BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem dapat di definisikan sebagai penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan. Analisis bertujuan untuk mendapatkan pemahaman secara keseluruhan tentang sistem yang akan dibuat berdasarkan masukan dari pihak-pihak dan juga pengalaman analis yang berkepentingan dengan sistem tersebut.

3.2 Analisis Masalah

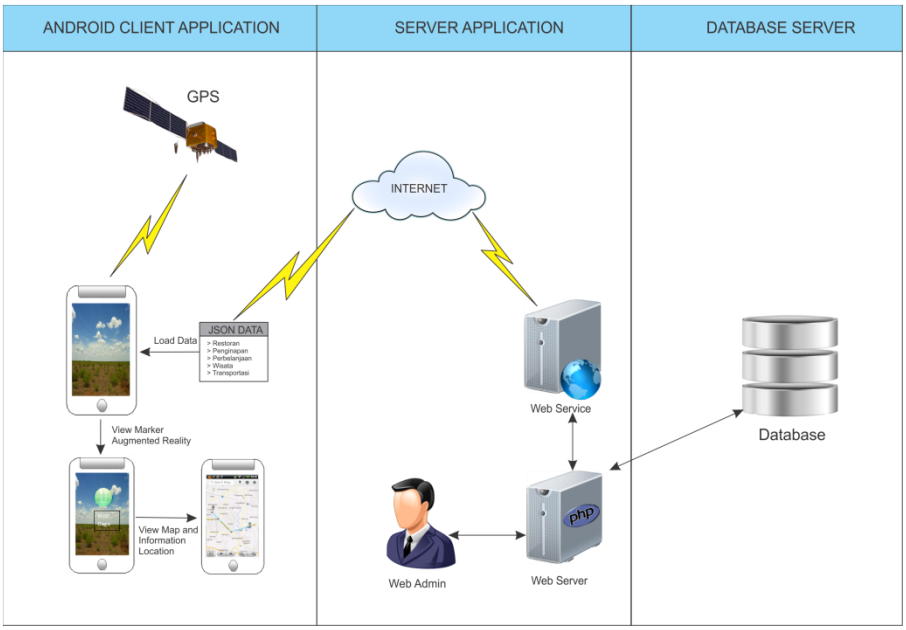
Bagi setiap turis yang baru mengenal wilayah kota Bandung dirasa sangat menyulitkan dengan mendapatkan informasi dan data mengenai lokasi-lokasi yang ingin dikunjungi. Pemandu turis(*tour guide*) dapat menjadi solusi bagi para turis yang baru mengenal wilayah Bandung agar dapat memberikan informasi-informasi umum yang dibutuhkan turis tersebut, tetapi peranan pemandu turis belum tentu dapat memenuhi kebutuhan maupun keinginan turis tersebut.

Untuk menggantikan peran pemandu turis tersebut maka dibangun sistem yang dapat memberikan informasi secara efektif, cepat dan akurat. Agar solusi tersebut dapat dicapai maka sistem dibangun pada *smartphone* yang kemudian diterapkan

teknologi *augmented reality* sehingga dapat memandu pengguna terhadap keadaan(lingkungan) sekitar dengan informasi yang ingin dituju. Karena keunggulan yang terdapat pada teknologi *augmented reality* tersebut maka peranan pemandu turis dapat digantikan. Selain itu sistem yang dibangun mencakup data-data yang biasanya dibutuhkan para turis yaitu restoran, wisata, penginapan, perbelanjaan dan transportasi. Data restoran, wisata, penginapan, perbelanjaan dan transportasi ditampilkan dengan visualisasi teknologi *augmented reality* sebagai penunjuk arah, sedangkan data transportasi hanya menampilkan informasi angkot di kota Bandung. Dengan demikian data-data informasi lokasi yang diterima turis dapat sesuai dengan yang diharapkan.

3.2.1 Deskripsi Sistem

Sistem yang dibangun menggunakan metode *client server* dalam arsitekturnya, dimana arsitektur pada sistem ini terdiri dari 3 komponen utama yaitu *android client application, server application* dan *database server*.



Gambar 3. 1 Arsitektur *markerless augmented reality* dengan GPS *based tracking*

Android client application merupakan aplikasi yang digunakan oleh *user* sebagai pemandu wisata untuk mengetahui lokasi yang diimplementasikan teknologi *augmented reality*. Pada *android client application* akan muncul visualisasi berupa *marker augmented reality* lokasi-lokasi secara *real-time* berdasarkan kategori yang diakses. Kategori yang menampilkan data melalui *augmented reality* yaitu restoran, penginapan, perbelanjaan dan wisata. Ke-4 data yang tergabung dalam POI ditampilkan melalui *augmented reality* tersebut juga dapat menampilkan informasi detail mengenai lokasi tersebut, selain itu juga dapat memberikan navigasi melalui map antara lokasi pengguna berada dengan marker POI yang telah dipilih.

