

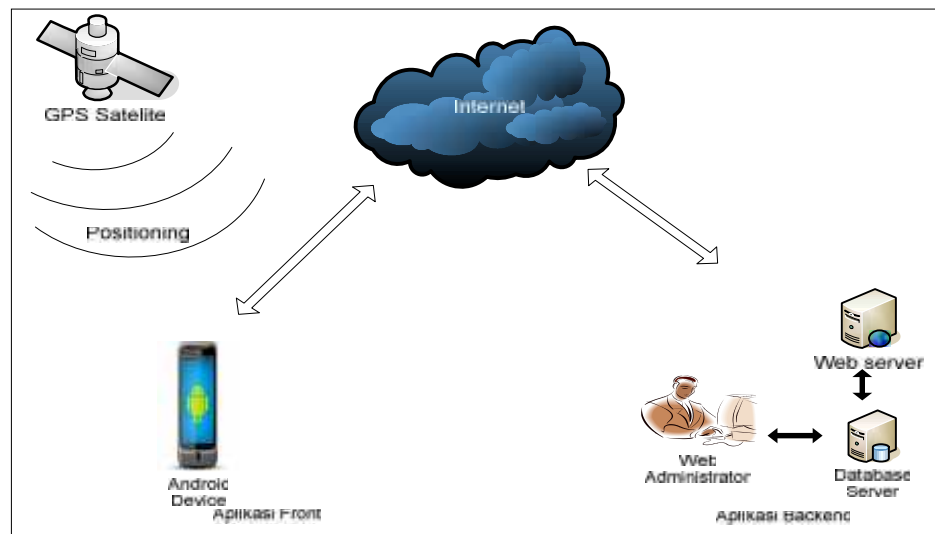
## BAB IV

### ANALISA DAN PERANCANGAN

#### 4.1 Deskripsi Umum Sistem

Sistem informasi kantor pemerintahan kota Pekanbaru yang dibangun merupakan aplikasi *mobile* menggunakan konsep *client server*. Aplikasi ini berfungsi untuk mempermudah masyarakat dalam melakukan pencarian kantor pemerintahan yang ada di kota Pekanbaru melalui perangkat berbasis Android.

Aplikasi ini menggunakan sistem LBS (*Location Based Service*) sehingga memudahkan pengguna untuk melihat dimana saja lokasi kantor pemerintahan kota Pekanbaru berada dengan tampilan peta Google Map. Sistem ini juga memanfaatkan teknologi GPS yang berguna dalam penentuan rute dari *user* berada menuju lokasi kantor pemerintahan kota Pekanbaru. Secara sederhana deskripsi umum sistem dapat dilihat melalui Gambar 4.1



Gambar 4.1 Deskripsi umum sistem

Gambar 4.1 menjelaskan bahwa android *device* melakukan komunikasi dengan GPS satelit untuk mendapatkan nilai koordinat yang kemudian diterjemahkan

menjadi nama lokasi berdasarkan posisi pengguna android. Sistem melakukan komunikasi dengan *database server* menggunakan jaringan GPRS/EDGE/HSDPA/3G. Semua data kantor pemerintahan dan data pemilik kantor pemerintahan dimasukkan oleh admin ke *database* yang ada di database server dengan perantara berupa halaman website. Komunikasi aplikasi android dengan *database server* berupa *update* data dari *database server* (MySQL) ke *database lokal* (SQLite), dan kemudian menghasilkan informasi secara lengkap mengenai kantor pemerintahan. Data awal sistem dikelola oleh administrator sistem yang berada pada aplikasi *server* dan diinputkan ke database. Pada aplikasi *front*, pengelola kantor pemerintahan melakukan update jumlah kamar kantor pemerintahan yang tersedia apabila terjadi perubahan.

#### **4.2 Analisa Kebutuhan User**

Sebelum melakukan perancangan sistem, terlebih dahulu dilakukan analisa kebutuhan *user* yang akan menggunakan sistem tersebut. Hal ini dimaksudkan agar dapat mengatasi ketidaksesuaian antara aplikasi yang dirancang dengan kebutuhan pengguna. Adapun kebutuhan sistem yang diperlukan antara lain :

1. Menampilkan informasi kantor pemerintahan yang di kota Pekanbaru
2. Menampilkan informasi mengenai lokasi keberadaan kantor pemerintahan yang di kota Pekanbaru
3. Memberikan rute jalan agar dapat mempermudah *user* menemukan rute terdekat menuju kantor pemerintahan kota Pekanbaru.

#### **4.3 Analisa Kebutuhan Data**

Dalam melakukan pembuatan aplikasi ini, ada beberapa data yang diperlukan agar sistem aplikasi ini bisa memberikan informasi yang bermanfaat bagi *user*. Data tersebut dapat dikelompokkan sebagai berikut :

1. Data kantor pemerintahan

Data kantor pemerintahan yang menjadi objek dalam sistem ini merupakan data awal yang harus ada dalam sistem ini. Data kantor pemerintahan meliputi nama kantor, area kantor pemerintahan, alamat, gambar kantor, dan lokasi. Data inilah yang akan menjadi acuan untuk melakukan pencarian kantor pemerintahan dan melihat rute/navigasi menuju kantor pemerintahan. Dalam melakukan perekaman terhadap data lokasi kantor pemerintahan, dilakukan dengan menggunakan bantuan GPS (*Global Positioning System*) yang kemudian akan disimpan berupa titik koordinat, *longitudinal* dan *latitudinal*-nya. Data tersebut yang kemudian akan dimasukkan ke dalam database.

2. Posisi pengguna

Data posisi pengguna didapat dari sinyal GPS yang diterima oleh GPS *receiver* pada *android device* kemudian di terjemahkan oleh menjadi informasi lokasi berdasarkan posisi pengguna. Data posisi pengguna ini bersifat dinamis, selalu berubah sesuai dengan lokasi akses pengguna terhadap sistem. Agar fitur navigasi menuju kantor pemerintahan dapat berfungsi, sebelumnya perangkat android harus memiliki aplikasi Map.

