

## **BAB III**

### **ANALISIS MASALAH DAN RANCANGAN PROGRAM**

#### **III.1. Analisis Masalah**

Proses analisa sistem merupakan langkah kedua pada pengembangan sistem. Analisa sistem dilakukan untuk memahami informasi-informasi yang didapat dan dikeluarkan oleh sistem itu sendiri. Sistem informasi geografis lokasi taxi belum begitu banyak diketahui oleh seorang pengguna (*user*), seorang pengguna ingin melakukan analisa terhadap lokasi tersebut. Karena dengan berkembangnya teknologi informasi secara otomatis akan menambah wawasan pribadi.

Untuk itu, sistem yang penulis rancang adalah sistem yang melakukan memberikan informasi tentang sistem informasi geografis media melalui media yang penulis lakukan, agar data lokasi taxi dapat diketahui atau dianalisa yang terjadi pada sistem informasi geografis tersebut. Dalam tahap pengembangan sistem SIG tempat lokasi tersebut, analisa sistem merupakan hal yang harus dilakukan sebelum proses perancangan sistem.

#### **III.2. Strategi Pemecahan Masalah**

Adapun strategi pemecahan masalah dari sistem informasi GIS lokasi taxi yang dirancang adalah sebagai berikut :

1. Data teori sistem informasi geografis yang dikemas melalui media, apabila data informasi tentang SIG tersebut dapat digunakan sebagai analisa untuk mengetahui lokasi yang ditujuh, oleh karena itu perlu

dibuat melalui media sistem informasi geografis berbasis web sebagai ilustrasinya.

2. Agar informasi tentang lokasi taxi yang mudah dipahami dan dimengerti setiap pengguna yang menggunakan, harus dilengkapi dengan lokasi tersebut.

### **III.3. Evaluasi Sistem Yang Berjalan**

Sistem informasi geografi saat ini hanya sebatas teori seperti pemberitahuan informasi geografis lokasi. Kelemahan dari sistem ini adalah data tentang informasi letak lokasi tidak akurat karena sistem informasi tersebut hanya sebatas simulasi teori.

Maka solusi yang penulis buat untuk mengatasi masalah tersebut adalah membuat suatu sistem informasi geografis tentang letak lokasi yang dilengkapi sesuai dengan analisa penulis di lapangan dan dilengkapi dengan sistem memasukan data ke dalam *database* di MySQL.

### **III.4. Analisa Kebutuhan *Hardware* Dan *Software***

Kebutuhan non fungsional menjabarkan apa-apa saja yang harus dimiliki oleh sistem agar dapat berjalan. Analisis kebutuhan non fungsional bertujuan untuk mengetahui sistem seperti apa yang cocok diterapkan, perangkat keras dan perangkat lunak apa saja yang dibutuhkan serta siapa saja pengguna yang akan menggunakan sistem ini.

1. Aspek Perangkat Keras

Perangkat keras adalah semua bagian fisik komputer dan dibedakan dengan data yang berada di dalamnya atau yang beroperasi di dalamnya, dan dibedakan dengan perangkat lunak yang menyediakan instruksi untuk perangkat keras dalam menyelesaikan tugasnya.

Adapun kebutuhan *hardware* untuk menciptakan aplikasi atau perangkat lunak keamanan data tersebut terdiri dari *Prosesor Intel Core I3, harddisk, Memory RAM 2 GB, Monitor 15, Keyboard dan Mouse, Internet.*

## 2. Aspek Perangkat Lunak (*software*)

Perangkat lunak adalah program yang digunakan untuk menjalankan perangkat keras. Tanpa adanya perangkat lunak ini komponen perangkat keras tidak dapat berfungsi, adapun aplikasi dan *software* yang digunakan dalam pembuatan keamanan data tersebut terdiri dari OS windows 7, mozilla firefox.

### **III.5. Algorithma Dijkstra**

Algorithma Dijkstra adalah suatu algorithma untuk menentukan jalur terpendek antar node dengan bedasar pada basis penghitungan “dari satu node keseluruhan node“. Algorithma Dijkstra termasuk dalam jenis algorithma Link State, yaitu memperhatikan total jarak dan rute yang akan dilalui. Pada dasarnya, terdapat beberapa notasi utama dalam pengerjaan Algorithma Dijkstra :

## 1. Inisialisasi

$$N = \{s\}$$

$$D_j = C_{sj}, \forall j \neq s$$

$$D_s = 0$$

Untuk proses inisialisasi, dibentuk suatu array/himpunan  $N$  dengan anggota  $s$  ( $s$  adalah lambang untuk suatu node sumber). Nilai  $D$  adalah jarak yang akan tersedia pada tabel hasil algoritma, sementara  $C$  adalah nilai jarak pada map yang tersedia. Maka pada tahap inisialisasi ini nilai  $D_j$  (jarak pada hasil tabel antara node  $s$  dengan node  $j$ , dengan  $j$  tidak sama dengan  $s$ ) dimasukkan nilai yang sebenarnya. Jika tidak tersambung secara langsung maka akan dianggap tak terdefinisi. Untuk jarak  $D_s$  tentu saja bernilai 0.

