BAB III
ANALISIS DAN PERANCANGAN SISTEM

III.1 Analisis Sistem

Analisis sistem merupakan kegiatan penguraian suatu sistem informasi yang utuh dan nyata ke dalam bagian-bagian atau komponen-komponen komputer yang bertujuan untuk mengidentifikasi serta mengevaluasi masalah-masalah yang muncul, hambatan-hambatan yang mungkin terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Tahapan yang dilakukan dalam analisis ini yaitu :

1. Analisis Kasus
2. Analisis Struktur Data
3. Analisis Strategi
4. Analisis Algoritma
5. Analisis Kebutuhan Non Fungsional
6. Analisis Data
7. Analisis Kebutuhan Fungsional

III.1.1 Analisis Kasus

Rute terpendek merupakan salah satu solusi ketika kita melakukan suatu perjalanan dari suatu lokasi menuju lokasi lain baik untuk melakukan perjalanan wisata ataupun perjalanan dikota tempat kita tinggal khususnya di kota Tasikmalaya karena dengan rute terpendek, kita dapat menghemat biaya dan menghemat waktu. Namun terkadang kita tidak mengetahui rute terpendek untuk menuju suatu lokasi di kota Tasikmalaya karena banyaknya rute yang terdapat pada kota tersebut.

Perangkat lunak untuk mencari rute terpendek di kota Tasikmalaya sudah ada sebelumnya dengan menggunakan algoritma *ant colony* yang merupakan

penelitian dari Danardi Gunawan **[4]**. Namun pada algoritma *ant colony*, terdapat beberapa kelemahan yaitu:

1. Pencarian optimalnya tergantung dari jumlah iterasi perhitungan yang digunakan.
2. Semakin banyak data yang diolah, tingkat akurasinya pun semakin menurun.
3. Membutuhkan nilai-nilai pada implementasinya.
4. Masih memiliki kesalahan dalam pencarian titik, karena di dalam sistem ini menggunakan *random* yang kurang optimal.

Dengan kelemahan yang ada pada algoritma *ant colony*, diperlukan untuk menganalisis performansi algoritma yang diharapkan dapat lebih baik dari algoritma *ant colony* dalam mencari rute terpendek di kota Tasikmalaya, salah satunya algoritma pencarian beam. Dengan menganalisis algoritma pencarian beam, maka akan diketahui perfomansi algoritma pencarian beam pencarian rute terpendek di kota Tasikmalaya.

III.1.2 Analisis Struktur Data

Graph merupakan struktur data yang tersusun atas simpul-simpul atau titik-titik (Simpul) yang terhubung satu sama lain dengan garis-garis (*edge*). Secara matematis, graph dinyatakan sebagai berikut:

$$G = (V,E)$$

Dimana:

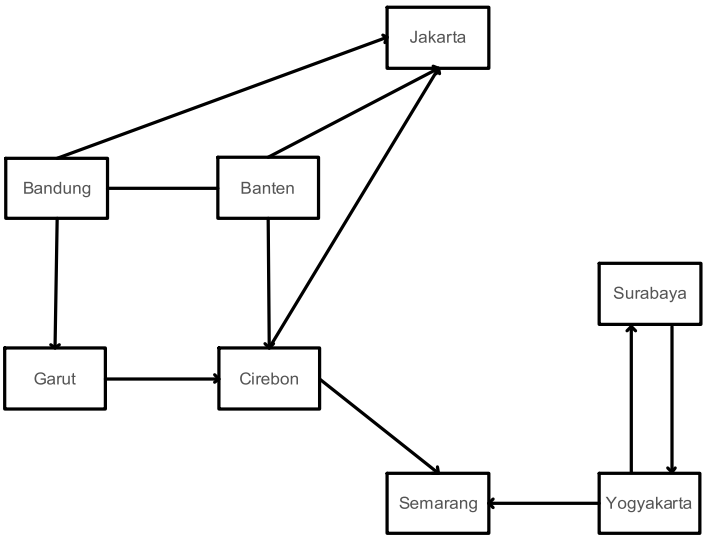
G = Graph

V= Simpul atau vertex, simpul

E = Sisi, busur atau edge

Terdapat banyak sekali jenis graph. Namun secara orientasi arah, graph dapat dibedakan menjadi dua, pertama graph berarah. Graph berarah merupakan graph yang mana sisinya memiliki arah. Sisi ini biasa disebut busur (*arc*). Pada

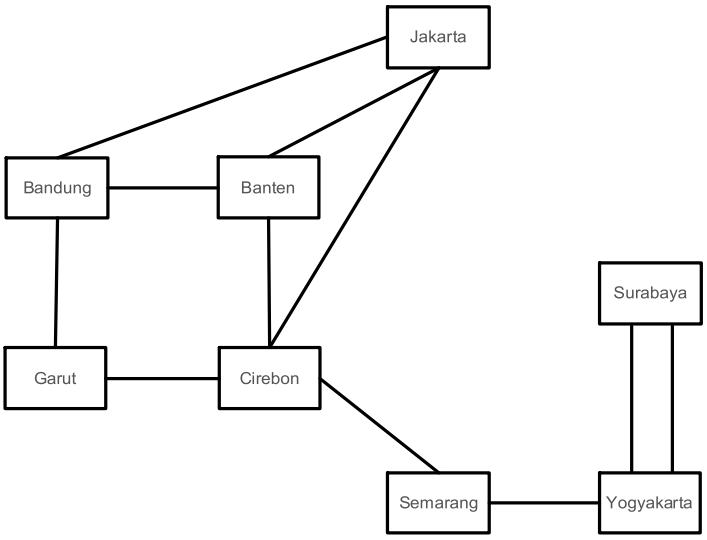
graph berarah, urutan pasangan simpul sangat diperhatikan karena dapat menyatakan hal yang berbeda.



Gambar III-1 Graph Berarah

Pada gambar Gambar III-1 Graph Berarah, dapat dilihat bahwa dua busur yang menghubungkan (Yogyakarta, Surabaya) dan (Surabaya, Yogyakarta) menyatakan hal yang berbeda. Untuk busur yang menghubungkan simpul Yogyakarta dan Surabaya, simpul Yogyakarta dinamakan sebagai simpul awal (*initial suksesor*) dan simpul Surabaya dinamakan sebagai simpul terminal. Sebaliknya, pada busur yang menghubungkan simpul Surabaya dan Yogyakarta, simpul Surabaya merupakan simpul awal dan simpul Yogyakarta adalah simpul terminal.

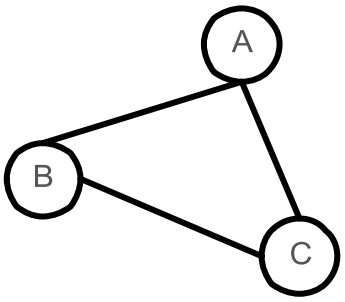
Dan yang kedua adalah graph tidak berarah. Graph tidak berarah merupakan graph yang sisinya tidak memiliki arah (kebalikan dari graph berarah). Pada graph tidak berarah, urutan pasangan simpul yang berhubungan tidak diperhatikan.



Gambar III-2 Graph Tidak Berarah

Dari gambar Gambar III-2 Graph Tidak Berarah dapat dilihat bahwa pasangan simpul (Bandung, Garut) dan (Garut, Bandung) menyatakan hal yang sama karena busur yang ada tidak memiliki orientasi arah. Graph dapat dinyatakan sebagai sebuah struktur data, atau secara spesifik dinamakan sebagai ADT (abstract data type) yang terdiri dari kumpulan simpul dan sisi yang membangun hubungan antar simpul. Secara umum terdapat dua macam representasi dari struktur data graph, yang pertama *adjacency list* dan yang kedua *adjacency matrix.*

Adjacency list merupakan bentuk representasi dari busur atau sisi dalam suatu graph sebagai satu senarai. Simpul yang terhubung oleh busur atau sisi dinyatakan sebagai simpul yang terkait. Dalam implementasinya, tabel hash digunakan untuk menghubungkan simpul dengan senarai yang berisi simpul-simpul yang saling terkait.



Gambar III-3 *Adjency List*

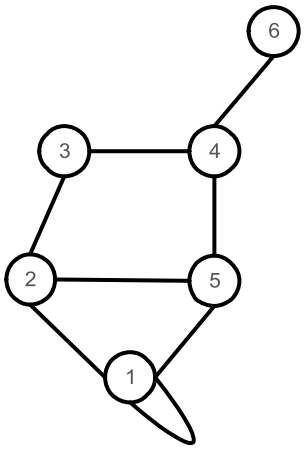
Graph pada Gambar III-3 *Adjency List* dapat digambarkan sebagai senarai {A,B}, {A,C} dan {B,C}. Representasi dari *adjacency list* dapat dilihat pada Tabel III-1 Representasi Adjacency List.

Tabel III-1 Representasi Adjacency List

Simpul	Kedekatan	Senarai Dari Simpul yang Berhubungan
A	Dekat dengan	B,C
B	Dekat dengan	A,C
C	Dekat dengan	A,B

Salah satu kekurangan dari representasi *adjacency list* adalah tidak adanya tempat untuk menyimpan nilai yang melekat pada sisi yang mana contoh nilai ini adalah nilai jarak simpul.

Yang kedua adalah *adjacency matrix. Adjacency matrix* merupakan representasi matriks $n \ x \ n$ yang menyatakan hubungan antar simpul dalam graph. Kolom dan baris dari matriks merepresentasikan simpul-simpul, dan nilai entri dalam matriks ini menyatakan hubungan antar simpul. Berikut ini adalah representasi *adjacency matrix* dari graph tak berarah.



Gambar III-4 Graph Tak Berarah

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Gambar III-5 Representasi Adjacency Matrix Dari Gambar III-4

Kelebihan dari adjacency matrix ini adalah elemen matriksnya dapat diakses langsung melalui indeks. Sedangkan kekurangannya bila graph memiliki jumlah sisi atau busur yang relatif sedikit, kebutuhan ruang hash tabel untuk matriks menjadi boros dan tidak efisien karena komputer menyimpan elemen yang tidak perlu.

III.1.3 Analisis Strategi

Algoritma pencarian beam merupakan pencarian heuristik yang menyimpan dan mengembangkan simpul yang paling menjanjikan, sedangkan simpul-simpul yang tidak terpilih langsung dihapus. Dalam menemukan solusi, langkah-langkah yang dilakukan oleh algoritma pencarian beam adalah mengecek seluruh simpul tetangga, menentukan simpul selanjutnya untuk diperluas pencariannya dan mendapatkan tujuan.

Algoritma ini merupakan optimasi dari algoritma *best-first search*. Algoritma *best-first search* merupakan gabungan dengan mengambil kelebihan dari algoritma *breadth-first search* dan algoritma *dept-first search*. Pada setiap langkah proses algoritma *best-first search*, kita akan memilih simpul-simpul dengan menggunakan fungsi heuristik yang memadai pada setiap simpul yang kita pilih dengan menggunakan aturan-aturan tertentu untuk menghasilkan penggantinya. Algoritma *best-first search* memperbolehkan pencarian ke level yang lebih rendah jika nilai heuristik di simpul level yang lebih tinggi memiliki nilai yang lebih buruk. Cara kerja algoritma *best-first search* adalah sebagai berikut:

1. Asumsikan senarai untuk simpul yang sudah dibangkitkan namun belum diuji sebagai list OPEN dan senarai untuk simpul yang telah diuji sebagai CLOSE.
2. Masukan simpul awal pada list OPEN.
3. Kerjakan langkah-langkah berikut hingga tujuan ditemukan atau antrian OPEN sudah kosong:
   a. Ambil simpul terbaik dari list OPEN.
   b. Bangkitkan semua suksesor dari simpul yang ada di list OPEN.
   c. Untuk setiap suksesor kerjakan:
      - Jika simpul tersebut belum pernah dibangkitkan, periksa simpul tersebut dan masukkan ke OPEN;
      - Jika simpul tersebut sudah pernah dibangkitkan, ubah parent jika lintasan baru lebih menjanjikan. Hapus simpul tersebut dari antrian OPEN.