#### 4.4 Perencanaan Spesifikasi Sistem

Beberapa fitur yang akan dibangun pada sistem ini adalah :

1. Sistem dikembangkan dengan model arsitektur *two tier*. Aplikasi dikembangkan disisi *client* atau *frontend* sebagai aplikasi *On Device Portal* (ODP). Sedangkan disisi server atau *server* akan dikembangkan aplikasi web *Content Managemen System*(CMS).
2. Konektivitas antara aplikasi pada *platform* Android ini dengan aplikasi *server* di *server* menggunakan koneksi *protocol* HTTP dengan memanfaatkan jaringan GPRS/EDGE/UMTS.
3. Fitur sistem di sisi *user* yaitu:

a. Menampilkan list nama kantor dan jarak terhadap user

Pada fitur ini, sistem akan menampilkan list seluruh nama kantor pemerintahan yang ada. Dalam penggunaan fitur ini, GPS receiver pada android device harus sudah dalam keadaan aktif sehingga aplikasi bisa mendapatkan nilai koordinat user berada untuk menentukan jarak terhadap tiap-tiap kantor pemerintahan yang ditampilkan dalam listview. Jika salah satu nama kantor pemerintahan dipilih, maka akan masuk ke form informasi detail kantor pemerintahan yang dipilih tersebut.

b. Menampilkan peta lokasi kantor pemerintahan

Selain menampilkan seluruh kantor pemerintahan dalam bentuk listview, aplikasi juga memberikan fitur untuk menampilkan seluruh lokasi kantor pemerintahan berada dalam tampilan peta. Pada fitur ini, titik-titik lokasi seluruh kantor pemerintahan yang ada akan ditampilkan dalam peta sesuai dengan nilai koordinatnya masing-masing.

c. Informasi Detail Kantor pemerintahan

Pada saat list kantor pemerintahan ditampilkan, user bisa memilih salah satu kantor pemerintahan untuk menampilkan informasi secara detail. Adapun informasi tersebut yaitu: id kantor pemerintahan, nama kantor pemerintahan, area kantor pemerintahan, dan alamat. Pada form ini juga ditampilkan informasi mengenai kantor pemerintahan tersebut. Data pemilik kantor pemerintahan yang ditampilkan dalam fitur ini adalah nama, alamat, dan nomor telepon.

d. Rute Menuju Kantor pemerintahan

Pada form detail kantor pemerintahan, terdapat button “Rute Menuju Kantor pemerintahan”. Pada fitur ini, user dapat menampilkan peta lokasi kantor pemerintahan dalam 2 rute. Rute pertama akan memberikan jalan tercepat menuju lokasi yang akan dituju dan rute yang ke dua menggunakan rute terjauh. akan memberikan informasi daerah rawan macet apabila jalur yang dilalui user melewati titik tersebut. *GPS receiver*

pada android *device* harus sudah dalam keadaan aktif untuk menggunakan fitur ini. Fitur ini menggunakan aplikasi lain diluar sistem yang dibangun seperti Maps atau webbrowser yang ada pada perangkat android. Aplikasi hanya memberikan nilai *start* dan nilai *finish*.

e. *About*

Pada menu utama, terdapat pilihan menu *about*. Menu ini berfungsi untuk menampilkan informasi tentang pengembang sistem.

4. Fitur sistem di sisi *server* adalah:

Sistem dapat memfasilitasi Admin untuk melakukan tambah, ubah, dan hapus data kantor pemerintahan dan data pemilik kantor pemerintahan.

#### 4.5 Analisa Kebutuhan Sistem

Dalam membangun aplikasi, sebelum merancang perlu diketahui apa saja perangkat baik itu perangkat lunak maupun perangkat keras yang harus dipenuhi agar sistem yang dibangun bisa berjalan dengan semestinya. Dalam pembangunan aplikasi Kantor pemerintahan yang menggunakan layanan berbasis lokasi dan menggunakan konsep *client-server*, aplikasi yang akan dibangun memiliki kebutuhan yang harus dipenuhi yaitu:

1. Perangkat android dapat melakukan koneksi ke jaringan internet
2. Perangkat android memiliki fitur GPS *receiver*
3. Fitur GPS pada perangkat android harus sudah dalam keadaan aktif dan telah terhubung dengan baik ke pusat GPS *server*
4. Perangkat android menggunakan *Operating System* (OS) versi 2.3.6 ( *Gingerbread* ) ke atas.

#### 4.6 Analisa Fungsional

Aplikasi dibangun menggunakan konsep *client-server*. Aplikasi ini dibangun dengan bahasa pemrograman yang berbeda dan fungsional berbeda sehingga analisa

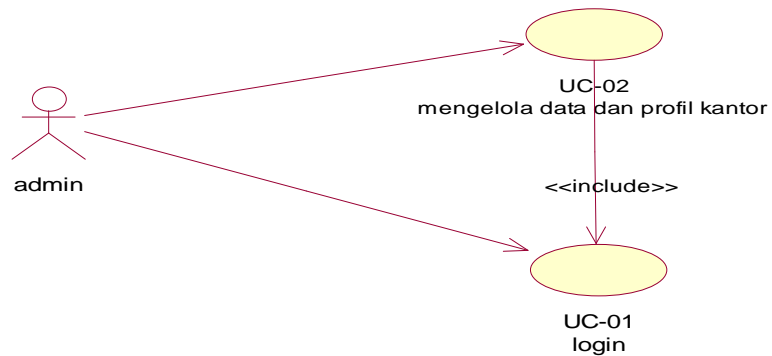
fungsional pada sistem website dari sisi *server* dipisahkan dengan analisa fungsional pada sistem android. Analisa fungsional sistem merupakan pemaparan model menggunakan pemodelan UML (*Unified Modelling Language*) yang terdiri dari *usecase diagram*, *class diagram*, *collaboration diagram*, *sequence diagram*, dan *activity diagram*.

#### 4.7 Use Case Diagram

*Use Case Diagram* adalah sesuatu yang penting untuk menspesifikasikan dan mendokumentasikan kebutuhan sistem. *Use Case Diagram* digunakan untuk menjelaskan kegiatan apa saja yang dapat dilakukan oleh pengguna sistem. Untuk *Use Case Diagram* diperlukan *actor*, *use case*, dan hubungannya. Pada *Use Case Diagram* *actor* yang diperlukan untuk membangun sistem.

##### 4.7.1. Use Case Diagram Aplikasi Server

Interaksi antara *use case* dengan aktor terhadap sistem digambarkan menggunakan *Use case diagram*. Terdapat dua aktor yang terlibat dalam aplikasi website, yaitu Admin dan Pengelola kantor. Berikut adalah gambar *usecase diagram* pada aplikasi *server*:



Gambar 4.2 *Usecase Diagram* kelola data oleh admin

### 1. Definisi Aktor

Berikut adalah pendefinisian aktor pada aplikasi :

Tabel 4.1 Definisi aktor-aktor dalam sistem *server*

No	Aktor	Keterangan
1.	Admin	Orang yang bertugas dan memiliki hak akses untuk melakukan operasi pengelolaan data kantor dan data pemilik kantor melalui halaman website.

### 2. Deskripsi *Usecase*

Deskripsi *usecase* menggambarkan kebutuhan fungsional sistem, kemudian dibuat skenario (*flow of event*) yang menggambarkan urutan skenario.