## 2. Temukan simpul tetangga (node selain sumber)

$$D_i = \min_{j \in N} D_j$$

Pokoknya nanti untuk perulangan tiap baris dimasukkan node  $i$  yang belum termasuk pada array/himpunan  $N$  untuk nanti node  $i$  tersebut dijadikan sebagai "perpanjangan" dari node  $s$ , dengan node  $i$  juga merupakan node tetangga dari node  $s$ . Node  $i$  yang dimasukkan pada himpunan  $N$  berdasarkan pada jarak

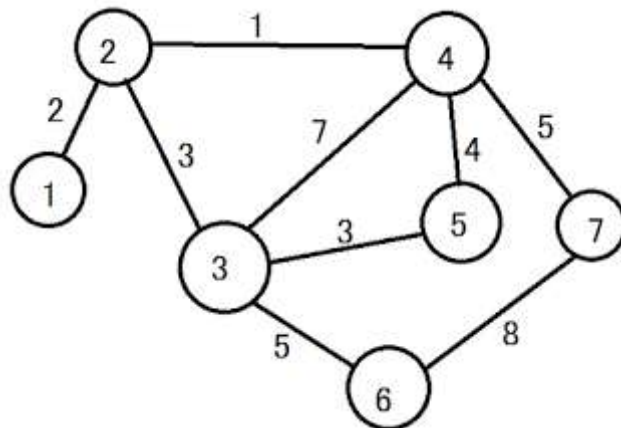
terkecil dengan node  $s$ . Dan jika seluruh node sudah masuk dalam himpunan  $N$ , maka iterasi akan berhenti.

### 3. Update untuk setiap $j \in N$

$$D_j = \min \{D_j, D_i + C_{ij}\}$$

Untuk setiap node  $j$  (dalam tabel hasil: tiap kolom) diperbaharui nilainya yang paling kecil yaitu membandingkan antara nilai  $D_j$  sebelumnya dengan hasil penjumlahan ( $D_i + C_{ij}$ ), yaitu penjumlahan jarak node  $s$  ke node  $i$  dengan jarak sebenarnya dari node  $i$  ke node  $j$ .

Jaringan yang sama dengan yang ada pada postingan Bellman-Ford yang lalu, dapat dilihat pada gambar III.1.



**Gambar III.1. Algoritma Dijkstra Untuk mencari Jalur Terpendek.**

**Contoh Kasus :**

Untuk mempermudah, buatlah tabel seperti berikut ini:

(Notasi dalam tabel algoritma Dijkstra memiliki format (s-j,D), dimana s-j menunjukkan rute dari node s menuju node j, sementara D menunjukkan jarak total antara kedua node tersebut)

N	D2	D3	D4	D5	D6	D7
{1}	(1-2, 2)	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$

Baris pertama masih berupa inisialisasi, yaitu  $D_j$  akan memiliki nilai jika tersambung langsung dan tidak memiliki nilai jika tidak tersambung langsung.

Karena node 1 kebetulan hanya memiliki 1 tetangga yaitu node 2, maka  $i = 2$  dimasukkan pada himpunan N.

N	D2	D3	D4	D5	D6	D7
{1}	(1-2, 2)	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
{1,2}	(1-2, 2)	(1-2-3, 5)	(1-2-4, 3)	$\infty$	$\infty$	$\infty$

Node 2 sudah berperan sebagai "perpanjangan" node sumber (node 1), sehingga sekarang node-node yang terhubung dengan node 2 sudah bisa "dijangkau" oleh node 1 via node 2. Diketahui node 3 dan node 4 terhubung langsung dengan node 2, sehingga rutanya ditulis (1-2-3) dan (1-2-4).

Untuk langkah selanjutnya dipilih node  $i$  yang telah tersambung dengan node  $s$  namun belum masuk dalam himpunan N. Diketahui yaitu node 3 dan node 4.

Node yang dipilih adalah yang memiliki jumlah jarak yang paling minimum, yaitu node 4. Sehingga didapat baris tabel berikutnya seperti berikut.

N	D2	D3	D4	D5	D6	D7
{1}	(1-2, 2)	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
{1,2}	(1-2, 2)	(1-2-3, 5)	(1-2-4, 3)	$\infty$	$\infty$	$\infty$
{1,2,4}	(1-2, 2)	(1-2-3, 5)	(1-2-4, 3)	(1-2-4-5, 7)	$\infty$	(1-2-4-7, 8)

Seperti langkah yang sebelumnya, sekarang node 4 ikut berperan sebagai "perpanjangan" dari node 1 sehingga node 5 dan node 7 yang terhubung langsung dengan node 4 sudah bisa "dijangkau" oleh node 1 dengan rute yang tertampil.

Sebenarnya disini mulai terjadi perbandingan nilai jarak. Dengan dimasukkannya node 4 dalam himpunan N, maka node 4 berlaku sebagai "penembak sementara". Maksudnya adalah node 4 dapat melakukan perhitungan minimum menuju node tertentu meskipun node tersebut telah diketahui rute dan jaraknya.

Dalam kasus ini dapat diambil node 3. Node 3 sudah diketahui rute dan jaraknya pada baris ke-2. Dengan masuknya node 4 pada himpunan N, maka node 4 akan menghitung jarak minimum antara *entry* D3 yang terdahulu dengan yang baru. (Perbandingan via node 2 dengan via node 4):

$$\text{Via node 2} \rightarrow D2 + C23 = 2 + 3 = 5$$

$$\text{Via node 4} \rightarrow D4 + C43 = 3 + 7 = 10$$

Maka *entry* yang dipertahankan adalah via node 2 dengan jarak 5.

Sebelumnya telah dipilih node 4 sebagai anggota N karena bertetangga dengan node 2. Sekarang dipilih tetangga node 2 yang lainnya yaitu node 3.

N	D2	D3	D4	D5	D6	D7
{1}	(1-2, 2)	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
{1,2}	(1-2, 2)	(1-2-3, 5)	(1-2-4, 3)	$\infty$	$\infty$	$\infty$
{1,2,4}	(1-2, 2)	(1-2-3, 5)	(1-2-4, 3)	(1-2-4-5, 7)	$\infty$	(1-2-4-7, 8)
{1,2,3,4}	(1-2, 2)	(1-2-3, 5)	(1-2-4, 3)	(1-2-4-5, 7)	(1-2-3-6, 10)	(1-2-4-7, 8)

Perbandingan yang terjadi:

Pada D4

$$\text{Via node 2} \rightarrow D2 + C24 = 2 + 1 = 3$$

$$\text{Via node 3} \rightarrow D3 + C34 = 5 + 7 = 12$$

Maka *entry* yang dipertahankan adalah via node 2 dengan jarak 3.