Server application merupakan 3 unit bagian dari *web admin*, *web server* dan *web service*, dimana tugas *web admin* sebagai media untuk mengolah keseluruhan

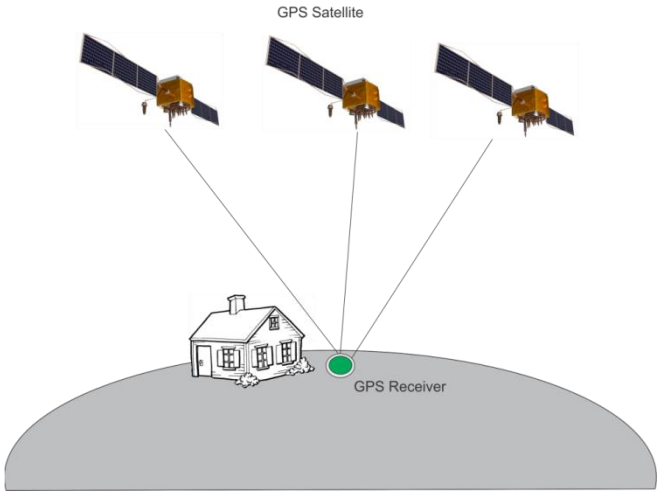
data yang dibutuhkan oleh android *client. Web server merupakan* layanan data yang berfungsi menerima permintaan HTTP dari web admin dan mengirimkannya kembali ke dalam bentuk halaman-halaman web, sedangkan *web service* berfungsi sebagai media jembatan penerjemah data agar dapat dibaca oleh *android client*. *Database server* merupakan media penampung data dari keseluruhan data yang diolah di *web admin*.

GPS berfungsi sebagai penentu posisi pengguna berada pada saat itu, agar informasi lokasi yang diterima relevan dengan lokasi pengguna berada. Untuk dapat diimplementasikan secara maksimal GPS dikombinasikan dengan kompas dan akselerometer yang telah terintegrasi dengan perangkat *handset*. Kompas berfungsi untuk mendapatkan orientasi horizontal *handset*. Sedangkan akselerometer berfungsi melakukan pembacaan terhadap orientasi handphone yang dipegang pengguna. Jika pengguna memegang dengan posisi layar menghadap depan dan rata dengan tanah maka visualisasi *augmented reality* akan ditampilkan sejajar terhadap horizon. Namun jika pengguna memegang *device* pada posisi 45 derajat ke bawah maka visualisasi *augmented reality* muncul pada layar bagian atas.

3.2.2 Analisis Permasalahan Penentuan Lokasi terdekat.

Penentuan posisi lokasi pengguna dapat ditentukan dengan adanya teknologi GPS. Selama pengguna tersebut berada dalam *line of sight* yaitu lingkaran tringulasi dari 3 satelit untuk mendapatkan posisi pengguna. Selama pengguna berada di dalam

line of Sight dimanapun pengguna tersebut berada maka lokasi tersebut akan terus *update*.



Gambar 3. 2 *Line of sight*

GPS bertugas membaca penentu titik lokasi koordinat-koordinat pengguna dan lokasi. Agar jarak lokasi terdekat dengan pengguna dapat divisualisasikan diperlukan persamaan agar dapat melakukan perhitungan untuk mendapatkan jarak tujuan terdekat dengan lokasi saat ini berada. Persamaan haversine merupakan sebuah solusi agar lokasi-lokasi terdekat tampak di sekitar pengguna dengan mengabaikan bentuk geografis bukit maupun lembah.

Jarak = ACos(Sin(Lat1) x Sin(Lat2) x Cos(Lat1) x Cos(Lat2) x Cos(Long2-Long1)) x R

Dimana :

Lat1 = Nilai latitude lokasi pengguna

Lat2 = Nilai latitude lokasi poi

Long1 = Nilai longitude lokasi pengguna

Long2 = Nilai longitude lokasi poi

R = Radius bumi (rata-rata radius = 6,371kilometer)

Contoh :

Misalkan untuk mencari jarak antara lokasi *user* di unikom ke pusat perbelanjaan Bandung Indah Plaza Jika diketahui :

1. Latitude 1 (Unikom) -6.886776 dan Longitude 1 (Unikom) 107.615246.
2. Latitude 2 (Bandung Indah Plaza) -6.908358 dan Longitude 2 (Bandung Indah Plaza) 107.610745.

Jarak = ACos(Sin(Lat1) x Sin(Lat2) x Cos(Lat1) x Cos(Lat2) x Cos(Long2-Long1)) x R

= ACos(Sin(-6.886776 x pi / 180) x Sin(-6.908358 x pi / 180) + Cos(-6.886776 x pi / 180) x Cos(-6.908358 x pi / 180) x Cos((107.610745 x pi / 180) – (107.615246 x pi / 180)) x 6.371

= Acos(0.897524175) x 6.371

= 2.909 km

3.2.3 Analisis Permasalahan *Markerless Augmented Reality*

Analisis *markerless augmented reality* merupakan analisis yang berfungsi menggambarkan secara rinci pembentukan sebuah *marker* yang divisualisasikan menjadi *augmented reality*. Analisis ini meliputi :

1. GPS

GPS merupakan penentu titik lokasi pengguna berada dengan lokasi yang akan dituju sehingga data informasi yang diterima aplikasi relevan terhadap lokasi pengguna.