Tabel 4.2 Definisi *Usecase* dalam sistem *server*

ID	Nama <i>Usecase</i>	Keterangan
UC-01	Login	Merupakan proses pengecekan hak akses siapa yang berhak mengakses proses pengelolaan data kantor.
UC-02	Mengelola data dan profil kantor	Merupakan proses generalisasi yang meliputi tiga buah proses pengelolaan data kantor yaitu memasukkan data, merubah data, menghapus data

### 3. Skenario

Sebelum masuk pada tahap pembuatan *usecase diagram*, perlu dijabarkan skenario dari penggunaan sistem ini. Berikut adalah skenario berdasarkan dari tiap *usecase*:

Tabel 4.3 Skenario UC01-Login

Nama Usecase	: Login
Aktor	: Admin
Pre-Condition	: Admin/Pengelola berada pada halaman login
Post-Condition	: 1. User berhasil login ke sistem

2. Sistem menampilkan halaman menu utama berdasarkan hak akses aktor	
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan Id dan Password  4. Berhasil masuk ke sistem	2. Mengecek valid tidaknya data masukan 3. Menampilkan pesan login berhasil
Skenario Alternatif 1: Login Gagal	
1. Memasukkan Id dan Password	2. Mengecek valid tidaknya data masukan 3. Data masukan tidak valid 4. Menampilkan pesan login gagal

Tabel 4.4 Skenario UC02-Mengelola data kantor pemerintahan

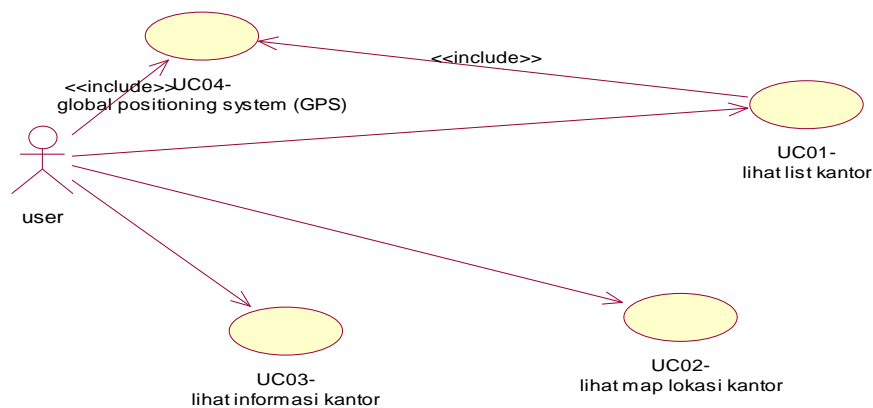
Nama Usecase	: Mengelola data dan profil kantor
Aktor	: Admin
Pre-Condition	: Admin telah masuk ke menu utama
Post-Condition	: User berhasil mengelola data kantor
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Admin memilih menu tambah titik kantor  3. Memasukkan data kantor    7. User memilih menu update titik	2. Sistem menampilkan form input titik  4. Mengecek valid tidaknya data masukan 5. Sistem memasukkan data kantor ke database 6. Menampilkan pesan data berhasil dimasukkan  8. Menampilkan form edit kantor 9. Mengecek valid tidaknya data masukan 10. Meng- <i>update</i> data kantor ke



12. User memilih menu hapus kantor	database 11. Menampilkan pesan data berhasil diedit  13. Menampilkan konfirmasi hapus data 14. Data dihapus dari database 15. Menampilkan pesan data berhasil dihapus
<b>Skenario Alternatif 1: Gagal mengelola data kantor</b>	
1. User salah memasukkan data	2. Sistem akan memvalidasi dan menampilkan pesan error.

#### 4.7.2. Use Case Diagram Aplikasi Client

Pada aplikasi *user* yang dibangun untuk perangkat android hanya ada satu aktor, yaitu aktor pencari kantor. Berikut adalah gambar *usecase diagram* pada aplikasi *user*:



Gambar 4.3 Usecase Diagram pencarian kantor

## 1. Definisi Aktor

Berikut adalah pendefinisian aktor pada aplikasi :

Tabel 4.5 Definisi aktor-aktor dalam sistem *user*

No	Aktor	Keterangan
1.	Pencari kantor	Pengguna yang menggunakan sistem pada android <i>device</i> , melakukan pencarian kantor.

## 2. Definisi *Usecase*

Deskripsi *usecase* menggambarkan kebutuhan fungsional sistem, kemudian dibuat skenario (*flow of event*) yang menggambarkan urutan skenario. Tabel dibawah ini merupakan definisi dari *usecase* dalam aplikasi *user*.

Tabel 4.6 Definisi *Usecase* dalam sistem *user*

ID	Nama <i>Usecase</i>	Keterangan
UC-01	Lihat list kantor	Melihat nama-nama kantor yang ada dengan tampilan list
UC-02	Lihat map lokasi kantor	Melihat titik lokasi kantor berada dengan tampilan peta
UC-03	Lihat informasi detail kantor	Melihat informasi secara detail terhadap salah satu kantor yang dipilih. Informasi yang ditampilkan yaitu, nama kantor, alamat, gambar eksterior bangunan.
UC-04	<i>Global Positioning System (GPS)</i>	Memberikan nilai koordinat lokasi user berada dengan penggunaan perangkat GPS yang ada pada android device.

### 3. Skenario

Sebelum masuk pada tahap pembuatan *usecase diagram*, perlu dijabarkan skenario dari penggunaan sistem ini. Berikut adalah skenario penggunaan aplikasi *user* berdasarkan dari tiap *usecase*:

Tabel 4.7: Skenario UC01-Lihat *list* kantor

Nama Usecase	: Lihat list kantor
Aktor	: Pencari Kantor
Pre-Condition	: User berada pada menu utama aplikasi pada android device
Post-Condition	: User berhasil melihat semua data kantor berupa listview
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih button List Kantor	2. Mengupdate data dari database dan GPS 3. Menampilkan list nama kantor dan jarak
Skenario Alternatif 1: Gagal koneksi database & GPS	
1. Memilih button List Kantor	2. Tidak ada koneksi, tidak dapat mengupdate data dari database dan GPS 3. Menampilkan pesan error.