Pada D5

$$\text{Via node 4} \rightarrow D4 + C45 = 3 + 4 = 7$$

$$\text{Via node 3} \rightarrow D3 + C35 = 5 + 3 = 8$$

Maka *entry* yang dipertahankan adalah via node 4 dengan jarak 7.

tabel hasil akhirnya adalah sebagai berikut.

N	D2	D3	D4	D5	D6	D7
{1}	(1-2, 2)	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
{1,2}	(1-2, 2)	(1-2-3, 5)	(1-2-4, 3)	$\infty$	$\infty$	$\infty$
{1,2,4}	(1-2, 2)	(1-2-3, 5)	(1-2-4, 3)	(1-2-4-5, 7)	$\infty$	(1-2-4-7, 8)
{1,2,3,4}	(1-2, 2)	(1-2-3, 5)	(1-2-4, 3)	(1-2-4-5, 7)	(1-2-3-6, 10)	(1-2-4-7, 8)
{1,2,3,4,5}	(1-2, 2)	(1-2-3, 5)	(1-2-4, 3)	(1-2-4-5, 7)	(1-2-3-6, 10)	(1-2-4-7, 8)
{1,2,3,4,5,7}	(1-2, 2)	(1-2-3, 5)	(1-2-4, 3)	(1-2-4-5, 7)	(1-2-3-6, 10)	(1-2-4-7, 8)
{1,2,3,4,5,6,7}	(1-2, 2)	(1-2-3, 5)	(1-2-4, 3)	(1-2-4-5, 7)	(1-2-3-6, 10)	(1-2-4-7, 8)

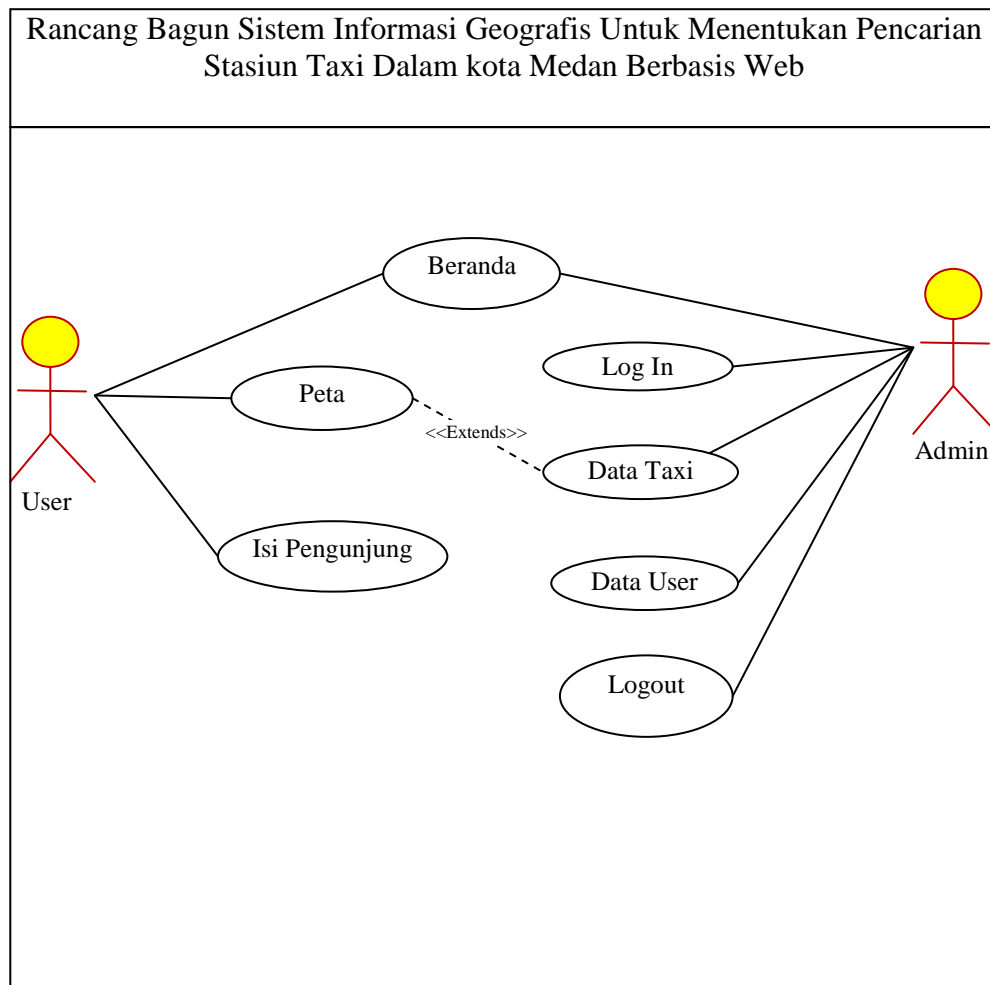
### **III.6. Desain Sistem**

Setelah tahapan analisis sistem, maka selanjutnya dibuat suatu rancangan sistem. Perancangan sistem adalah tahapan yang berguna untuk memperbaiki efisiensi kerja suatu sistem yang telah ada. Pada perancangan sistem ini terdiri dari tahap perancangan yaitu :

1. Perancangan *Use Case Diagram*
2. Perancangan *Sequence Diagram*
3. Perancangan *Activity Diagram*
4. Perancangan *Output dan Input*

#### **III.6.1. Diagram *Use Case***

*Use case* menjelaskan urutan kegiatan yang dilakukan aktor dan sistem untuk mencapai suatu tujuan tertentu. Sebuah *Use Case* mempresentasikan sebuah interaksi antara aktor dengan sistem dan menggambarkan fungsionalitas yang diharapkan dari sebuah sistem informasi geografis letak stasiun taxi di kota medan. Diagram *Use Case* tersebut dapat dilihat pada gambar III.2.



