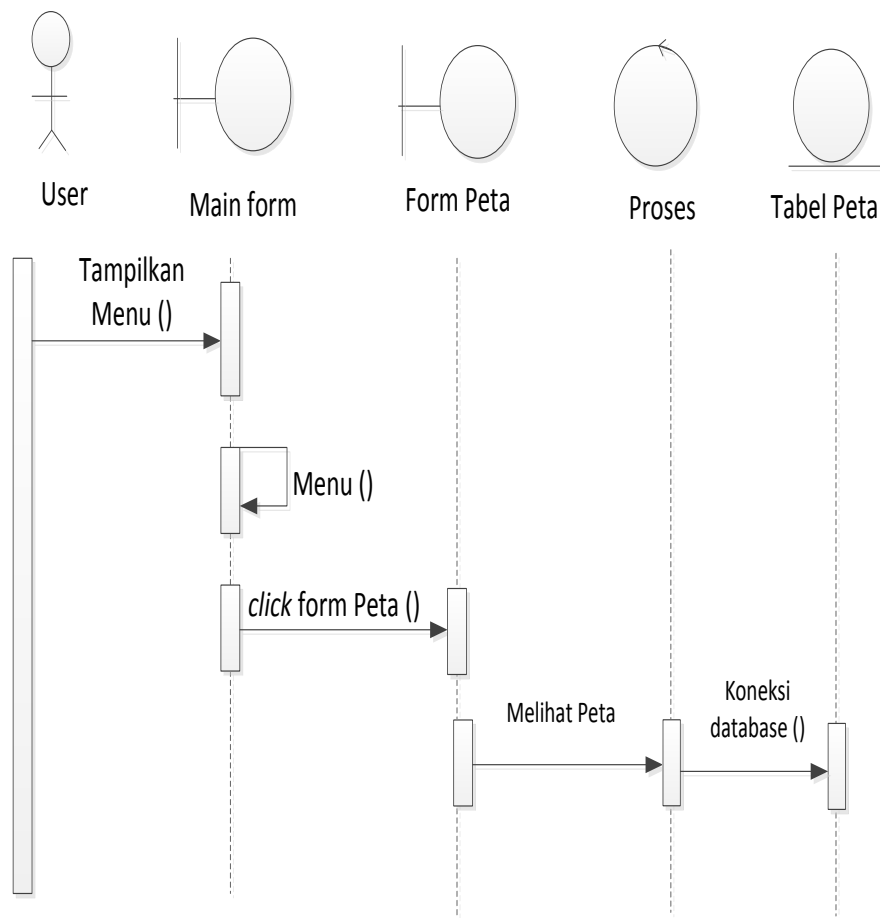
Serangkaian kerja melihat data Apotek dapat terlihat seperti pada gambar III.6 berikut:



Gambar III.6. Sequence Diagram Lihat Data Apotek

6. Sequence Diagram Lihat Peta

Serangkaian kerja melihat peta dapat terlihat seperti pada gambar III.7 berikut:



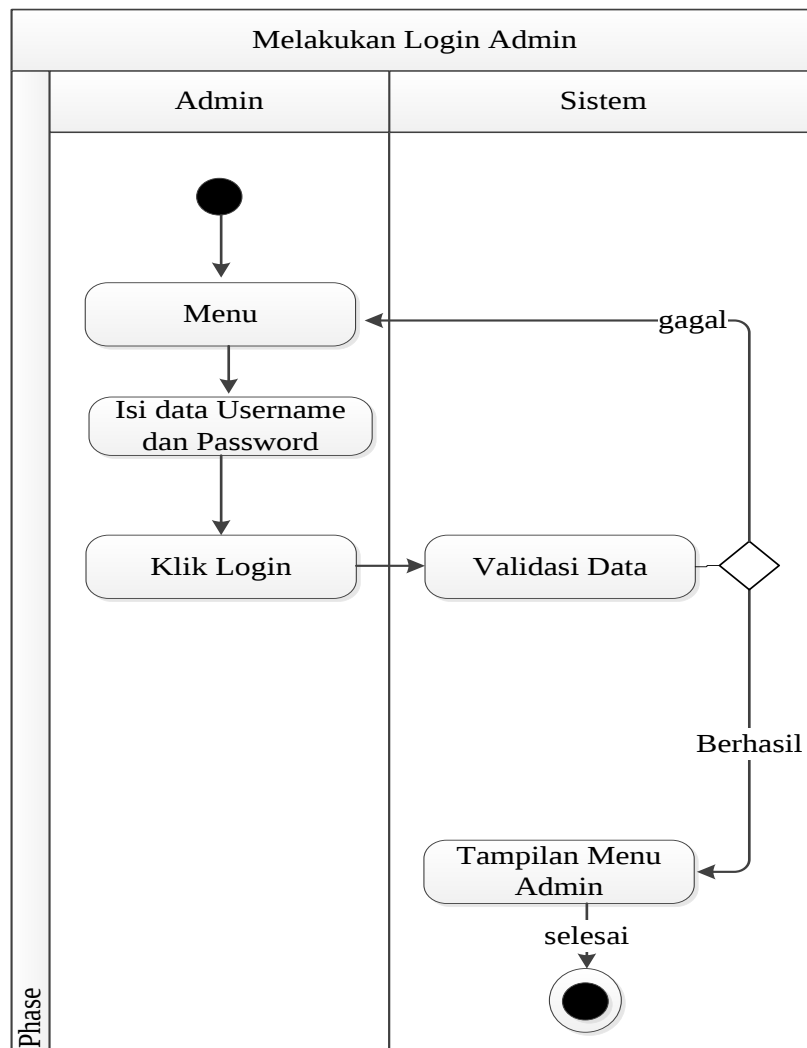
Gambar III.7. Sequence Diagram Lihat Data Peta

III.3.4. Activity Diagram

Bisnis proses yang telah digambarkan pada *use case* diagram di atas dijabarkan dengan Activity Diagram.

1. Activity Diagram Login Admin

Aktivitas proses *login* admin diterangkan dalam langkah-langkah *state*, dimulai dari memasukkan *username*, memasukkan *password*, jika profil *valid* maka sistem akan mengaktifkan menu admin, sedangkan jika tidak *valid*, maka tampilkan pesan kesalahan yang ditunjukkan pada gambar III.8 berikut :

