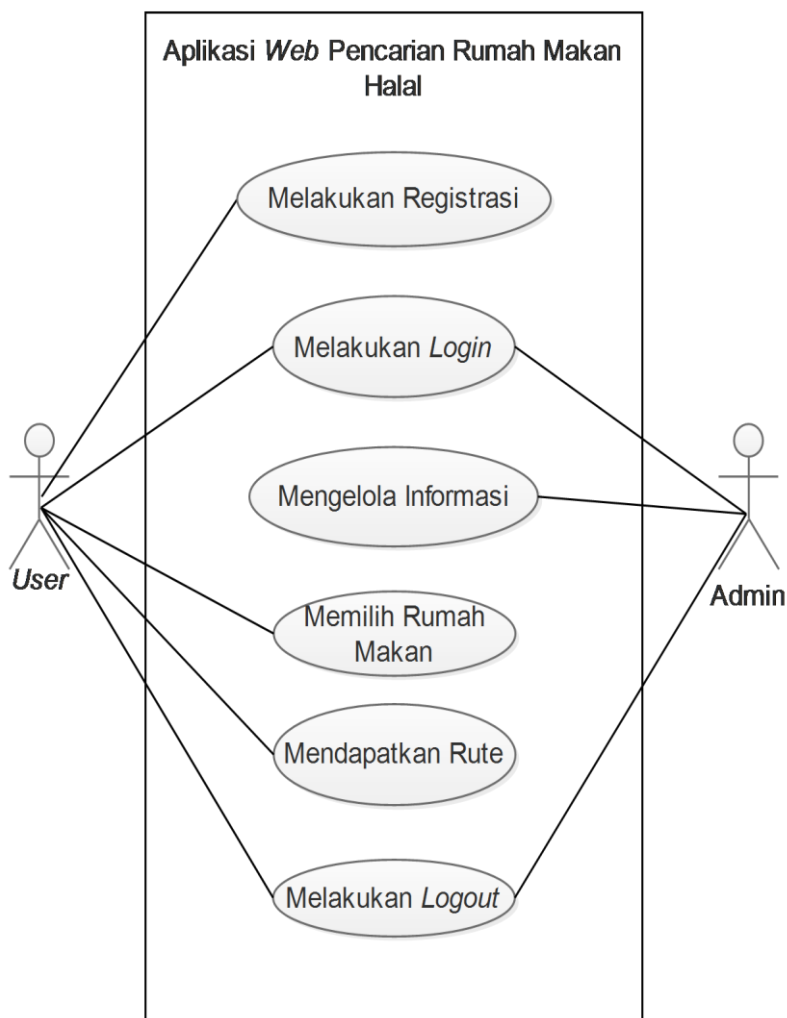


## BAB IV PERANCANGAN

Tahap Desain memodelkan rancangan sistem berdasarkan hasil analisis dari tahap analisis. Tahap ini memodelkan rancangan sistem baru yang berbentuk aplikasi. Rancangan sistem baru berupa struktur menu, rancangan basis data dan fitur-fitur. Untuk memodelkan data digunakan UML (*Unified Modelling Language*) berupa *Use Case Diagram*, *Class Diagram* dan *Activity Diagram*.

### 4.1. Memodelkan *Use Case Diagram* Pada Sistem Baru

Tahap ini bertujuan untuk memodelkan alur kerja pada sistem yang akan dibangun dengan menggunakan kaskas pemodelan *use case diagram*.



Gambar 4.1 *Use Case Diagram* Sistem Baru

#### 4.1.1. Use Case Table pada Sistem Baru

Untuk menjelaskan *use case diagram* pada Gambar 4.1 maka dibuatlah *use case table*. Pada Tabel 4.1 merupakan *use case table* untuk *use case* Melakukan Registrasi.

