

## BAB III

### METODE PENELITIAN

#### 1.1 Metode Pengembangan SDLC

Metode pengembangan sistem yang penulis gunakan adalah SDLC (*Sistem Development Life Cycle*) model *Waterfall*. Alur dari metode pengembangan sistem SDLC dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Metode Pengembangan SDLC (Roberta M. Roth, 2012)

##### 1.1.1 Perencanaan

Pada tahap perencanaan akan dilakukan proses pengumpulan data yang bertujuan untuk mengetahui data apa saja yang dibutuhkan untuk menyelesaikan permasalahan ini. Dalam penelitian ini, objek yang dijadikan bahan studi pada penelitian ini adalah kendaraan umum jenis taksi. Dari objek tersebut diperoleh data-data yang berupa data jalan atau jalur yang akan dilewati oleh kendaraan umum jenis taksi, data kecepatan, dan data persimpangan. Data kecepatan pada penelitian ini di tinjau dari lebar jalan, kondisi jalan dan lingkungan sekitar dari jalan.

Jenis data yang digunakan pada penelitian dari cara memperolehnya di dapat data dengan kategori primer dan sekunder:

1. **Data primer** ialah data yang berasal dari sumber asli atau pertama. Data ini tidak tersedia dalam bentuk terkompilasi ataupun dalam bentuk file-file. Data ini harus dicari melalui narasumber atau dalam istilah teknisnya responden, yaitu orang yang kita jadikan objek penelitian atau orang yang kita jadikan sebagai sarana mendapatkan informasi ataupun data (Umi Narimawati, 2008:98). Data primer pada penelitian ini diperoleh dari data penelitian sebelumnya oleh Aditya Setiawan (2016) yang didapatkan dari hasil wawancara dengan Dinas Perhubungan Kota Denpasar mengenai jarak

tempuh, jalan dan juga data kecepatan di setiap jalan di daerah Denpasar Utara. Data kecepatan kendaraan taksi berdasarkan lebar jalan, kondisi jalan dan lingkungan sekitar seperti sekolah, kantor dan pasar yang bersifat dinamis. Kemudian untuk menunjang penelitian ini juga dilaksanakan survei ke lapangan untuk mengetahui kondisi dari jalan yang akan dimodelkan ke dalam bentuk *graph*.

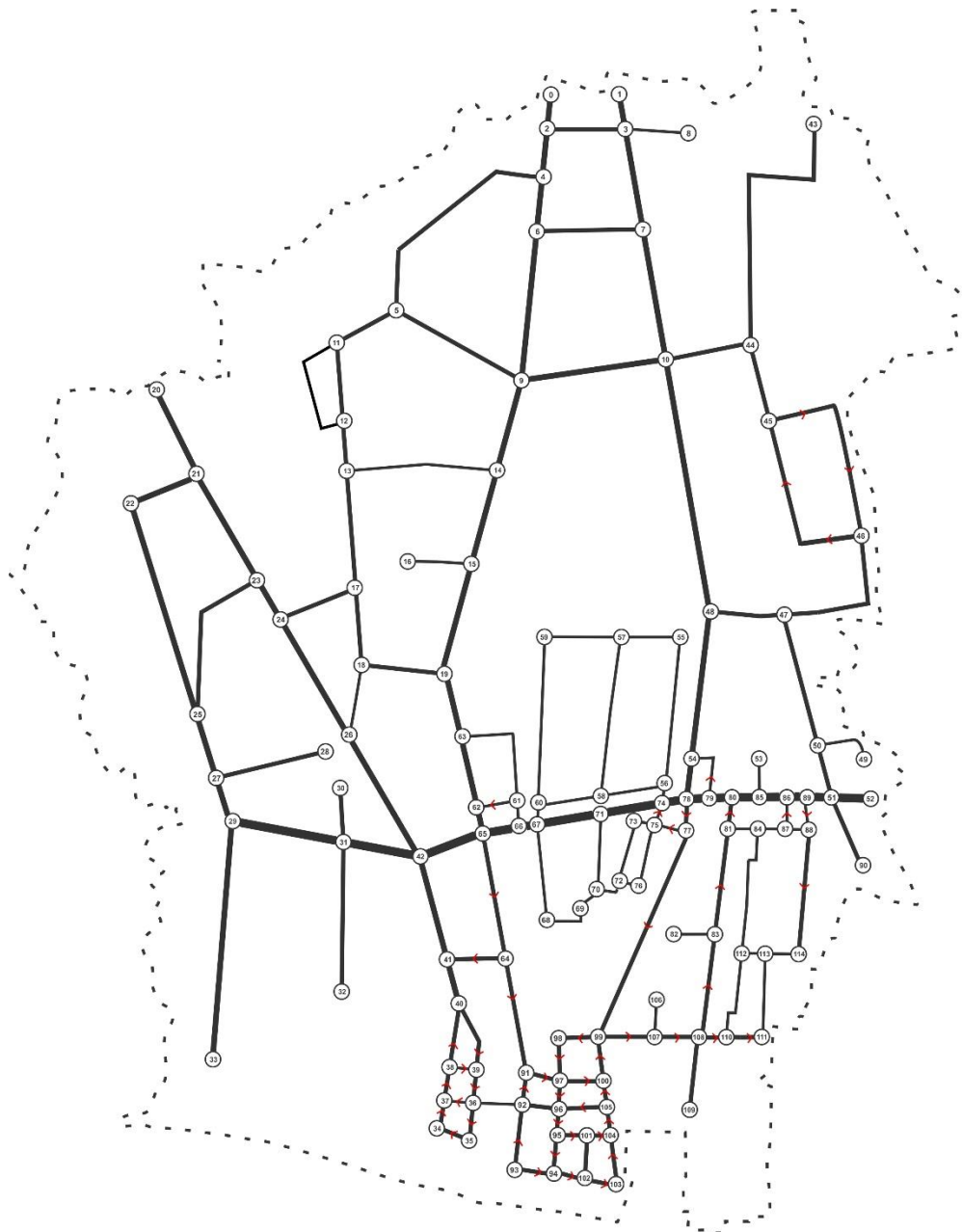
2. **Data sekunder** adalah sumber data yang tidak langsung memberikan data kepada pengumpul data, misalnya lewat orang lain atau lewat dokumen (Sugiyono, 2008). Data sekunder yang akan digunakan untuk menunjang penelitian ini adalah data kecepatan kendaraan yang didapatkan Dinas Perhubungan Kota Denpasar dan. data kondisi dan panjang jalan dari Dinas Pekerjaan Umum Kota Denpasar.