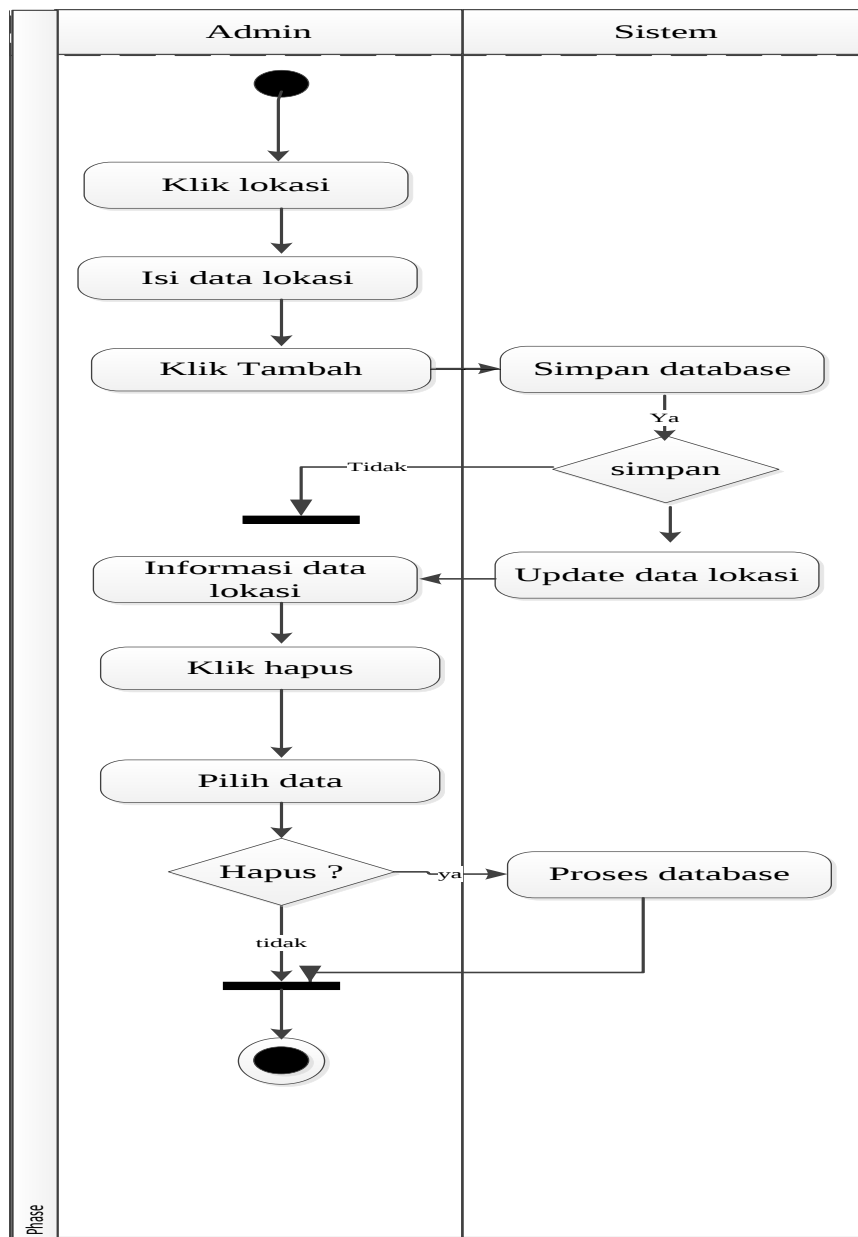


Gambar III.8. Activity Diagram Login Admin

2. Activity Diagram Mengolah Data Lokasi

diterangkan dalam langkah-langkah *state* yang ditunjukkan pada gambar III.9

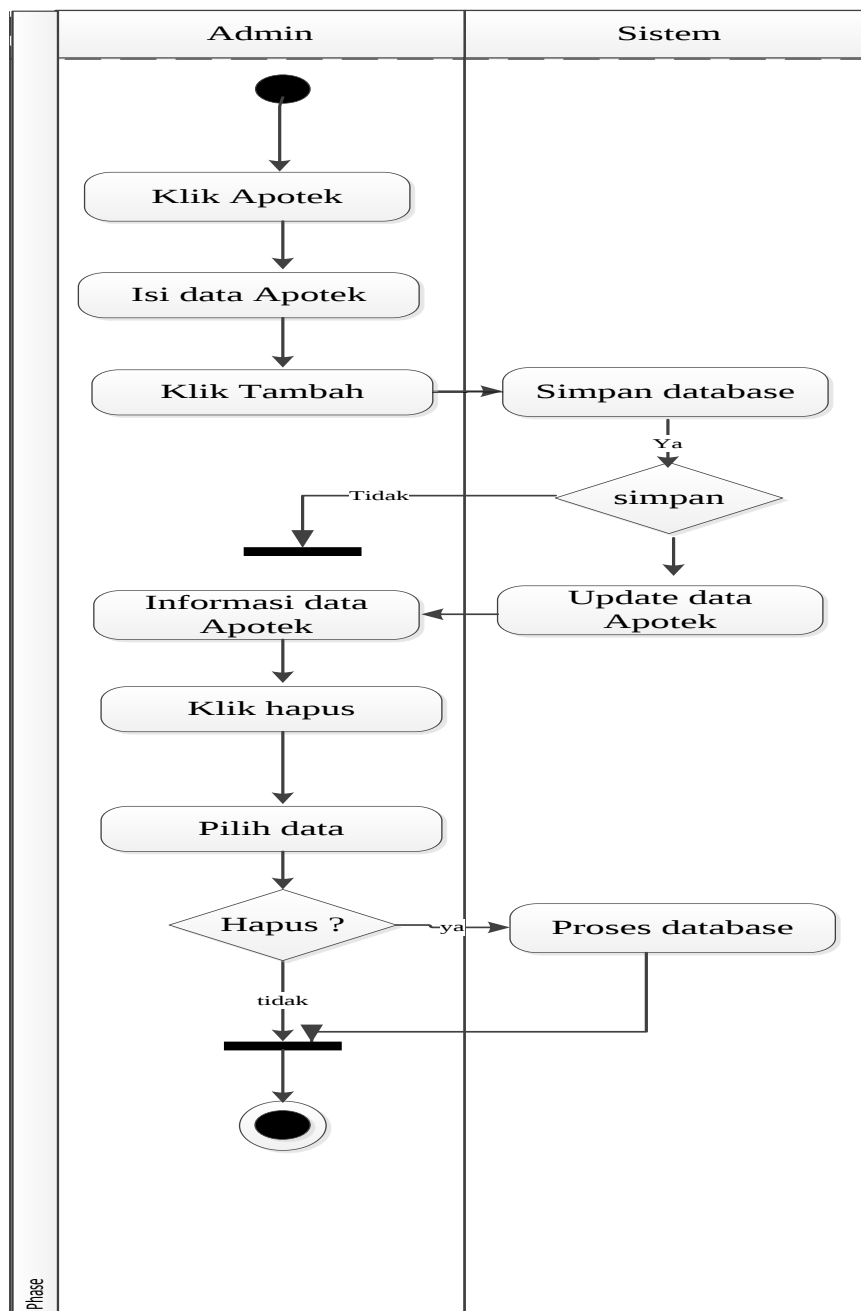
berikut :