Tabel 4.8 Skenario UC02-Lihat map lokasi kantor

Nama Usecase	: Lihat map lokasi kantor
Aktor	: Pencari Kantor
Pre-Condition	: User berada pada menu utama aplikasi pada android device
Post-Condition	: User berhasil melihat semua lokasi kantor berupa mapview
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih button MapView	2. Mengupdate data dari database dan GPS 3. Menampilkan peta dan lokasi semua

	kantor
<b>Skenario Alternatif 1: Gagal koneksi database &amp; GPS</b>	
1. Memilih button MapView	<ol style="list-style-type: none"> <li>2. Tidak ada koneksi, tidak dapat mengupdate data dari database dan GPS</li> <li>3. Menampilkan pesan error</li> </ol>

Tabel 4.9 Skenario UC03-Lihat informasi detail kos

Nama Usecase	: Lihat informasi detail kantor
Aktor	: Pencari Kantor
Pre-Condition	: User berada pada form List kantor atau Map Kantor
Post-Condition	: 1. User memilih salah satu kantor 2. User berhasil masuk ke form informasi detail kantor yang dipilih
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
<b>Skenario Normal</b>	
1. Memilih salah satu kantor	<ol style="list-style-type: none"> <li>2. Mengupdate data dari database</li> <li>3. Menampilkan informasi detail kantor yang dipilih</li> </ol>
<b>Skenario Alternatif 1: Gagal koneksi database</b>	
1. Memilih salah satu kantor	<ol style="list-style-type: none"> <li>2. Tidak ada koneksi, tidak dapat mengupdate data dari database</li> <li>3. Menampilkan pesan error</li> </ol>

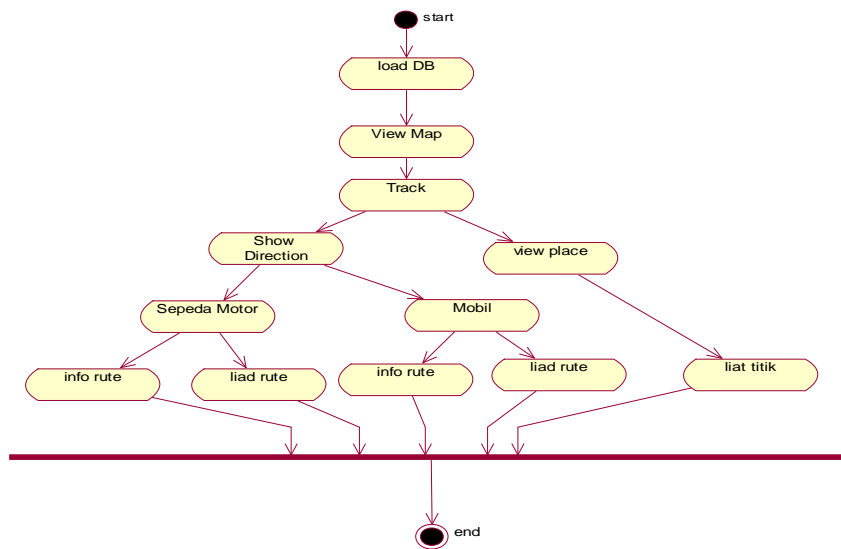
Tabel 4.10: Skenario UC04-Global Positioning System (GPS)

Nama Usecase	: Global Positioning System (GPS)
Aktor	: Pencari Kos
Pre-Condition	: Data koordinat user tidak diketahui
Post-Condition	: Data koordinat user didapat
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
<b>Skenario Normal</b>	
1. Memilih fitur-fitur yang menggunakan GPS	<ol style="list-style-type: none"> <li>2. GPS Receiver mengirim sinyal ke satelit GPS</li> </ol>

	3. Mendapatkan nilai koordinat user berada
<b>Skenario Alternatif 1: Gagal koneksi GPS Satelit</b>	
1. Memilih fitur-fitur yang menggunakan GPS	2. GPS Receiver tidak aktif 3. Gagal mendapatkan nilai koordinat user.

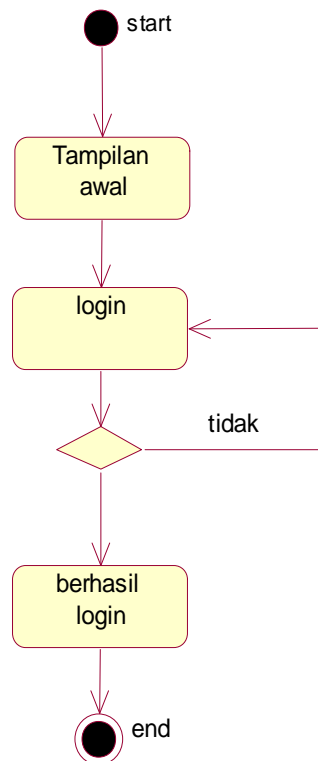
### 4.7.3. Activity Diagram

Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.



Gambar 4.4 Activity Diagram Aplikasi

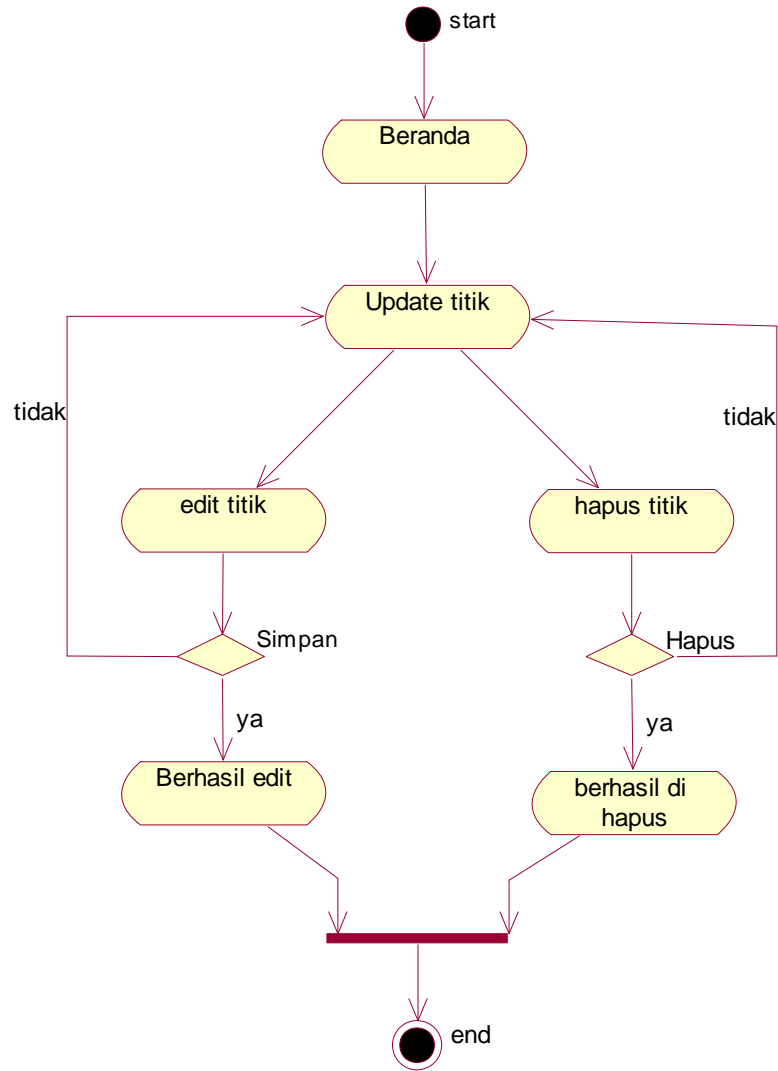
Pada Gambar 4.4 menjelaskan tentang aktivitas yang terjadi pada aplikasi pencarian lokasi kantor pemerintahan kota Pekanbaru. Di awali dengan proses pemanggilan database, kemudian menampilkan Map. Setelah map muncul maka pengguna dapat mencari daftar kantor yang terdapat pada *button track*. Didalam *button track* terdapat *list* kantor, setelah memilih salah satu kantor maka akan muncul *button show direction* dan *button view place*. Pada *button show direction* terdapat 2 buah pilihan *button* sepeda motor dan mobil. Isi dari terdapat info rute liat rute.