III.1.4 Analisis Algoritma

Analisis algoritma yang dilakukan dalam penelitian ini adalah meneliti bagaimana cara kerja dan performansi algoritma pencarian beam dalam mencari rute terpendek di kota Tasikmalaya. Tahapan yang dilakukan dalam menganalisis algoritma pencarian beam adalah:

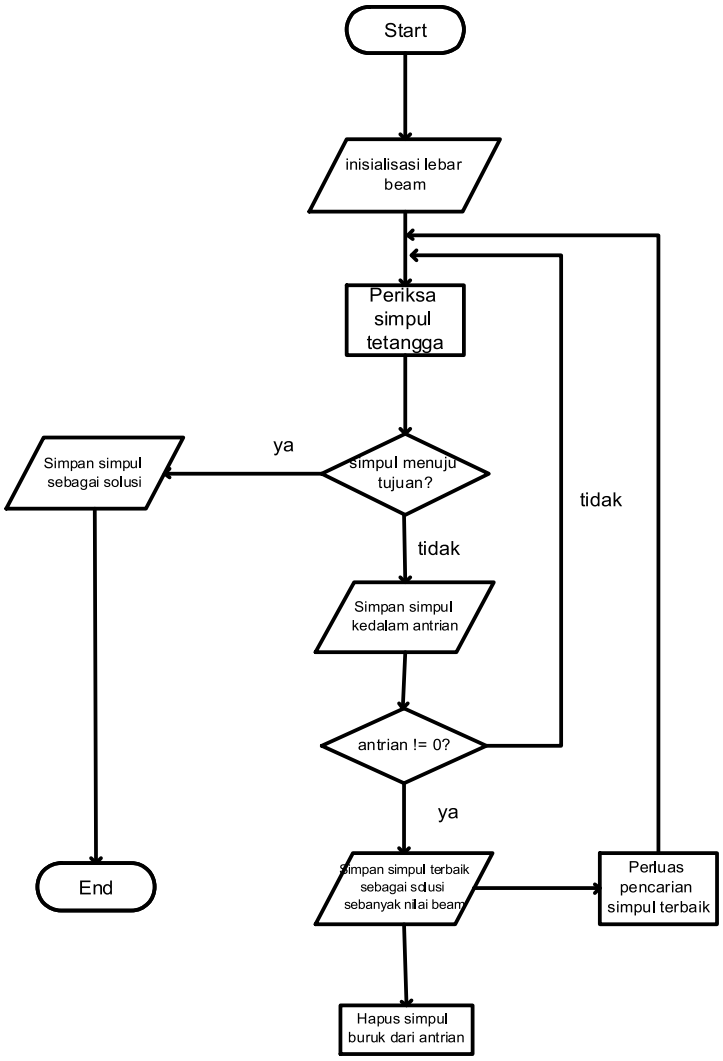
1. Langkah kerja algoritma beam
2. Flowchart
3. Pseudocode

III.1.4.1 Algoritma Pencarian beam

Algoritma pencarian beam merupakan algoritma yang hanya menyimpan nilai terbaik dan menghapus simpul yang tidak digunakan karena memiliki nilai yang besar. Cara kerja algoritma pencarian beam adalah sebagai berikut:

1. Masukan besar nilai beam.
2. Periksa simpul tetangga dari simpul awal hingga tidak ada lagi simpul yang dapat diperiksa dan simpan semua simpul tersebut ke dalam antrian. Bila pada pencarian ini terdapat simpul tujuan, simpan simpul sebagai solusi dan pencarian selesai.
3. Bandingkan nilai dari simpul-simpul pada array antrian. Simpul dengan nilai terkecil adalah simpul yang akan dipilih dan di simpan sebagai solusi dan simpul yang disimpan tersebut sebesar nilai beam.
4. Bila solusi belum ditemukan, pencarian akan dilakukan lagi dengan memperluas pencarian dari simpul-simpul yang bertetangga dengan simpul yang dipilih sebelumnya hingga tujuan ditemukan.

Algoritma beam memiliki kemampuan mengurangi konsumsi memori yang dibutuhkan dalam melakukan pencarian yang mana kemampuan ini didapatkan karena pemangkasan simpul-simpul yang tidak digunakan. Namun dengan pemangkasan ini, memungkinkan tidak didapatkannya solusi terbaik karena pemangkasan tersebut.



Gambar III-6 Flowchart Algoritma Pencarian beam

Pseudocode algoritma pencarian beam dapat dilihat pada Tabel III-2 Pseudocode Algoritma Pencarian beam.

Tabel III-2 Pseudocode Algoritma Pencarian beam

No	Algoritma Pencarian Beam
1	**Procedure** Beam()
2	{IS : jumlah simpul dan besar nilai beam telah didapatkan
3	FS : Mendapatkan solusi terbaik dari pencarian algoritma beam}
4	
5	**Kamus**
6	const maxsize = 100
7	type solusi : array[0..maxsize] of integer

No	Algoritma Pencarian Beam
8	type antrian : array[0..maxsize] of integer
9	beam, i, j, h : integer
10	**Algoritma**
11	i ← 0
12	input beam
13	**while**(solusi[i] != null) **do**
14	j ← 0
15	**for** j = 0 to j < beam **do**
16	**if**(simpul = solusi[j]) then
17	solusi[j] ← simpul
18	j← beam-1
19	**else**
20	antrian[j] ← simpul
21	endif
22	j ← j+1
23	**endfor**
24	
25	j←0
26	**while** (antrian[j] != null) **do**
27	compare(h)
28	simpul ← h
29	delete(antrian[j])
30	solusi[j] ← simpul
31	j ← j+1
32	**endwhile**
33	i ←i+1
34	**endwhile**
35	**endprocedure**

Untuk perhitungan Big-O algoritma pencarian beam, dapat dilihat pada Tabel III-3 Perhitungan Big-O Algoritma Beam.

Tabel III-3 Perhitungan Big-O Algoritma Beam

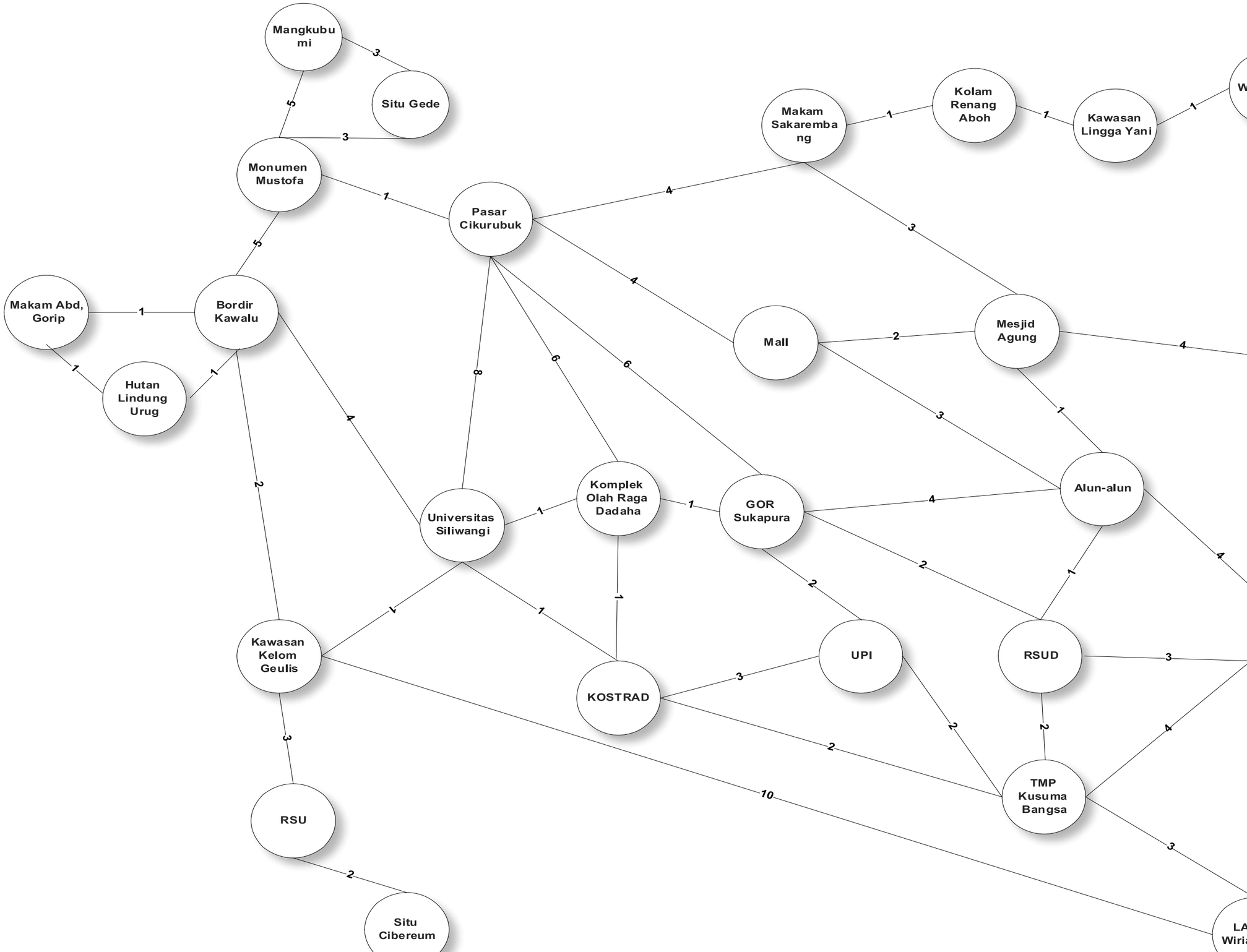
No	Pseudocode	Nilai Big-O
1	i ← 0	O(1)
2	input beam	O(1)
3	**while**(solusi[i] != null) **do**	O(n)
4	j ← 0	O(1)
5	**for** j = 0 to j < beam **do**	O(n)
6	**if**(simpul = solusi[j]) then	O(1)
7	solusi[j] ← simpul	O(1)
8	j← beam-1	O(1)
9	antrian[j] ← simpul	O(1)

No	Pseudocode	Nilai Big-O
10	`j ← j+1`	O(1)
11	`j←0`	O(1)
12	**`while`** `(antrian[j] != null)` **`do`**	O(n)
13	`compare(h)`	O(1)
14	`simpul ← h`	O(1)
15	`delete(antrian[j])`	O(1)
16	`solusi[j] ← simpul`	O(1)
17	`j ← j+1`	O(1)
18	`i ←i+1`	O(1)

Berdasarkan hasil perhitungan performansi algoritma pencarian beam menggunakan notasi Big-O, didapat kompleksitas waktu dengan **$O(n^2)$**.

III.1.4.2Kasus Algoritma Pencarian beam

Seseorang ingin melakukan wisata dari Karang Resik menuju sebuah Mall di kota Tasikmalaya dengan peta lokasi dapat dilihat pada Gambar III-7 Lokasi Wisata Tasikmalaya. Pencarian dilakukan dengan menggunakan nilai beam sebesar 2.


Mangkubumi
Situ Gede
Monumen Mustofa
Pasar Cikurubuk
Makam Sakarembang
Kolam Renang Aboh
Kawasan Lingga Yani
Makam Abd, Gorip
Bordir Kawalu
Hutan Lindung Urug
Mall
Mesjid Agung
Universitas Siliwangi
Komplek Olah Raga Dadaha
GOR Sukapura
Alun-alun
Kawasan Kelom Geulis
KOSTRAD
UPI
RSUD
TMP Kusuma Bangsa
RSU
Situ Cibereum

Langkah Peancarian:

1. Pencarian dimulai dengan menelusuri seluruh lokasi yang bertetangga dengan Karang Resik, dimana simpul yang bertetangga pada kondisi ini adlah Terminal Pancasila dan Tugu Koperasi. Karena hanya dua simpul yang bertetangga dan nilai beam yang digunakan adalah dua, maka kedua lokasi tersebut merupakan simpul terbaik yang akan disimpan ke dalam solusi dan diperluas pencariannya.

Tabel III-4 Kondisi Awal Penelusuran

Simpul Yang Diperiksa	Urutan Simpul
Karang Resik	(Karang Resik)

Tabel III-5 Kondsi Penelusuran Pertama

Simpul Yang Diperiksa	Urutan Simpul
Karang Resik	(Karang Resik)
	(Karang Resik, Terminal Pancasila), (Karang Resik, Tugu Koperasi)

2. Pencarian dilanjutkan dengan memperluas pencarian dengan memeriksa seluruh simpul yang bertetangga dengan Terminal Pancasila dan Tugu Koperasi. Pada kondisi ini, lokasi yang bertetangga dengan Terminal pancasila adalah Alun-alun, RSUD, TMP Kusuma Bangsa. Sedangkan yang bertetangga dengan Tugu Koperasi adalah Batik Tasik dan Karang Resik. Dengan memperhatikan jarak, maka lokasi yang diambil adalah Batik Tasik dan RSUD.