### **1.1.2 Analisis**

Terdapat beberapa analisis yang dilakukan oleh penulis untuk mengembangkan Sistem Penentuan Rute Optimum Kendaraan umum Menggunakan Algoritma Particle Swarm Optimization.

#### **1.1.2.1 Pemodelan Jalur Kendaraan Umum Taksi**

Dari data yang diperoleh dari penelitian Aditya Setiawan (2016) beserta analisa yang sudah dilakukan maka dipilih jalan atau jalur yang dapat dilalui oleh angkutan umum taksi yang berada pada wilayah Denpasar Utara. Pemodelan jalur taksi direpresentasikan dalam bentuk graf, pada Gambar 3.2.



Gambar 3. 2 Pemodelan Graf Denpasar Utara (Aditya Setiawan, 2016)

Graf pada Gambar 3.2 tersebut, nama persimpangan diwakili oleh nomer atau *ID* yang dimiliki oleh setiap *node*. Daftar lengkap nama persimpangan dapat dilihat pada lampiran A. Pemberian bobot pada tiap simpul menggunakan rata – rata waktu tempuh kendaraan umum Taksi. Pemberian bobot dilakukan dengan melakukan perhitungan dari data yang sudah di dapatkan di Dinas Perhubungan Kota Denpasar. Pemberian bobot dilakukan dengan menggunakan perhitungan :

$$t = \frac{s}{v} \dots \dots \dots (3.1)$$

Dimana:

- t = Waktu tempuh
- s = Jarak / Panjang jalan
- v = Kecepatan

Berikut ini adalah rata - rata kecepatan kendaraan umum taksi secara umum yang dipengaruhi oleh lebar jalan di Denpasar Utara. Data kecepatan pada penelitian ini di tinjau dari lebar jalan, kondisi jalan dan lingkungan sekitar dari jalan. Tabel 3.1 menunjukkan kecepatan rata-rata kendaraan taksi menurut lebar jalan, kondisi dan lingkungan sekitar.

Tabel 3. 1 Rata – Rata Kecepatan Kendaraan Umum (Aditya Setiwan,2016)

No.	Lebar Jalan (m)	Kecepatan Rata-Rata (km/h)
1.	5	20
2.	6	22
3.	7	24
4.	8	26
5.	9	28
6.	10 – 12	30 - 32
7.	13 – 16	34 - 36
8.	17 – 20	38 - 40

Kemudian dilakukan pengolahan data dari panjang jalan dan juga kecepatan rata-rata untuk bisa mendapatkan waktu tempuh di setiap ruas jalan. Sebagai contoh, panjang jalan dari *node* 17 ke *node* 18 adalah 400 m (0.4 km) dengan lebar jalan 7 m (Jl.Warmadewa), maka waktu tempuh rata – rata yang dimiliki adalah

$$\frac{0.4}{24} = 0.016 \frac{km}{h} (0.96 \text{ menit}) \dots \dots \dots (3.2)$$

Jadi waktu tempuh yang dibutuhkan kendaraan umum taksi untuk melintas dari *node* 17 ke *node* 18 adalah rata-rata 0.96 menit. Daftar bobot untuk setiap ruas

jalan yang ada pada pemodelan jalur angkutan umum taksi di Denpasar Utara dapat dilihat pada Lampiran B.

#### **1.1.2.2 Analisis Sistem**

Tahap analisis sistem merupakan tahapan menentukan apa saja yang dibutuhkan oleh sistem. Pada tahap ini adapun beberapa komponen yang akan membantu dalam proses menganalisis sistem, yaitu daftar masukan (*input*) pada sistem, proses yang berlangsung pada sistem serta daftar keluaran (*output*) dari sistem sebagai berikut :

##### **1. Masukan untuk Sistem**

Adapun *input* atau masukan data yang akan diproses di dalam sistem penentuan rute optimum kendaraan umum dengan mengimplementasikan algoritma *Particle Swarm Optimization* (PSO) adalah sebagai berikut :

###### **A. Data Persimpangan**

Data persimpangan yang dimaksud adalah data dari lokasi-lokasi yang akan dilewati dari kendaraan umum taksi pada rutanya, seperti nama persimpangan dan koordinat persimpangan yang didapatkan dari *Google Maps*.

###### **B. Data Jalan**

Data jalan yang dimaksud adalah data dari nama jalan yang menghubungkan persimpangan-persimpangan dari kendaraan umum taksi pada rutanya, seperti nama jalan, persimpangan yang dihubungkan, serta bobot dari jalan yang berupa waktu tempuh kendaraan umum jenis taksi.

###### **C. Data wilayah**

Data wilayah yang dimaksud adalah wilayah dari persimpangan yang dilalui oleh taksi. Wilayah yang dimaksud adalah kelurahan yang berada di Denpasar Utara.

##### **2. Proses yang berlangsung dalam Sistem**

Sistem ini akan terbagi menjadi dua bagian yaitu pada *server* dan *mobile*. Proses pembuatan rute dilakukan pada *server*, sedangkan pada aplikasi *mobile* berfungsi untuk mengirimkan persimpangan awal dan akhir ke server serta menampilkan rute yang dihasilkan oleh *server* pada Api Google Maps. Untuk

menghasilkan rute yang optimal terdapat beberapa proses yang dilakukan pada *server* yaitu sebagai berikut :

- A. Menghitung jarak antar masing-masing persimpangan.
- B. Optimasi rute dengan algoritma *Particle Swarm Optimization* (PSO).
- C. Menampilkan rute dalam bentuk peta dengan bantuan Google maps API.

### 3. Keluaran dari Sistem

Keluaran (*output*) dari sistem ini berupa rute dari masing-masing kendaraan umum taksi dengan biaya yang optimal, dimana rute yang diperoleh nanti akan divisualisasikan kedalam bentuk peta pada Google Maps Api pada aplikasi mobile dan dalam bentuk tampilan *Infografis*.

#### 1.1.2.3 Analisis Kebutuhan Fungsional

Kebutuhan Fungsional sistem adalah kebutuhan yang berisi proses-proses apa saja yang nantinya dilakukan oleh sistem. Berikut adalah bagan kebutuhan fungsional Sistem Penentuan Rute Optimum Angkutan Umum Berbasis Web Menggunakan Metode Particle Swarm Optimization.