**Tabel 4.1 Use Case Melakukan Registrasi**

Nama Use Case :	Melakukan Registrasi
Aktor :	User
Deskripsi :	Melakukan Registrasi untuk menggunakan fitur yang disediakan oleh aplikasi.
Normal Course :	<ol style="list-style-type: none"> <li>1. User mengakses halaman registrasi</li> <li>2. User memasukkan data sesuai dengan form yang disediakan dengan benar.</li> <li>3. Registrasi berhasil.</li> </ol>
Alternate Course :	<ol style="list-style-type: none"> <li>1. User mengakses halaman registrasi.</li> <li>2. User salah memasukkan data sesuai dengan form yang disediakan.</li> <li>3. Registrasi gagal.</li> </ol>
Pre-Condition :	-
Post-Condition :	Use Case Melakukan Login
Asumsi :	Pengunjung berhasil melakukan registrasi.

Pada Tabel 4.2 merupakan *use case table* untuk *use case* Melakukan Login.

**Tabel 4.2 Use Case Melakukan Login**

Nama Use Case :	Melakukan Login
Aktor :	Admin dan User
Deskripsi :	User atau admin Melakukan Login ke halaman sesuai dengan hak akses pada aplikasi.
Normal Course :	<ol style="list-style-type: none"> <li>1. User atau admin mengakses halaman login.</li> <li>2. User atau admin memasukkan username dan password dengan benar.</li> <li>3. Login berhasil.</li> </ol>
Alternate Course :	<ol style="list-style-type: none"> <li>1. User atau admin mengakses halaman login.</li> <li>2. User atau admin salah memasukkan username dan password sesuai form yang disediakan.</li> <li>3. Login gagal.</li> </ol>
Pre-Condition :	-
Post-Condition :	Use Case Mengelola Informasi
Asumsi :	Aplikasi menampilkan halaman sesuai dengan hak akses pengguna.

Pada Tabel 4.3 merupakan *use case table* untuk *use case* Mengelola Informasi.

**Tabel 4.3 Use Case Mengelola Informasi**

Nama <i>Use Case</i> :	Mengelola Informasi
Aktor :	Admin
Deskripsi :	Mengelola Informasi di dalam aplikasi berupa daftar rumah makan, perempatan, pertigaan dan pengguna dengan hak akses admin.
Normal <i>Course</i> :	1. Admin melakukan <i>login</i> . 2. Admin mengelola informasi pada aplikasi.
<i>Alternate Course</i> :	-
<i>Pre-Condition</i> :	<i>Use Case</i> Melakukan <i>Login</i> .
<i>Post-Condition</i> :	<i>Use Case</i> Mengelola Informasi
Asumsi :	Aplikasi berhasil memproses fungsi mengelola informasi pada halaman admin aplikasi.

Pada Tabel 4.4 merupakan *use case table* untuk *use case* Memilih Rumah Makan.

**Tabel 4.4 Use Case Memilih Rumah Makan**

Nama <i>Use Case</i> :	Memilih Rumah Makan.
Aktor :	<i>User</i>
Deskripsi :	Memilih rumah makan halal yang ingin dikunjungi.
Normal <i>Course</i> :	<i>User</i> memilih rumah makan.
<i>Alternate Course</i> :	-
<i>Pre-Condition</i> :	<i>Use Case</i> Melakukan <i>Login</i> .
<i>Post-Condition</i> :	<i>Use Case</i> Mendapatkan Rute.
Asumsi :	Aplikasi berhasil menampilkan daftar rumah makan halal.

Pada Tabel 4.5 merupakan *use case table* untuk *use case* Mendapatkan Rute.

**Tabel 4.5 Use Case Memilih Mendapatkan Rute**

Nama <i>Use Case</i> :	Mendapatkan Rute
Aktor :	<i>User</i>
Deskripsi :	Mendapatkan rute terpendek menuju rumah makan yang dipilih.
Normal <i>Course</i> :	1. <i>User</i> memilih rumah makan. 2. <i>User</i> mendapatkan rute terpendek.
<i>Alternate Course</i> :	-
<i>Pre-Condition</i> :	<i>Use Case</i> Memilih Rumah Makan.
<i>Post-Condition</i> :	<i>Use Case</i> Melakukan <i>Logout</i>
Asumsi :	Aplikasi berhasil menampilkan rute tependek menuju rumah makan yang dipilih.

