

## BAB III

### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1 Analisa Sistem

Sistem yang di bangun pada penelitian ini secara umum merupakan sistem yang di gunakan untuk membantu warga di kota gresik dalam mencari rumah sakit di wilayah gresik. Sistem aplikasi pencarian rumah sakit berbasis web ini merupakan suatu sistem penentuan rute terpendek dari lokasi pengguna dan lokasi rumah sakit yang dituju dengan menggunakan algoritma *Dijkstra*. Dalam aplikasi pencarian lokasi rumah sakit ini akan menampilkan hasil pencarian dalam bentuk peta digital yang dilengkapi dengan rute untuk menemukan lokasi yang akan dituju oleh pengguna.

Data lokasi rumah sakit diperoleh dari salah satu *website* ([Gresikkab.go.id](http://Gresikkab.go.id)). Data yang diperoleh akan di inputkan kedalam sistem sehingga pengguna juga bisa melihat informasi berupa alamat rumah sakit, no. telpon, dan juga *website* rumah sakit yang ada di kota gresik.

#### 3.2 Hasil Analisis

Berdasarkan hasil analisis yang terkumpul dari penelitian yang dilakukan adalah menghasilkan informasi dan memberikan rute menuju rumah sakit yang akan dituju oleh pengguna dan diharapkan mampu dalam menentukan rute terbaik yang dipilih sesuai dengan rekomendasi dari sistem. Secara umum sistem yang akan dibuat dalam penelitian ini adalah sebagai berikut:

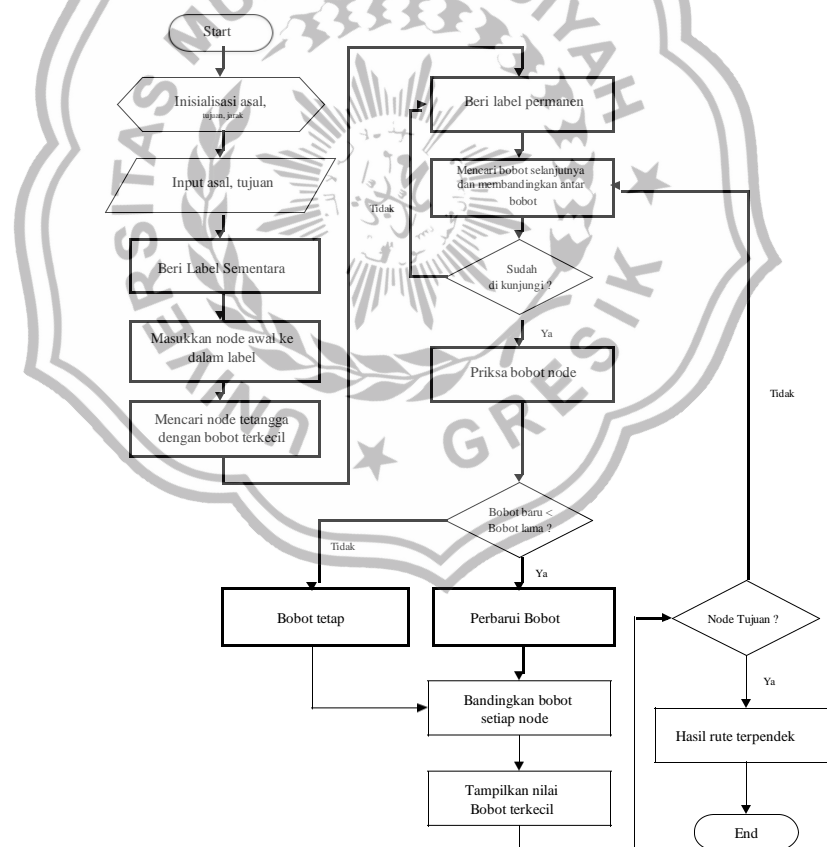
- a. Admin memasukkan data informasi dan lokasi rumah sakit kedalam sistem
- b. Pengguna menentukan lokasi awal kemudian menentukan lokasi tujuan rumah sakit yang telah ditentukan oleh sistem

- c. Sistem akan melakukan kalkulasi untuk memberikan rute terpendek bagi pengguna.

Sistem aplikasi pencarian rute menuju rumah sakit menggunakan metode algoritma *dijkstra*. *Dijkstra* merupakan salah satu varian bentuk algoritma populer dalam pemecahan persoalan yang terkait dengan masalah optimasi dan bersifat sederhana. Algoritma ini menyelesaikan masalah mencari sebuah lintasan terpendek.

### 3.2.1 Flowchart Perhitungan Dijkstra

Flowchart Perhitungan *Dijkstra* adalah langkah-langkah perhitungan dalam bentuk diagram alir. Seperti pada gambar 3.1.



**Gambar 3.1** Flowchart Algoritma *Dijkstra*

Pada *flowchart* di atas dapat dijelaskan proses algoritma dijktra sebagai berikut :

Masukkan : Graf berbobot.

Proses :

1. Inisialisasi vertex.
2. Inisialisasi jarak antar vertex.
3. Tentukan vertex awal (s) dan vertex tujuan (t).
4. Beri label permanen = 0 ke verteks awal (s) dan label sementara =  $\infty$  ke verteks lainnya.
5. Untuk setiap verteks V yang belum mendapat label permanen, mendapat label sementara =  $\min \{ \text{label lama V (label lama V + D)} \}$ .
6. Cari nilai minimum diantara semua verteks yang masih berlabel sementara.
7. Jadikan verteks minimum yang berlabel sementara menjadi verteks dengan label permanen, jika lebih dari satu verteks pilih sembarang.
8. Ulangi langkah 5 sampai 7 hingga verteks tujuan mendapat label permanen.
9. Simpan hasil perhitungan Tampilkan hasil perhitungan.

### 3.3 Representasi Model

#### 3.3.1 Perancangan Model Sistem

Perancangan Model Sistem memberikan informasi mengenai tahapan-tahapan dalam pembangunan Sistem dan menjelaskan mengenai komponen apa saja yang dibutuhkan dalam penggambaran sebuah informasi. Tahapan-tahapan dalam pembangunan Sistem adalah sebagai berikut:

- a. Penentuan objek. objek yang melingkupi sistem ini adalah rumahsakit di Kabupaten Gresik.
- b. Penentuan *latitude longitude* lokasi rumah sakit yang digunakan dalam sistem.

- c. Menentukan rute dengan menggunakan metode algoritma *dijkstra*.

### 3.3.2 Penentuan *Latitude* dan *Longitude* Rumah Sakit

Untuk mendapatkan data *Latitude* dan *Longitude* rumah sakit dengan cara membuka google maps dan mencari lokasi rumah sakit di kabupaten gresik yang nantinya *latitude* dan *longitude* nya akan muncul. *Latitude* dan *longitude* yang nantinya akan digunakan sebagai penentuan lokasi pada sistem. Dibawah ini adalah tabel data *latitude* dan *longitude* Rumah sakit di wilayah gresik.