Tabel 3. 2 Kebutuhan Fungsional Sistem

Kode	Pengguna	Kebutuhan	Deskripsi Kebutuhan
KF1	User	Input Persimpangan	User memilih nama persimpangan yang akan di jadikan titik awal keberangkatan dan titik akhir sebagai tujuan akhir
KF2	User	Mencari jalur optimal	User mendapatkan hasil jalur optimal dari perhitungan algoritma PSO dalam bentuk simulasi. Hasil ditampilkan di Google maps atau ditampilkan

			nama persimpangan secara berurutan ( <i>Infografik</i> ).
KF3	User	Melihat daftar persimpangan	User dapat melihat daftar persimpangan beserta kordinat lokasinya.
KF4	User	Melihat Panduan	Ketika mengalami kesulitan atau mempunyai kendala dalam mengoprasikan sistem, user dapat melihat panduan bagaimana menggunakan sistem tersebut
KF5	Admin	Login Sistem	Sebagai autentifikasi admin unuk dapat memberikan hak akses dalam mengolah dan menggunakan sistem.
KF6	Admin	Create Data	Admin dapat menambah data pada sistem. Data yang dapat di tambah diantaranya: - Data node - Data jalan - Data edge - Data wilayah
		Read Data	Admin dapat melihat keseluruhan data-data yang ada pada sistem. Data yang dapat dilihat antaranya: - Data node - Data edge - Data jalan

			- Data wilayah
		Delete Data	Admin dapat melakukan hapus (delete) data pada sistem. Data yang dapat dihapus diantaranya: <ul style="list-style-type: none"> <li>- Data node</li> <li>- Data egde</li> <li>- Data Jalan</li> <li>- Data Wilayah</li> </ul>
KF7	Admin	Melihat Laporan	Admin dapat melihat laporan ( <i>report</i> ) data hasil perhitungan menggunakan Algortima PSO pada sistem dalam bentuk tampilan diagram

### 1.1.2.3 Kebutuhan Non Fungsional Sistem

Analisis Kebutuhan Non Fungsional yang harus dipenuhi oleh sistem diluar dari kebutuhan fungsional untuk melengkapi sistem secara keseluruhan, diantaranya ditinjau dari:

#### 1. Kebutuhan Pengguna

Ditinjau dari segi fungsi, sistem dirancang dan dibangun mengimplementasikan fungsi-fungsi antarmuka yang mudah dipahami oleh pengguna.

Fungsi antarmuka yang diimplementasikan berupa desain yang mudah dipahami oleh penggunanya. Seperti tombol fungsi untuk melakukan *insert*, *update*, dan *delete* data pada sistem admin yang diimplentasikan dalam bentuk gambar. Kemudian tombol fungsi melihat simulasi peta dan tombol menu pada aplikasi mobile yang juga direpresentasikan dalam bentuk gambar.



### 1.1.3 Tahap Perancangan

Design (Perancangan, yaitu melakukan perancangan agar dapat menyediakan rancangan yang diharapkan. Pada tahap ini dilakukan *Uses Case*, *Entity Relational Diagram* (ERD), perancangan struktur menu serta perancangan tampilan aplikasi.

#### 3.1.3.1 Rancangan Use Case

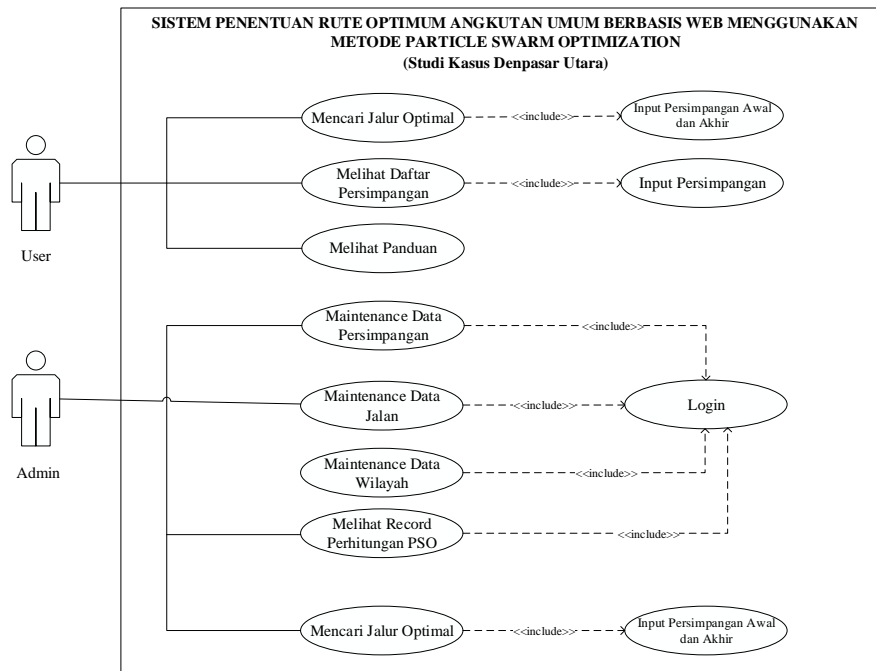
##### 1. Pendefinisian Use Case

Tabel berikut merupakan dekripsi actor pada sistem rekomendasi warung khas bali:

Tabel 3. 3 Deskripsi Aktor Use Case

No.	Aktor	Deskripsi
1.	User	Merupakan pengendara angkutan umum taksi yang dapat menggunakan aplikasi langsung tanpa harus melakukan login terlebih dahulu
2	Admin	Merupakan pengguna yang memiliki hak akses untuk melihat melakukan management data persimpangan, jalan dan wilayah serta melihat laporan hasil pencarian rute angkutan umum taksi

Berdasarkan analisis kebutuhan yang telah dibuat, maka dapat dirancang sebuah *use case* diagram seperti berikut ini:



Gambar 3. 3 *Use Case Diagram*

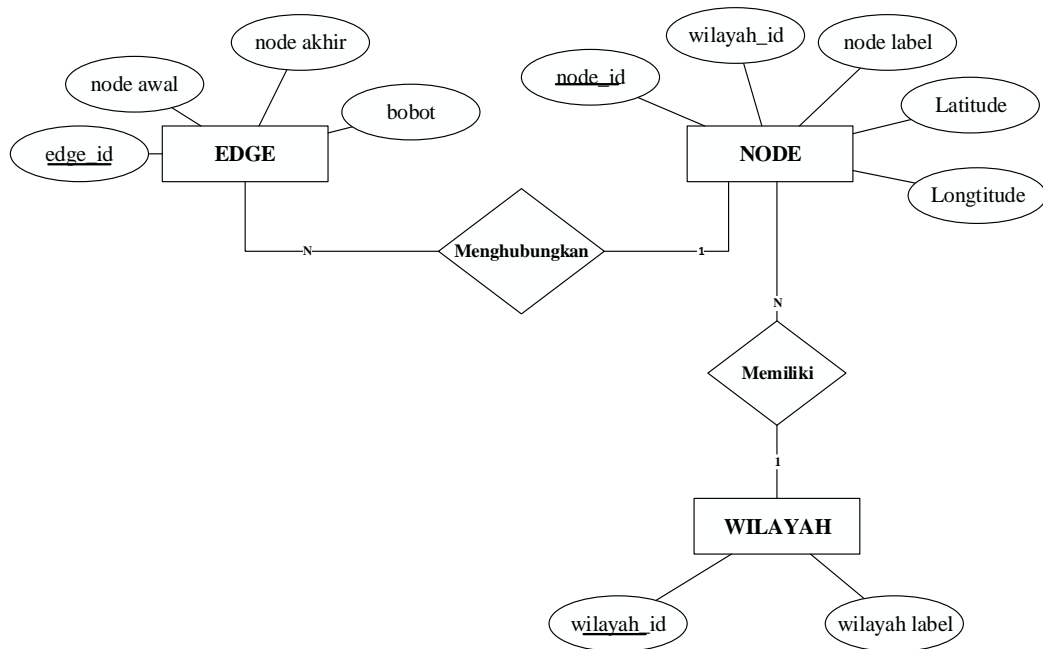
*Use case diagram* pada Gambar 3.3 menggambarkan interaksi antara *actor* dengan sistem informasi yang dibuat. Dalam sistem ini terdapat dua buah *actor* yaitu admin dan juga *user*. *User* merupakan Pengendara Taksi. Sebelum sistem melakukan proses pencarian rute, sebelumnya *user* harus memilih nama persimpangan yang akan dijadikan titik asal dan titik tujuan. Selanjutnya *user* akan dapat melihat hasil jalur optimal dari persimpangan yang sudah dipilih dengan perhitungan tersebut dan dapat melihat hasil dari jalur optimal dalam bentuk tampilan informasi rute-rute yang harus dilalui, atau akan ditampilkan di Google Maps. Jika *user* memiliki kendala dalam melihat nama persimpangan, atau pengoprasian aplikasi, *user* dapat memilih menu panduan untuk melihat bagaimana menggunakan aplikasi.

Pada *use case* admin, disini admin dapat melakukan pengolahan data yang ada pada sistem. Pengolahan data tersebut berupa menambahkan (*create*) data, melakukan perubahan (*update*), dapat melakukan hapus (*delete*) data pada sistem dan melihat laporan yang ada pada sistem. Data-data tersebut adalah data nama(*node*) atau persimpangan, data nama bobot (*edge*), data nama wilayah, data

nama jalan, dan data admin. Namun untuk dapat melakukan pengolahan data, terlebih dahulu admin harus melakukan proses *login* pada sistem.

### 3.1.3.2 Entity Relationship Diagram (ERD)

Entity Relationship Diagram digunakan untuk memodelkan basis data yang ada pada sistem, hubungan anatra satu entitas dengan entitas yang lainnya dalam sistem. Berikut ini adalah rancangan ERD dari sistem yang dibuat:



Gambar 3. 4 Entity Relationship Diagram

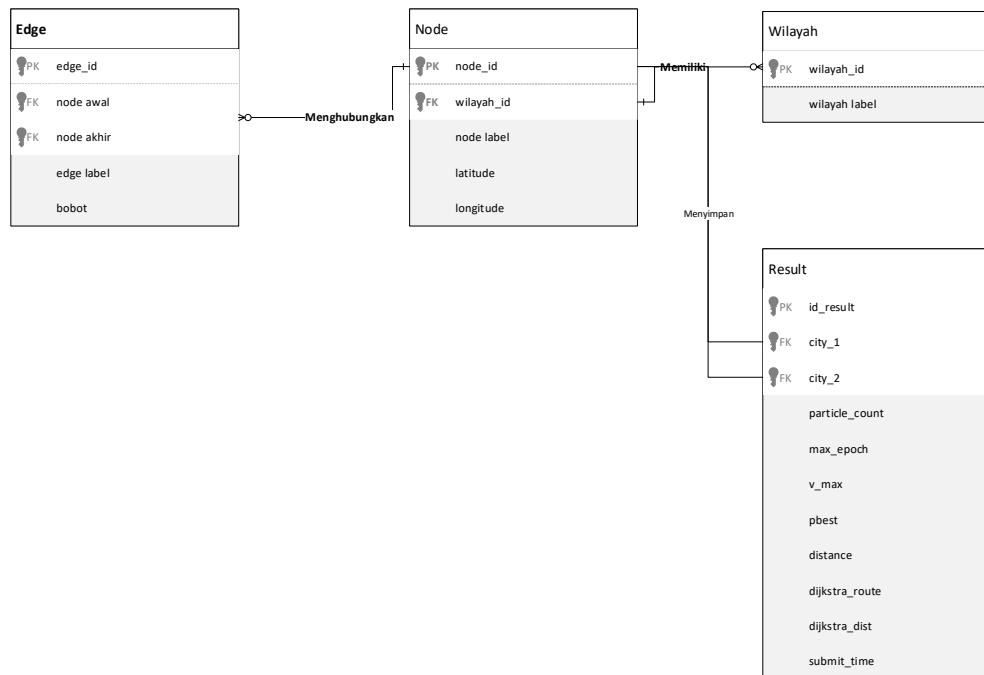
Pada rancangan *Entity Relationship Diagram* pada Gambar 3.4 terdapat tiga buah entitas yaitu entitas *node* , *edge*, dan wilayah. Rancangan ERD ini nantinya akan diimplementasikan ke dalam bentuk tabel pada database. Adapun hubungan atau relasi yang terdapat dalam rancangan ERD diatas yaitu:

1. Relasi antara entitas *Node* dengan entitas *Wilayah* yang memiliki kardinalitas *many to one* karena relasi ini memiliki makna bahwa terdapat banyak node yang memiliki suatu wilayah. Begitu juga sebaliknya dimana satu wilayah bisa merepresentasikan banyak node di dalam wilayah tersebut.
2. Relasi antara entitas *Node* dengan entitas *edge* yang memiliki kardinalitas *one to many* dimana suatu *Node* dapat menghubungkan banyak *Edge* di dalam graf.

Antara node satu dengan lainnya dihubungkan dengan sebuah edge. Setiap data egde menyimpan data node awal dan node akhir yang dihubungkannya. Edge A dan Egde B dikatakan terhubung apabila node akhir Egde A sama dengan node awal Egde B begitu juga sebaliknya.