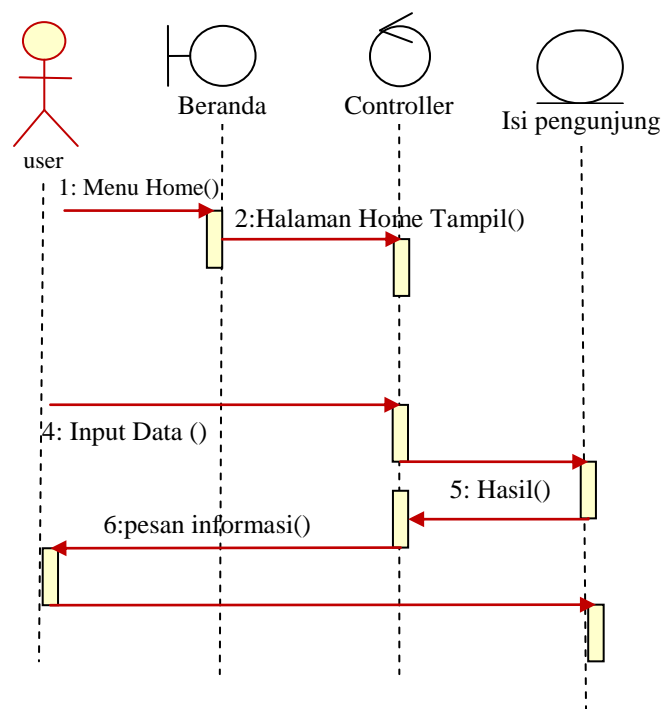
**Gambar III.2. Diagram Use Case**

### III.6.2. Sequence Diagram

*Sequence* diagram menunjukkan bagaimana operasi yang dilakukan secara detail. *Sequence* diagram menjelaskan interaksi obyek yang disusun dalam suatu urutan waktu. Urutan waktu yang dimaksud adalah urutan kejadian yang dilakukan oleh seorang *actor* dalam menjalankan sistem, adapun *sequence* yang dilakukan adalah sebagai berikut.

### 1. *Sequence Isi Pengunjung*

Isi Pengunjung digunakan halaman pembuka program sistem informasi geografis mengetahui lokasi dan salam pembuka, tentang defisini sistem informasi geografis dan nama seorang programmer, untuk lebih jelasnya dapat dilihat pada gambar III.3.

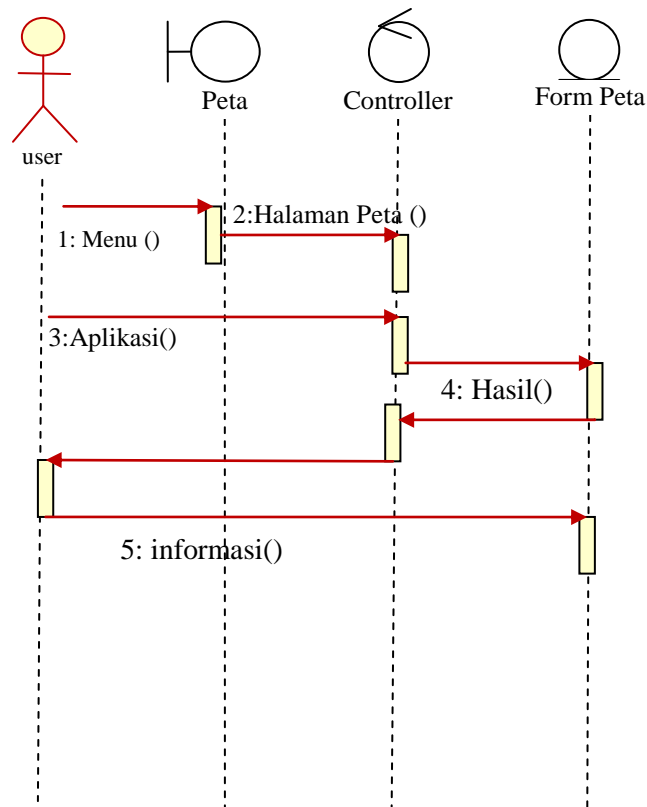


**Gambar III.3. *Sequence Isi Pengunjung***

Dari gambar III.2 menunjukkan bahwa seorang *user* jika ingin masuk ke halaman beranda harus terlebih dahulu melakukan klik pada menu beranda. Maka menu beranda tampil dilayar monitor dengan menampilkan salam pembuka dan menampilkan pengisian data pengunjung.

## 2. *Sequence Peta*

Peta digunakan sebagai informasi tentang lokasi taxi, untuk lebih jelasnya dapat dilihat pada gambar III.4.