Gambar III.9. Activity Diagram Lokasi

3. Activity Diagram Apotek

Aktivitas proses mengolah data Apotek diterangkan dalam langkah-langkah *state* yang ditunjukkan pada gambar III.10 berikut :

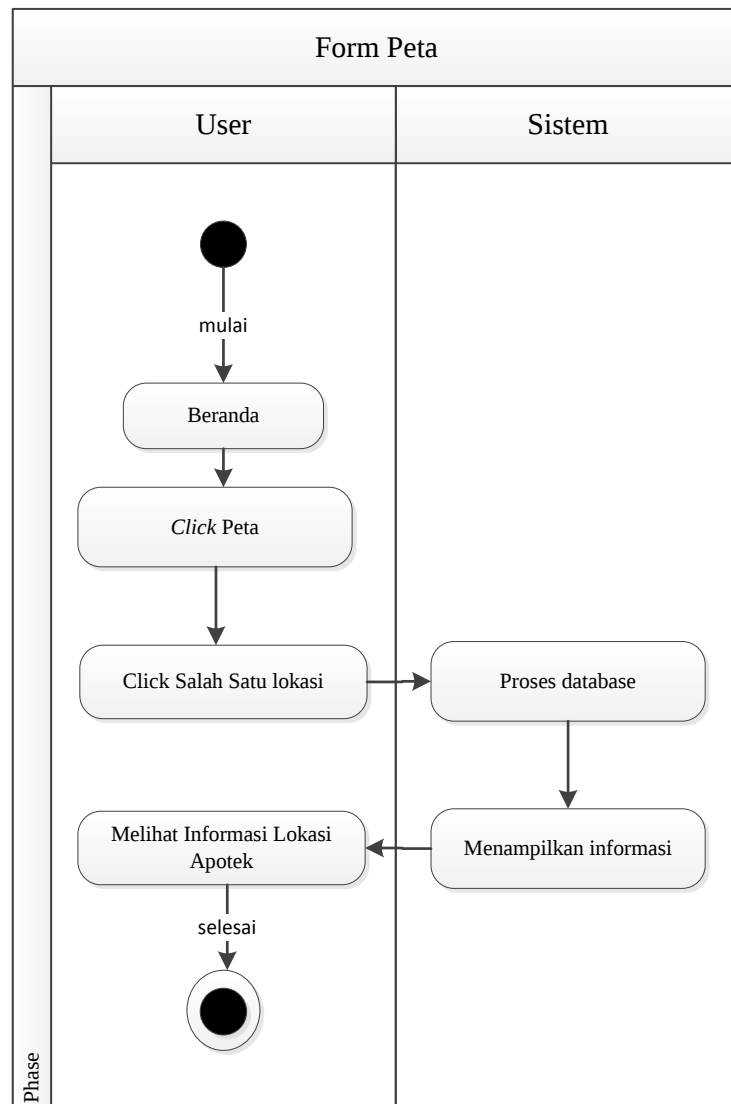


Gambar III.10. Activity Diagram Apotek

4. Activity Diagram Melihat Peta

Aktivitas melihat peta diterangkan dalam langkah-langkah *state*, dimulai dari kegiatan melihat panel peta kemudian mencari artikel peta, selanjutnya menekan