### 3.1.3.3 Conceptual Data Model (CDM)

Menurut Connolly (2010, p467) *conceptual data model* (CDM) adalah proses membangun model data yang digunakan di dalam suatu medesain database bersifat independent dari semua pertimbangan fisikal. CDM merupakan penggambaran yang lebih lengkap dari ERD dalam penggambaran data serta hubungan antara data – data atau objek. Berikut merupakan conceptual data model dari ERD pada gambar 3.5

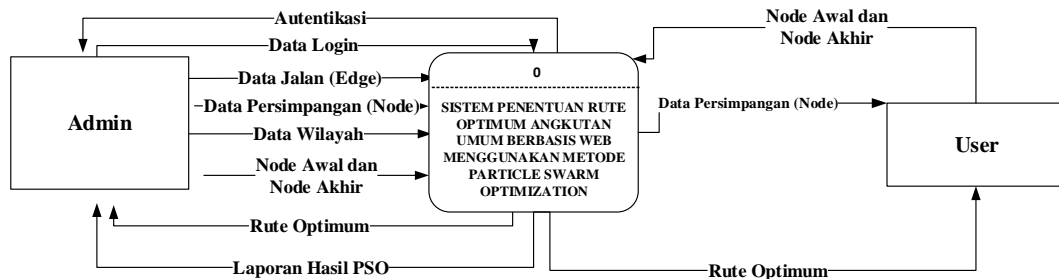


Gambar 3. 5 Conceptual Data Model

### 3.1.3.4 Data Flow Diagram

Dalam perancangan sistem informasi manajemen ini, *data flow diagram* digunakan sebagai representasi atau gambaran dari aliran data pada sistem. Berikut akan dipaparkan lebih lanjut mengenai perancangan aliran data pada sistem.

## 1. Context Diagram



Gambar 3. 6 Context Diagram Sistem

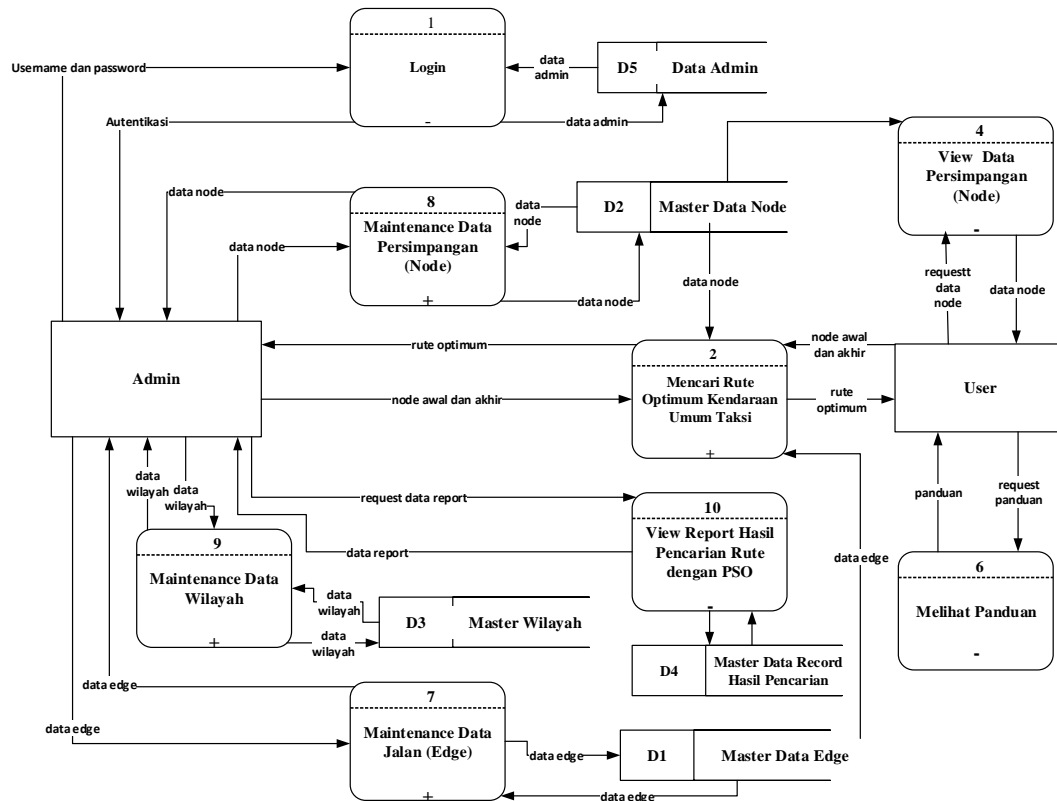
Perancangan Context Diagram pada Gambar 3.6 Terdapat dua buah entitas yang berhubungan dengan sistem yaitu Admin dan User. Admin dapat memberikan inputan data jalan, persimpangan, dan wilayah karena admin diberikan akses untuk maintenance data sistem. Admin juga diberikan fitur untuk mencari rute optimum, dengan memberikan inputkan ke sistem berupa Node awal dan node akhir. Admin juga diberikan akses untuk melihat *record* hasil pencarian rute yang sebelumnya telah dilakukan, lengkap dengan parameter yang digunakan. Untuk autentifikasi admin, diperlukan username dan password yang sesuai dengan data di database.

User hanya diberikan akses untuk melihat daftar persimpangan dan melakukan proses pencarian rute dengan sebelumnya menginputkan persimpangan awal dan akhir pada aplikasi *mobile*

## 2. DFD Level 0

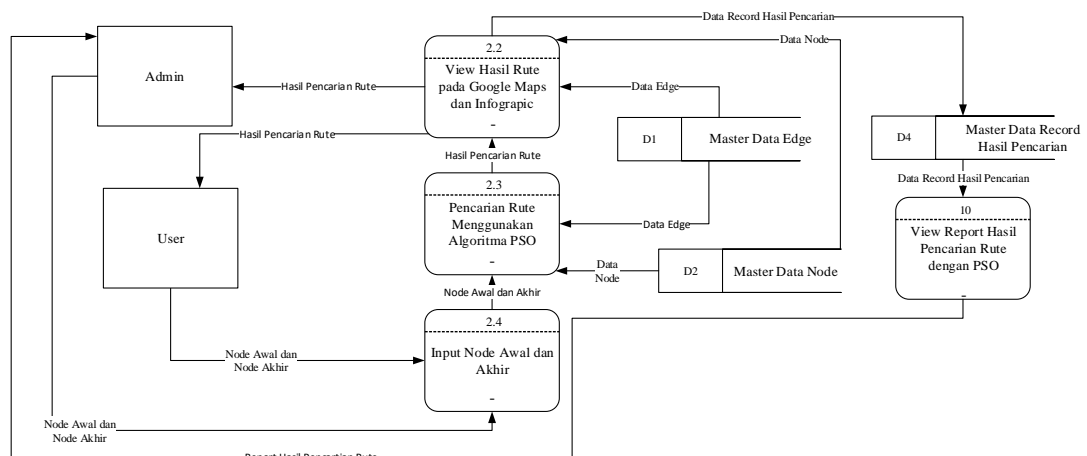
Pada rancangan DFD Level 0 pada Gambar 3.7 menunjukkan seluruh proses yang berada di dalam sistem. Total terdapat 8 proses yang terjadi yaitu proses login, proses mencari rute optimum kendaraan umum taksi, proses view data persimpangan, proses dan melihat panduan, maintainace data persimpangan, edge, dan wilayah serta proses melihat record hasil dari Algoritma PSO sebelumnya. Terdapat 5 database yang terlibat didalam proses yaitu *Master Data Edge*, *Master Data Node*, *Master Data Wilayah*, *Master Data Record Hasil Pencarian* dan *Master Data Admin*. Tanda plus (+) dan (-) menunjukkan proses tersebut apakah