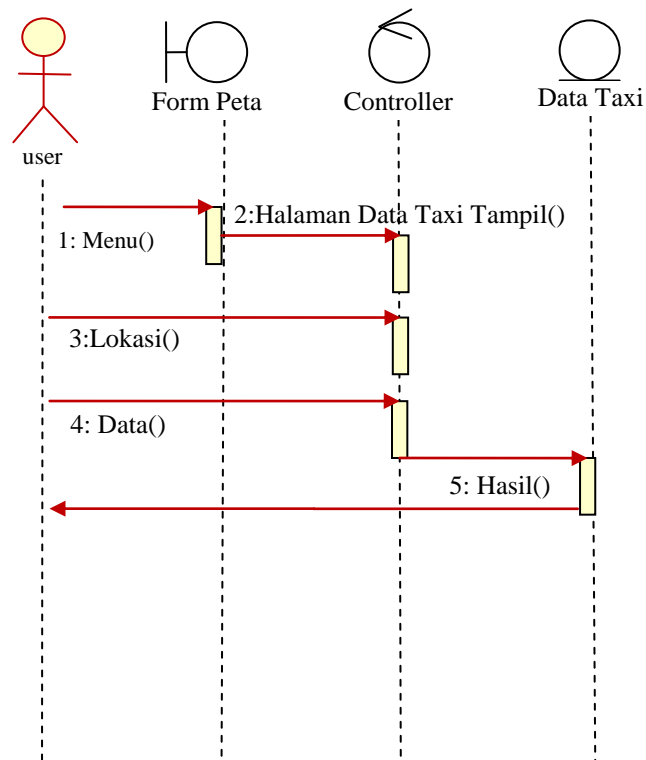


**Gambar III.4. *Sequence Peta***

Dari gambar III.3 menunjukkan bahwa seorang *user* dapat melakukan mendapatkan informasi dengan melakukan mengklik dari masing-masing menu aplikasi, dan hasil.

## 3. *Sequence Data Taxi*

*Sequence* data taxi adalah halaman untuk memasukkan data peletakan stasiun taxi yang ada di Medan, untuk lebih jelasnya dapat dilihat pada gambar III.5.

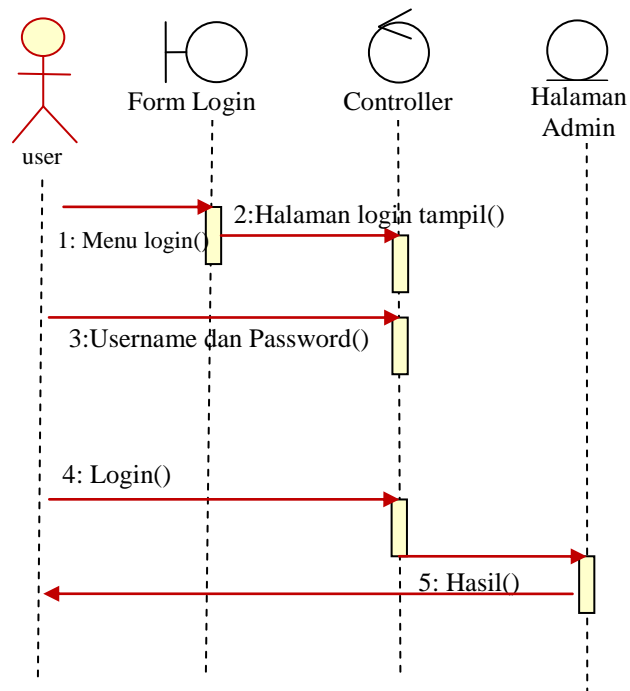


**Gambar III.5. Sequence Data Taxi**

Dari gambar III.4 jika pengguna ingin melihat informasi geografis tentang peta tentang lokasi taxi yang ada di Kota Medan.

#### 4. Sequence Login

Sequence login terdiri dari beberapa label, teksbox, dan tombol. Dari masing-masing tool tersebut mempunyai tugas masing-masing, label memberikan informasi tentang nama objek, teksbox untuk memasukkan data, sedangkan tombol untuk mengeksekusi perintah. *Login* digunakan menampilkan menu login administrator. untuk lebih jelasnya dapat dilihat pada gambar III.6.

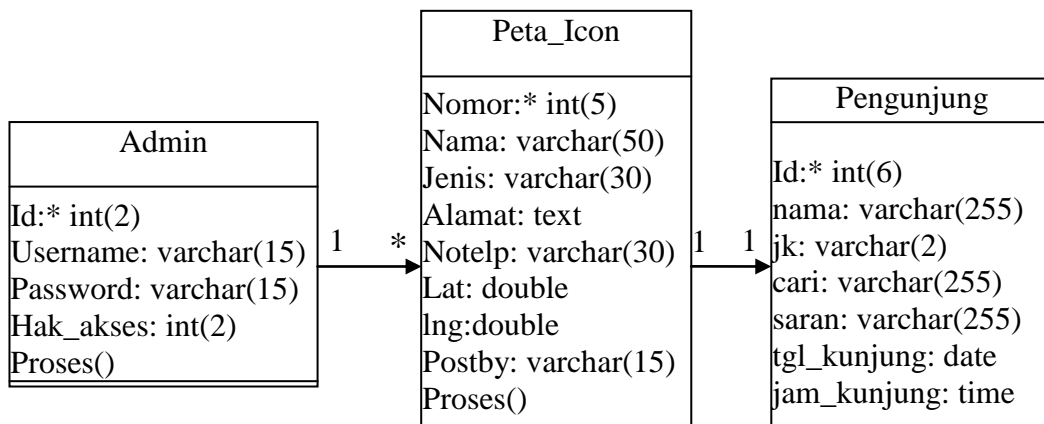


**Gambar III.6. Sequence Login**

Dari gambar III.5 menunjukkan bahwa seorang pengguna dapat masuk kehalaman administrator dengan memasukkan *username* dan *password* yang benar dan melakukan eksekusinya dengan mengklik tombol login..

### III.6.3. Class Diagram

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).