**Tabel 3.1** Tabel Data Informasi Rumah Sakit

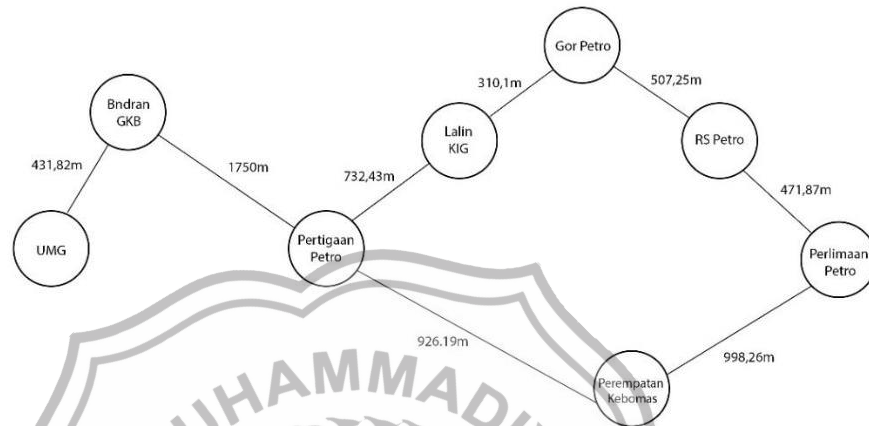
No	Nama	Alamat	<i>Latitude</i> <i>Longitude</i>
1	RSUD Ibnu Sina	Jl. Dr.Wahidin Sudiro Husodo 243-B	-7.168342, 112.601057
2	RS Muhammadiyah Gresik	Jl. KH.Kholil 88, Kebungson	-7.150690, 112.654796
3	RS Petrokimia Gresik	Jl. Jendral Ahmad Yani No.69	-7.160491, 112.642649
4	RSPG Driyorejo	Jl. Raya Legundi Km. 05, Driyorejo	-7.378301, 112.574671
5	RS Semen Gresik	Jl. RA. Kartini 280, Sidomoro	-7.166289, 112.642243
6	RSIA Nyai Ageng Pinatih	Jl. KH. Abdul Karim 76-78, Bedilan	-7.157262, 112.652136
7	RSU Rachmidewi	Jl. Jawa 79-81, Perumahan GKB	-7.150846, 112.613105

8	RS Denisa	Jl. Dr. Wahidin Sudirohusodo	-7.167634, 112.597898
9	RS PKU Muhammadiyah Sekapuk	Jl. Raya Deandels Sekapuk	-6.956639, 112.504159
10	RS Fathma Medika Gresik	Jl. Pendopo 45, Sembayat	-7.074715, 112.574320
11	RSI Mabarrot MWC NU	Jl. Raya Bungah No.63	-7.052809, 112.575074
12	RS Wates Husada Gresik	Jl. Raya Wates Utara Kedung Pring	-7.270843, 112.442767
13	RS Islam Almunawaroh	Jl. Raya Bringkang, Menganti	-7.288055, 112.562912
14	RS Wali Songo 1	Jl. Raya Balongpanggung	-7.290190, 112.435788
15	RS PKG Graha Husada Gresik	Jl. Padi No. 3 Komplek Perumahan PT Petrokimia Gresik	-7.153915, 112.646278
16	RS Surya Medika	Jl. Raya Lakarsantri No. 58, laban Menganti	-7.301676, 112.620889

Dari tabel diatas sudah ada daftar lokasi atau tempat tujuan. Untuk menentukan jalur pengguna harus menginputkan lokasi awal kemudian menentukan lokasi tujuan, jika sudah sistem akan mengkalkulasi dengan menggunakan metode *dijkstra* untuk menentukan rute menuju ke tempat tujuan.

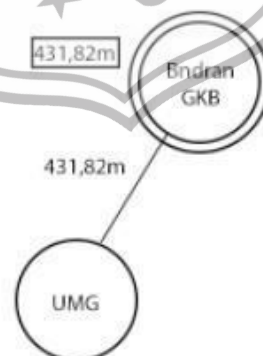
Dibawah ini merupakan penjelasan langkah per langkah pencarian jalur terpendek secara rinci dimulai dari *node* awal sampai *node* tujuan dengan nilai jarak terkecil.

- ada Gambar 3.2, *node* awal berlabel UMG, *node* tujuan Berlabel RS Petro. Setiap *edge* yang terhubung antar *node* telah diberi nilai



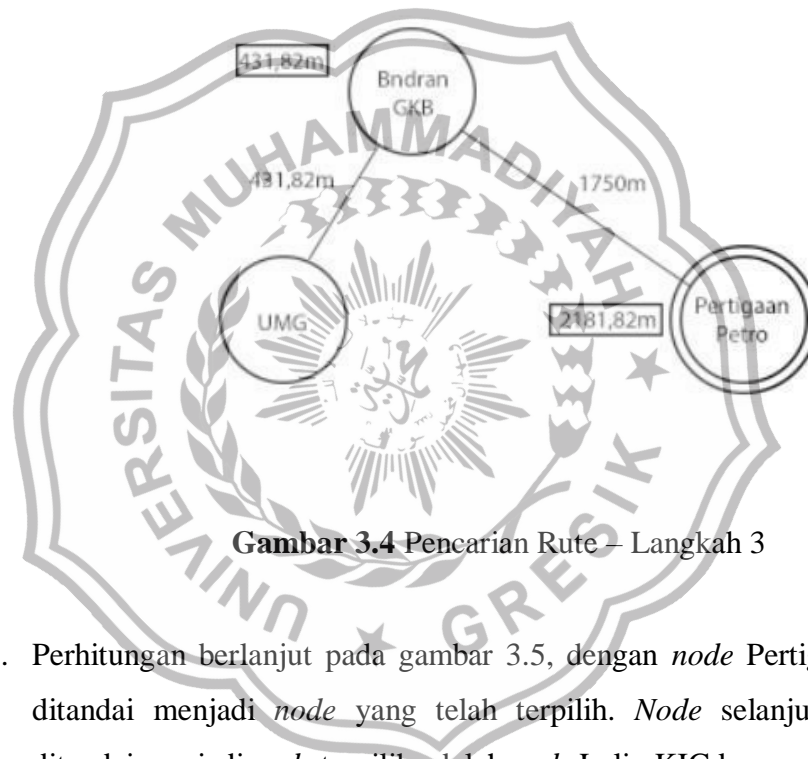
Gambar 3.2 Pencarian Rute - Langkah 1

- Dijkstra* melakukan kalkulasi terhadap *node* yang terhubung langsung dengan *node* keberangkatan (*node* UMG), terlihat pada Gambar 3.3, hasil yang didapat adalah *node* Bundaran Gkb, nilai bobot *node* Bundaran GKB adalah =  $431,82\text{m}$  ( $0+431,82$ )