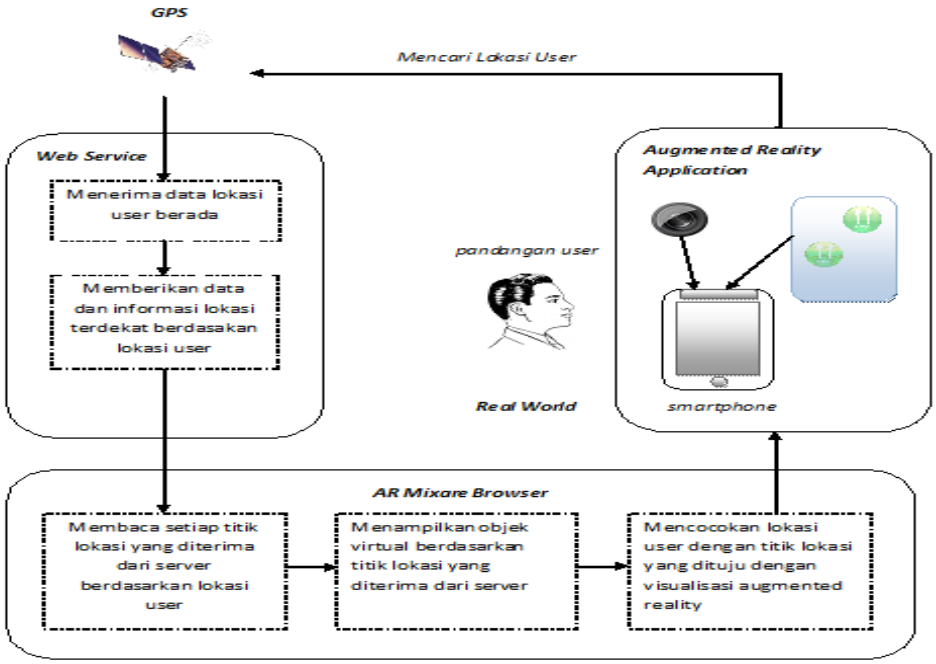
2. AR Mixare Browser

AR Mixare *Browser* merupakan *engine browser augmented reality* sebagai pembentuk keseluruhan sistem. Mixare berperan sebagai pembaca lokasi yang sebelumnya telah ditentukan oleh GPS sebagai penentu lokasi *user*. Setelah GPS menentukan lokasi *user* maka sistem akan menampilkan titik koordinat lokasi terdekat *user* dengan menggunakan visualiasi objek dunia maya yang disebut *augmented reality*. Visualiasi tersebut berupa *marker* yang berisi informasi mengenai nama, alamat dan nomor telepon lokasi tersebut.

3. *Web Service*

Web Service merupakan sarana transportasi antara database *server* dengan aplikasi. *Web service* menerjemahkan data sesuai dengan format antara database *server* dan aplikasi.

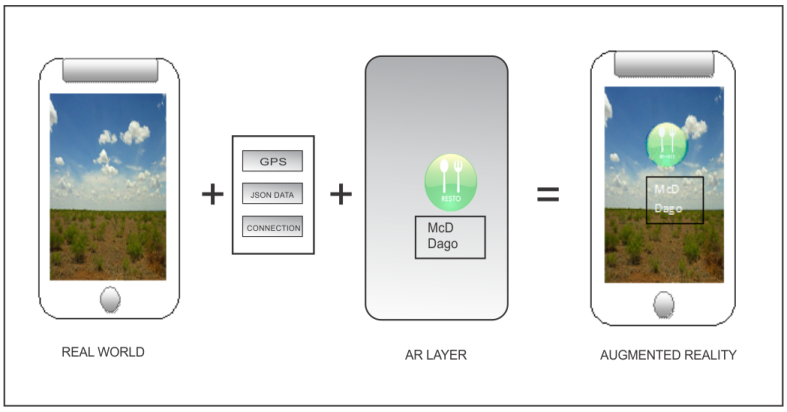
3 pembentuk sistem tersebut memiliki peranan yang vital dalam proses terbentuknya sebuah *marker augmented reality*. Dapat dilihat dengan jelas peranan dari 3 pembentuk sistem tersebut pada gambar 3.3.



Gambar 3. 3 Arsitektur sistem pembentukan *marker augmented reality*

Pada gambar 3.3 diasumsikan user telah berada pada wilayah yang telah ditentukan. Aplikasi akan menentukan lokasi pengguna dengan memindai melalui

GPS, setelah lokasi pengguna ditentukan maka aplikasi akan menerima data dari web *server* dengan syarat *smartphone* yang digunakan memiliki jaringan internet. Data yang telah diterjemahkan oleh *web service* ke aplikasi kemudian akan ditentukan setiap titik koordinat lokasi yang terdekat dengan *user* berada. Setelah data informasi yang terdekat dengan *user* telah diketahui, maka tugas mixare *browser* selanjutnya yaitu memvisualisasikan titik-titik koordinat lokasi yang berada di sekitar *user*. Bentuk visualisasi tersebut berupa *marker* yang menentukan keberadaan lokasi tersebut berdasarkan koordinat(latitude dan longitude) yang diterima. Setelah data divisualisasikan maka sistem akan menyesuaikan orientasi antara gambar dunia nyata(*real world*) yang ditangkap kamera dengan dunia maya(AR *layer*) yang menghasilkan objek *virtual* berupa *marker augmented reality*. Objek *virtual* yang dilacak dikombinasikan dengan sensor yang terdapat dalam perangkat(*smartphone*) yaitu GPS, *accelerometer*, *digital compass*, dan kamera.