Tabel III-6 Kondisi Penelusuran ke Dua

Simpul Yang Diperiksa	Urutan Simpul
Karang Resik dan Tugu Koperasi	(Karang Resik, Terminal Pancasila), (Karang Resik, Tugu Koperasi)
	(Karang Resik, Terminal Pancasila, RSUD), (Karang Resik, Tugu Koperasi, Batik Tasik)

3. Pencarian dilanjutkan dengan memperluas pencarian dengan memeriksa seluruh simpul yang bertetangga dengan Batik Resik dan RSUD. Pada kondisi ini, lokasi yang bertetangga dengan Batik Resik adalah Payung Tasik dan Mesjid Agung. Sedangkan yang bertetangga dengan RSUD adalah Alun-alun, GOR Sukapura, Terminal Pancasila dan TMP Kusuma Bangsa. Berdasarkan jarak, maka lokasi yang dipilih adalah Alun-Alun dan Payung Tasik.

Tabel III-7 Kondisi Penelusuran ke Tiga

Simpul Yang Diperiksa	Urutan Simpul
RSUD dan Batik Tasik	(Karang Resik, Terminal Pancasila, RSUD), (Karang Resik, Tugu Koperasi, Batik Tasik)
	(Karang Resik, Terminal Pancasila, RSUD, Alun-Alun), (Karang Resik, Tugu Koperasi, Batik Tasik, Payung Tasik)

4. Pencarian dilanjutkan dengan memperluas pencarian dengan memeriksa seluruh lokasi yang bertetangga dengan Payung Tasik dan Alun-alun. Pada kondisi ini, lokasi yang bertetangga dengan Payung Tasik adalah Waterboom. Sedangkan yang bertetangga dengan Alun-alun adalah GOR Sukapura, Mall, Mesjid Agung dan RSUD. Berdasarkan jarak, maka lokasi yang dipilih adalah Mesjid Agung dan Waterboom.

Tabel III-8 Kondisi Penelusuran ke Empat

Simpul Yang Diperiksa	Urutan Simpul
Alun-alun dan Payung Tasik	(Karang Resik, Terminal Pancasila, RSUD), (Karang Resik, Tugu Koperasi, Batik Tasik)
	(Karang Resik, Terminal Pancasila, RSUD, Alun-Alun, Mesjid Agung), (Karang Resik, Tugu Koperasi, Batik Tasik, Payung Tasik, Waterboom)

5. Pencarian dilanjutkan dengan memperluas pencarian dengan memeriksa seluruh lokasi yang bertetangga dengan Waterboom dan Masjid Agung. Pada kondisi ini, lokasi yang bertetangga dengan Waterboom adalah Kawasan Lingga Yani. Sedangkan lokasi yang bertetangga dengan Masjid Agung adalah Alun-alun, Batik Tasik, Makam Sakarembang dan Mall. Karena lokasi tujuan telah ditemukan pada penelusuran dari Mesjid Agung, maka pencarian di hentikan. Jadi rute terpendek dari Karang Resik menuju Mall adalah melalui Terminal Pancasila, menuju RSUD, menuju Alun-alun, menuju Mesjid Agung hingga sampai di lokasi tujuan yaitu Mall.

III.1.5 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak diperlukan untuk mempelajari kebutuhan dan menetapkan fungsi perangkat lunak. Pada spesifikasi perangkat lunak disini dibagi menjadi dua yaitu SKPL-F (Spesifikasi Perangkat Lunak Fungsional) dan SKPL-NF (Spesifikasi Perangkat Lunak Non Fungsional. Pada aplikasi simulasi pencarian rute terpendek di kota Tasikmalaya, spesifikasi kebutuhan perangkat lunak fungsional dapat dilihat pada Tabel III-9 Spesifikasi Kebutuhan Perangkat Lunak Fungsional dan spesifikasi kebutuhan perangkat lunak non fungsional dapat dilihat pada

Tabel III-10 Spesifikasi Kebutuhan Perangkat Lunak Non **Fungsional**.

.

Tabel III-9 Spesifikasi Kebutuhan Perangkat Lunak Fungsional

Kode	Kebutuhan
SKPL-F001	Perangkat lunak dapat menerima inputan lokasi awal dan destinasi lokasi tujuan.
SKPL-F002	Perangkat lunak dapat mengukur waktu proses pencarian rute terpendek.
SKPL-F003	Perangkat lunak dapat menampilkan hasil pencarian rute terpendek.

Tabel III-10 Spesifikasi Kebutuhan Perangkat Lunak Non Fungsional

Kode	Kebutuhan
SKPL-NF001	Perangkat lunak berbasis mobile Android
SKPL-NF002	Sistem operasi minimal android 2.3
SKPL-NF003	Bahasa pemrograman yang digunakan adalah Java
SKPL-NF004	Pemetaan menggunakan Google Map
SKPL-NF005	*Web service* hanya menyimpan nama lokasi dan koordinat lokasi

III.1.6 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional adalah langkah dimana seorang pembangun perangkat lunak (*software developer*) menganalisis sumber daya yang akan digunakan dan menggunakan perangkat lunak yang dibangun. Perangkat keras dan perangkat lunak yang dimiliki harus sesuai dengan kebutuhan atau permintaan, sehingga dapat ditentukan kompabilitas aplikasi yang dibangun terhadap sumber daya yang ada.

Apabila terjadi ketidakcocokan antara fakta dan kebutuhan maka perlu dilakukan penyesuaian fakta terhadap kebutuhan yang ada. Apabila kebutuhan tidak dipenuhi maka sistem yang dibangun tidak akan berjalan baik sesuai yang diharapkan.

Analisis kebutuhan non fungsional yang dilakukan dibagi dalam tiga tahap, yaitu:

1. Analisis perangkat keras (hardware).
2. Analisis perangkat lunak (software).
3. Analisis pengguna (user).

III.I.6.1Analisis Kebutuhan Perangkat Keras

Perangkat keras yang digunakan untuk membangun aplikasi simulasi pencarian rute terpendek di kota Tasikmalaya memiliki spesifikasi sebagai berikut:

1. Processor 2.5GHz

2. VGA 2 Gb
3. RAM 4 Gb
4. Harddisk 500 Gb

Sedangkat perangkat lunak minimal agar dapat menjalankan aplikasi simulasi pencarian rute terpendek di kota Tasikmalaya adalah sebagai berikut:

1. Processor 800 Hz
2. RAM 512 Mb
3. Memori kosong 5 Mb
4. Sistem operasi Android 2.3

III.I.6.2 Analisis Kebutuhan Perangkat Lunak

Adapun perangkat lunak yang digunakan untuk membangun aplikasi simulasi pencarian rute terpendek di kota Tasikmalaya adalah sebagai berikut:

1. Eclipse Indigo
2. Macromedia Dreamweaver 5
3. Wamp Server 5

III.I.6.3 Analisis Kebutuhan Perangkat Pikir

Analisis kebutuhan perangkat pikir diperlukan agar kemampuan perangkat lunak yang dibangun menjadi jelas. Beberapa analisa dan kebutuhan yang berkaitan dengan perangkat lunak yang akan dibangun nanti yaitu analisa dan kebutuhan pengguna. Adapun yang diperlukan pada aplikasi android.

1. Mengerti penggunaan mobile dengan sistem operasi Android.
2. Mengerti secara umum penggunaan internet pada mobile Android.

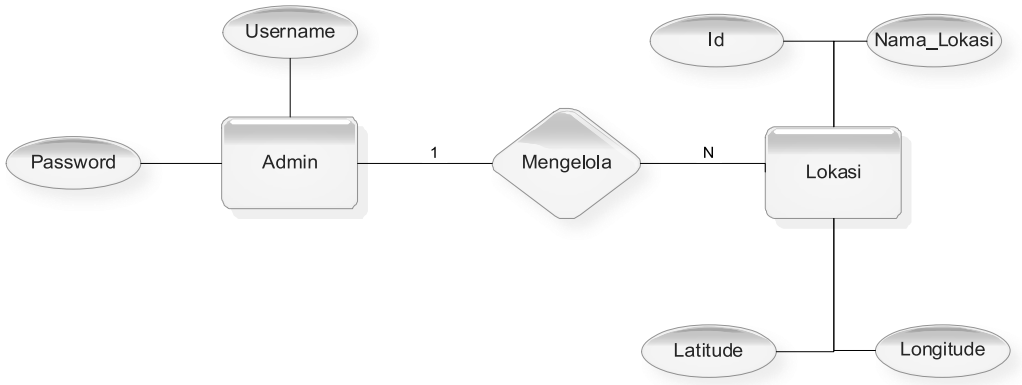
Sedangkan kebutuhan untuk *web service* adalah sebagai berikut:

1. Mengerti menggunakan komputer.

2. Dapat mengolah data yang dengan fitur yang disediakan didalam web.

III.1.7 Analisis Data / File

Analisis data adalah kegiatan menganalisis data yang akan diolah dan disimpan dalam database. Dalam analisis ini direpresentasikan darimana data berasal dan atribut dari data tersebut. Berikut adalah ERD dari basis data yang digunakan dalam aplikasi simulasi pencarian rute terpendek di kota Tasikmalaya yang dapat dilihat pada Gambar III-8 Entitas Aplikasi Simulasi Pencarian Rute Terpendek di Kota Tasikmalaya.



Gambar III-8 Entitas Aplikasi Simulasi Pencarian Rute Terpendek di Kota Tasikmalaya

Penjelasan untuk entitas-entitas pada gambar Gambar III-8 adalah sebagai berikut.

Tabel III-11 Entitas ERD

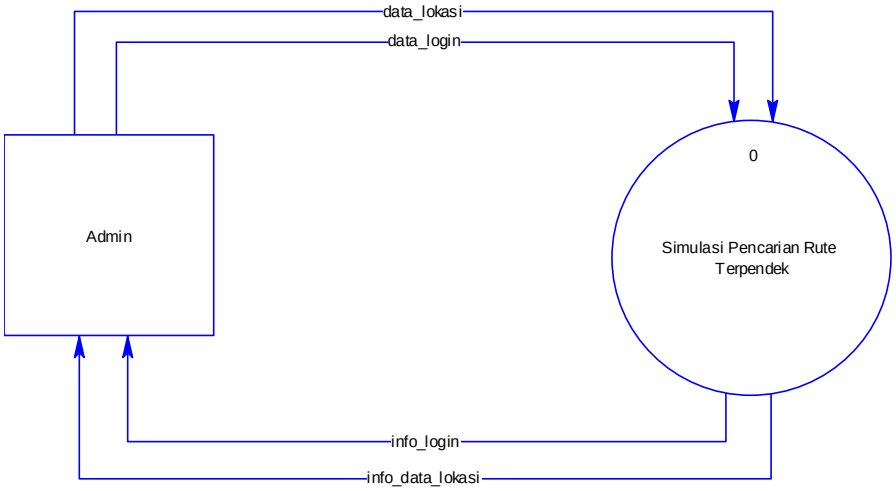
No	Entitas	Atribut	Keterangan
1.	Admin	Username	Nama pengguna admin
		Password	Sandi admin
2.	Lokasi	Id	Nilai kunci data lokasi
		Nama_Lokasi	Nama lokasi
		Latitude	Garis lintang
		Longitude	Garis Bujur

III.1.8 Analisis Kebutuhan Fungsional

Analsis kebutuhan fungsional mendefinisikan aksi dasar yang ada dalam perangkat lunak yang dibangun untuk menerima dan memproses masukan dan menghasilkan keluaran.