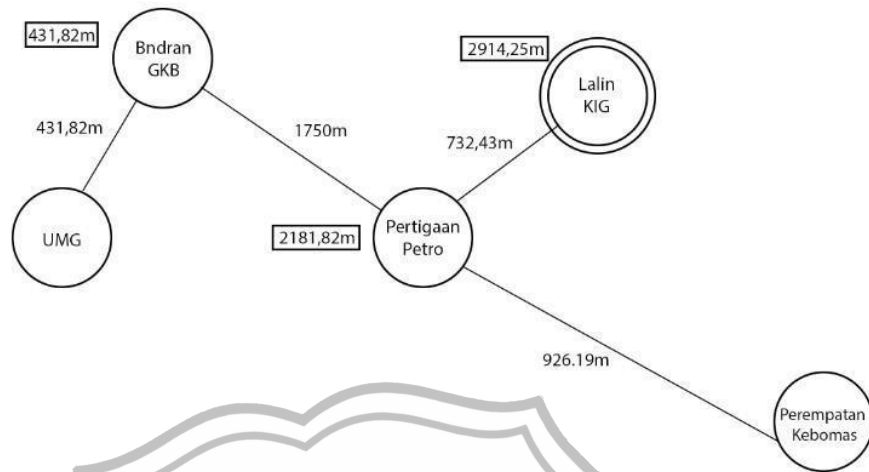
Gambar 3.3 Pencarian Rute Langkah 2

3. *Node* Bundaran GKB diset menjadi *node* keberangkatan dan ditandai sebagai *node* yang terpilih. terlihat pada Gambar 3.4, hasil yang didapat adalah *node* Pertigaan Petro, karena tidak ada *node* tetangga dan nilai bobot *node* Pertigaan Petro adalah = 2181,82m (431,82+1750)



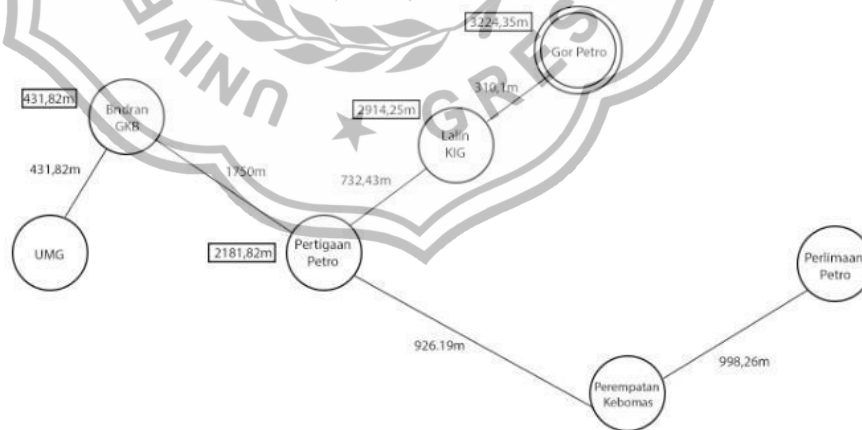
**Gambar 3.4** Pencarian Rute – Langkah 3

4. Perhitungan berlanjut pada gambar 3.5, dengan *node* Pertigaan Petro ditandai menjadi *node* yang telah terpilih. *Node* selanjutnya yang ditandai menjadi *node* terpilih adalah *node* Lalin KIG karena nilai bobot yang terkecil dibandingkan Nilai bobot *node* tetangga. Nilai bobot *node* Lalin KIG adalah = 2914,25m (2181,82+732,43).



**Gambar 3.5** Pencarian Rute – Langkah 4

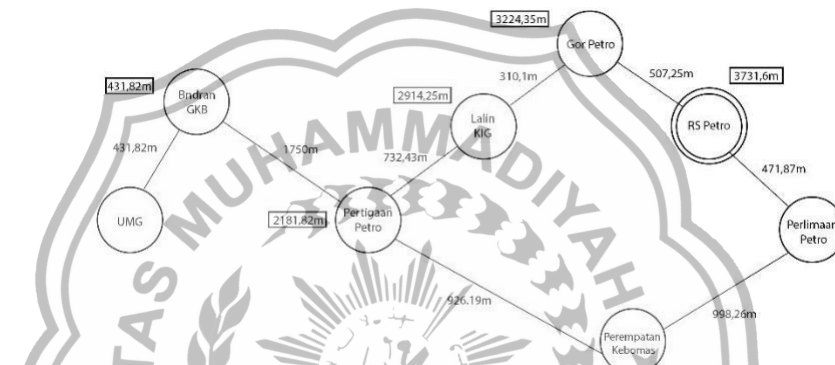
5. *Node* Lalin KIG ditandai sebagai *node* yang terpilih. Dari semua *node* tetangga belum terlewati yang terhubung langsung dengan *node* terlewati, *node* selanjutnya yang ditandai menjadi *node* terpilih adalah *node* Gor Petro terlihat pada gambar 3.6. Nilai bobot *node* Gor Petro adalah = 3224,35m (2914,25+310,1).



**Gambar 3.6** Pencarian Rute – Langkah 5



6. *Node* Gor Petro menjadi *node* terpilih, *Dijkstra* melakukan kalkulasi kembali, dan menemukan bahwa *node* RS Petro (*node* tujuan) telah tercapai lewat *node* Gor Petro. Pada Gambar 3.7, alur terpendeknya adalah UMG - Bundaran GKB - Pertigaan Petro – Lalin KIG - Gor Petro – RS Petro, dan nilai bobot yang didapat adalah 3731,6m (3224,35+507,25). Bila *node* tujuan telah tercapai maka kalkulasi *Dijkstra* dinyatakan selesai.



Gambar 3.7 Hasil Pencarian Rute

Dibawah ini adalah perhitungan hasil pencarian rute dalam bentuk tabel :

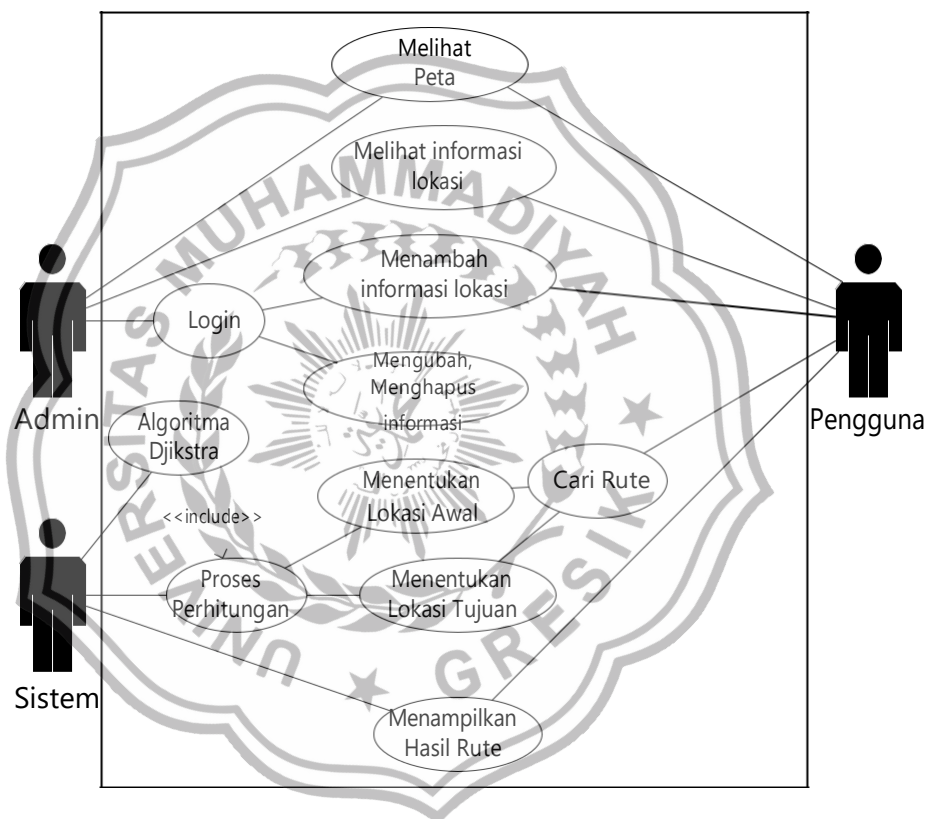
Tabel 3.2 Perhitungan Hasil Pencarian Rute