Gambar 3. 4 Penyesuaian dunia nyata(*real world*) dengan dunia maya(AR *layer*).

Pada gambar 3.4 tergambar bahwa dunia nyata(*real world*) menjadi keadaan awal sebelum objek *augmented reality* tampil, untuk memenuhi syarat agar *augmented reality* dapat menyatu dengan dunia nyata melalui kamera maka dibutuhkan GPS, JSON data dan jaringan internet(*connection*). GPS sebagai penentu lokasi pengguna, JSON data sebagai kerangka pemberi informasi agar *augmented reality* dapat tampil dan *connection* sebagai jaringan internet untuk mengakses data ke server agar dapat menerima JSON data yang berada di server. Setelah seluruh syarat terpenuhi maka objek dibentuk ke dalam sebuah *layer* kemudian di satukan ke dalam dunia nyata melalui kamera yang nantinya menghasilkan objek yang disebut dengan *augmented reality*.

3.2.4 Analisis *Web Service*

Web service adalah media penerjemah data yang bertujuan untuk menghubungkan berbagai macam *platform* aplikasi dengan berbagai macam *platform* yang berbeda-beda (contohnya platform JAVA, C#, PASCAL) agar seluruh *platform* tersebut dapat saling berkomunikasi. *Web service* berperan penting dalam implementasi *augmented reality* GPS dengan cara menerjemahkan data yang ada pada *web admin* agar dapat dibaca oleh android *client*. Bahasa pemrograman untuk membangun *web admin* berbeda dengan bahasa pemrograman untuk membangun android *client* sehingga *web service* menjadi solusi sebagai media komunikasi 2 *platform* aplikasi yang berbeda tersebut.



Gambar 3. 5 *Alur request* dan *response* data

Analisis dalam *web service* merupakan gambaran rinci dari komunikasi data antara *client* dan *server*. Sehingga dapat dijelaskan dengan rinci sebagai berikut :

1. *Android client application*

Android client application adalah aplikasi yang meminta data dari *database server* dengan *web service* sebagai media komunikasinya. Data yang diminta oleh *client* sudah berbentuk format pertukaran data *web service* yaitu JSON(*Java Script Object Notation*) sehingga *Android client* dapat bekerja dengan memparsing data berbentuk format JSON tersebut.

2. *Web Service*

Web service berperan sebagai penerjemah data antara android *client* dengan database *server*. Format pertukaran data *web service* yang dipakai yaitu JSON. *Web service* bekerja dengan cara mengambil data dari *database server* untuk dikonversikan ke dalam bentuk JSON yang kemudian diterima android *client*. Bentuk format JSON dapat dilihat seperti pada contoh berikut :


```
        "result":[
    {
      "id_poi":"1",
      "nama":"90 Gourmet",
      "telpon":"(022)4203295",
      "latitude":"-6.9083786",
      "longitude":"107.62365415"
    }
```


3.3 Analisis dan Kebutuhan Non-Fungsional

Analisis kebutuhan non fungsional menggambarkan kebutuhan sistem yang menitik-beratkan pada properti perilaku yang dimiliki oleh sistem, diantaranya kebutuhan perangkat keras, perangkat lunak, serta *user* sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan.

3.3.1 Kebutuhan Perangkat Lunak

Analisis perangkat lunak terdiri dari spesifikasi minimum perangkat yang dibutuhkan dan elemen google maps api.

a. Spesifikasi perangkat lunak

Berikut ini adalah spesifikasi minimum perangkat lunak yang digunakan dalam membangun aplikasi Bandung *tour guide* ini.

Tabel 3. 1 Spesifikasi perangkat lunak

Komputer	Smartphone
Sistem Operasi Windows 7	Platform Android versi. 2.1 keatas.
IDE(I*ntegrated Development Environment*) Eclipse Galileo	
JDK(*Java Development Kit*) v.1.6	
Android SDK (*Standard Developmenet Kit*) windows	
ADT (*Android Development Tools*) v.8.0.0	
Mixare AR Browser (*Augmented Reality Browser*)	

b. Elemen Google Maps API

Berikut ini elemen dan objek penting pada *Google Maps API* yang diterapkan pada aplikasi Bandung Tour Guide.

1. *Markers*

   Marker adalah fungsionalitas simbol yang menandakan suatu titik koordinat lokasi. Sebagai contoh pada penelitian *Gmarker* digunakan untuk menandai lokasi seperti ATM, SPBU, Restoran dan Rumah Sakit pada peta *Google Maps*.

2. *Directions*

   Adalah *utility* yang menangani masalah pencarian rute. Sebagai contoh pada penelitian *Gdirections* digunakan untuk memandu rute jalan dari pengguna menuju tempat lokasi seperti ATM, SPBU, Restoran dan Rumah Sakit pada peta *Google Maps*.

3. *GeoCoding and Reverse GeoCoding*

   Adalah *utility* untuk memproses konversi alamat (seperti "07 Tubagus Ismail Dalam, Bandung") ke koordinat geografis (seperti lintang dan bujur 37,423021 -122,083739) atau fungsionalitas sebaliknya, yang dapat digunakan untuk menempatkan penanda atau posisi peta.

4. *Location Manager*

   adalah API yang paling penting, dalam memberikan akses ke layanan sistem lokasi dengan pemetaan dan GPS. Dalam hal ini *Location Manager*

berfungsi untuk mendapatkan update periodik lokasi geografis perangkat dengan memetakan nilai latitude dan longitude kedalam *Google Maps*.