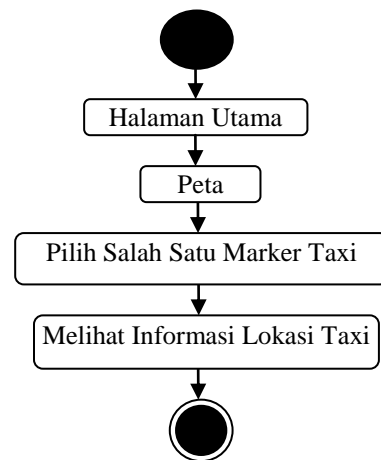
**Gambar III.7. Class Diagram Taxi di Medan**

#### III.6.4. Activity Diagram

Bentuk proses yang digambarkan pada *use case* diagram di atas dijabarkan dengan *activity diagram* :

##### 1. Activity Diagram Melihat Peta

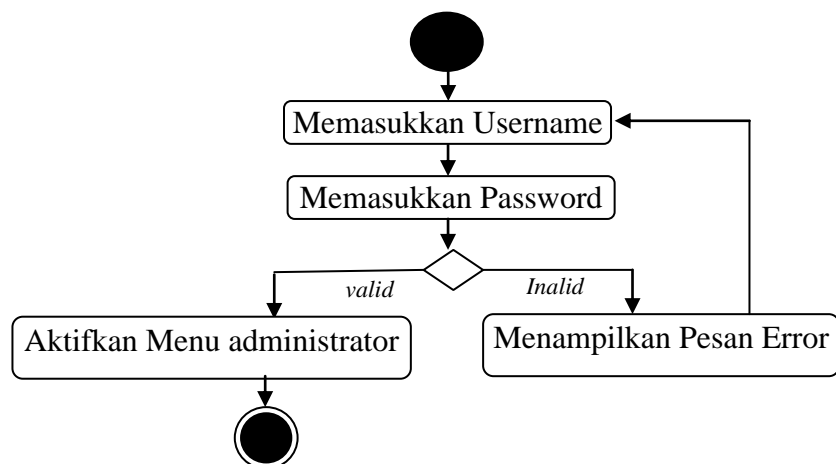
Aktivitas melihat peta diterangkan dalam langkah-langkah *state*, dimulai dari kegiatan melihat panel peta kemudian mencari artikel lokasi tempat stasiun Taxi , selanjutnya menekan tombol atau *link* yang ada pada peta dan yang terakhir melihat informasi yang di sajikan dalam peta yang ditunjukkan pada gambar III.8.



**Gambar III.8. Activity Diagram Melihat Peta**

## 2. Activity Diagram Login Admin

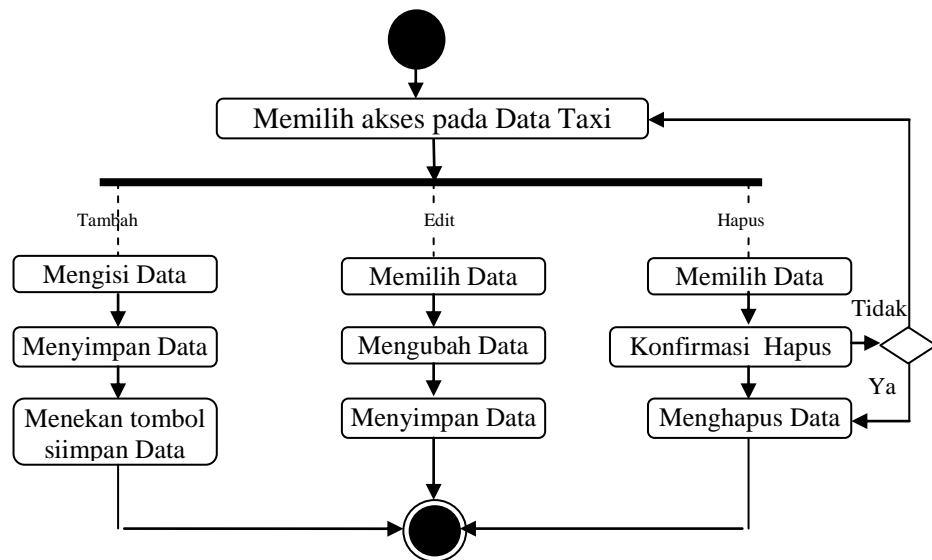
*Activity* diagram login merupakan *activity* diagram untuk proses login admin. Dimulai dari memasukkan username, memasukkan password, jika profil valid maka sistem akan mengaktifkan administrator, sedang jika tidak valid maka tampilan pesan kesalahan yang ditunjukkan. *Activity* diagram login admin ditunjukkan pada gambar III.9. berikut ini:



**Gambar III.9. Activity Diagram login**

## 2. Activity Tambah, Edit Dan Hapus Data Objek

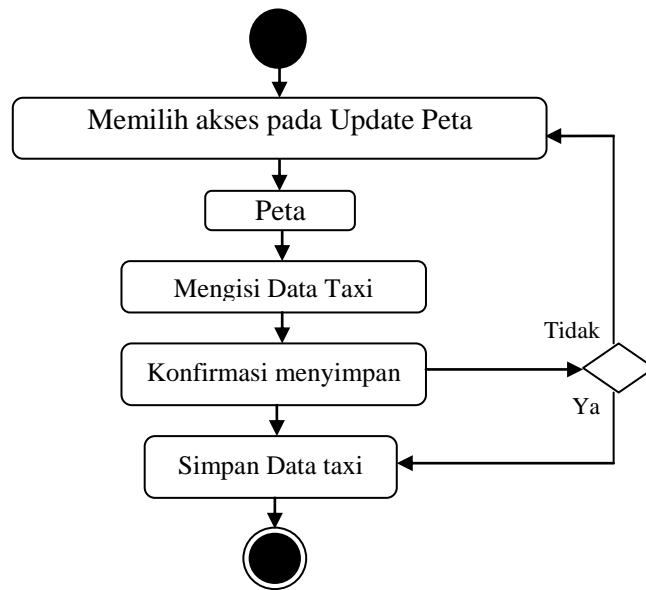
Aktifitas proses mengolah data diterangkan dalam langkah-langkah *state* yang ditunjukkan pada gambar III.10.