Titik terkunjungi	Umg (A)	Bundaran GKB (B)	P3 Petro (C)	Lalin KIG (D)	Gor (E)	P4 Kebomas (F)	P5 kebomas (G)	RS PETRO (H)
	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
{A}	0	431,82	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
{A,B}	0	431,82	2181,82	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
{A,B,C}	0	431,82	2181,82	2914,25	$\infty$	3108,01	$\infty$	$\infty$
{A,B,C,D}	0	431,82	2181,82	2914,25	3224,35	3108,01	$\infty$	$\infty$
{A,B,C,D,E}	0	431,82	2181,82	291,25	3224,35	$\infty$	$\infty$	3731,6

### 3.4 Perancangan Sistem

#### 3.4.1 Usecase Diagram

*Usecase* diagram merupakan interaksi yang saling berkaitan antara sistem dengan aktor, *usecase* digunakan untuk membentuk perilaku (*behavior*) sistem yang akan dibuat. *Usecase* diagram pada gambar 3.8 dan 3.9 ini akan menggambarkan interaksi sistem pencarian rute dengan pengguna (aktor).



**Gambar 3.8** *Usecase Diagram sistem*

*Definisi* dan deskripsi *usecase diagram* dapat dilihat pada tabel di bawah ini.

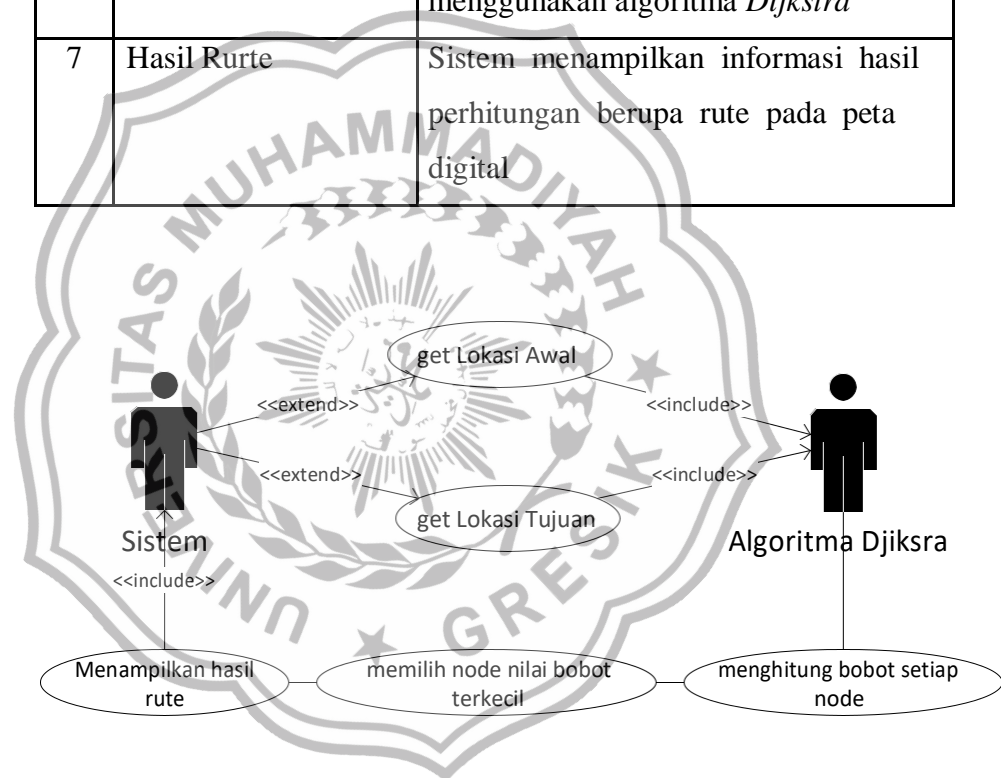
**Tabel 3.3** Definisi Aktor

No	Aktor	Deskripsi
1	Admin	Melakukan <i>login</i> kemudian mengelola data rumah sakit dan data peta kemudian mempublishkan, Jika ada data yang salah maka admin dapat mengganti dengan informasi yang benar ( <i>editing/hapus</i> )
2	Pengguna	<i>User</i> dapat langsung mencari rute menuju rumah sakit dengan melakukan klik pada halaman pencarian rute, selanjutnya <i>user</i> akan mengisi lokasi awal kemudian menentukan lokasi rumah sakit yang akan dituju
3	Sistem	Sistem yang akan melakukan proses perhitungan, dan menampilkan rute dalam bentuk peta digital sebagai informasi kepada <i>user</i>

**Tabel 3.4** Deifinisi *Usecase*

No	Skenario	Deskripsi
1	Login	Proses admin dalam memberikan informasi
2	Informasi lokasi	Admin akan memasukkan data informasi lokasi ke sistem
3	<i>Edit</i> atau Hapus informasi lokasi	Jika ada data yang salah, maka admin dapat mengganti dengan data informasi yang benar

4	Cari Rute	Halaman <i>user</i> untuk melakukan pencarian rute
5	Set Lokasi Awal dan Tujuan	<i>User</i> menentukan lokasi awal kemudian memilih lokasi rumah sakit yang akan dituju
6	Proses Perhitungan	Sistem melakukan proses perhitungan data yang telah di ambil dari <i>user</i> menggunakan algoritma <i>Dijkstra</i>
7	Hasil Rurte	Sistem menampilkan informasi hasil perhitungan berupa rute pada peta digital



**Gambar 3.9** Usecase proses perhitungan rute

*Definisi* dan deskripsi *usecase diagram* dapat dilihat pada table di bawah ini.