III.1.8.1 Diagram Konteks

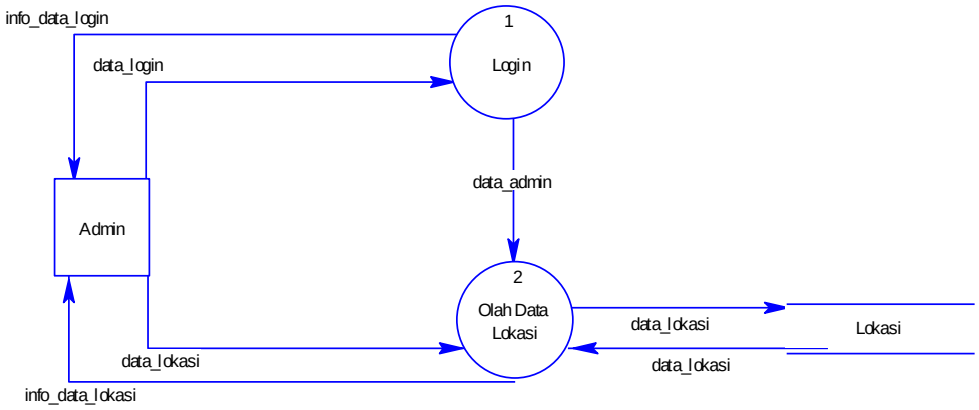
Diagram konteks adalah diagram yang menggambarkan secara umum yang menjadi masukkan, proses, dam keluaran yang terjadi pada sebuah sistem. Diagram konteks untuk perangkat lunak yang akan dibangun sebagai berikut :



Gambar III-9 Diagram Konteks *Web Service*

III.1.8.2 DFD Level 1 *Web service*

Pada DFD level 1 terdapat dua proses utama yaitu proses login dan proses olah data lokasi. Pada proses login, admin memasukan username dan password agar dapat melakukan pengolahan data lokasi sedangkan pada proses olah data lokasi, admin dapat melakukan proses tambah data lokasi, ubah data lokasi dan hapus data lokasi yang akan dijelaskan secara lebih terinci pada DFD level selanjutnya. Untuk lebih jelas mengenai DFD level 1 *web service*, proses dapat dilihat pada Gambar III-10 DFD Level 1 *Web Service*.



Gambar III-10 DFD Level 1 *Web Service*

III.1.8.3 DFD Level 2 *Web Service*

Pada DFD level 2 *web service*, terdapat tiga proses utama yaitu tambah data lokasi, ubah data lokasi dan hapus data lokasi.

1. Tambah Data Lokasi

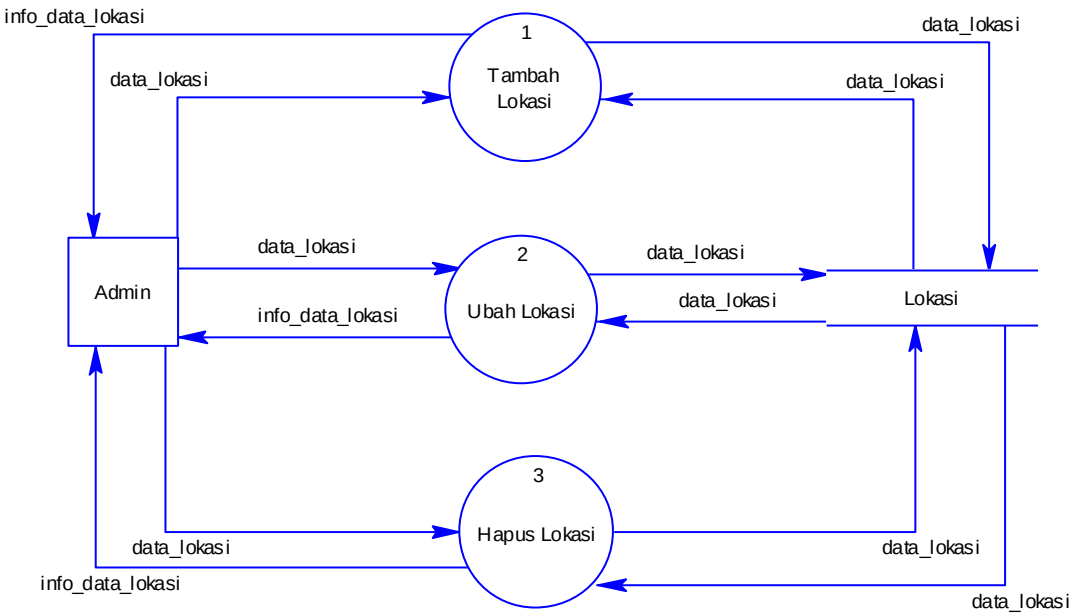
   Pada Proses ini, admin menambahkan data lokasi dengan menginputkan nama lokasi, latitude dan longitude lokasi.
2. Ubah Data Lokasi

   Pada Proses ini, admin mengubah data lokasi dengan menginputkan nama lokasi, latitude dan longitude lokasi yang baru.

3. Hapus Data Lokasi

   Pada Proses ini, admin menghapus data lokasi.

Untuk lebih jelas, dapat dilihat pada Gambar III-11 DFD Level 2 *Web Service*.



Gambar III-11 DFD Level 2 *Web Service*

Berikut ini adalah spesifikasi proses yang mana spesifikasi proses ini akan menjelaskan proses-proses pada DFD dengan lebih terperinci. Spesifikasi proses pada aplikasi simulasi ini dapat dilihat pada Tabel III-12 Spesifikasi Proses.

Tabel III-12 Spesifikasi Proses

NO	Proses	Keterangan
1	No Proses	1
	Nama Proses	Login
	Input	Data Login
	Output	Info Login
	Destination (Tujuan)	Admin
	Logika Proses	Jika Username dan Password benar maka admin masuk ke sistem, jika salah akan muncul info login invalid
2	No Proses	2
	Nama Proses	Olah Lokasi Data
	Input	Data Lokasi
	Output	Info data lokasi
	Destination (Tujuan)	Admin
	Logika Proses	1. Admin dapat mengelola data lokasi 2. Sistem menyimpan semua data-data lokasi baik penambahan, pengurangan dan penghapusan data

3	No Proses	2.1
	Nama Proses	Tambah Lokasi
	Input	Data Lokasi
	Output	Info Data Lokasi
	Destination (Tujuan)	Admin
	Logika Proses	1. Admin menambahkan data lokasi 2. Sistem menyimpan data yang ditambahkan oleh admin
4	No Proses	2.2
	Nama Proses	Ubah Lokasi
	Input	Data Lokasi
	Output	Info Data Lokasi
	Destination (Tujuan)	Admin
	Logika Proses	1. Admin mengubah data lokasi 2. Sistem menyimpan perubahan data yang dilakukan oleh admin
5	No Proses	2.3
	Nama Proses	Hapus Lokasi
	Input	Data Lokasi
	Output	Info Data Lokasi
	Destination (Tujuan)	Admin
	Logika Proses	1. Admin menghapus data lokasi 2. Sistem menghapus data yang ingin dihilangkan oleh admin

III.1.8.4 Kamus Data DFD

Kamus data dapat membantu mendefinisikan data yang mengalir dengan lengkap diantara proses, penyimpanan dan entitas luar. Data tersebut dapat berupa masukan ataupun keluaran dari proses sistem. Kamus data pada aplikasi simulasi pencarian rute terpendek di kota Tasikmalaya dapat dilihat padaTabel III-13 Kamus Data DFD.

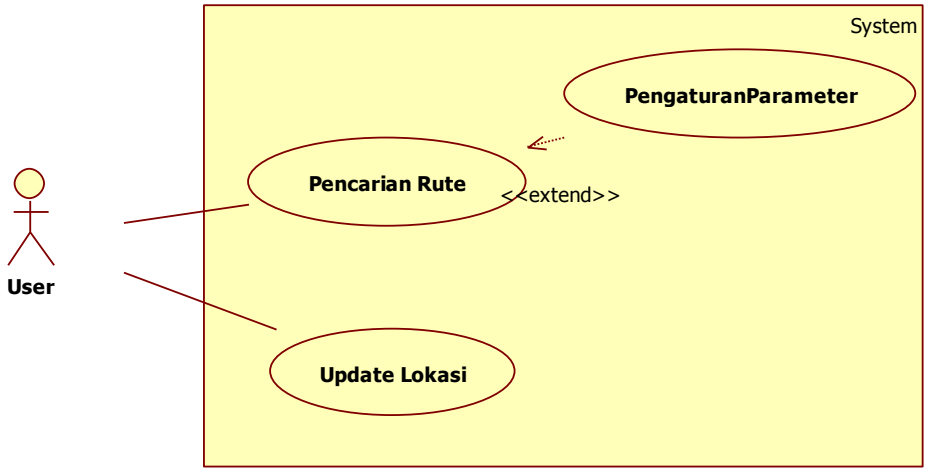
Tabel III-13 Kamus Data DFD

NO.	Data	Keterangan
1.	**Nama**	data_login
	Deskripsi	Bersisi data username dan password untuk admin
	Struktur Data	username+password
	Username	[A-Z|a-z|0-9]
	Password	[A-Z|a-z|0-9]
2	**Nama**	data_lokasi
	Deskripsi	Berisi data lokasi seperti id lokasi, nama

		lokasi, dan koordinat lokasi
	Struktur Data	id,nama_lokasi, latitude, longitude
	Id	[0-9]
	nama_lokasi	[A-Z|a-z|0-9]
	Latitude	[0-9]
	Longitude	[0-9]

III.1.8.5 Usecase Diagram Aplikasi Simulasi Android

Use case merupakan gambaran skenario dari interaksi antara *user* dengan sistem. Sebuah diagram use case menggambarkan hubungan antara aktor dan kegiatan yang dapat dilakukan terhadap aplikasi. Berikut adalah *usecase* model untuk aplikasi simulasi android.



Gambar III-12 Usecase Diagram Aplikasi Simulasi

III.1.8.5.1 Definisi Aktor

Definisi aktor berfungsi menjelaskan peran setiap aktor yang terdapat pada use case. Definisi aktor pada aplikasi simulasi ini adalah sebagai berikut.

Tabel III-14 Definisi Aktor

No.	Aktor	Deskripsi
1	User	Pengguna yang menggunakan aplikasi simulasi pencarian rute terpendek.

III.1.8.5.2 Definisi Use Case

Definisi Use Case berfungsi untuk menjelaskan proses yang terdapat pada setiap use case. Definisi usecase pada aplikasi simulasi ini dapat dilihat padaTabel III-15 Definisi Use Case.

Tabel III-15 Definisi Use Case

No.	Use Case	Deskripsi
1	Pencarian Rute	Proses untuk mencari rute terpendek
2	Update Lokasi	Proses untuk memperbaharui data-data lokasi
3	Pengaturan	Proses untuk mengatur parameter algoritma pencarian beam

III.1.8.5.1 Skenario Usecase

Skenario setiap bagian pada use case menunjukkan proses apa yang terjadi pada setiap bagian didalam use case tersebut, dimana user memberikan perintah pada setiap bagian dan respon apa yang diberikan oleh sistem kepada user setelah user memberikan perintah pada setiap bagian-bagian use case. Berikut adalah skenario usecase pada aplikasi simulasi android.

Tabel III-16 Skenario Use Case Pencarian Rute

Identifikasi		
Nomor	1	
Nama	Pencarian Rute	
Tujuan	Menampilkan form pencarian lokasi	
Deskripsi	Proses untuk memanggil form pencarian lokasi	
Aktor	User	
Skenario Utama		
Kondisi Awal	User berada menu awal	
Aksi Aktor		Reaksi Sistem
1. Memilih menu Pencarian Rute		
		2. Menampilkan form pencarian lokasi
3.Memasukan lokasi awal dan lokasi tujuan		
4.Menekan tombol cari		5.Membaca inputan lokasi awal dan tujuan
		6.Mencari rute dengan algoritma pencarian beam
		7. Menampilkan hasil pencarian rute
Kondisi Akhir	Sistem menampilkan hasil pencarian rute	

Tabel III-17 Skenario Use Case Update Lokasi

Identifikasi	
Nomor	2
Nama	Update Lokasi
Tujuan	Menampilkan kembali menu awal
Deskripsi	Proses untuk memperbaharui data lokasi
Aktor	User