memiliki sub proses lagi di dalamnya. Jika plus (+) maka memiliki sub proses begitupun sebaliknya.



Gambar 3. 7 Data Flow Diagram Level 0

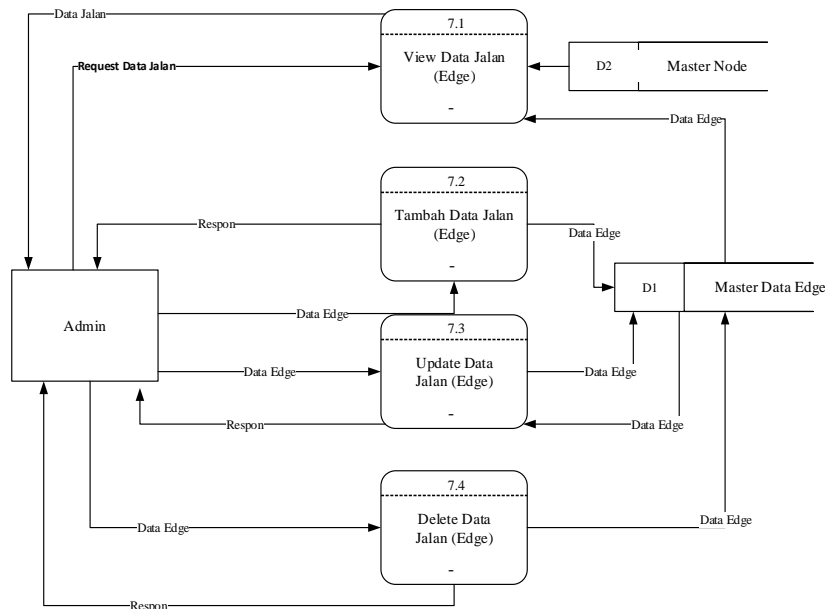
### 3. DFD Level 1 Sub Proses Mencari Rute Optimum Kendaraan Umum (Taksi)



Gambar 3. 8 DFD Level 1 Sub Proses Mencari Rute Optimum Kendaraan Umum

Gambar 3.8 merupakan subproses dari pencarian rute optimum. Terdapat 3 proses yang terjadi yaitu User dan Admin menginputkan data node awal dan akhir yang kemudian akan masuk ke dalam proses 2.3 yaitu perhitungan menggunakan Algoritma PSO. Untuk proses 2.3, diperlukan data yang bersumber dari master data node dan edge. Hasil yang didapatkan dari proses 2.3 akan masuk ke proses 2.2 yaitu proses menampilkan hasil perhitungan dari proses 2.3 ke bentuk infographic atau akan ditampilkan di google maps. Hasil dari proses akan disimpan ke dalam master data *Record Hasil Pencarian*

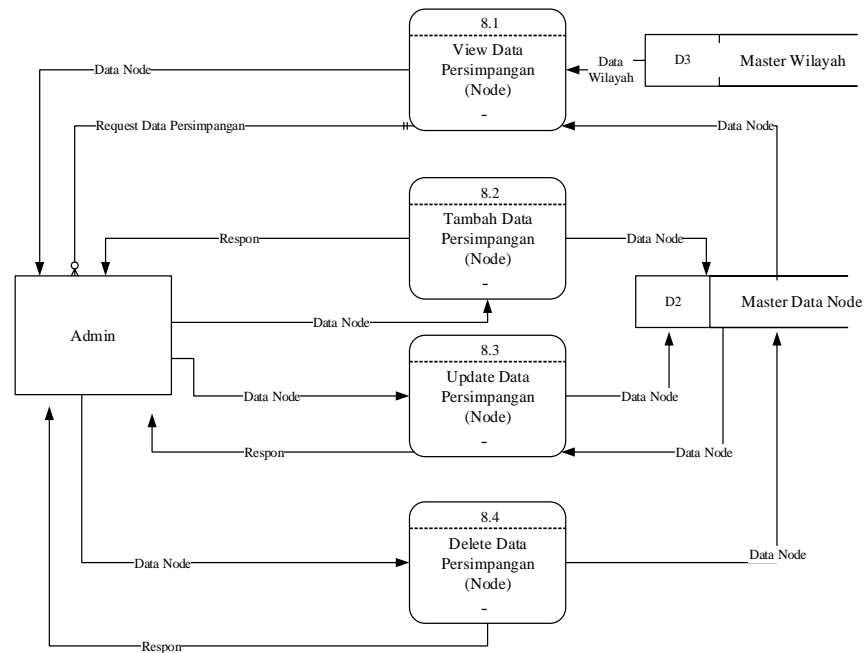
#### 4. DFD Level 1 Sub Proses Maintenance Data Jalan



Gambar 3. 9 DFD Level 1 Sub Proses Maintenance Data Jalan

Pada gambar 3.9, terdapat 4 proses yang terjadi yaitu proses *View*, *Tambah*, *Update*, dan *Delete* Data jalan yang hanya bisa dilakukan oleh Admin. Proses *Tambah* dan *Update* mengambil data dari Master Data Edge, serta proses *tambah* dan *delete* mengirimkan data ke Master Data Edge. Proses *View* data edge juga mengambil data dari Master Data Node untuk menampilkan node-node yang dihubungkan oleh edge.