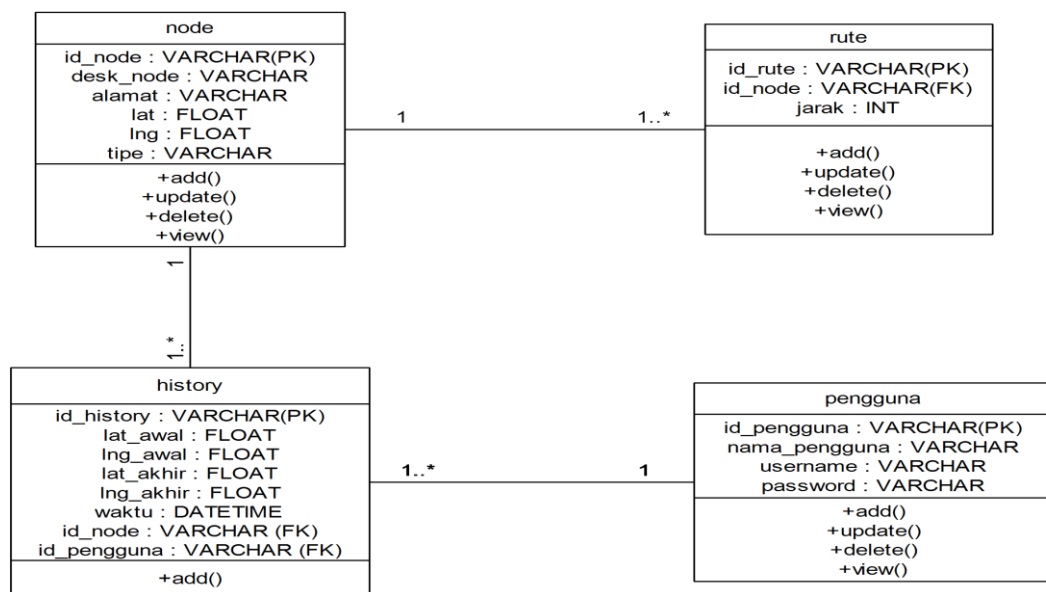
Pada Tabel 4.6 merupakan *use case table* untuk *use case* Melakukan *Logout*.

**Tabel 4.6 Use Case Memilih Melakukan Logout**

Nama Use Case :	Melakukan <i>Logout</i>
Aktor :	<i>User</i> dan Admin
Deskripsi :	Melakukan <i>Logout</i> dari halaman sesuai dengan hak akses pengguna di dalam aplikasi.
Normal Course :	1. <i>User</i> atau Admin memilih <i>button logout</i> . 2. <i>Logout</i> berhasil
Alternate Course :	-
Pre-Condition :	<i>Use Case</i> Melakukan <i>Login</i> .
Post-Condition :	-
Asumsi :	Aplikasi berhasil melakukan proses <i>logout</i> .