**Tabel 3.5** Definisi Aktor Proses Perhitungan

No	Aktor	Deskripsi
1	Sistem	Sistem mendapatkan lokasi awal dan tujuan dari pengguna
2	<i>Dijkstra</i>	Melakukan Proses perhitungan

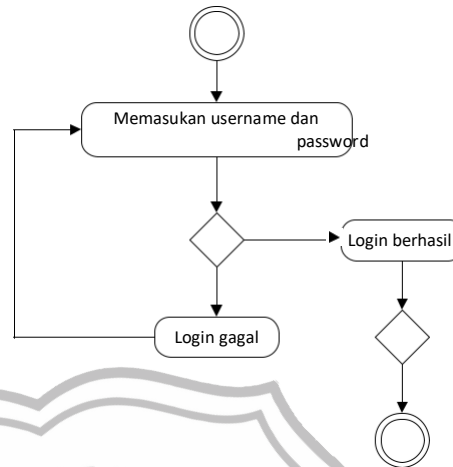
**Tabel 3.6** Deskripsi *Usecae* Proses Perhitungan

No	Skenario	Deskripsi
1	Get Lokasi Awal	Sistem menerima lokasi awal dari pengguna
2	Get Lokasi Tujuan	Sistem menerima lokasi tujuan dari pengguna
3	Menghitung Bobot Setiap <i>Node</i>	<i>Dijkstra</i> menerima <i>node</i> awal dan tujuan dari sistem kemudian <i>dijkstra</i> menghitung nilai bobot dr setiap <i>node</i>
4	Memilih <i>Node</i> Nilai Bobot Terkecil	<i>Dijkstra</i> mengkalkulasi setiap <i>Node</i> dan memilih <i>node</i> yang memiliki bobot terkecil
5	Menampilkan Hasil Rute	Setelah mendapatkan nilai bobot terkecil <i>dijkstra</i> kemudian menampilkan hasil perhitungannya berupa rute ke sistem

### 3.4.2 Activity Diagram

Diagram aktivitas menggambarkan aliran kerja atau aktivitas dari sebuah sistem, tetapi bukan aktivitas aktor. Diagram aktivitas juga menggambarkan bagaimana alur sistem berawal, pilihan (*decission*) yang mungkin terjadi, dan bagaimana akhir alur sistem tersebut. Berikut ini diagram aktivitas pada sistem informasi yang digunakan.

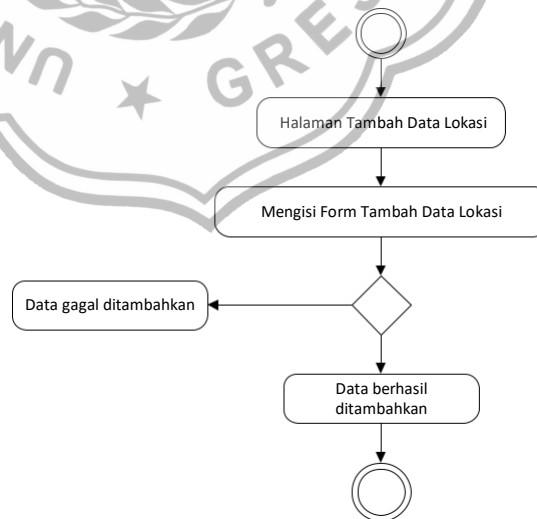
### 1. Activity Diagram Login



**Gambar 3.10** Activity Diagram Login

Diagram diatas menerangkan alur proses *login* pada *admin*, dimulai dari memasukkan *username* dan *password* jika *username* dan *password* benar *login* berhasil, jika *username* dan *password* salah *login* gagal.

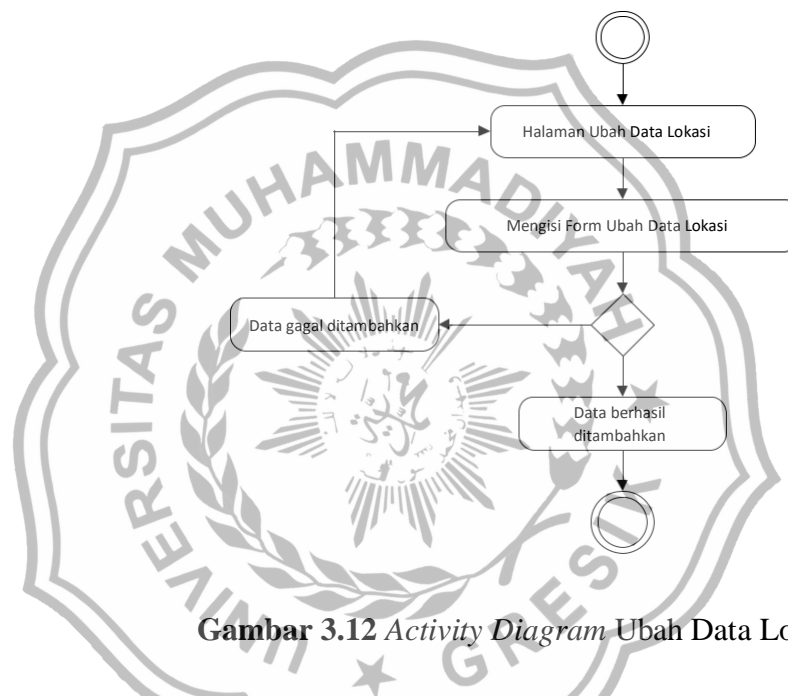
### 2. Activity Diagram Tambah Data Lokasi



**Gambar 3.11** Activity Diagram Tambah Data Lokasi

Diagram diatas menerangkan tentang alur proses tambah data lokasi dimulai dari halaman tambah data lokasi, kemudian mengisi form tambah data lokasi jika data sudah benar data lokasi berhasil di tambah, jika data lokasi salah data lokasi akan gagal ditambahkan.