Gambar 4.5 Activity Diagram Login pada Admin

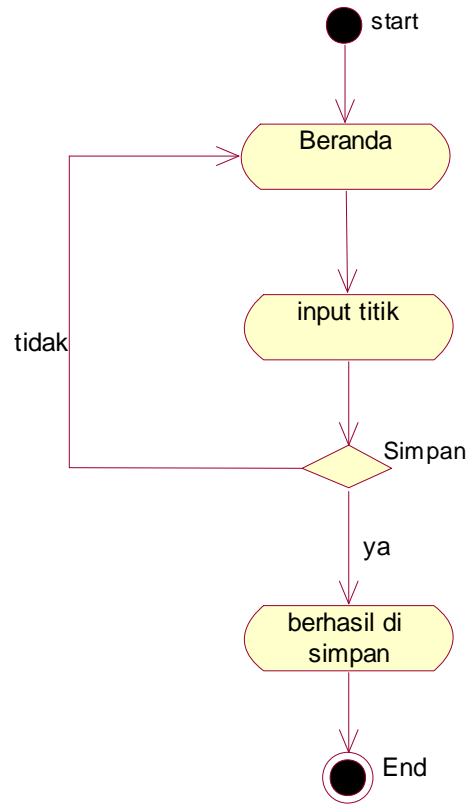
Pada gambar 4.5 menjelaskan tentang aktivitas pada saat admin akan login. Setelah menampilkan menu utama maka admin harus login ke sistem server dengan

memasukan username dan password, apabila tiusername dan password tidak valid maka admin harus login kembali.



Gambar 4.6 Activity Diagram Update pada Admin

Pada gambar 4.6 menjelaskan tentang aktifitas pada saat admin ingin mengupdate data pada server. User admin masuk pada halaman beranda dan memilih update titik, setelah itu memilih edit titik atau menghapus titik.



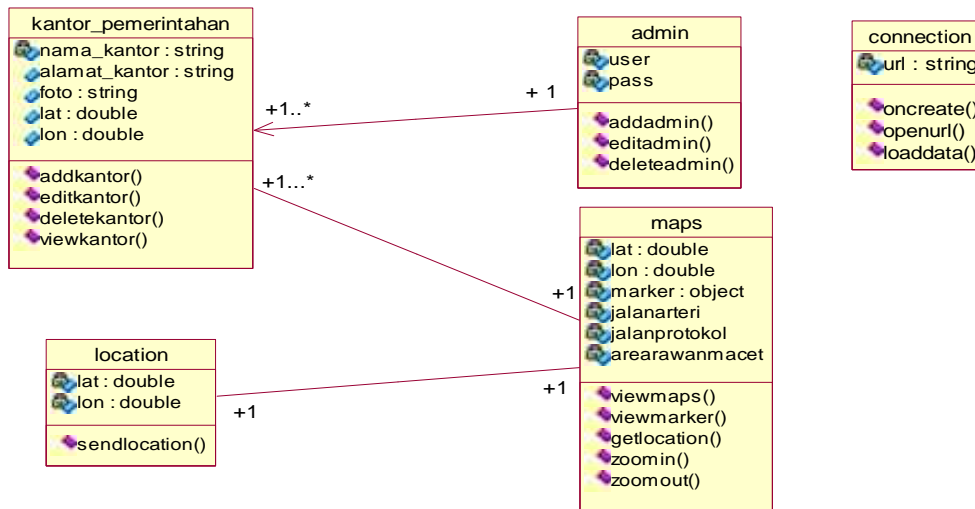
Gambar 4.7 *Activity Diagram* input titik pada Admin

Pada gambar 4.7 menjelaskan tentang input data pada server dimulai dengan input titik, setelah data data di simpan maka muncul pemberitahuan data berhasil disimpan.

#### 4.7.4. *Class Diagram*

*Class diagram* dalam sistem aplikasi menggambarkan hubungan antara class yang ada pada . Penggambaran *Class diagram* dibawah ini akan menampilkan *atribut* dan *method* yang ada pada suatu *class*.





Gambar 4.8 *Class Diagram* aplikasi

Pada *class diagram* di atas terlihat beberapa class yang ada pada aplikasi dan hubungan antar class-class yang ada di database aplikasi tersebut.

Tabel 4.11 Definisi class-class pada *class diagram*

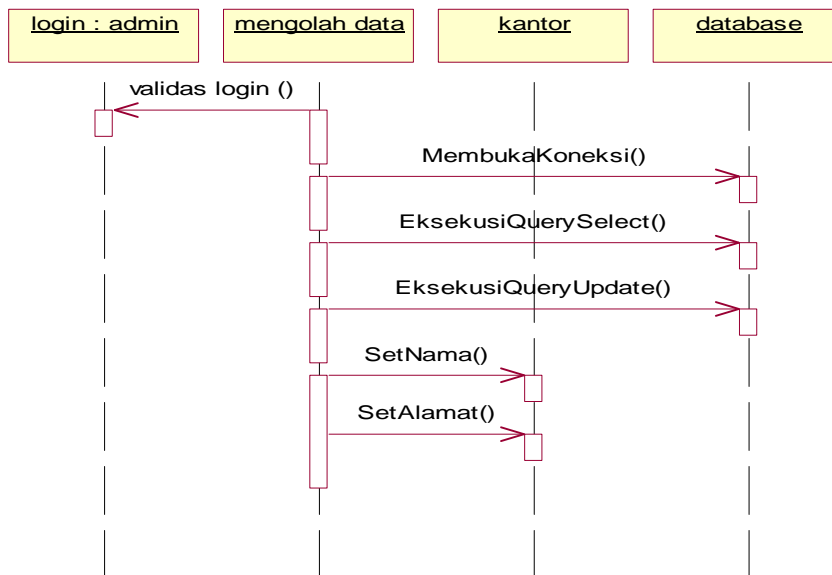
<b>Class</b>	<b>Keterangan</b>
Kantor pemerintahan	Entitas kantor yang berisi atribut-atribut yang dimiliki pada tiap-tiap kantor. Operasi yang dapat dilakukan pada class kantor adalah melakukan penambahan data kantor, merubah data kantor, dan menghapus data kantor. Operasi untuk menampilkan data kantor juga dapat dilakukan terutama pada aplikasi android.
admin	Class admin merupakan entitas admin sistem. Admin memiliki atribut yang hanya dibutuhkan dalam pengoperasian sistem seperti username dan password. Operasi pada class admin yaitu menambah, mengedit, dan

	menghapus.
Maps	Class maps merupakan class dari peta lokasi kantor yang akan ditampilkan pada aplikasi .
Location	Class Location merupakan class yang berisi objek-objek lokasi <i>user</i> dan lokasi kantor berada.
Connection	Class Connection merupakan class yang menjadi penghubung antara aplikasi <i>server</i> , aplikasi <i>user</i> dengan <i>database server</i> .