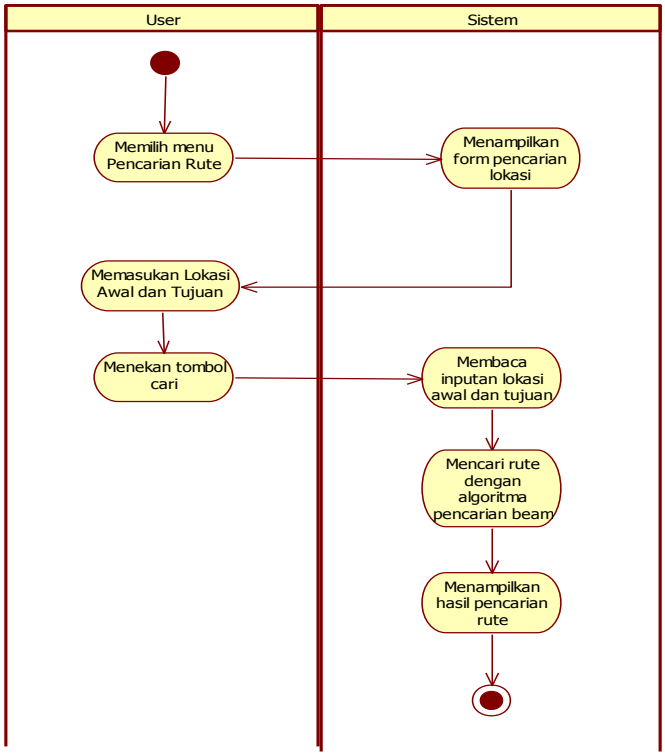
Skenario Utama		
Kondisi Awal	User berada menu awal	
Aksi Aktor		Reaksi Sistem
1. Memilih menu update lokasi		
		2. Mengambil data lokasi dari *web service*
		3. Menampilkan menu utama
Kondisi Akhir	Sistem memperbaharu i data lokasi dan kembali ke menu utama	

Tabel III-18 Skenario Use Case Pengaturan

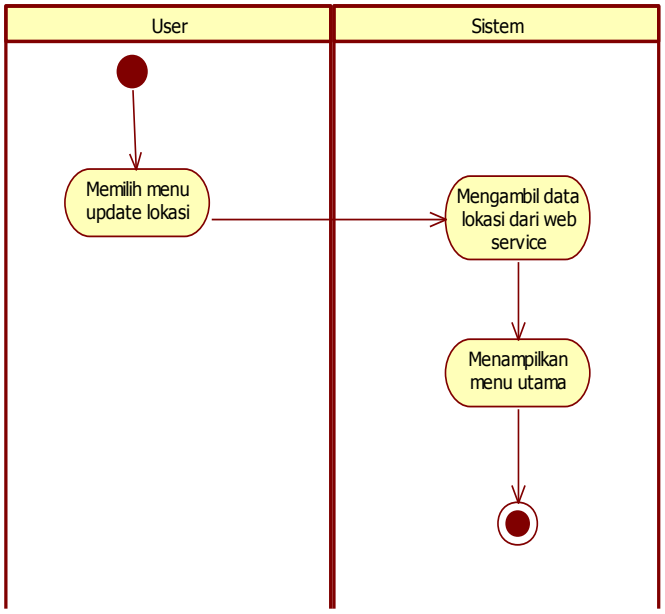
Identifikasi		
Nomor	3	
Nama	Pengaturan Parameter	
Tujuan	Mengatur Parameter Algoritma Pencarian Beam	
Deskripsi	Proses untuk mengatur nilai parameter algoritma pencarian beam	
Aktor	User	
Skenario Utama		
Kondisi Awal	Berada di form pencarian lokasi	
Aksi Aktor		Reaksi Sistem
1.Menekan tombol pengaturan		2.Menampilkan form pengaturan parameter
3.Memasukan nilai parameter		
4.Menekan tombol simpan		5.Menyimpan parameter
Kondisi Akhir	Sistem menyimpan nilai parameter	

III.1.8.6 Activity Diagram

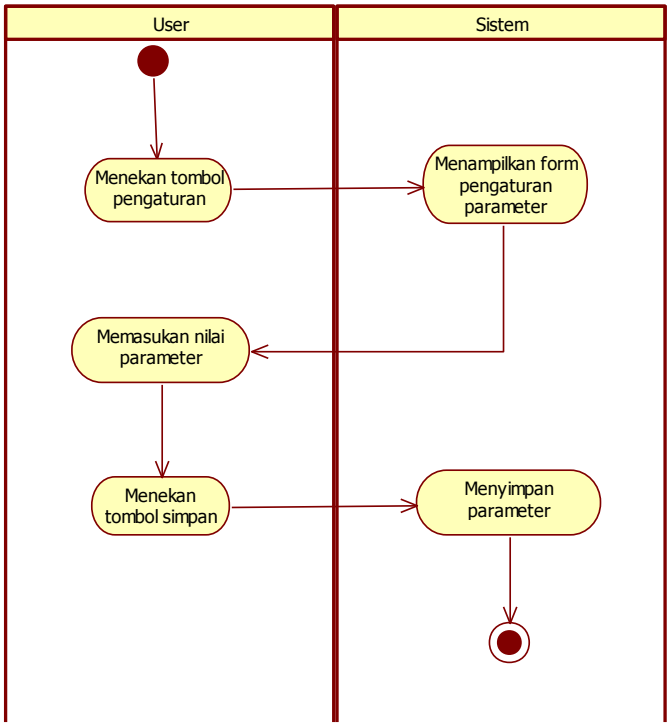
Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses pararel yang mungkin terjadi pada eksekusi.



Gambar III-13 Activity Diagram Pencarian Lokasi



Gambar III-14 Activity Diagram Update Lokasi

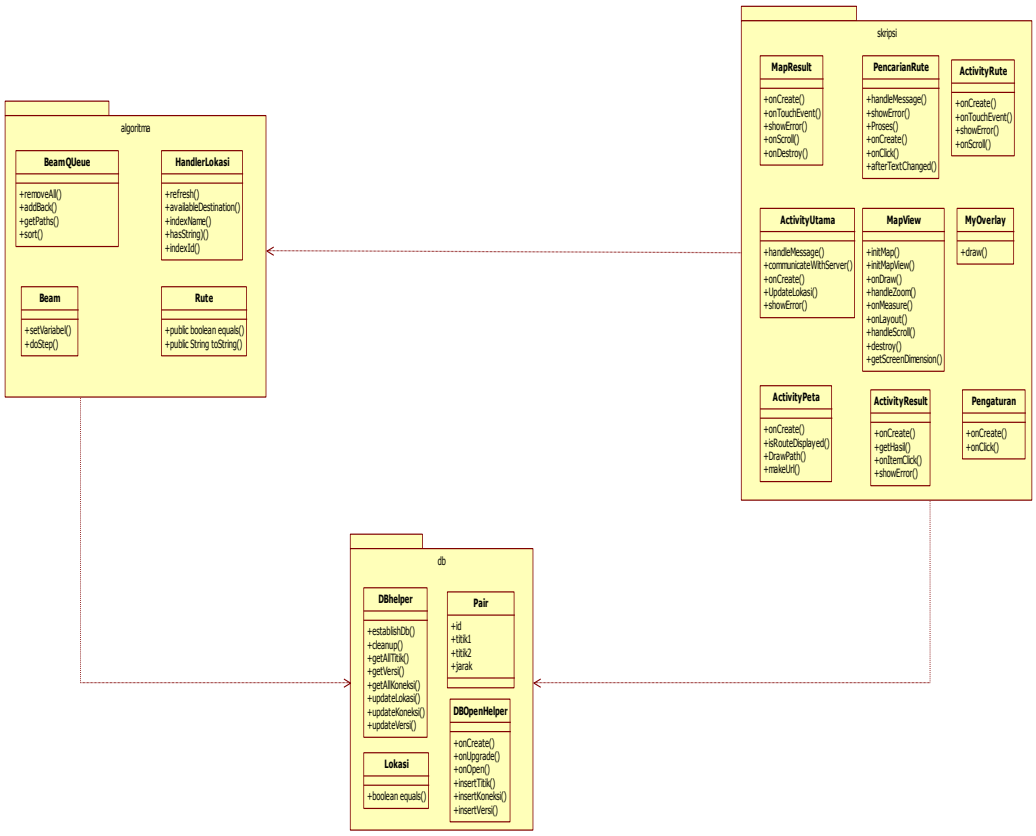


Gambar III-15 Activity Diagram Pengaturan Parameter

III.1.8.7 Class Diagram

Kelas merupakan satu set objek yang memiliki atribut dan perilaku yang sama (Whitten, 2004: 410). Kelas dapat juga diartikan sebagai suatu definisi sumber daya yang termasuk didalamnya informasi-informasi yang menggambarkan fitur suatu entitas dan bagaimana penggunaanya.

Berikut adalah *class diagram* untuk aplikasi simulasi pencarian rute terpendek di kota Tasikmalaya.

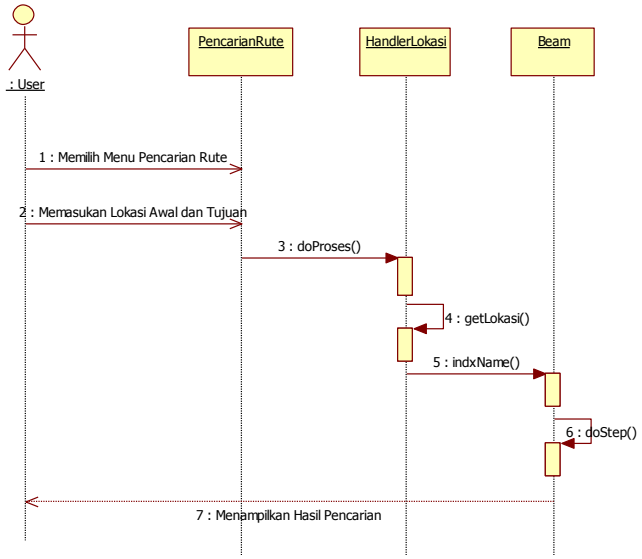


Gambar III-16 Class Diagram Aplikasi Simulasi

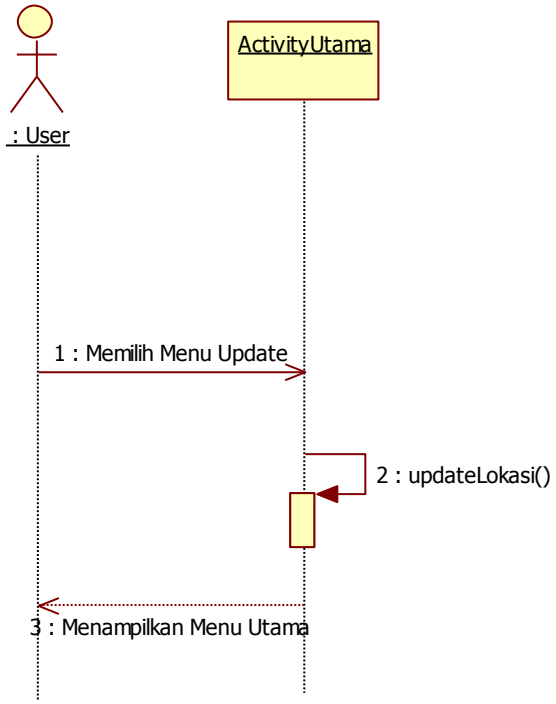
III.1.8.8 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan disekitar sistem (termasuk pengguna, *display*, dan sebagainya). *Sequence diagram*

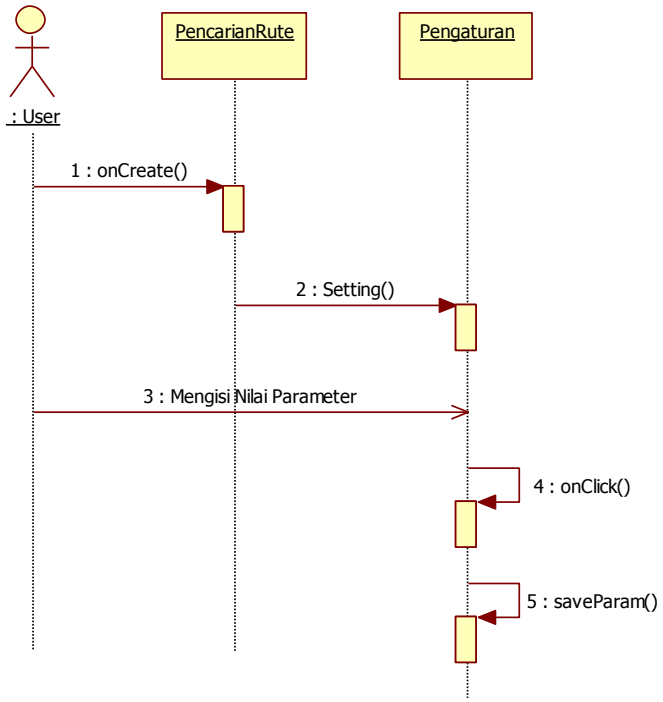
terdiri atas dimensi *vertical* (waktu) dan dimensi *horizontal* (objek-objek yang terkait).