## 5. DFD Level 1 Sub Proses Maintenance Data Persimpangan

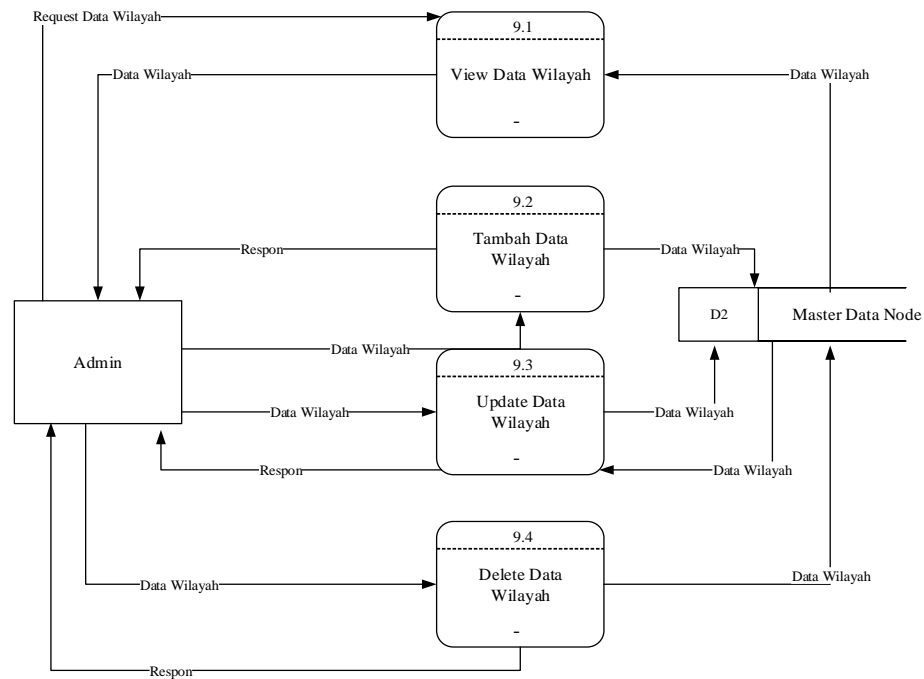


Gambar 3. 10 DFD Level 1 Sub Proses Maintenance Data Persimpangan

Pada gambar 3.10, terdapat 4 proses yang terjadi yaitu proses *View*, *Tambah*, *Update*, dan *Delete* Data Node atau persimpangan yang hanya bisa dilakukan oleh Admin. Proses *Tambah* dan *Update* mengambil data dari Master Data Persimpangan, serta proses *tambah* dan *delete* mengirimkan data ke Master Data Persimpangan. Proses *View* data persimpangan juga mengambil data dari Master Data Wilayah untuk menampilkan wilayah dari node yang ditampilkan.



## 6. DFD Level 1 Maintenance Data Wilayah



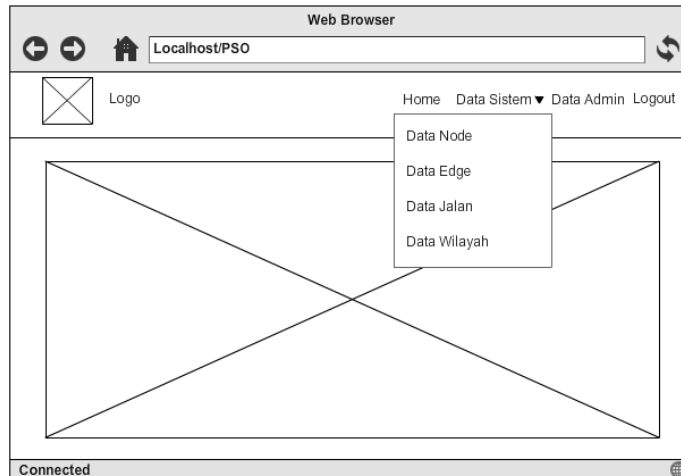
Gambar 3. 11 DFD Level 1 Maintenance Data Wilayah

Pada gambar 3.11, terdapat 4 proses yang terjadi yaitu proses *View*, *Tambah*, *Update*, dan *Delete* Data Wilayah yang hanya bisa dilakukan oleh Admin. Proses *Tambah* dan *Update* mengambil data dari Master Data Wilayah, serta proses *tambah* dan *delete* mengirimkan data ke Master Data Wilayah.

### 3.1.3.5 Rancangan Antramuka

#### 1. Tampilan Halaman Utama Admin

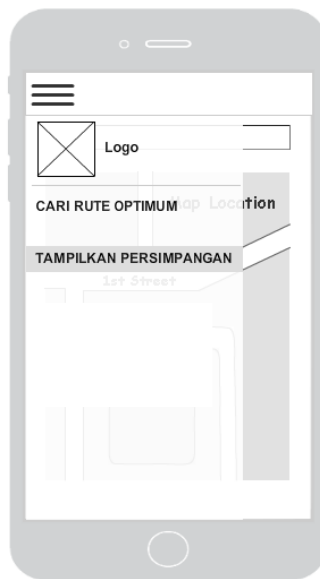
Tampilan halaman utama pada Gambar 3.12 adalah tampilan antar muka pada sistem yang menampilkan seluruh menu atau fitur dari program. Beberapa menu yang dapat dipilih oleh admin antara lain adalah Home, Data Sistem, Data Admin dan Logout. Pada menu Data Sistem terdapat empat sub menu yang menampilkan data masing-masing Data Node, Data Edge dan Data Wilayah.



Gambar 3. 12 Halaman Utama Admin

## 2. Tampilan Menu Utama User

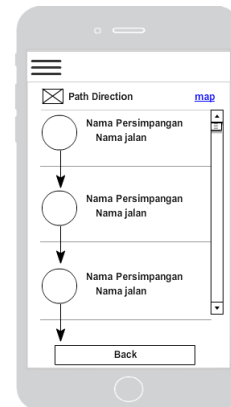
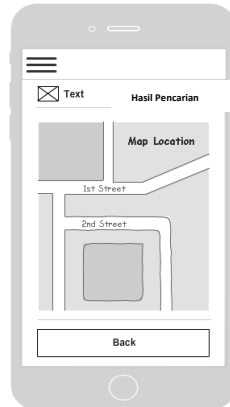
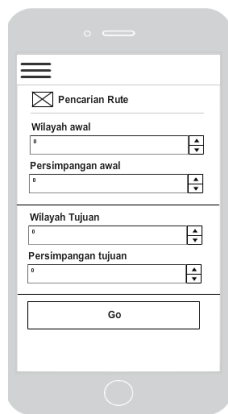
Pada menu utama pada Gambar 3.13, jika user menekan tombol menu di sebelah kiri atas, maka akan muncul *slider* yang akan menampilkan seluruh fitur dari sistem. Terdapat empat menu utama yaitu Cari Rute Optimum dan Tampilkan Persimpangan.