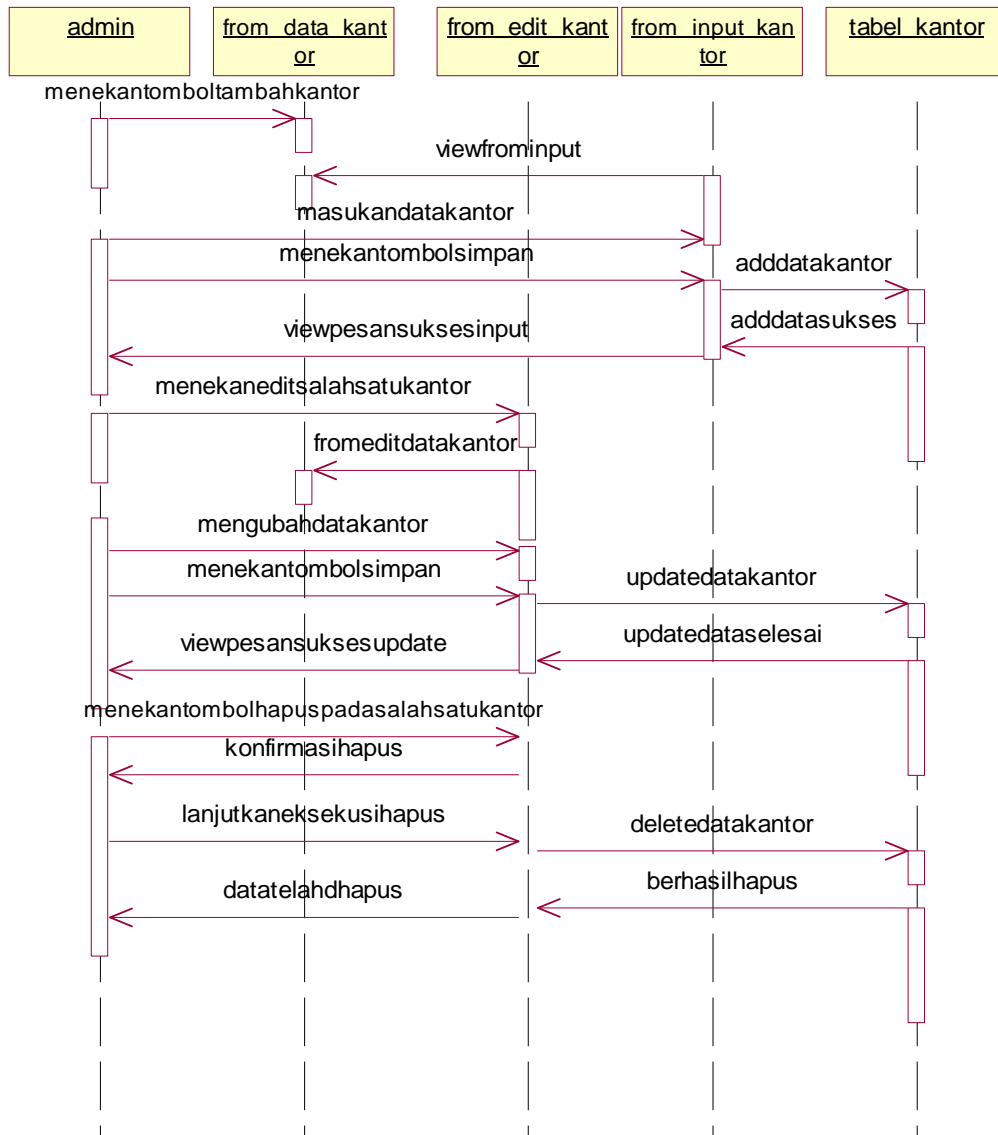
#### 4.7.5. Sequence Diagram

*Sequence diagram* pada menjelaskan pemodelan sistem yang menunjukkan interaksi antara objek yang disusun berdasarkan urutan waktu. Diagram ini sangat berkaitan dengan *use case diagram* karena merupakan tahapan proses (apa yang dihasilkan) pada *use case*.