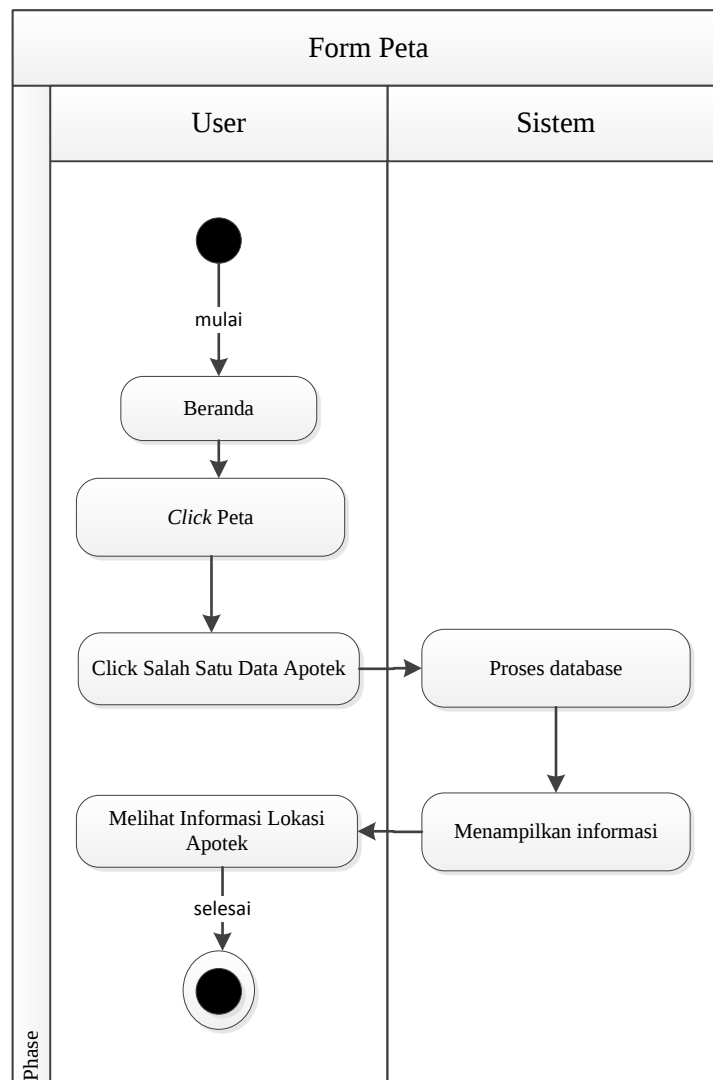
tombol atau *link* yang ada pada peta dan yang terakhir melihat informasi yang disajikan dalam peta yang ditunjukkan pada gambar III.11:



Gambar III.11. User Activity Diagram Peta

5. Activity Diagram Melihat Data Apotek

Aktivitas melihat peta diterangkan dalam langkah-langkah *state*, dimulai dari kegiatan melihat panel peta kemudian mencari Apotek, selanjutnya menekan tombol atau *link* data Apotek dan yang terakhir melihat informasi yang ditunjukkan pada gambar III.12:



Gambar III.12. User Activity Diagram Apotek

III.3.6. Desain Tabel

Merancang struktur tabel pada basis data sistem yang akan dibuat, berikut ini merupakan rancangan struktur tabel tersebut.

1. Struktur Tabel Admin

Tabel admin digunakan untuk menyimpan data *id_admin*, *nama*, *username*, *password*, selengkapnya mengenai struktur table ini dapat dilihat pada tabel III.6 di bawah ini :

Tabel III.6. Rancangan Tabel Admin

Nama <i>Database</i>		DbApotek		
Nama Tabel		Admin		
No	Nama Field	Tipe Data	Boleh Kosong	Kunci
1.	<i>id_admin</i>	int(10)	Tidak	<i>Primary Key</i>
2.	<i>Nama</i>	varchar(30)	Tidak	-
3.	<i>Username</i>	varchar(15)	Tidak	-
4.	<i>Password</i>	varchar(15)	Tidak	-

2. Struktur Tabel Lokasi

Tabel lokasi digunakan untuk menyimpan data *id_lokasi*, *Nama_lokasi*, *Longitude*, *Longitude*, selengkapnya mengenai struktur table ini dapat dilihat pada tabel III.7 di bawah ini :

Tabel III.7. Rancangan Tabel Lokasi

Nama <i>Database</i>		DbApotek		
Nama Tabel		Lokasi		
No	Nama Field	Tipe Data	Boleh Kosong	Kunci
1.	<i>id_Lokasi</i>	int(11)	Tidak	<i>Primary Key</i>
2.	<i>Nama_lokasi</i>	varchar(50)	Tidak	-
3.	<i>Longitude</i>	decimal(10,8)	Tidak	-
4.	<i>Latitude</i>	decimal(10,8)	Tidak	-