**Gambar III.10. Activity Diagram Tambah, Edit, Hapus Data**

## 3. Activity Diagram Update Peta

Activity diagram *Update* peta merupakan *activity* diagram untuk proses update peta. Activity diagram tersebut ditunjukkan pada gambar III.11. berikut ini:



**Gambar III.11. Activity Update Peta**

### III.7. Rancangan Program

Rancangan program simulasi tentang sistem informasi geografis untuk mengetahui letak lokasi taxi, teknis sistem secara fisik yaitu perancangan bentuk fisik atau bagan arsitektur sistem yang diusulkan. Dalam merancang suatu sistem perlu diketahui hal yang akan menunjang sistem, agar dapat mempermudah pengolahan data nantinya. Pengolahan data ini diharapkan dapat mempermudah dalam hal penyajian, pelayanan dan pembuatan berbagai laporan data yang dibutuhkan.

### III.7.1. Rancangan Output

#### 1. Rancangan *Output* Menu Utama

Form *output* menu utama ini dirancang sebagai pusat dari semua menu pada aplikasi SIG, untuk jelasnya dapat dilihat pada gambar III.12.

Peta	User Name <input type="text"/>	Password <input type="text"/>	Login
Header			
Selamat Datang Di GIS Pencarian Stasiun Taxi			
Kalender	Nama	<input type="text"/>	
	Jenis Kelamin	<input type="radio"/> Laki-laki <input type="radio"/> Perempuan	
	Anda Pengunjung Ke:-	<input type="text"/>	
	Hari Ini	<input type="text"/>	
Bulan Ini	Informasi Yang DiCari	<input type="text"/>	
	Saran-Saran	<input type="text"/>	
Keterangan Harus Diisi			

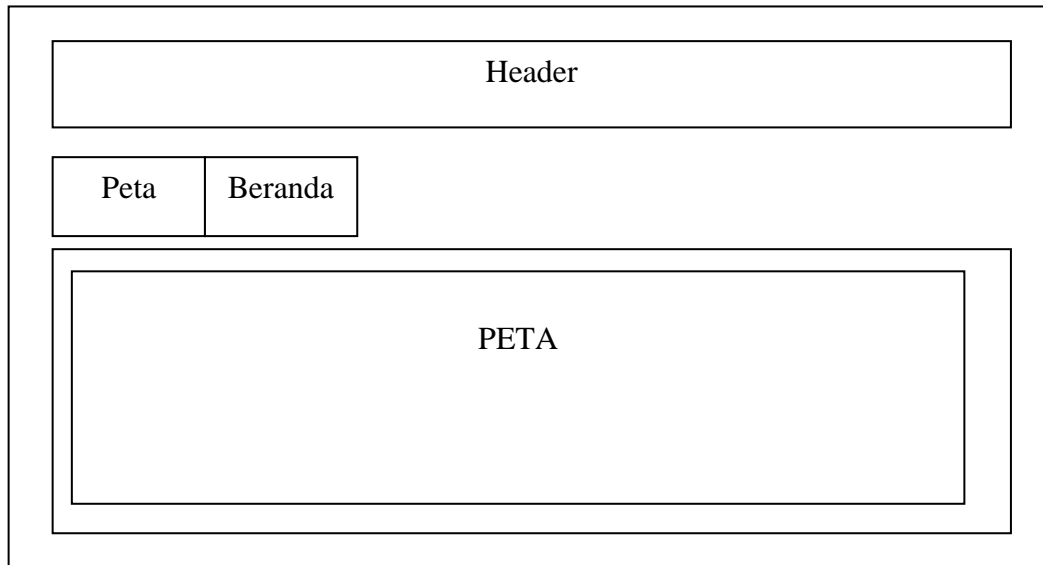
**Gambar III.12. Rancangan Form Menu Utama**

Rancangan menu utama yang terdapat pada gambar III.12 terdiri dari peta, login dan pengisian pengunjung, untuk masuk ke halaman administrator user harus login terlebih dahulu.

#### 2. Rancangan *Output* Peta

Dalam perancangan *form* peta ini terdiri peta dan untuk menentukan node yaitu awal dan tujuan. Tombol kiri menampilkan halaman menurun, jika pengguna ingin melihat halaman berikutnya pengguna mengklik tombol kanan

yang menampilkan halaman menaik. Kedua tombol kiri dan kanan letaknya di sebelah kanan atas. Untuk lebih jelasnya dapat dilihat pada gambar III.13.



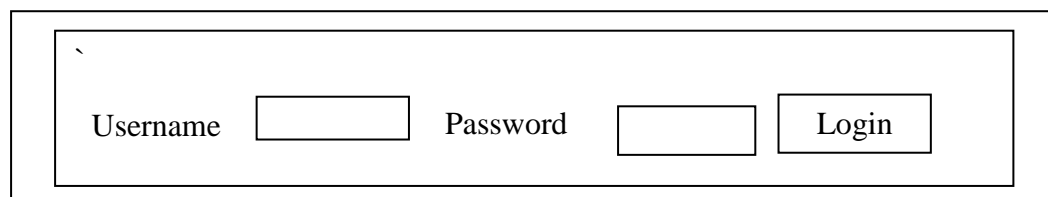
The diagram shows a web form layout for a map. It consists of a main container with a header section at the top. Below the header, there are two buttons labeled 'Peta' and 'Beranda'. The main content area is a large rectangle labeled 'PETA'.