#### 4.2. Memodelkan Interaksi antara Objek didalam Kelas

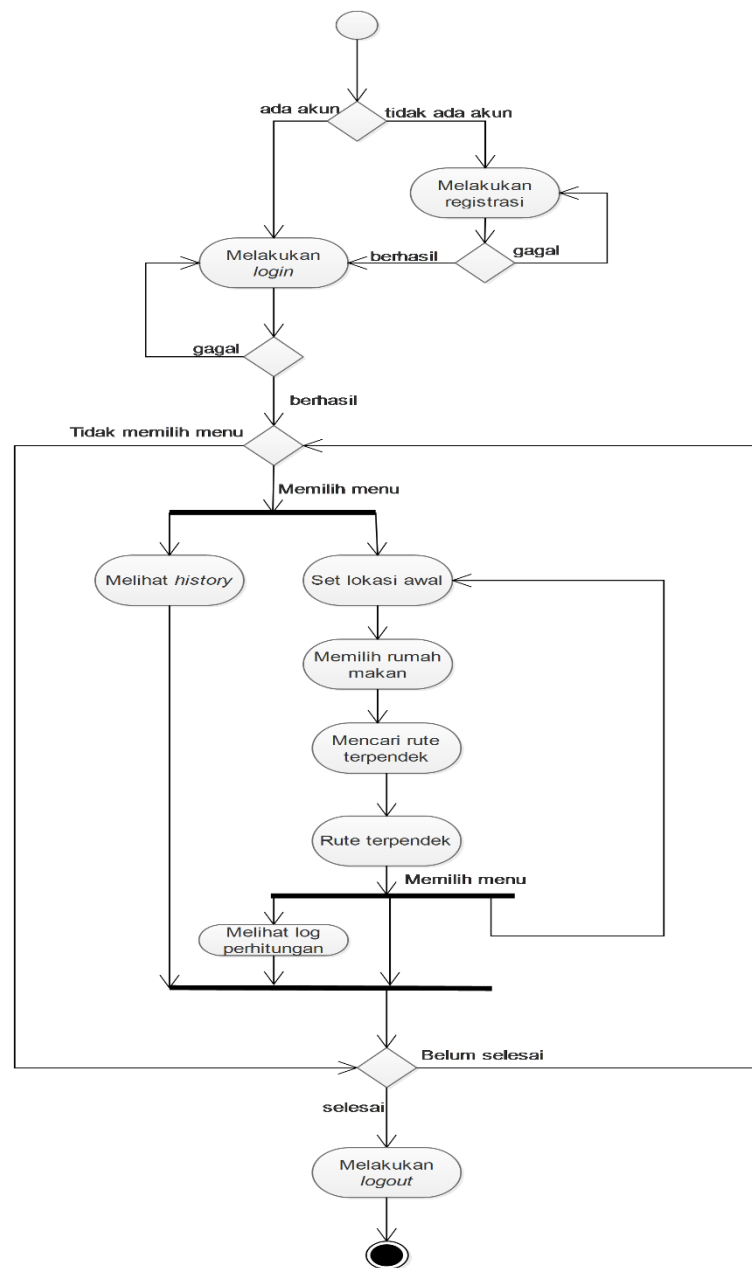
Tahap ini bertujuan untuk menggambarkan struktur sistem dari kelas-kelas yang akan dibangun. Tujuan dari tahap ini yaitu mengidentifikasi interaksi antar objek beserta atribut-atributnya dan mengklasifikasikannya demi mempermudah pembuatan basis data pada tahap implementasi. Tujuan pembuatan *Class diagram* untuk menggambarkan bentuk basis data kepada *programmer* pada tahap implementasi. *Class diagram* akan dilanjutkan pada tahap implementasi yaitu pembuatan basis data. *Class diagram* sistem baru ditunjukkan pada Gambar 4.2.