3. Struktur Tabel Apotek

Tabel Apotek digunakan untuk menyimpan data *id_apotek*, *Nama_apotek*, *Id_lokasi*, *Nama_lokasi*, *longitude*, *latitude*, selengkapnya mengenai struktur table ini dapat dilihat pada tabel III.8 di bawah ini :

Tabel III.8.Rancangan Tabel Apotek

Nama <i>Database</i>		DbApotek		
Nama Tabel		Apotek		
No	Nama Field	Tipe Data	Boleh Kosong	Kunci
1.	<i>id_apotek</i>	<i>int(11)</i>	Tidak	<i>Primary Key</i>
2.	<i>Nama_apotek</i>	<i>varchar(50)</i>	Tidak	-
3.	<i>Id_lokasi</i>	<i>Text</i>	Tidak	<i>Foreign Key</i>
4.	<i>Nama_lokasi</i>	<i>varchar(50)</i>	Boleh	-
6.	<i>Longitude</i>	<i>decimal(10,8)</i>	Boleh	-
7.	<i>Latitude</i>	<i>decimal(10,8)</i>	Boleh	-

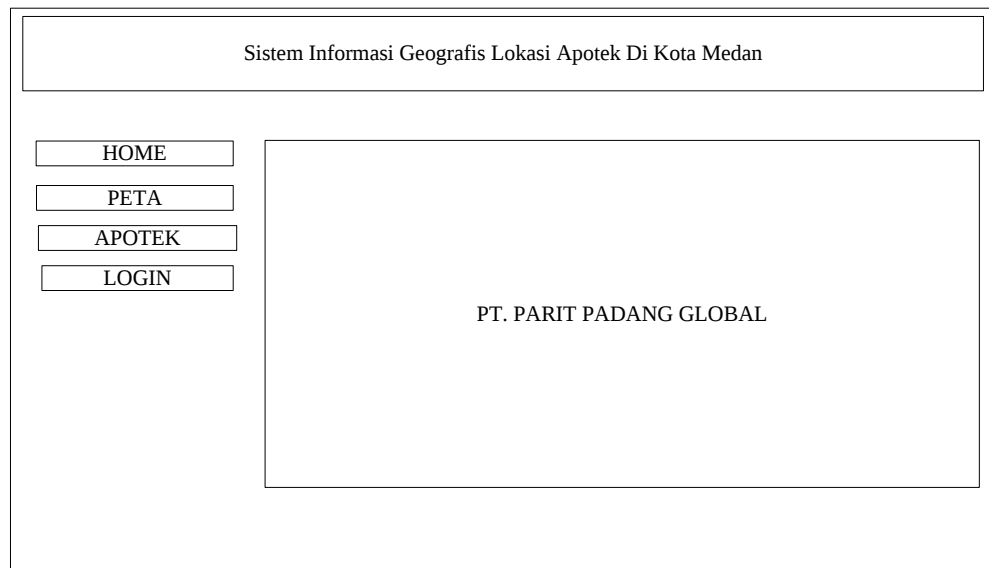
III.4. Desain User Interface

Tahap perancangan berikutnya yaitu desain sistem secara detail yang meliputi desain *output* sistem dan desain *input* sistem.