##### 1. Sequence Diagram aplikasi server

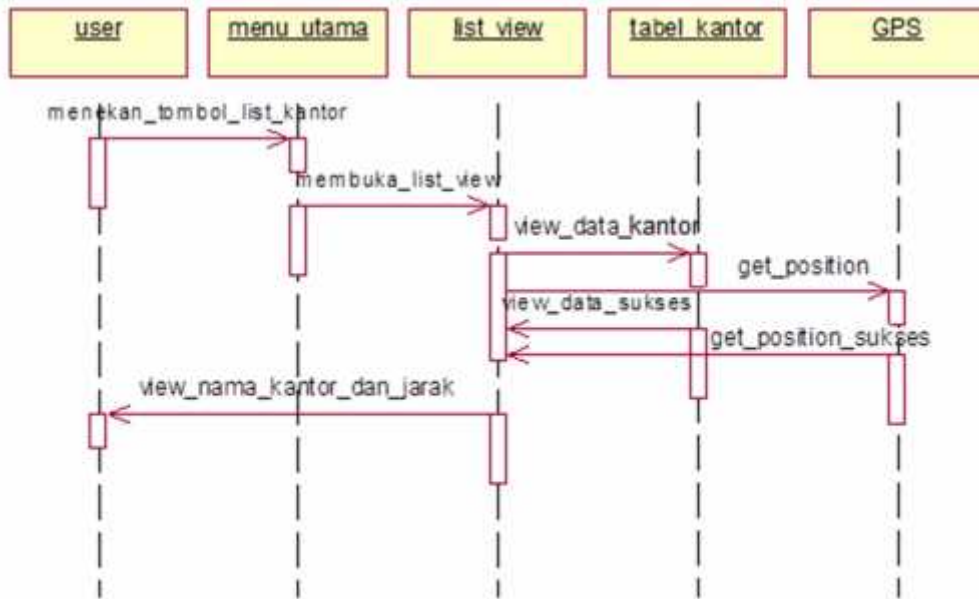


Gambar 4.9 *Sequence Diagram* Skenario UC01-Login

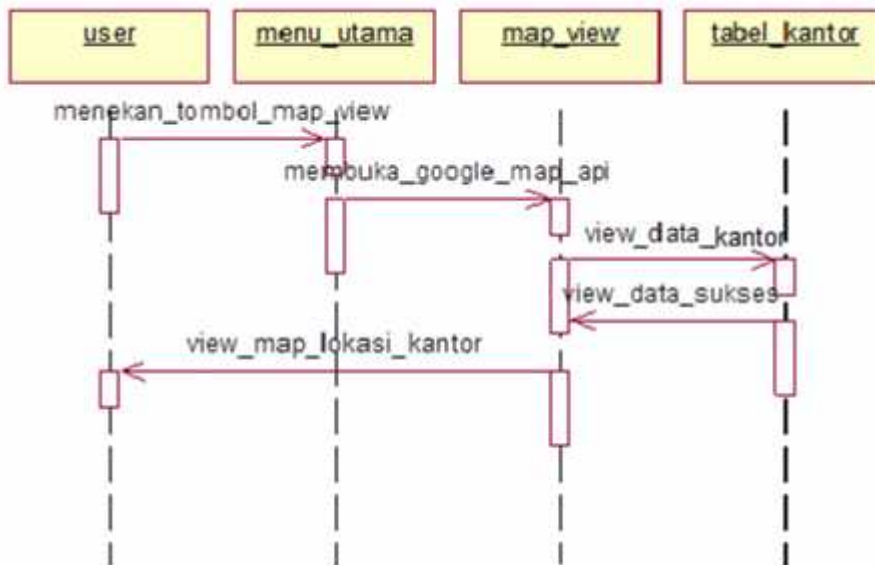


Gambar 4.10 Sequence Diagram Skenario UC02-Login

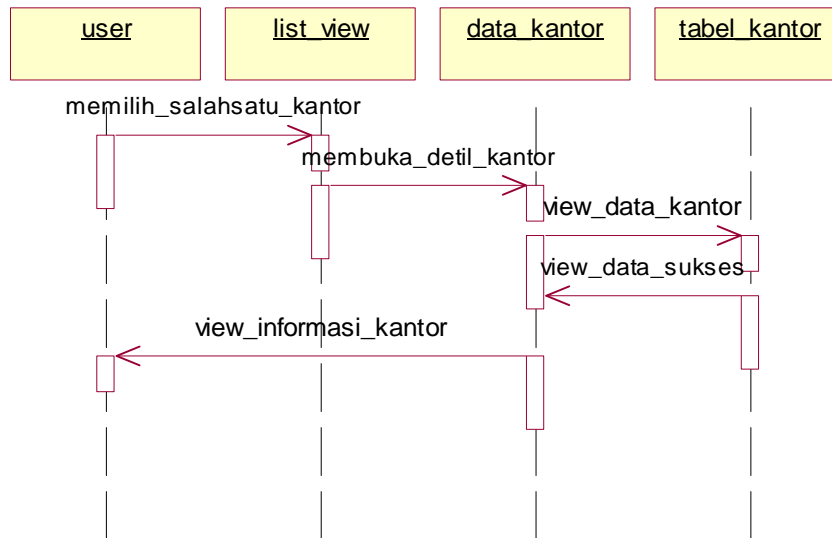
2. *Sequence Diagram* aplikasi user



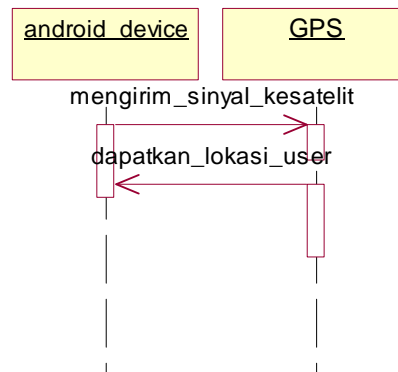
Gambar 4.11 *Sequence Diagram* Skenario UC01- Lihat list kantor



Gambar 4.12 *Sequence Diagram* Skenario UC02- Lihat map lokasi kantor



Gambar 4.13 *Sequence Diagram* Skenario UC03- Lihat informasi detail kantor



Gambar 4.14 *Sequence Diagram* Skenario UC04- *Global Positioning System*

#### 4.8 Perancangan *Database*

*Database* merupakan bagian penting dalam . Seluruh fitur yang ada menggunakan *database* untuk melakukan fungsinya. Perancangan *database* pada meliputi perancangan *database server* dan *database* lokal. *Database server* adalah *database* utama yang ada di *server*. *Database* lokal merupakan *database* pendukung yang di simpan di *device* android.

Penggunaan dua *database* ini bertujuan untuk efisiensi pemanfaatan *bandwidth* jaringan. Dengan adanya *database* lokal maka tidak harus selalu *online* dengan *database server* untuk menggunakan fitur yang ada. *Database* lokal merupakan *database client* yang akan selalu melakukan sinkronisasi data ketika aplikasi pertama kali di buka.

Nama Tabel : kantor  
 Organisasi File : Index  
 Primary Key : -

Tabel 4.12 Keterangan atribut tabel kantor

No.	Nama field	Tipe data	Null	Keterangan
1	nama_kantor	String	Not null	Nama kantor
2	alamat_kantor	String	Not null	Alamat kantor
3	Foto	String	Not null	Foto kantor
4	Lat	double	Not null	Koordinat Latitude
5	Lon	double	Not null	Koordinat Longitude

Nama Tabel : admin  
 Organisasi File : Index

Tabel 4.13 Keterangan atribut tabel admin

No.	Nama field	Tipe data	Null	Keterangan
1	user	Varchar(20)	Not null	Username admin
2	password	Varchar(50)	Not null	Password admin

#### 4.9 Perancangan Antar Muka

Aplikasi dibangun dengan dua jenis pemrograman yaitu pemrograman android untuk *user* pencari kantor dan pemrograman website untuk pengguna dari sisi administrator dan pengelola kantor. Oleh karena itu, perancangan antarmuka dibuat berdasarkan ketiga unit fungsional tersebut.

#### 4.9.1 Antarmuka *User* Pencari Kantor

Berikut dibawah ini merupakan penjelasan perancangan antarmuka untuk *user* pencari kantor.