Gambar III-17 Pemilihan Menu Pencarian Lokasi



Gambar III-18 Pemilihan Menu Update



Gambar III-19 Pengaturan Parameter

III.2 Perancangan Sistem

Tahap perancangan merupakan kelanjutan dari proses analisis. Perancangan sistem ini meliputi parancangan basis data, pencangan aristektur perangkat lunak, perancangan struktur menu, perancangan antarmuka, perancangan pesan, jaringan semantik, perancangan prosedural. Semua perancangan yang telah disebutkan diatas harus saling berhubungan agar menciptakan suatu sistem yang baik.

III.2.1 Perancangan Basis Data

Perancangan basis data dalam *web service* aplikasi simulasi pencarian rute terpendek di kota Tasikmalaya adalah sebagai berikut:

Tabel III-19 Perancangan Basis Data

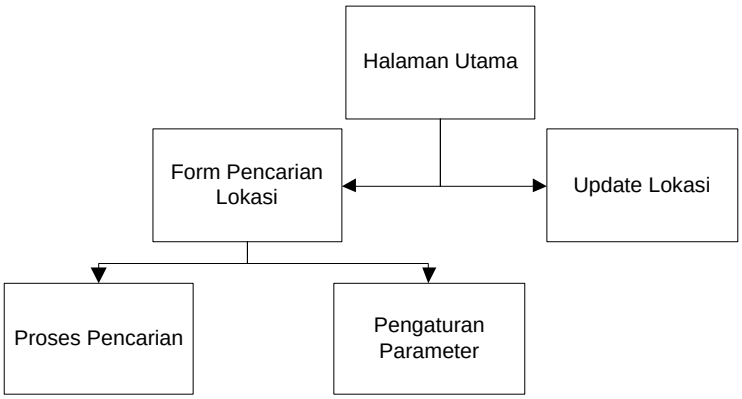
No.	Nama Basis Data		Atribut
1	Admin	:	<username>, <password>
2	Lokasi	:	<id>, <nama_lokasi>, <latitude>,<longitude>

III.2.2 Perancangan Struktur Menu

Dalam pembangunan aplikasi diperlukan sebuah struktur menu untuk menunjukan keterkaitan dan tingkatan antara menu utama dan sub menu, sehingga dapat lebih mudah di mengerti. Pada aplikasi simulasi pencarian rute terpendek ini, perancangan struktur menu terbagi menjadi dua yaitu struktur menu pada aplikasi simulasi android dan struktur menu pada *web service*.

III.2.2.1 Aplikasi Android

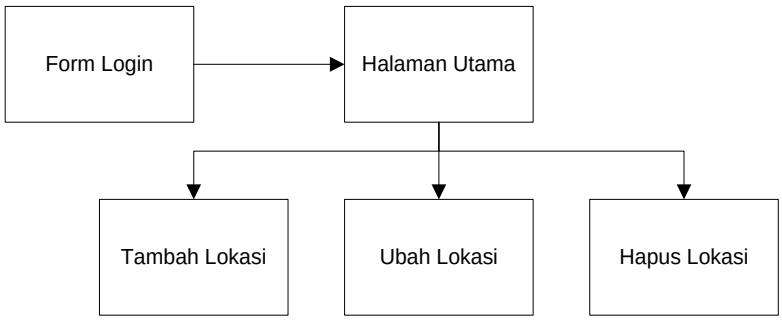
Strukrur menu yang terdapat pada aplikasi simulasi android dapat dilihat pada Gambar III-20 Struktur Menu Aplikasi Simulasi Android.



Gambar III-20 Struktur Menu Aplikasi Simulasi Android

III.2.2.2 Web service

Strukrur menu yang terdapat pada *web service* adalah sebagai berikut:



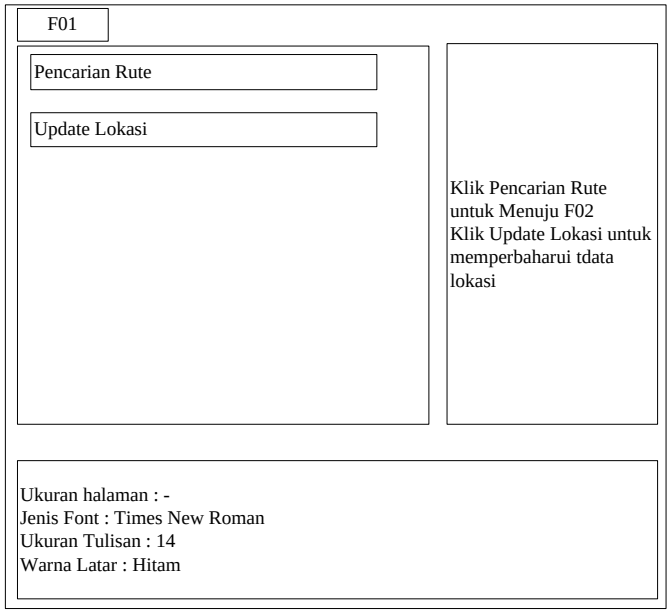
Gambar III-21 Struktur Menu *Web service*

III.2.3 Perancangan Antarmuka

Perancangan antarmuka merupakan salah satu bagian penting dalam perancangan sistem karena nantinya antarmuka tersebut akan menjadi fasilitas yang menjembatani interaksi manusia dengan sistem. Perancangan antarmuka yang akan dibangun pada aplikasi simulasi ini terbagi dua yaitu perancangan antarmuka pada aplikasi simulasi android dan perancangan antarmuka *web service*.

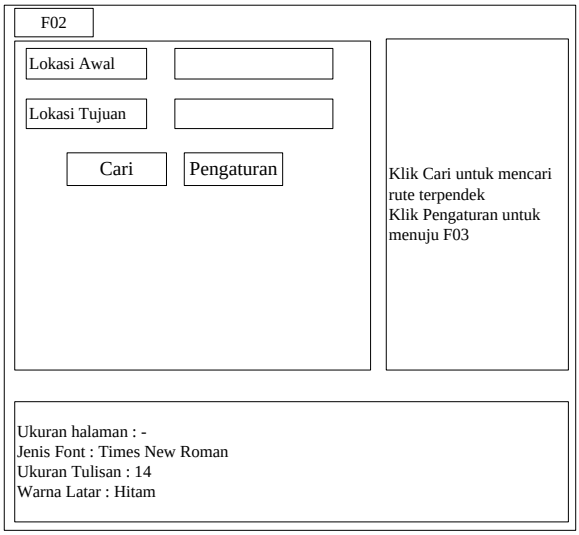
III.2.3.1 Perancangan Antarmuka Aplikasi Simulasi Pada Android

Halaman yang akan muncul pertama pada aplikasi simulasi android ini adalah halaman utama. Rancangan halaman utama aplikasi simulasi pencarian rute terpendek pada android adalah sebagai berikut.



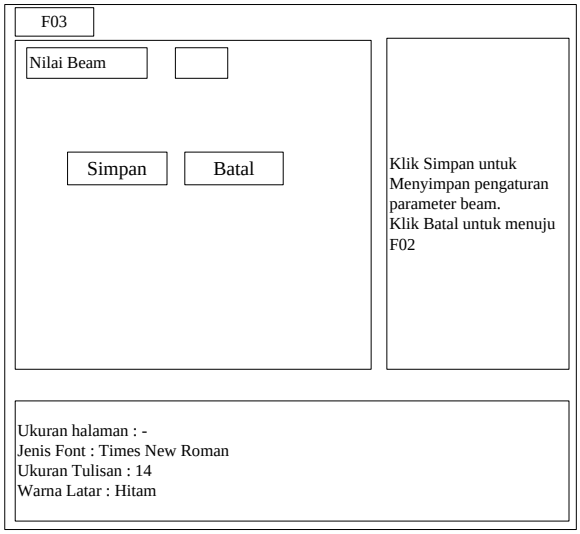
Gambar III-22 Halaman Utama Aplikasi Simulasi Pada Android

Perancangan antarmuka untuk menginputkan nilai-nilai yang dibutuhkan pada aplikasi simulasi pencarian rute terpendek dapat di lihat pada Gambar III-23.



Gambar III-23 Form Input Nilai Pencarian

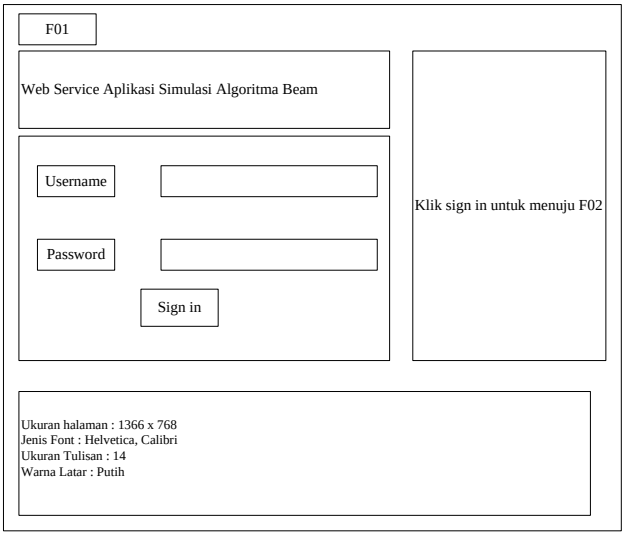
Dan perancangan antarmuka untuk memasukan nilai paramenter yang dibutuhkan dalam aplikasi simulasi pencarian beam adalah sebagai berikut.



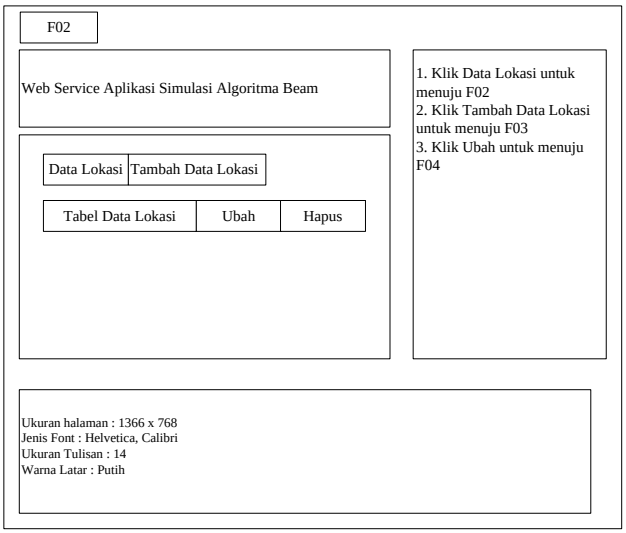
Gambar III-24 Form Input Nilai Parameter

III.2.3.2Perancangan Antarmuka Web service

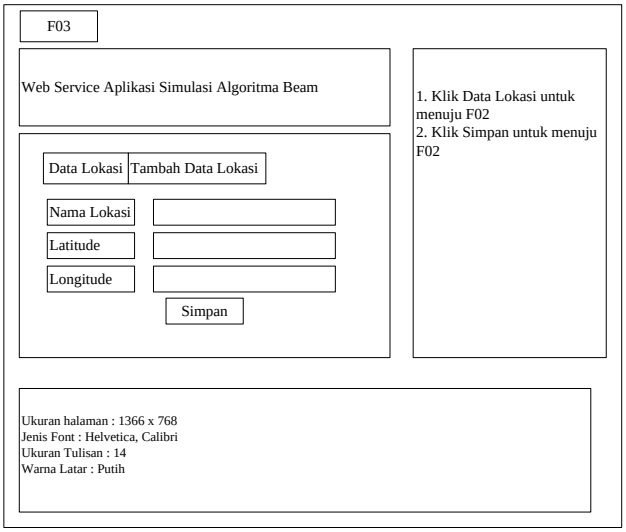
Pada aplikasi *web service*, halaman pertama yang akan muncul adalah halaman login. Setelah login berhasil, halaman selanjutnya yang dapat diakses adalah halaman utama, halaman tambah data dan halaman ubah data. Berikut adalah perancangan antarmuka pada *web service*.