III.4.1. Desain Output

1. Tampilan *Form Home* Atau Awal

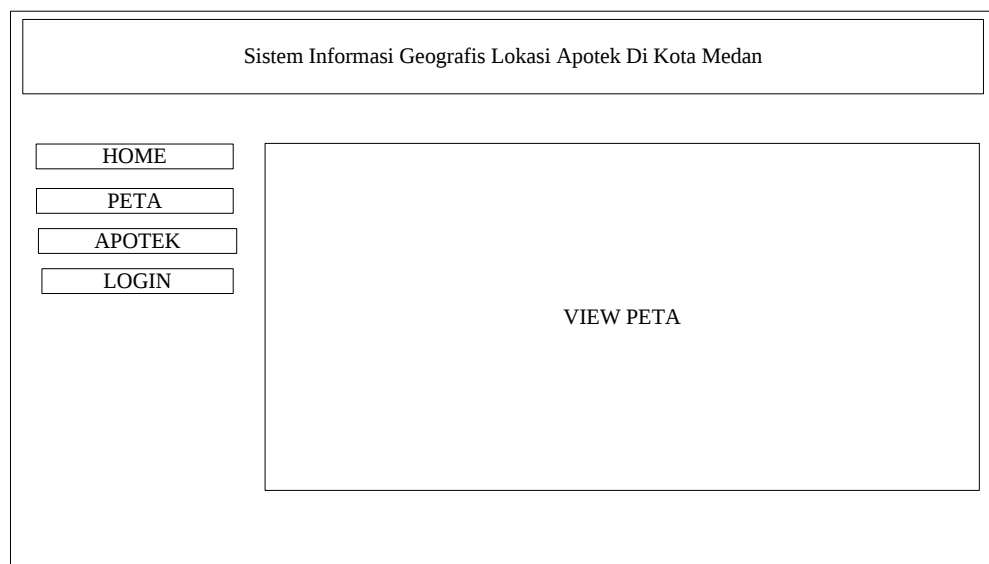
Berikut ini adalah rancangan tampilan desain awal atau home akan dihasilkan oleh sistem, *form* home ini berfungsi untuk menampilkan tampilan awal program yang dilihat oleh user :



Gambar III.13. Desain Tampilan Awal atau Home

2. Tampilan *Form* Peta

Berikut ini adalah rancangan tampilan desain Peta akan dihasilkan oleh sistem, *form* peta ini berfungsi untuk menampilkan tampilan peta yang dilihat oleh user :



Gambar III.14. Desain Tampilan PETA

3. Tampilan *Form* Data Apotek

Berikut ini adalah rancangan tampilan data Apotek yang akan dihasilkan oleh sistem, *form* data Apotek ini berfungsi untuk menampilkan tampilan data Apotek yang dilihat oleh user :

SISTEM INFORMASI GEOGRAFIS LOKASI APOTEK DI KOTA MEDAN					
HOME	SISTEM INFORMASI GEOGRAFIS LOKASI APOTEK				
PETA	DATA APOTEK				
APOTEK	ID APOTEK	NAMA APOTEK	ALAMAT	LONGITUDE	LATITUDE
LOGIN					

Gambar III.15. Desain Tampilan Data Apotek

III.4.2. Desain *Input*

Berikut ini adalah rancangan atau desain *input* sebagai antarmuka pengguna:

1. Desain *Form Login*

Desain *form login* berfungsi untuk mengaktifkan menu atau halaman *admin* yang dapat mengelola data yang berhubungan dengan data Apotek, *form* tersebut dapat dilihat pada gambar III.16:

SISTEM INFORMASI GEOGRAFIS LOKASI APOTEK DI KOTA MEDAN

HOME
PETA
APOTEK
LOGIN

SISTEM INFORMASI GEOGRAFIS LOKASI APOTEK

LOGIN

USERNAME

PASSWORD

LOGIN

Gambar III.16. Desain Form Login

2. Desain Form Tambah Lokasi

Desain *form* tambah lokasi berfungsi untuk menambahkan data lokasi pada peta yang di akses oleh admin setelah melakukan aktivitas login, *form* tersebut dapat dilihat pada gambar III.17:

SISTEM INFORMASI GEOGRAFIS LOKASI APOTEK DI KOTA MEDAN

HOME
PETA
APOTEK
LOGOUT

SISTEM INFORMASI GEOGRAFIS LOKASI APOTEK

TAMBAH DATA LOKASI

PETA

IDLOKASI

NAMA LOKASI

LONGITUDE

LATITUDE

TAMBAH

Gambar III.17. Desain Form Tambah Lokasi

3. Desain Form Tambah Data Apotek

Desain *form* tambah data Apotek berfungsi untuk menambahkan data Apotek yang lokasinya telah di tambahkan terlebih dahulu melalui tambah data

lokasi yang di akses oleh admin setelah melakukan aktivitas login, *form* tersebut dapat dilihat pada gambar III.18:

SISTEM INFORMASI GEOGRAFIS LOKASI APOTEK DI KOTA MEDAN		
SISTEM INFORMASI GEOGRAFIS LOKASI APOTEK		
HOME	TAMBAH DATA APOTEK	
PETA	ID APOTEK	
APOTEK	NAMA APOTEK	
LOGOUT	ID LOKASI	
	NAMA LOKASI	
	LONGITUDE	
	LATITUDE	
	TAMBAH	

Gambar III.18. Desain *Form* Tambah Data Apotek