5. *Overlays Overview*

Adalah objek di *Google Maps* yang terkait dengan koorddinat lintang dan bujur, sehingga objek ini bergerak saat ditarik atau zoom pada peta. *Overlays* mencerminkan benda yang ada pada peta untuk menunjukan titik, garis, area, atau *markers* benda.

6. *Loading Google Maps API*

Koneksi script yang dibuat keserver Google Map API dengan menggunakan *API key* yang didapatkan pada saat mendaftar ke *Google Maps API*.

7. *Map DOM Elements*

Untuk menampilkan map pada halaman web, umumnya menggunakan *div tag* untuk menampungnya. Diharuskan membuat sebuah *div tag* dengan nama "*map_canvas*", dan mereferensikan *element* ini ke *Document Object Model* (DOM).

3.3.2 Kebutuhan Perangkat Keras

Komputer dan *smartphone* terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi. Perangkat lunak memberikan instruksi-instruksi kepada perangkat keras untuk melakukan tugas tertentu. Pada pembangunan aplikasi Bandung *tour guide* ini, perangkat keras yang digunakan untuk membangun aplikasi adalah sebagai berikut:

Tabel 3. 2 Spesifikasi perangkat keras

Komputer	Smartphone
Processor AMD Dual Core 1GHz	HTC Desire HD (OS android 2.2)
RAM 2GB	
Harddisk 250GB	
Mouse dan *Keyboard*	

3.3.3 Spesifikasi *User*

User dibutuhkan sebagai pengguna aplikasi, spesifikasi pengguna dalam aplikasi Bandung *tour guide* dibagi menjadi dua bagian, yaitu :

1. Pengguna aplikasi Bandung *tour guide* (*user client*) yaitu para pengguna *smartphone* dengan *platform* android versi 2.2 atau lebih tinggi. Dalam menggunakan sistem ini, pengguna diharuskan terhubung dengan koneksi jaringan internet, dan sensor dalam smartphone aktif seperti GPS *receiver*,

digital compass, *accelerometer*, dan kamera. Untuk bisa menggunakan aplikasi ini.

2. Admin sebagai pengelola data memiliki kapabilitas dalam mengelola database seperti menambah, mengubah, dan menghapus data melalui web admin.

3.4 Analisis dan Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan. Analisis kebutuhan fungsional ini meliputi analisis kebutuhan sistem, analisis kebutuhan data, spesifikasi sistem dan pemodelan sistem.

3.4.1 Analisis Kebutuhan Sistem

Sebelum melakukan perancangan sistem, terlebih dahulu dilakukan analisis kebutuhan sistem yaitu dengan cara melakukan wawancara terhadap responden. Hal ini dimaksudkan agar dapat mengatasi ketidaksesuaian antara aplikasi yang dirancang dengan kebutuhan pengguna. Adapun kebutuhan sistem yang diperlukan antara lain :

1. Memberikan visualiasi informasi POI melalui objek *marker augmented reality* terdekat yang berada di sekitar pengguna.
2. Menampilkan informasi POI mengenai tempat-tempat yang dibutuhkan oleh para turis dengan lengkap dan rinci.
3. Mengatur jarak radius lokasi terdekat dengan pengguna.

4. Menampilkan navigasi antara lokasi pengguna berada dengan lokasi tujuan yang diterima melalui *marker* POI *augmented reality*.
5. Memberikan informasi data transportasi beserta dengan detail dari transportasi tersebut dengan lengkap dan rinci.

3.4.2 Analisis Kebutuhan Data

Setelah melakukan proses analisis kebutuhan sistem, maka analisis selanjutnya adalah analisis kebutuhan data. Analisis ini bertujuan untuk memudahkan dalam perancangan informasi. Adapun analisis kebutuhan data yang diperlukan antara lain :

1. Data POI

   Data POI sebenarnya terbagi ke dalam 4 kategori yang meliputi data restoran, wisata, penginapan dan perbenlajaan. Dalam 4 kategori data POI tersebut memiliki informasi penunjang setiap masing-masing kategori diantaranya nama tempat, alamat, nomor telepon, deskripsi singkat dan deskripsi lengkap lokasi tersebut.

2. Data Transportasi

   Data transportasi merupakan data yang terpisah dari POI, data ini meliputi nama angkot, warna angkot dan rute angkot.

3.4.3 Spesifikasi Sistem

Adapun spesifikasi sistem yang dibutuhkan antara lain :

1. Sistem dikembangkan dengan menggunakan metode *client server*. Aplikasi Bandung *Tour Guide* dikembangkan disisi *client* atau *frontend*. Sedangkan disisi *server* atau *backend* akan dikembangkan aplikasi web *Content Managemen System* (CMS) yang merupakan sumber data dari aplikasi *frontend*.
2. Konektivitas antara aplikasi Bandung *Tour Guide* pada *platform* android ini dengan aplikasi *backend* di s*erver* menggunakan koneksi protocol HTTP dengan memanfaatkan jaringan internet yang ada dalam perangkat(*smartphone*).
3. Spesifikasi sistem disisi *frontend* yaitu :

    a. Perangkat yang digunakan harus bisa digunakan dimana saja dan kapan saja selama perangkat(*smartphone*) memiliki jaringan internet.
    b. Agar aplikasi dapat berjalan dengan maksimal, sebaiknya berada dalam cakupan *line of sight* agar data yang diterima relevan dengan lokasi pengguna berada. Cakupan *line of sight* yang baik yaitu ketika berada di lingkungan terbuka atau tidak berada di ruangan tertutup.