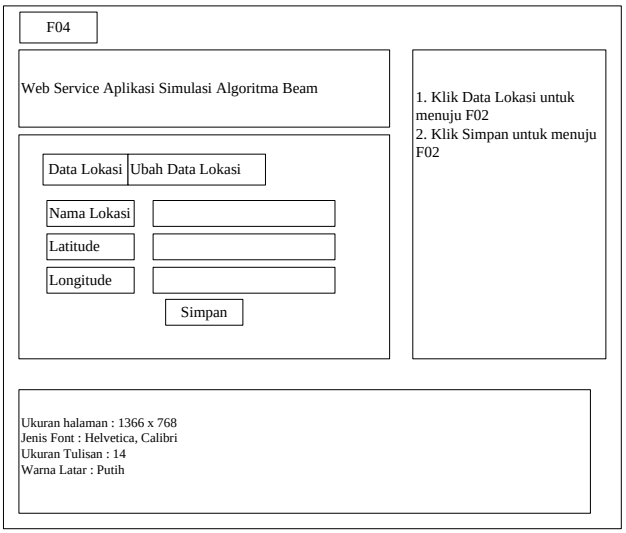
Gambar III-25 Perancangan Antarmuka Halaman Login



Gambar III-26 Perancangan Antarmuka Halaman Utama



Gambar III-27 Perancangan Antarmuka Halaman Tambah Data Lokasi



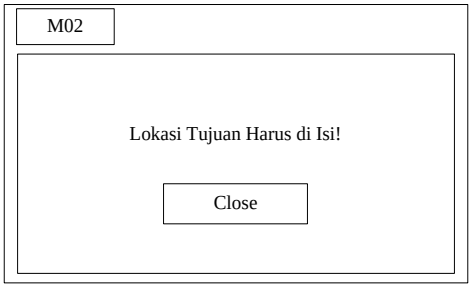
Gambar III-28 Perancangan Antarmuka Halaman Ubah Data Lokasi

III.2.4 Perancangan Pesan

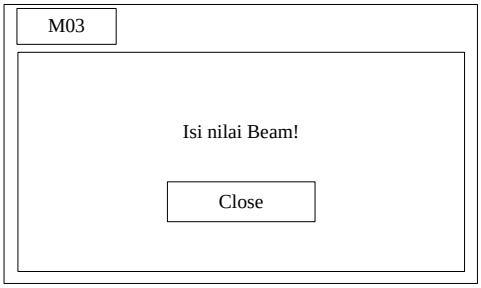
Rancangan pesan adalah rancangan antar muka dari pesan yang akan ditampilkan apabila terjadi kesalahan dan pemberitahuan dalam proses yang dilakukan. Berikut adalah perancangan pesan yang ada pada aplikasi android.



Gambar III-29 Pesan Kesalahan Lokasi Awal

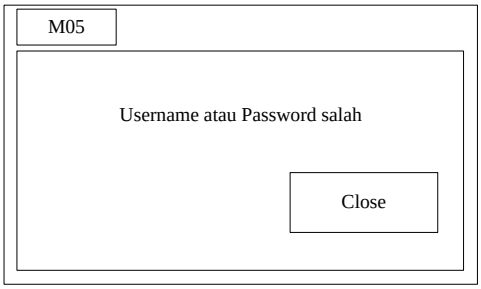


Gambar III-30 Pesan Kesalahan Lokasi Tujuan

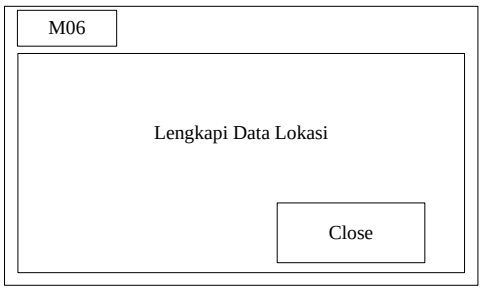


Gambar III-31 Pesan Kesalahan Nilai Beam Tidak di Isi

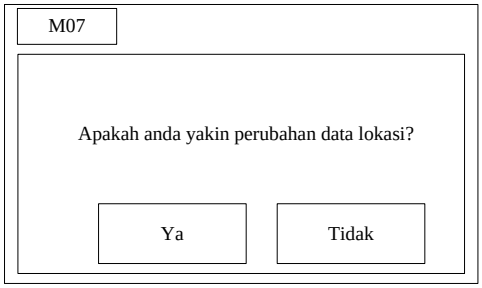
Sedangkan perancangan pesan pada *web service* dapat dilihat pada gambar Gambar III-32 Pesan Kesalahan Login, Gambar III-33 Pesan Kesalahan Pengisian Data Belum Lengkap, Gambar III-34 Pesan Konfirmasi Perubahan Data Lokasi dan Gambar III-35 Pesan Konfirmasi Hapus Data Lokasi.



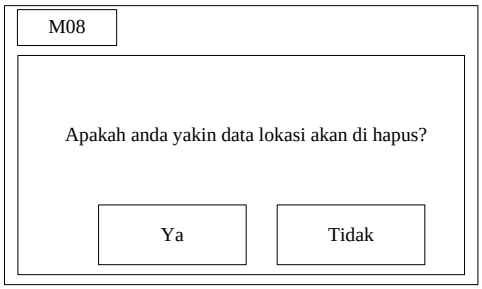
Gambar III-32 Pesan Kesalahan Login



Gambar III-33 Pesan Kesalahan Pengisian Data Belum Lengkap



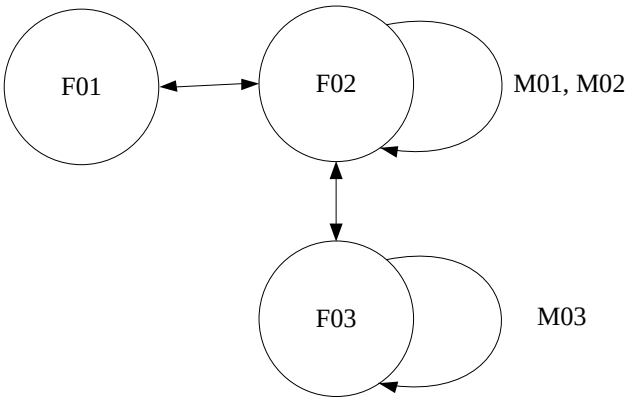
Gambar III-34 Pesan Konfirmasi Perubahan Data Lokasi



Gambar III-35 Pesan Konfirmasi Hapus Data Lokasi

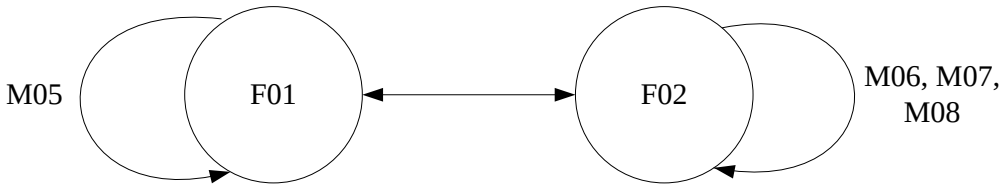
III.2.5 Jaringan Semantik

Jaringan semantik menggambarkan keterhubungan navigasi menu dari satu halaman ke halaman lainnya. Jaringan semantic untuk aplikasi android adalah sebagai berikut.



Gambar III-36 Jaringan Semantik Aplikasi Android

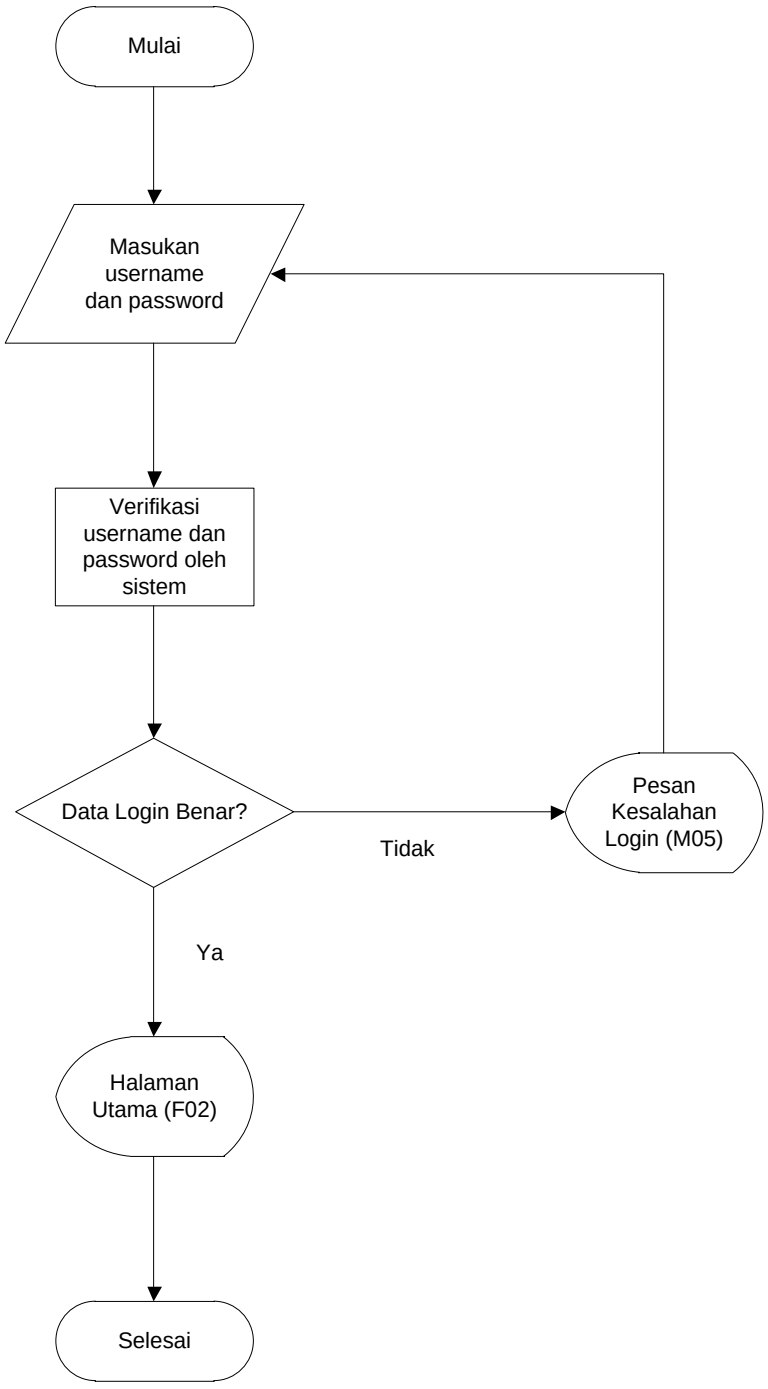
Sedangkan jaringan semantik *web service* adalah sebagai berikut.



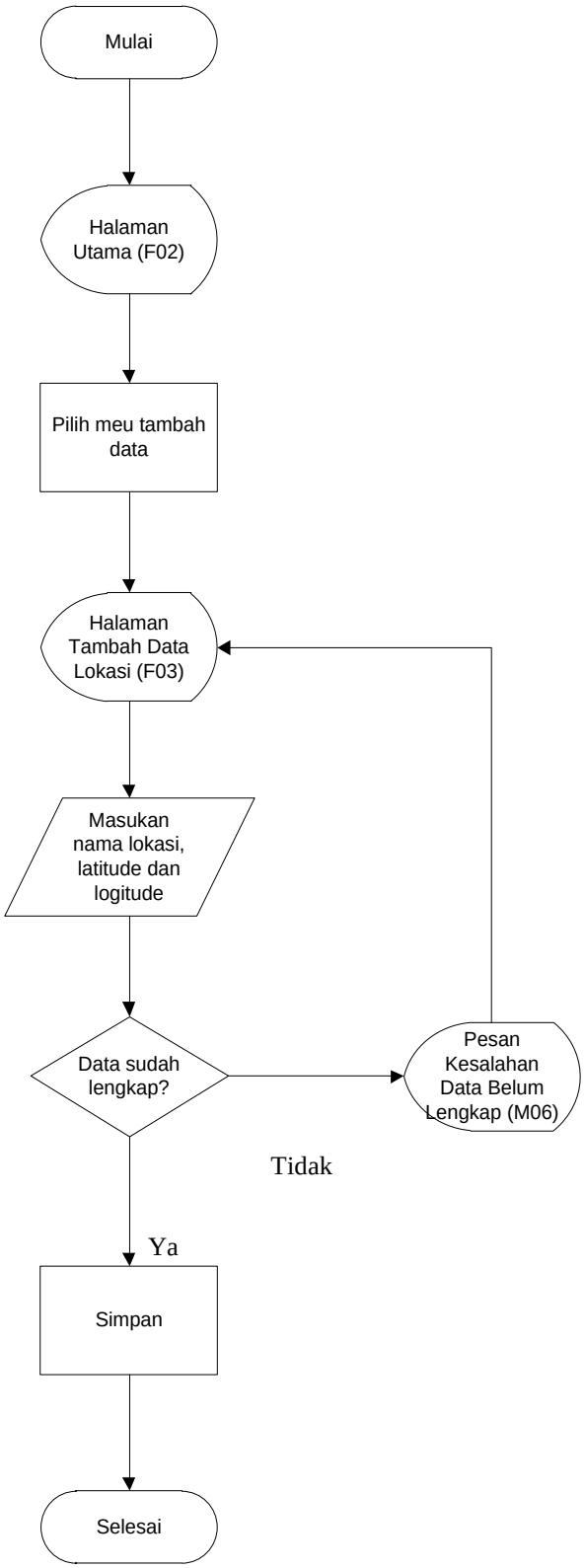
Gambar III-37 Jaringan Semantik *Web service*