**Gambar III.13. Rancangan *Form Peta***

Rancangan *form* peta untuk memunculkan peta dan untuk mengetahui letak taxi di Medan dengan melakukan memilih dari masing-masing tempat stasiun taxi tersebut.

### 3. Rancangan *Output Login*

Form login dirancang agar user dapat masuk kedalam halaman administrator. Untuk lebih jelasnya dapat dilihat pada gambar III.14.



The diagram shows a login form layout. It features a main container with a sub-container inside. The sub-container contains the labels 'Username' and 'Password', each followed by an input field. To the right of the input fields is a button labeled 'Login'.

**Gambar III.14. Rancangan *Form Login***

Rancangan *form* login yang terdapat pada gambar III.14 terdiri *username* dan *password* berikut dengan tombol eksekusi perintah yaitu *login* dan *batal*.

### III.7.2. Rancangan Input Pengunjung

#### 1. Rancangan Input Data Pengunjung

Rancangan *Input* pengunjung, dapat dilihat pada gambar III.15.

Header	
Nama	<input type="text"/>
Jenis Kelamin	<input type="radio"/> Laki-laki <input type="radio"/> Perempuan
Agama	<input type="text" value="v"/>
Informasi Yang DiCari	<input type="text"/>
Saran-Saran	<input type="text"/>
	<input type="button" value="Submit"/> <input type="button" value="Reset"/>
Keterangan Harus Diisi	

**Gambar III.15. Rancangan Input Pengunjung**

#### 2. Rancangan Input Peta

Rancangan input Peta yang terdapat pada gambar III.16 terdiri dari identitas para pengunjung.

Header						
PETA	<div style="text-align: center; margin-bottom: 10px;"><b>Data Taxi</b></div> <p>LATITUDE <input style="width: 100%;" type="text"/></p> <p>LONGTITUDE <input style="width: 100%;" type="text"/></p> <p>Nama Taxi <input style="width: 100%;" type="text"/></p> <p>Alamat <input style="width: 100%;" type="text"/></p> <p>Telepon Taxi <input style="width: 100%;" type="text"/></p> <div style="text-align: center; margin-top: 10px;"><input type="button" value="Simpan"/></div>					
No	Nama Taxi	Alamat	Telepon Taxi	Latitude	Longitude	Aksi

**Gambar III.16. Rancangan Input Peta**

Pada gambar III.14 menampilkan rancangan *input* peta yang terdiri dari beberapa *field* yaitu latitude, longitude, nama taxi, alamat, telepon taxi dan tombol simpan. Dan dilengkapi dengan tabel grid untuk menampilkan data yang tersimpan di *database*.

### III.8. Desain Database

*Database* merupakan kumpulan dari data yang saling berhubungan satu dengan yang lain. Untuk merancang *database* secara konseptual tentunya diperlukan alat bantu, baik untuk menggambarkan keterhubungan antar data maupun pengoptimalan rancangan *database*. Alat bantu adalah kamus data dan disain tabel. Dengan dibuatnya desain *database* maka proses *database* lebih jelas dan hubungan antar tabel dapat mudah dimengerti dan dipahami.

#### III.8.1 Kamus Data

Kamus data adalah kumpulan elemen-elemen atau simbol-simbol yang digunakan untuk membantu dalam penggambaran atau pengidentifikasian setiap *field* atau *file* di dalam sistem. Kamus Data berfungsi antara lain untuk menjelaskan arti aliran data dan penyimpana data, mendeskripsikan komposisi paket data yang bergerak melalui aliran data dan menjelaskan spesifikasi nilai dan satuan yang relevan dengan data. Didalam kamus data ini menjelaskan nama tabel berisi dengan nama-nama field yang berkaitan dengan informasi data. Berikut adalah Kamus Data dari sistem yang penulis bahas.

1. Admin = [{id\_admin} + username + password + level]
2. Athor = [{id\_author} + isi + page + postby]
3. Peta = [{id} + nomor + nama + jenis + alamat + no\_telp + postby]

#### III.8.2. Desain Tabel

Desain tabel pada aplikasi gis untuk mengetahui lokasi stasiun taxi. *Field-field* beserta rancangan tabel *database* diperlihatkan dalam tabel berikut :

## a. Tabel Admin

Tabel admin berfungsi untuk menyimpan data lokasi kedalam *database*.

Nama Database : taxi

Nama Tabel : admin

Primary Key : id\_admin

**Tabel III.1. Struktur Tabel Admin**

No	Nama Field	Tipe Data	Size
1	Id_admin	int	11
2	Username	Varchar	20
3	Password	Varchar	100
4	Level	Int	11

## b. Tabel Peta

Tabel peta berfungsi untuk menampilkan data-data taxi berdasarkan titik pusat kota Medan tujuan kedalam *database*.

Nama Database : taxi

Nama Tabel : peta\_icon

Primary Key : nomor

**Tabel III.2. Struktur Tabel Peta**

No	Nama Field	Tipe Data	Size
1	nomor	Int	11
2	nama	Varchar	100
3	jenis	Varchar	100
4	alamat	text	255
5	Notelp	varchar	-
6	lat	double	-
7	lang	double	10
8	Postby	varchar	10

c. Tabel Pengunjung

Tabel pengunjung berfungsi untuk menampilkan data-data pengunjung kedalam *database*.

Nama Database : taxi

Nama Tabel : pengunjung

Primary Key : id

**Tabel III.3. Struktur Tabel Pengunjung**

No	Nama Field	Tipe Data	Size
1	id	Int	11
2	nama	Varchar	255
3	jk	Varchar	2
8	cari	varchar	255
9	saran	varchar	255
10	Tgl_pengunjung	date	-
11	Jam_pengunjung	time	-