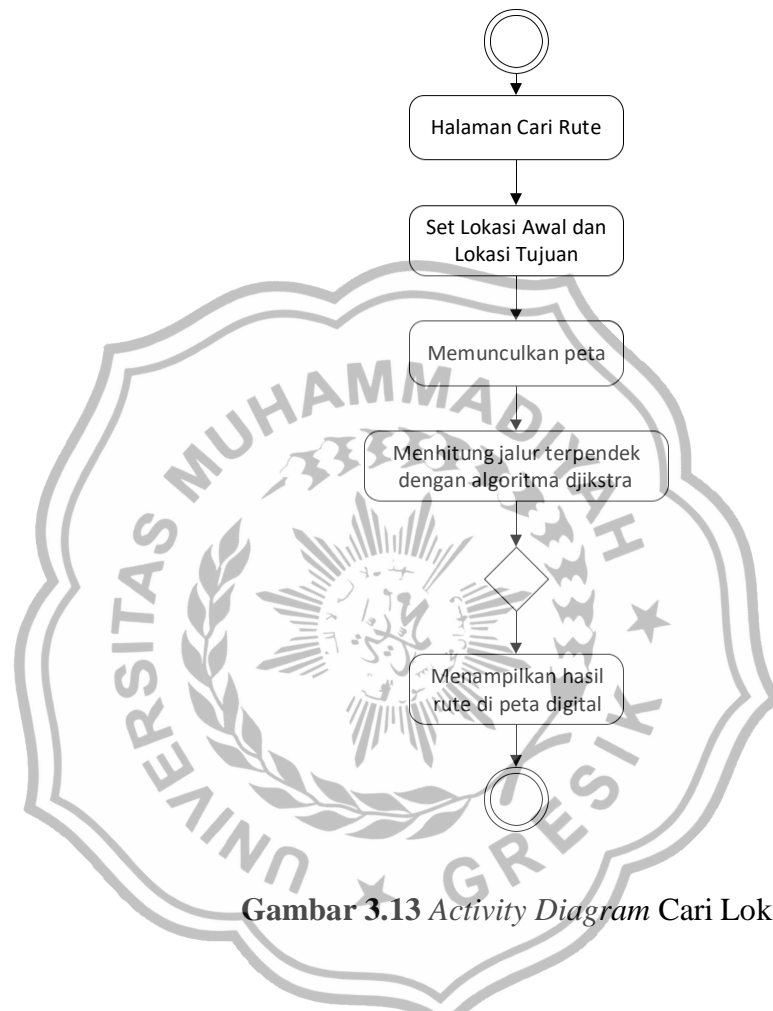
### 3. Activity Diagram Ubah Data Lokasi



**Gambar 3.12** Activity Diagram Ubah Data Lokasi

Diagram diatas menerangkan tentang alur proses ubah data lokasi dimulai dari halaman ubah data lokasi, kemudian mengisi form ubah data lokasi jika data sudah benar data lokasi berhasil di tambah, jika data lokasi salah data lokasi akan gagal ditambahkan.

#### 4. Activity Diagram Cari Rute

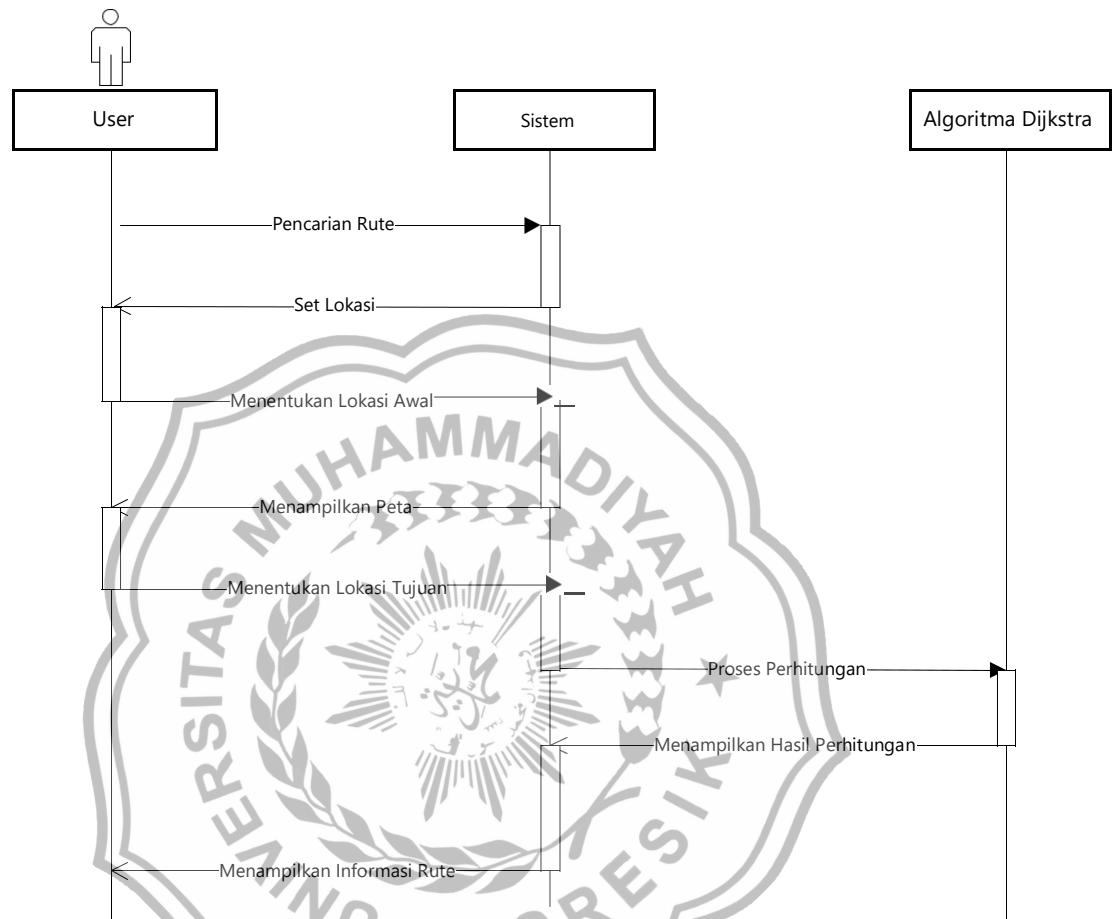


**Gambar 3.13** Activity Diagram Cari Lokasi

Diagram diatas menerangkan tentang alur sistem cari rute rumah sakit dimulai dari cari rute kemudian set lokasi *user* dan lokasi tujuan, dan selanjutnya di proses oleh sistem dan menghasilkan hasil rekomendasi rute dari sistem untuk *user* yang akan ditampilkan di peta digital.



### 3.4.3 Interaction Diagram



**Gambar 3.14** Interaction Diagram User pencarian rute

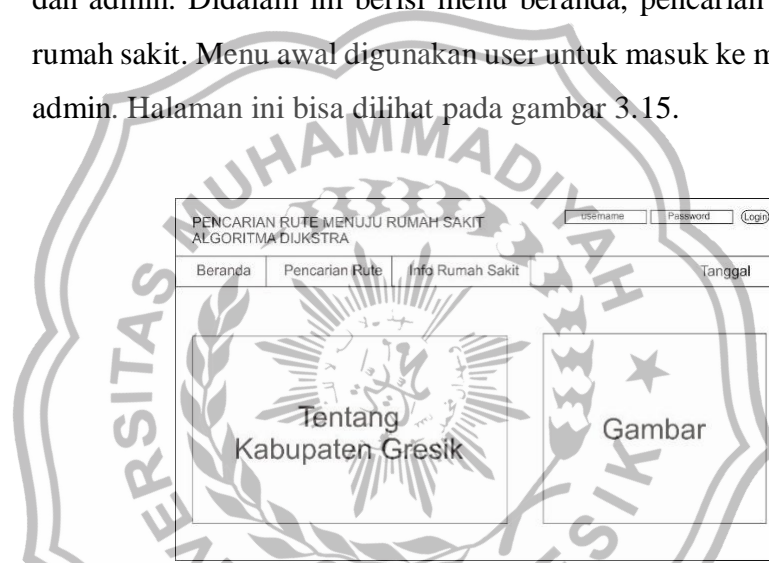
Interaksi Diagram pada Gambar 3.14 menjelaskan interaksi yang dilakukan oleh *user* dengan melakukan pencarian rute, kemudian *user* mendapatkan kembali informasi dari sistem, kemudian sistem mendapatkan lokasi awal dan tujuan yang telah dimasukkan *user*, selanjutnya sistem melakukan proses perhitungan dengan algoritma djikstra, dan djikstra menampilkan hasil rute yang selesai dihitung ke sistem, kemudian *user* mendapat kembali informasi berupa rute yang ditampilkan dalam bentuk peta digital.