4. Spesifikasi sistem di *backend* yaitu:

    a. Sistem dapat memfasilitasi pengelolahan konten informasi.

b. Sistem dapat memfasilitasi Admin untuk melakukan tambah, ubah, dan hapus konten informasi fasilitas umum serta kategorinya.

3.4.4 Pemodelan Sistem

Pemodelan sistem dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahap-tahap pemodelan dalam analisis tersebut antara lain *Use Case Diagram, Class Diagram, State Diagram, Activity Diagram, Sequence Diagram, Component Diagram,* dan *Deployment Diagram.*

Pemodelan sistem berfungsi untuk menentukan fungsi-fungsi yang dapat dilakukan oleh sistem pada aplikasi serta menentukan kelas yang dibutuhkan untuk realisasi fungsi-fungsi sistem yang telah dianalisis sebelumnya dan mendeskripsikannya kedalam bentuk diagram.

a. Use Case Diagram

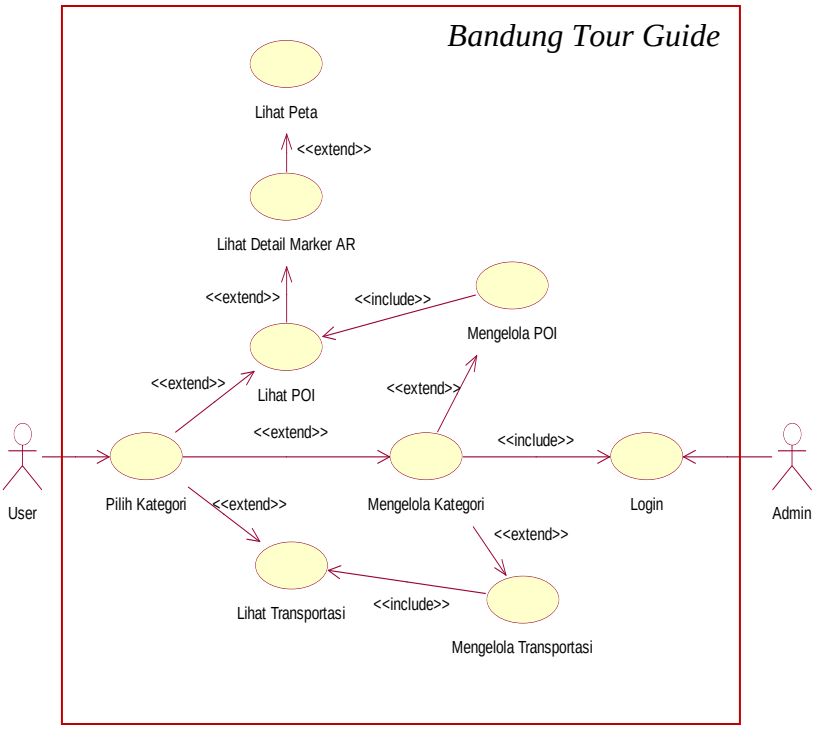
Digunakan untuk memodelkan atau menggambarkan batasan sistem dan fungsi-fungsi utamanya. Mendiskripsikan fungsi dari sebuah sistem dari perspektif pengguna, use case bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebauh sistem dipakai. Use *case* terdiri dari tiga bagian yaitu definisi *actor*, definisi *use case*, dan *scenario use case*.

1). Definisi Aktor

Definisi Aktor berfungsi untuk mendeskripsikan peran actor dalam perangkat lunak ini. Definisi tersebut dapat dilihat pada tabel 3.3.

Tabel 3. 3 Definisi aktor

No.	Aktor	Deskripsi
A-01	*User*	Merupakan aktor yang berperan dalam menggunakan aplikasi Bandung *Tour Guide* untuk mendapatkan lokasi tujuan dengan augmented reality beserta dengan informasi lokasi tersebut.
A-02	*Admin*	Merupakan aktor yang berperan dalam mengelola data informasi dalam database *server* melalui CMS berbasis web.



Gambar 3. 6 *Use case diagram*

2). Definisi *Use Case*

Definisi *Use Case* berfungsi sebagai penjelasan mengenai proses yang terdapat pada setiap *use case*. Dapat dilihat definisi tersebut pada tabel 3.4.