**Gambar 4.2 Class Diagram Sistem Baru**

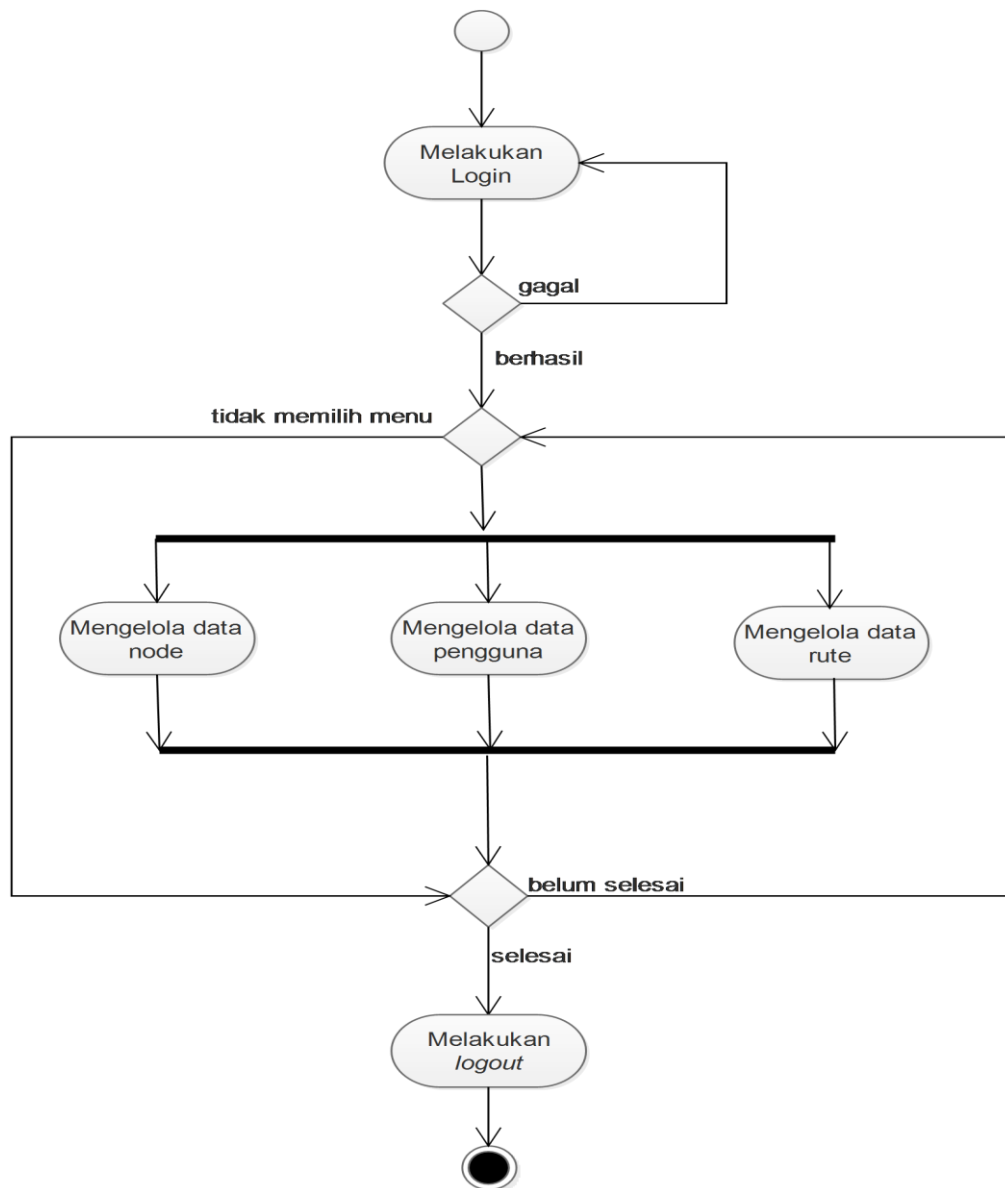
### 4.3. Menggambarkan Alur kerja Sistem Baru dengan *Activity Diagram*

Tahap ini menjelaskan alur kerja dari sistem baru yang akan dibangun. Tugas Akhir ini menggunakan kaskas pemodelan UML berupa *activity diagram*. Tahap ini bertujuan untuk mempermudah tahap implementasi. Untuk menjelaskan alur kerja sistem baru maka dibuatlah dua *activity diagram* sesuai hak akses *user*. *Activity diagram* untuk hak akses pengguna dapat dilihat pada Gambar 4.3.



Gambar 4.3 *Activity Diagram* Hak Akses User

Pada Gambar 4.4 menjelaskan tentang *activity diagram* dengan hak akses admin.



**Gambar 4.4 Activity Diagram Untuk Hak Akses Admin**

Mengelola data *node* maksudnya adalah menambahkan, mengubah atau menghapus data *node* pada aplikasi. Adapun yang termasuk *node* adalah patokan dan rumah makan. Mengelola data pengguna maksudnya adalah menambahkan, mengubah atau menghapus data pengguna pada aplikasi. Adapun yang termasuk jenis pengguna adalah admin dan pengunjung. Mengelola data rute adalah menambahkan, mengubah, atau menghapus matriks ketetangaan pada aplikasi. Adapun yang termasuk data rute adalah *node* patokan pada aplikasi.