### 3.5 Desain Interface

*Interface* adalah bagian yang menghubungkan antara sistem penentuan jenis dengan admin. *Interface* yang digunakan dalam sistem aplikasi pencarian rute menuju rumah sakit adalah sistem yang berbasis web dengan *source code* yang dipakai menggunakan php. Halaman yang dibuat adalah sebagai berikut:

#### 1. Halaman Awal

Halaman Awal merupakan halaman ketika aplikasi diakses oleh *user* dan admin. Didalam ini berisi menu beranda, pencarian rute dan info rumah sakit. Menu awal digunakan user untuk masuk ke menu *user* atau admin. Halaman ini bisa dilihat pada gambar 3.15.



Gambar 3.15 Halaman Awal

#### 2. Halaman Pencarian Rute

Halaman Pencarian Rute merupakan halaman yang digunakan *user* untuk melihat rute menuju rumah sakit. Dalam melakukan pencarian rute, pertama *user* harus menentukan lokasi awal yaitu tepatnya lokasi *user* berada, kemudian *user* harus menentukan lokasi rumah sakit yang dituju dan klik tombol “Lihat Rute” untuk menampilkan informasi terkait rute yang akan dituju. Halaman ini bisa dilihat pada gambar 3.16.

PENCARIAN RUTE MENUJU RUMAH SAKIT ALGORITMA DIJKSTRA			username	Password	(Login)
Beranda	Pencarian Rute	Info Rumah Sakit	Tanggal		
Rute					
Lokasi Awal : <input type="text"/>					
Lokasi Tujuan : <input type="text"/>					
<input type="button" value="Lihat Rute"/> <input type="button" value="Batal"/>					

**Gambar 3.16** Halaman Pencarian Rute

3. Halaman Hasil Pencarian Rute

Halaman Hasil Pencarian Rute adalah halaman yang menampilkan informasi dari hasil pencarian rute. Pada halaman ini menampilkan informasi mengenai lokasi awal, lokasi tujuan, rute perjalanan, jarak tempuh, dan pada tombol “Lihat Peta” digunakan untuk menampilkan rute dalam bentuk peta digital. Halaman ini bisa dilihat pada gambar 3.17.

PENCARIAN RUTE MENUJU RUMAH SAKIT ALGORITMA DIJKSTRA			
		username	Password
		Login	
Beranda	Pencarian Rute	Info Rumah Sakit	Tanggal
Hasil Pencarian Rute			
Lokasi Awal	:		
Lokasi Tujuan	:		
Rute Perjalanan	:		
Jarak Tempuh	:		
Lihat Peta		Lihat Jalur	
Peta			

**Gambar 3.17** Halaman Hasil Pencarian rute

4. Halaman Info Rumah Sakit

Halaman Info Rumah Sakit adalah halaman yang menampilkan informasi terkait rumah sakit yang berupa nama, no telepon, alamat, website, dan foto rumah sakit. Halaman ini bisa dilihat pada gambar 3.18.

PENCARIAN RUTE MENUJU RUMAH SAKIT ALGORITMA DIJKSTRA			
		username	Password
		Login	
Beranda	Pencarian Rute	Info Rumah Sakit	Tanggal
Info Rumah Sakit		Foto	

**Gambar 3.18** Halaman Info Rumah Sakit

5. Halaman admin
6. Halaman Kategori Peta

Halaman Kategori Peta adalah halaman yang digunakan admin untuk menambah, mengedit dan menghapus kategori peta yang berupa kategori tempat awal, dan tempat tujuan. Halaman ini bisa dilihat pada gambar 3.19.

PENCARIAN RUTE MENUJU RUMAH SAKIT ALGORITMA DIJKSTRA						
				username	Password	(Login)
Beranda	Kategori Peta	Peta	Setting Rute	Ubah Password	Logout	
Kategori Peta					Tambah Data	
No	Nama Kategori	Aktif	Jenis	Control		
				edit	Del	

Gambar 3.19 Halaman Kategori Peta

7. Halaman Informasi Peta

Halaman Informasi Peta adalah halaman yang digunakan admin untuk menambah mengedit dan menghapus informasi data rumah sakit berupa nama tempat, deskripsi, koordinat peta, dan foto rumah sakit. Halaman ini bisa dilihat pada gambar 3.20.

PENCARIAN RUTE MENUJU RUMAH SAKIT  
ALGORITMA DIJKSTRA

username Password Login

Beranda Kategori Peta Peta Setting Rute Ubah Password Logout

Data Peta Tambah Data

No	Nama Tempat	Deskripsi	Lat-Lng	Foto	Control
					<input type="button" value="Edit"/> <input type="button" value="Del"/>

Gambar 3.20 Halaman Informasi Peta

8. Halaman *Edit* Data Lokasi Peta

Halaman *edit* data Lokasi peta adalah halaman yang digunakan admin untuk mengedit data Lokasi rumah sakit dimana pada halaman sebelumnya ketika admin mengklik tombol "*edit*" maka halaman ini akan muncul. Halaman ini bisa dilihat pada gambar 3.21.

PENCARIAN RUTE MENUJU RUMAH SAKIT ALGORITMA DIJKSTRA		username	Password	Login	
Beranda	Kategori Peta	Peta	Setting Rute	Ubah Password	Logout
Edit Data Peta					
Map					
Nama Tempat	<input type="text"/>				
Kategori	<input type="text"/>				
Deskripsi	<input type="text"/>				
Latitude	<input type="text"/>	Longitude	<input type="text"/>		
Alamat Lokasi	<input type="text"/>				
Foto	<input type="text"/>				
Upload Gambar	<input type="button" value="Browse"/>				
<input type="button" value="Simpan"/> <input type="button" value="Batal"/>					

**Gambar 3.21** Halaman *Edit* Data Lokasi Peta

#### 9. Halaman *Setting* Rute

Halaman *Setting* Rute adalah halaman yang digunakan admin untuk menambah rute dengan menentukan lokasi awal dan lokasi tujuan pada peta digital sehingga dapat menampilkan titik awal dan titik tujuan. Halaman ini bisa dilihat pada gambar 3.22