III.2.6 Perancangan Prosedural *Web Service*

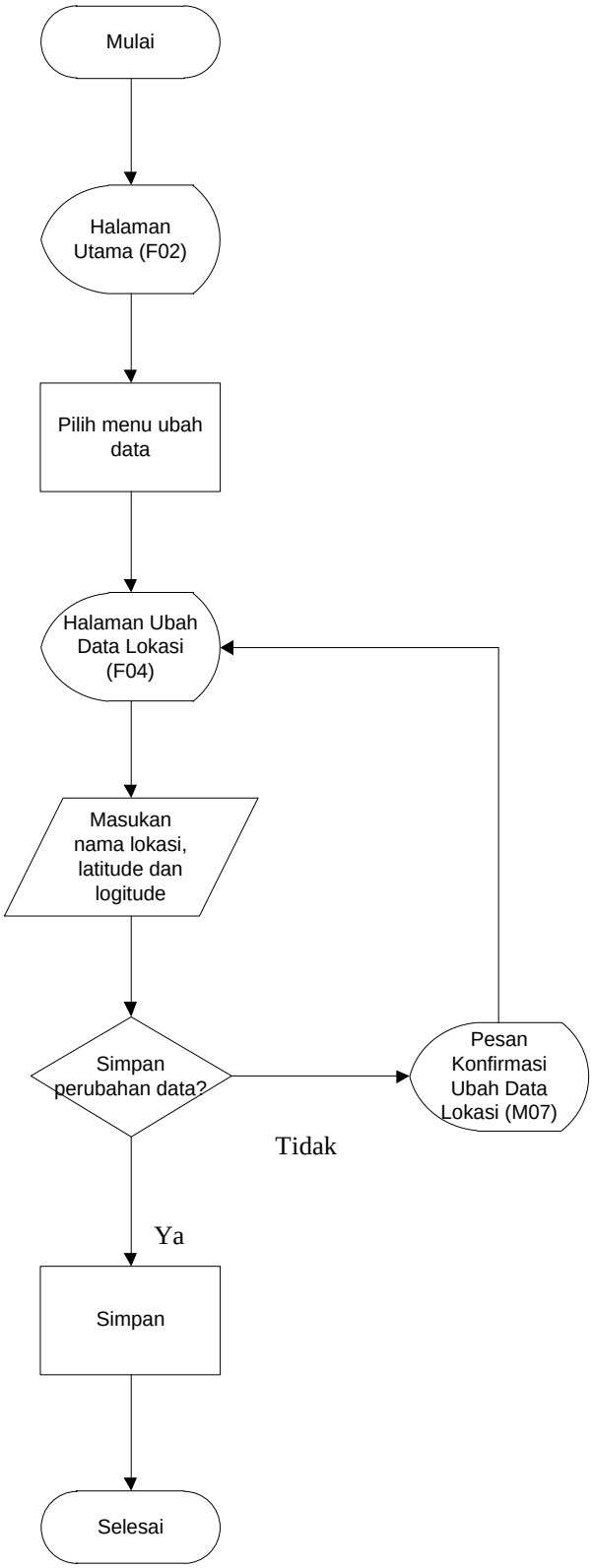
Perancangan prosedural *web service* menjelaskan langkah-langkah dalam mengelola data lokasi. Pada *web service*, terdapat beberapa perancangan prosedural yaitu prosedur login, prosedur tambah data, prosedur ubah data dan prosedur hapus data. Perancangan procedural *web service* dapat dilihat pada Gambar III-38 Prosedur Login, Gambar III-39 Prosedur Tambah Data Lokasi, Gambar III-40 Prosedur Ubah Data Lokasi dan Gambar III-41 Prosedur Hapus Data Lokasi.



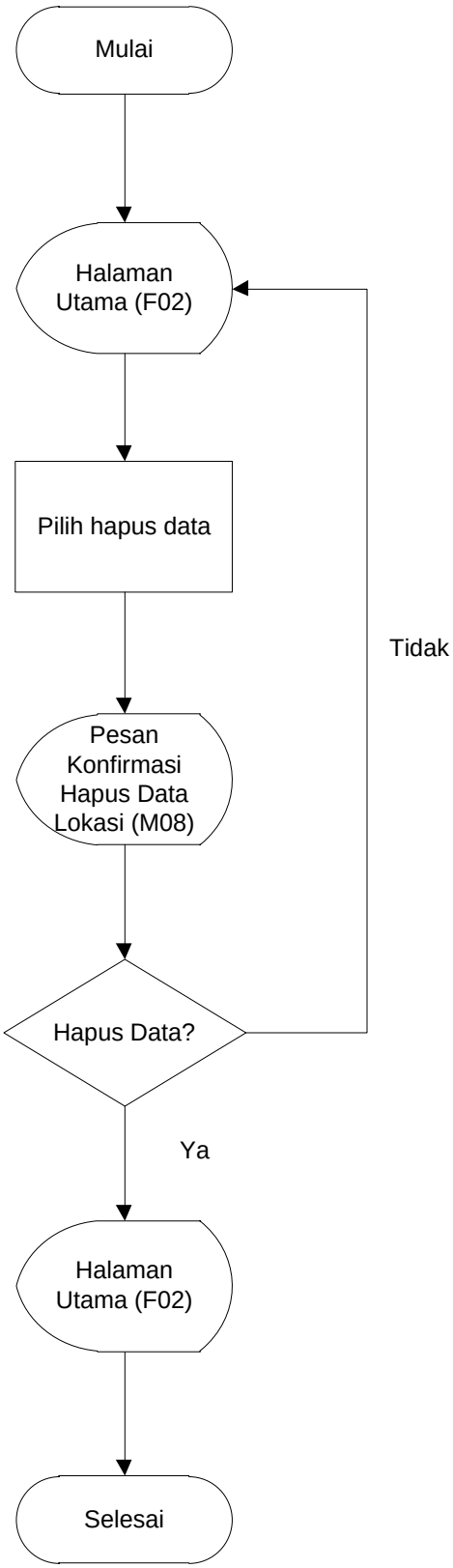
Gambar III-38 Prosedur Login



Gambar III-39 Prosedur Tambah Data Lokasi



Gambar III-40 Prosedur Ubah Data Lokasi



Gambar III-41 Prosedur Hapus Data Lokasi

III.2.7 Perancangan Method

Perancangan method merupakan pemaparan proses pada method-method yang digunakan pada setiap kelas dan digambarkan pada .

a. Method doUpdate()

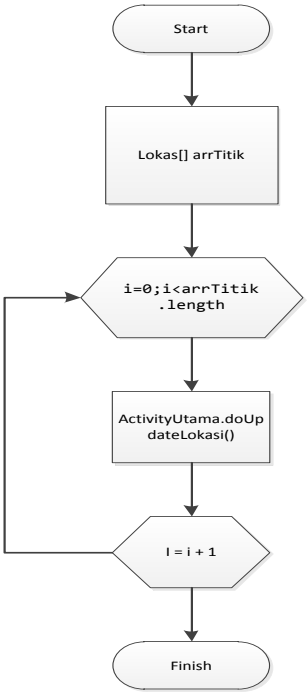
Method doUpdate() merupakan method yang digunakan untuk mengambil data lokasi yang tersedia pada *web service*. Method ini dapat dilihat pada gambar Gambar III-42 Perancangan Method doUpdate().

b. Method DrawPath()

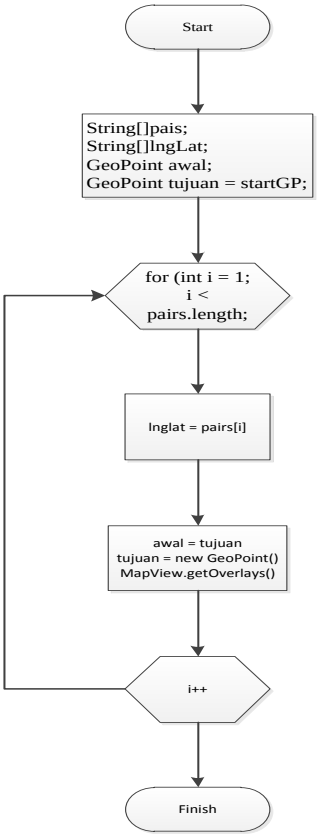
Method DrawPath merupakan method untuk menggambarkan hasil pencarian pada peta dan menghubungkan antara dua lokasi yang di cari dengan sebuah garis sebagai tanda rute terpendek ditemukan. Method ini dapat dilihat pada Gambar III-43 Perancangan Method DrawPath().

c. Method doStep()

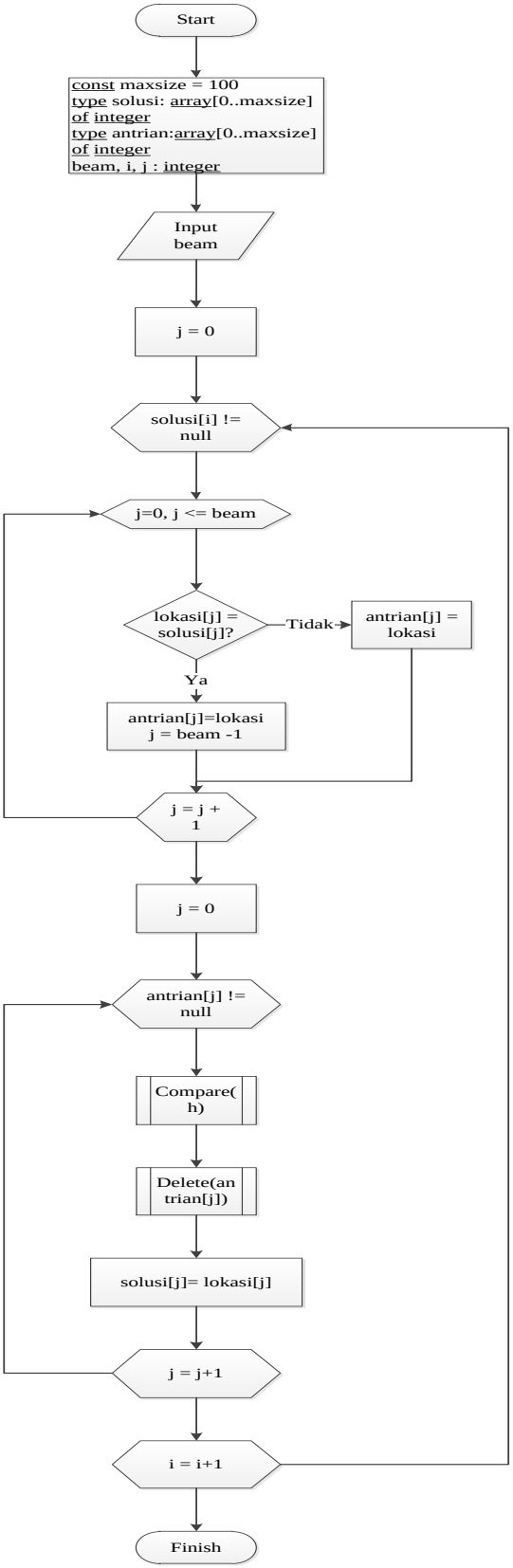
Method doStep() merupakan method yang berisi algoritma pencarian beam untuk mencari rute terpendek. Method ini dapat dilihat pada Gambar III-44 Perancangan Method doStep().



Gambar III-42 Perancangan Method doUpdate()



Gambar III-43 Perancangan Method DrawPath()



Gambar III-44 Perancangan Method doStep()