Gambar 3. 13 Menu Utama User

### 3. Tampilan Halaman Pencarian Rute Optimal

Pada Gambar 3.14 terdapat empat buah spinner untuk nama persimpangan yang akan dijadikan sebagai titik asal dan titik tujuan. *Spinner* dibuat sejajar simetris ke bawah untuk membuat user memilih imputan secara teratur dari atas ke bawah. Ketika mengklik tombol ‘Go’ user akan mendapatkan hasil berupa panduan arah rute optimal berupa mode tampilan *infographic* pada Gambar 3.16 dalam bentuk teks dan gambar. Kemudian ingin melihat simulasi dalam bentuk peta pada Google Maps, *user* tinggal memilih *link* “map” pada pojok kanan atas mode tampilan *infographic* untuk melihat simulasi jalur optimal yang di dapatkan dalam bentuk peta sesuai dengan Gambar 3.15.



Gambar 3. 14 Spinner    Gambar 3. 15 Tampilan Google Map    Gambar 3. 16 Tampilan Infografik

#### 3.1.5 Pengujian Sistem

Setelah perancangan antar muka berhasil dibuat, maka tahapan selanjutnya adalah melakukan perancangan scenario pengujian sistem. Dalam sub bab ini akan dipaparkan perancangan scenario pengujian sistem. Pada penelitian ini akan dilakukan tiga tahap pengujian yaitu pengujian *Black Box Testing*, *White Box Testing* dan Pengujian Akurasi Perbandingan Hasil Algoritma Particle Swarm Optimization dan Dijkstra.

##### 1. Black Box Testing

Pada pengujian *black box* akan dilakukan dengan menjalankan sistem dan menguji tingkah laku dari sistem apakah keluaran yang dihasilkan sistem sudah

sesuai dengan masukan yang diberikan. Sebelum melakukan pengujian, ditentukan terlebih dahulu skenario pengujian untuk memaparkan urutan yang akan diuji pada sistem. Adapun skenario dari pengujian dari *blackbox testing* yang dapat dilihat dari Tabel 3.4 dibawah ini.

Tabel 3. 4 Skenario Pengujian Black Box

Kelas Uji	Nama Kasus Uji	Kode Uji	Jenis Pengujian	Pengguna
Pengujian Antarmuka <i>User</i>	Pengujian Fungsi Menampilkan Rute Perjalanan Optimum dari titik awal ke akhir sesuai dengan inputan <i>User</i> pada Google Maps.	1-1	Black Box	User
	Pengujian Fungsi Menampilkan Rute Perjalanan Optimum dari titik awal ke akhir sesuai dengan inputan <i>User</i> pada tampilan Infografis	1-2	Black Box	User

	Pengujian Fungsi Menampilkan seluruh Persimpangan (Node) pada Wilayah Denpasar Utara.	1-3	Black Box	User
	Pengujian Fungsi Menampilkan Menu Detail Persimpangan	1-4	Black Box	User
	Pengujian Fungsi Menampilkan Menu Bantuan	1-5	Black Box	User
Pengujian Login	Pengujian Fungsi Validasi Login Admin	2-1	Black Box	User
Pengujian Antarmuka <i>Admin</i>	Pengujian Fungsi Menampilan Rute Perjalanan Optimum menggunakan Algoritma Particle Swarm Optimization	3-1	Black Box	Admin

	Menampilkan Rute Perjalanan Optimum menggunakan Algoritma Particle Swarm Optimization pada Google Maps	3-2	Black Box	Admin
	Menampilkan Tabel Perbandingan Hasil dari Algoritma PSO dengan Dijkstra	3-3	Black Box	Admin
Pengujian Fungsi Manajemen Data Persimpangan (Node)	Pengujian Fungsi Tambah Data Persimpangan	4-1	Black Box	Admin
	Pengujian Fungsi Edit Data Persimpangan	4-2	Black Box	Admin
	Pengujian Fungsi Hapus Data Persimpangan	4-3	Black Box	Admin

	Pengujian Fungsi Menampilkan Data Persimpangan	4-4	Black Box	Admin
Pengujian Fungsi Manajemen Data Jalan	Pengujian Fungsi Tambah Data Jalan	4-1	Black Box	Admin
	Pengujian Fungsi Edit Data Jalan	4-2	Black Box	Admin
	Pengujian Fungsi Hapus Data Jalan	4-3	Black Box	Admin
	Pengujian Fungsi Menampilkan Data Jalan	4-4	Black Box	Admin
Pengujian Fungsi Manajemen Data Jalan	Pengujian Fungsi Tambah Data Wilayah	5-1	Black Box	Admin
	Pengujian Fungsi Edit Data Wilayah	5-2	Black Box	Admin

	Pengujian Fungsi Hapus Data Wilayah	5-3	Black Box	Admin
	Pengujian Fungsi Menampilkan Data Wilayah	5-4	Black Box Admin	Admin

Tabel 3. 5 Rancangan Hasil Pengujian Black Box

Kode Uji			
Nama Kasus Uji			
Deskripsi			
Tujuan			
Pengguna			
Kondisi Awal			
Tanggal Pengujian			
Skenario			
Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan	

## 2. White Box Testing

*White box testing* merupakan sebuah pengujian yang dilakukan ketika *tester* memiliki *source code* dari sistem yang akan diuji sehingga dapat dilakukan pengujian terhadap *source code* dari program. *White Box Testing* dilakukan oleh tester dengan cara melakukan pengujian terhadap setiap fungsi *code* (Pressman, 2010).



Tabel 3. 6 Rencana Pengujian White Box

Keterangan	Flowgraph

### 3. Pengujian Perbandingan Hasil Algoritma PSO dan Dijkstra

Untuk proses pengujian perbandingan hasil dari Algoritma PSO dan Dijkstra, dilakukan dengan tabel berikut. Untuk menguji kemampuan dari Algoritma *Particle Swarm Optimization* dalam menentukan rute pada graph, diperlukan pembandingan untuk mendapatkan rute terbaik. Algoritma Dijkstra dengan konsep *greedy* yaitu menemukan solusi terbaik di setiap tahapannya, mampu untuk mendapatkan rute dengan hasil yang optimal. Untuk menguji keoptimalan dari Algoritma Particle Swarm Optimization, digunakan Algoritma Dijkstra sebagai pembandingan. Berikut adalah rencana pengujian perbandingan hasil Algoritma PSO dan Dijkstra. Akan dilakukan percobaan sebanyak 100 kali dengan node awal dan akhir yang berbeda. Kemudian akan dibandingkan bobot hasil dari Algoritma PSO dan Dijkstra.

Tabel 3. 7 Rencana Perbandingan Hasil Algoritma PSO dan Dijkstra

No.	Node Awal	Node Akhir	Rute PSO	Bobot PSO	Rute Dijkstra	Bobot Dijkstra	Keterangan