#### 4.4. Desain Antarmuka

Pada tahap ini menjelaskan detail rancangan tampilan untuk setiap halaman pada sistem yang akan dibangun. Hasil akhir dari tahap ini yaitu tampilan *story board* dari rancangan desain tampilan sistem yang akan dibangun.

##### 1. Tampilan awal Aplikasi

Tampilan Awal Aplikasi adalah tampilan yang muncul saat pengguna mengakses aplikasi. Tampilan ini berfungsi sebagai tempat mencari rute terpendek menuju rumah makan yang dipilih pengguna.

The screenshot shows the initial application interface. At the top, there is a header with the application title "Aplikasi Web Pencarian Rumah Makan Halal terdekat di Kota Manado Menggunakan Algoritma Dijkstra" and two placeholder boxes for "Gambar Logo Aplikasi" and "Gambar Logo Kota Manado". Below the header, there is a search input field labeled "Pilih Rumah Makan :" with a dropdown menu "Nama Rumah Makan". To the right of the input field are three buttons: "Button Cari Rute", "Button Lihat Jejak", and "Button Keluar". The main content area is labeled "Tampilan Peta". At the bottom, there is a "CopyRight" label.

**Gambar 4.5 Rancangan Desain Antarmuka Halaman Awal Aplikasi**

##### 2. Tampilan Jejak

Tampilan untuk melihat jejak (*History*) pencarian rute menuju

The screenshot shows the application interface for viewing search history. At the top, there is a header with the application title "Aplikasi Web Pencarian Rumah Makan Halal terdekat di Kota Manado Menggunakan Algoritma Dijkstra" and two placeholder boxes for "Gambar Logo Aplikasi" and "Gambar Logo Kota Manado". Below the header, there are three buttons: "Button Beranda", "Button Lihat Jejak", and "Button Keluar". The main content area contains a table with the following columns: "No", "Koordinat awal", "Tanggal", "Jam", "Tujuan", and "Rute". The table is currently empty. At the bottom, there is a "CopyRight" label.

No	Koordinat awal	Tanggal	Jam	Tujuan	Rute

rumah makan oleh pengguna selama menggunakan aplikasi.

### Gambar 4.6 Rancangan Desain Antarmuka Tampilan Jejak

#### 3. Tampilan Admin

Tampilan Admin adalah tampilan yang muncul saat pengguna ingin mengubah informasi. Tampilan ini berfungsi sebagai tempat untuk mengelola informasi pada aplikasi. Rancangan desain tampilan admin digunakan untuk ketiga menu pada hak akses admin.

Gambar Logo Aplikasi		Aplikasi Web Pencarian Rumah Makan Halal terdekat di Kota Manado Menggunakan Algoritma Dijkstra		Gambar Logo Kota Manado	
Button Awal	Button Pengguna	Button Rute	Button Keluar		
Button Tambah Data					
Nama Kolom	Nama Kolom	Nama Kolom	Nama Kolom	Ops	
				Button Ubah	Button Hapus
CopyRight					

Gambar 4.7 Rancangan Desain Antarmuka Halaman Admin

#### 4. Tampilan Tambah Data

Tampilan Tambah Data adalah tampilan yang saat pengguna menekan tombol Tambah Data. Tampilan tambah data digunakan untuk menambahkan data baru pada aplikasi.

Gambar Logo Aplikasi		Teks Aplikasi Web Pencarian Rumah Makan Halal terdekat di Kota Manado Menggunakan Algoritma Dijkstra		Gambar Logo Kota Manado	
Button Awal	Button Pengguna	Button Rute	Button Keluar		
<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: auto;"> <p>Label <input type="text" value="TextBox"/></p> <p>Label <input type="text" value="TextBox"/></p> <p>Label <input type="text" value="TextBox"/></p> <p>Label <input type="text" value="TextBox"/></p> <p style="text-align: center;">Button Kirim</p> </div>					
Teks CopyRight					

**Gambar 4.8 Rancangan Desain Antarmuka Halaman Tambah Data**