Tabel 3. 4 Definisi use case

No.	Nama *Use Case*	Deskripsi
UC-01	Pilih Kategori	Fungsionalitas yang dapat memilih kategori dari aplikasi pemandu turis yang terdiri dari restoran, wisata, penginapan perbelanjaan dan transportasi.
UC-02	Lihat POI	Fungsionalitas yang menampilkan kategori restoran, wisata, penginapan dan perbelanjaan.
UC-03	Lihat Transportasi	Fungsionalitas yang menampilkan kategori transportasi.
UC-04	Lihat Detail *Marker* AR	Fungsionalitas untuk menampilkan augmented reality berdasarkan POI yang dipilih.
UC-05	Lihat Peta	Fungsionalitas yang berfungsi menampilkan peta berada saat ini dengan lokasi tujuan yang akan dituju.
UC-06	Mengelola POI	Fungsionalitas ini berfungsi sebagai pengolaan data POI yang dilakukan oleh admin.
UC-07	Mengelola Transportasi	Fungsionalitas ini berfungsi sebagai pengolalaan data transportasi yang dilakukan oleh admin.
UC-08	Mengelola Kategori	Fungsionalitas ini berfungsi sebagai pengolalaan data kategori yang dilakukan oleh admin.
UC-09	*Login*	Fungsionalitas untuk proses autentifikasi hak akses kepada admin

3). Skenario *Use Case*

Skenario setiap bagian pada *use case* menunjukkan penjelasan setiap bagian-bagian di dalam *use case* tersebut. Di dalam skenario *use case* akan tergambar jelas,

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*.

Tabel 3. 5 Skenario use case pilih kategori

Identifikasi	
Nama *Use Case*	Pilih Kategori
Aktor	*User*
Tujuan	Memilih pilihan kategori
Keadaan Awal	Sistem menampilkan menu kategori
Sekenario Utama	
Aksi Aktor	**Reaksi sistem**
	1.Menampilkan pilihan beberapa pilihan kategori yaitu restoran, wisata, penginapan, perbelanjaan dan transportasi.
2. Memilih salah satu kategori yang ditampilkan aplikasi.	
	3.Jika memilih salah satu menu kategori maka sistem akan informasi berdasarkan kategori yang dipilih.
Skenario Alternatif – Verifikasi Gagal	
Aksi Aktor	**Reaksi sistem**
	1.” *Sorry!,* Bandung *Tour Guide has stopped unexpectedly. Please try again*. “
2. Menekan tombol “*Force Close*”	
	3. Sistem akan *terminate*.
Kondisi Akhir	Menampilkan informasi sesuai dengan kategori yang dipilih.

Tabel 3. 6 Skenario use case lihat POI

Identifikasi	
Nama *Use Case*	Lihat POI
Aktor	*User*
Tujuan	Menampilkan lokasi POI dengan *augmented reality*
Keadaan Awal	Sistem menampilkan menu pilih kategori
Sekenario Utama	
Aksi Aktor	**Reaksi sistem**
1. Memilih salah satu menu POI yaitu restoran, wisata, penginapan dan perbelanjaan.	
	2. menampilkan informasi lokasi yang telah ditentukan dengan *augmented reality*.
3.Mengarahkan handphone yang digunakan ke sekeliling area *user* berdiri dengan mengubah orientasi untuk mendapatkan letak POI yang akan muncul di layar handphone.	
Skenario Alternatif – Verifikasi Gagal	
Aksi Aktor	**Reaksi sistem**
	1.” Maaf, Koneksi Tidak Terdeteksi! Periksa Kembali Koneksi Jika Terhubung Dengan Internet. “
2.Memilih tombol pengaturan untuk mengaktifkan koneksi internet atau memilih tombol kembali untuk keluar aplikasi.	

	3. Sistem akan *terminate*.
Kondisi Akhir	Menampilkan lokasi POI dengan *augmented reality*.

Tabel 3. 7 Skenario use case lihat transportasi

Identifikasi	
Nama *Use Case*	Lihat Transportasi
Aktor	*User*
Tujuan	Menampilkan list informasi transportasi
Keadaan Awal	Sistem menampilkan menu pilih kategori
Sekenario Utama	
Aksi Aktor	**Reaksi sistem**
1. Memilih menu kategori transportasi.	
	2. Menampilkan list informasi jurusan trasportasi.
Skenario Alternatif – Verifikasi Gagal	
Aksi Aktor	**Reaksi sistem**
	1.” *Sorry!,* Bandung *Tour Guide has stopped unexpectedly. Please try again. “*
2. Menekan tombol “*Force Close*”	
	3. Sistem akan *terminate*.
Kondisi Akhir	Menampilkan *list* jurusan transportasi.

Tabel 3. 8 Skenario use case lihat *detail marker* AR

Identifikasi	
Nama *Use Case*	Lihat *Detail Marker AR*
Aktor	*User*
Tujuan	Menampilkan detail informasi lokasi
Keadaan Awal	Sistem menampilkan lokasi POI dengan *augmented reality*
Sekenario Utama	
Aksi Aktor	**Reaksi sistem**
1. Memilih salah satu marker augmented reality.	
	2. Menampilkan detail informasi lokasi yang dipilih.
Kondisi Akhir	Menampilkan detail informasi lokasi POI.

Tabel 3. 9 Skenario use case lihat peta

Identifikasi	
Nama *Use Case*	Lihat Peta
Aktor	*User*
Tujuan	Menampilkan rute navigasi ke arah lokasi yang akan dituju.
Keadaan Awal	Sistem menampilkan detail informasi lokasi
Sekenario Utama	
Aksi Aktor	**Reaksi sistem**
1. Memilih *button direction* pada *layout* informasi detail.	
	2. Menampilkan google map dengan memberikan navigasi lokasi user berada
Kondisi Akhir	Menampilkan rute navigasi lokasi yang akan dituju.