1. Perancangan antarmuka menu utama

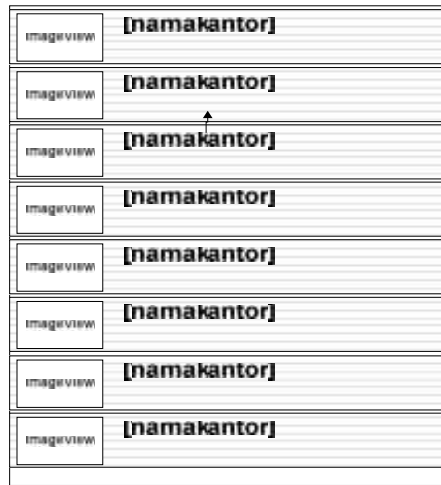
Perancangan antarmuka menu utama merupakan tampilan pilihan menu yang akan terbuka ketika proses *opening* sistem selesai.



Gambar 4.15 perancangan antar muka menu utama

2. Perancangan antarmuka daftar kantor

Perancangan antarmuka daftar kantor merupakan tampilan yang akan muncul setelah *user* memilih menu "tujuan" yang ada pada menu utama. Antarmuka daftar kantor memberikan tampilan berupa *listview* yang berisi nama, dan foto yang ada dalam database. Berikut adalah rancangan antarmuka daftar kantor.



Gambar 4.16 perancangan antar muka daftar kantor

3. Perancangan antarmuka peta kantor

Perancangan antarmuka peta kantor menampilkan peta dari Google Map yang berisi *icon-icon* tanda lokasi rumah kantor berada sesuai dengan titik koordinatnya masing-masing. Jika *icon* ini diklik, maka akan menampilkan nama dan alamat kantor yang dipilih tersebut. Dan jika *layout* nama kantor tersebut diklik lagi sebanyak dua kali, maka sistem akan masuk ke halaman informasi detail kantor. Peta kantor ini diberikan tombol yang dapat mengatur *zoom in* dan *zoom out* terhadap peta tersebut. Berikut adalah rancangan tampilan antarmuka peta kantor.





Gambar 4.17 Perancangan antar muka peta kantor

#### 4. Perancangan antarmuka rute menuju kantor

Rute menuju kantor merupakan salah satu fitur utama dalam aplikasi yang menggunakan konsep *Location Based Service* ini. Rute menuju kantor berguna untuk memberikan petunjuk arah bagi *user* untuk menemukan kantor dari tempat ia berada. Antarmuka rute menuju kantor berupa tampilan peta yang menampilkan rute terdekat menuju lokasi kantor dan akan memberitahu user apabila melewati daerah / titik yang rawan macet. Dalam google maps telah tersedia fitur *driving direction* sehingga aplikasi GAO di sinkronkan dengan *driving direction* google maps. Berikut rancangan antarmuka rute menuju kantor.




Gambar 4.18 Perancangan antarmuka rute menuju kantor

#### 4.9.2 Antarmuka *User Admin* Sistem

Admin sistem merupakan *user* yang memiliki hak akses tertinggi dalam melakukan manajemen data dalam aplikasi ini. Antarmuka sistem dirancang untuk diimplementasikan dalam pemrograman web. Berikut adalah tampilan perancangan antarmuka untuk *user* admin sistem pada tiap-tiap menu.

1. Perancangan antarmuka login

Perancangan antarmuka login merupakan halaman yang pertama kali akan muncul ketika admin mengakses sistem. Hak akses sistem perlu dibatasi untuk keamanan data. Berikut adalah rancangan antarmuka login untuk admin.

Header.jpg	
Home	[Navbar_area]
<p><b>Form Login Admin</b></p> <div style="text-align: center;">  </div> <p>Username : <input type="text" value="Input text"/></p> <p>Password : <input type="password" value="Input password"/></p> <div style="text-align: center;"> <input type="button" value="Login"/> </div>	

Gambar 4.19 Perancangan antarmuka login *user* admin

2. Perancangan antarmuka menu utama

Setelah admin menyelesaikan proses login, *user* akan masuk ke tampilan antarmuka menu utama. Menu utama menampilkan pilihan menu yang dapat dilakukan *user* dalam sistem tersebut meliputi menu data kantor, menu data profil dan logout. Berikut adalah rancangan antarmuka menu utama untuk *user* admin.

Header.jpg		
Home	Titik kantor	Logout
<p><b>Welcome admin</b></p>		

Gambar 4.20 Perancangan antarmuka menu utama *user* admin

### 3. Perancangan antarmuka kelola data kantor

Antarmuka kelola data kantor merupakan tampilan pilihan menu yang dapat dilakukan *user* untuk mengelola data kantor. Pada menu ini data kantor ditampilkan dalam bentuk tabel. Berikut rancangan antarmuka kelola data kantor.

Data Kantor					
Beranda		Titik Lokasi		Keluar	
No.	Nama Titik	Koordinat X	Koordinat Y	Informasi	Option
1.	Dinas Pendidikan Kota Pekanbaru	xxxxxxxxxx	xxxxxxxxxx	xxxxxxxxxx	
2.	Dinas Kesehatan Kota Pekanbaru	xxxxxxxxxx	xxxxxxxxxx	xxxxxxxxxx	
3.	Dinas Tenaga Kerja Kota Pekanbaru	xxxxxxxxxx	xxxxxxxxxx	xxxxxxxxxx	

Gambar 4.21 Perancangan antarmuka kelola data kantor *user* admin

### 4. Perancangan antarmuka tambah kantor

*User* admin adalah satu-satunya *user* yang memiliki hak akses untuk menambahkan data kantor. Untuk itu perlu disediakan antarmuka bagi *user* admin untuk memasukkan data kantor ke *database*. Berikut adalah rancangan antarmuka tambah kantor *user* admin. Jika data belum terisi dengan lengkap atau tidak sesuai dengan aturan pengisian, maka data tidak akan masuk ke *database* dan akan tampil pesan *error* data masukan.

The image shows a web form interface with a blue title bar labeled "Form Title". At the top, there are three buttons: "Beranda", "Titik Koordinat" (which has a mouse cursor over it), and "Keluar". Below these are three input fields labeled "Nama Titik", "Koordinat X", and "Koordinat Y". Underneath the input fields is a large text area labeled "Informasi" with vertical and horizontal scrollbars. At the bottom center of the form is a button labeled "Kirim".

Gambar 4.22 Perancangan antarmuka tambah data kantor *user* admin

##### 5. Perancangan antarmuka update data kantor

Apabila diperlukan perubahan terhadap data kantor, admin dapat melakukannya melalui menu edit. Oleh karena itu perlu disediakan antarmuka bagi *user* untuk melakukan perubahan data dalam *database*. Berikut adalah tampilan antarmuka edit data kantor untuk *user* admin.

Form Title

Beranda Titik Koordinat Keluar

Nama Titik

Koordinat X

Koordinat Y

Informasi

Gambar 4.23 Perancangan antarmuka update data kantor *user* admin.