PENCARIAN RUTE MENUJU RUMAH SAKIT ALGORITMA DIJKSTRA							
					username	Password	Login
Beranda	Kategori Peta	Peta	Setting Rute	Ubah Password	Logout		
Tambah Rute							
Map							
Lokasi Awal		<input type="text"/>					
Lokasi Tujuan		<input type="text"/>					
Latitude		<input type="text"/>	Longitude		<input type="text"/>		
Alamat Lokasi		<input type="text"/>					
<input type="button" value="Simpan"/> <input type="button" value="Batal"/>							

**Gambar 3.22** Halaman *Setting Rute*

#### 10. Halaman Ubah *Password*

Halaman Ubah *Password* adalah halaman yang digunakan untuk merubah *password* akun. Halaman ini bisa dilihat pada gambar 3.23

PENCARIAN RUTE MENUJU RUMAH SAKIT ALGORITMA DIJKSTRA							
					username	Password	Login
Beranda	Kategori Peta	Peta	Setting Rute	Ubah Password	Logout		
Ubah Password							
Password Anda		<input type="text"/>					
Password Baru		<input type="text"/>					
Ulangi		<input type="text"/>					
<input type="button" value="Simpan"/> <input type="button" value="Batal"/>							

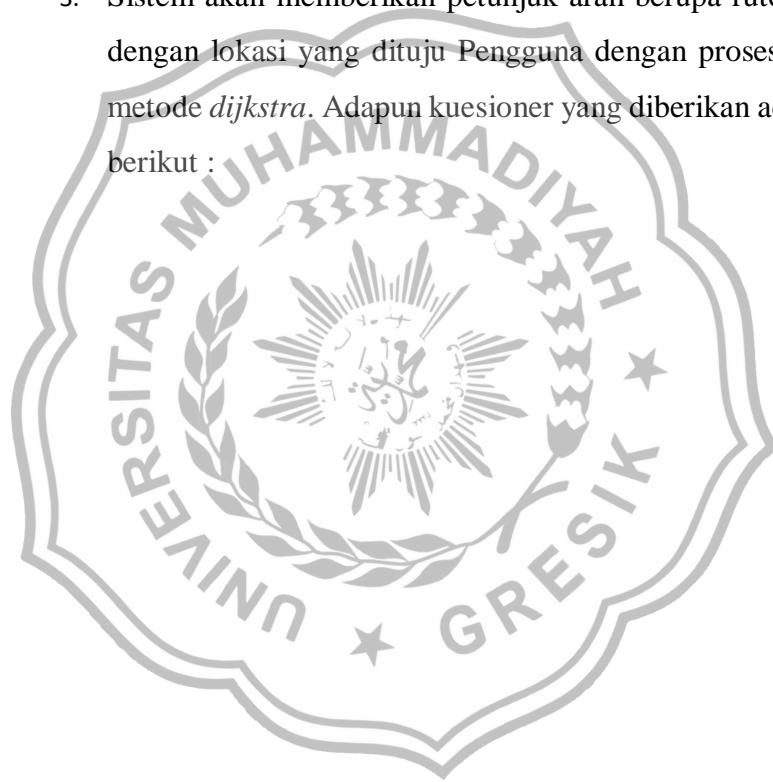
**Gambar 3.23** Halaman Ubah *Password*



### 3.6 Skenario Pengujian

Adapun tahapan dalam pengujian sistem pencarian rute menuju rumah sakit, sebagai berikut:

1. Sistem akan diuji dengan memberikan kuisisioner kepada 20 responden.
2. Respondes menguji sistem dengan memilih salah satu dari dua alternatif jawaban “ya” atau “tidak”.
3. Sistem akan memberikan petunjuk arah berupa rute yang sesuai dengan lokasi yang dituju Pengguna dengan proses perhitungan metode *dijkstra*. Adapun kuesioner yang diberikan adalah sebagai berikut :



**ANGKET PENGUJIAN SISTEM IMPLEMENTASI ALGORITMA *DIJKSTRA*  
PENCARIAN RUTE MENUJU RUMAH SAKIT DI KAB. GRESIK**

NAMA : \_\_\_\_\_

PEKERJAAN : \_\_\_\_\_

INSTANSI : \_\_\_\_\_

Berikan tanda centang (✓) pada salah satu alternatif jawaban yang tersedia

**Pengujian Fungsional Sistem Sebagai *User***

No	Pernyataan	Ya	Tidak
1	Sistem dapat menampilkan peta lokasi rumah sakit		
2	Sistem dapat menampilkan informasi mengenai rumah sakit		
3	Sistem dapat menemukan rute terpendek dan menampilkan detail rute		

**Pengujian *Interface***

No	Pernyataan	Ya	Tidak
1	Tampilan sistem menarik dan <i>user friendly</i>		
2	Sistem memiliki sistem navigasi (cara pengoprasian) yang mudah		
3	Menu sederhana dan tidak membingungkan		

**Responden:**

( \_\_\_\_\_ )

Pada penelitian ini, untuk mengukur kerja *accuracy* sistem yang dibangun dilakukan pengujian *precision*, dan *accuracy*.

**Tabel 3.3** Parameter menghitung *precision* dan *accuracy*

Keterangan	Relevan	Tidak Relevan
Terambil	True positive (tp)	False positive (fp)
Tidak terambil	False negative (fn)	True negative (tn)

Rumus perhitungan *precision*:

Rumus untuk menghitung *Precision* :

$$Precision = \frac{tp}{tp+fp} * 100\%$$

Rumus untuk menghitung *Accuracy* :

$$Accuracy = \frac{tp+tn}{tp+fp+tn+fn} * 100\%$$

Nilai *precision* dan *accuracy* dinyatakan dalam persen. Semakin tinggi kedua nilai tersebut menunjukkan semakin baiknya kinerja *query* pada model ontologi yang dibangun. Evaluasi yang akan dilakukan dalam penelitian ini adalah menghitung nilai dari *precision* dan *accuracy* berdasarkan *query* data ontologi yang berhasil dihasilkan. Penentuan nilai dari *precision* dan *accuracy* harus didapatkan jumlah data yang relevan terhadap suatu objek yang menghasilkan informasi yang dimaksud.

Menurut Rijsbergen (1979) relevansi merupakan sesuatu yang bersifat subjektif. Setiap orang mempunyai perbedaan dalam mengartikan suatu data yang relevan terhadap sebuah topik informasi. Sehingga dalam pelaksanaan pengujian sistem ini dibutuhkan seorang pakar yang dianggap mampu menilai apakah data yang dihasilkan dikatakan relevan atau tidak relevan. (Waluya, Onny Kartika. 2017).

### 3.1 Spesifikasi Pembangunan Sistem

Kebutuhan perangkat keras serta perangkat lunak dari sistem sebagai berikut:

#### 3.1.1 Kebutuhan Perangkat Keras

- a. CPU Intel N4000 setara atau lebih tinggi.
- b. RAM 2Gb setara atau lebih.
- c. HDD 500Gb setara atau lebih.
- d. *Mouse, Keyboard*, dan monitor standar.

#### 3.1.2 Kebutuhan Perangkat Lunak

- a. *OS Windows 7/8/8.1/10*
- b. *Server Apache*
- c. XAMPP v3.1.0