Tabel 3. 10 Skenario use case mengelola POI

Identifikasi	
Nama *Use Case*	Mengelola POI
Aktor	Admin
Tujuan	Menampilkan seluruh data POI beserta pengelohan data
Keadaan Awal	Menampilkan halaman utama Bandung *Tour Guide*
Sekenario Utama	
Aksi Aktor	**Reaksi sistem**
1. Memilih POI pada menu navigator atau memilih olah data poi pada halaman utama.	
	2. Menampilkan halaman olah data poi.
Kondisi Akhir	Menampilkan data POI.

Tabel 3. 11 Skenario use case mengelola transportasi

Identifikasi	
Nama *Use Case*	Transportasi
Aktor	Admin
Tujuan	Menampilkan seluruh data transportasi beserta pengelohan data
Keadaan Awal	Menampilkan halaman utama Bandung *Tour Guide*
Sekenario Utama	
Aksi Aktor	**Reaksi sistem**
1. Memilih transportasi pada menu navigator.	
	2. Menampilkan halaman olah data transportasi.
Kondisi Akhir	Menampilkan data transportasi.

Tabel 3. 12 Skenario use case mengelola kategori

Identifikasi	
Nama *Use Case*	Mengelola Kategori
Aktor	Admin
Tujuan	Menampilkan seluruh data kategori beserta pengelohan data

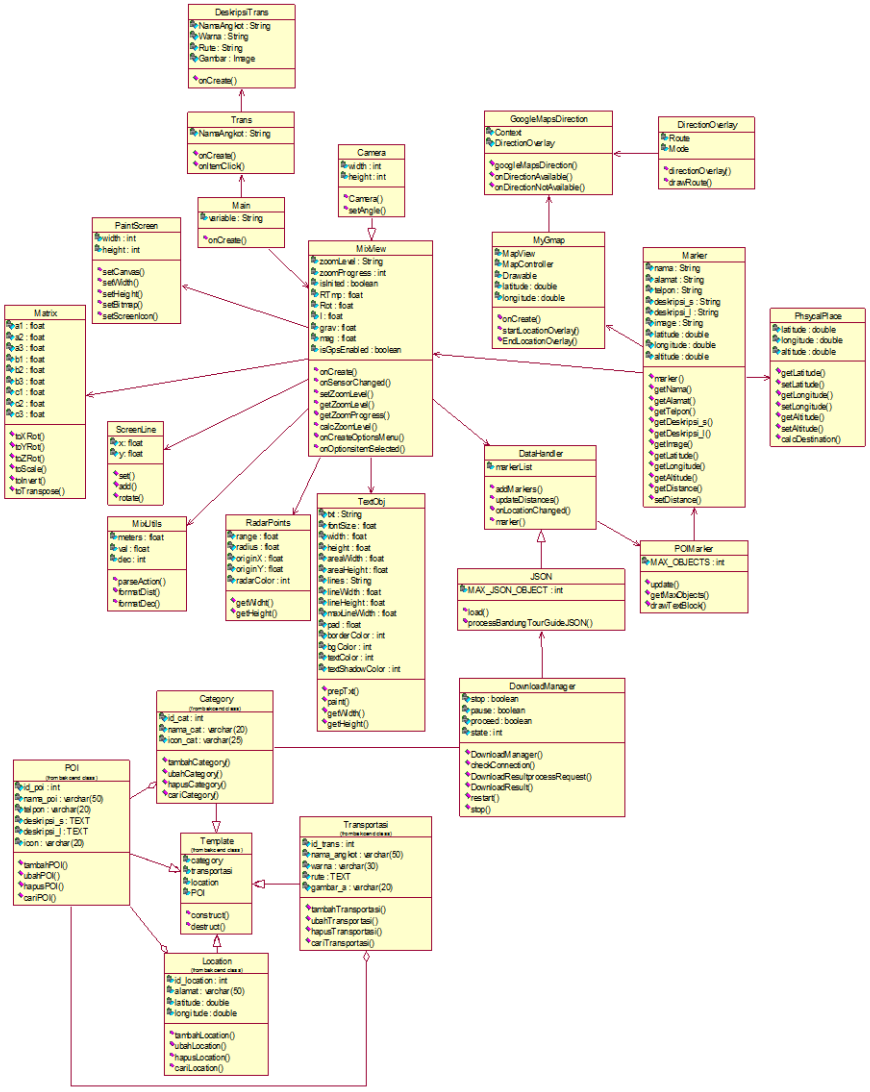
Keadaan Awal	Menampilkan halaman utama Bandung *Tour Guide*
Sekenario Utama	
Aksi Aktor	**Reaksi sistem**
1. Memilih kategori pada menu navigator atau memilih olah data kategori pada halaman utama..	
	2. Menampilkan halaman olah data kategori.
Kondisi Akhir	Menampilkan data kategori.

Tabel 3. 13 Skenario use case login

Identifikasi	
Nama *Use Case*	*Login*
Aktor	Admin
Tujuan	Menghapus data kategori
Keadaan Awal	Menampilkan halaman *login administrator*
Sekenario Utama	
Aksi Aktor	**Reaksi sistem**
1. Mengisi field *username* dan *password* pada halaman *login administrator.*	
	2. Melakukan autentifikasi kesamaan username dan password.
	3. Jika sesuai maka sistem akan menampilkan halaman menu utama CMS Bandung *Tour Guide*.
Skenario Alternatif – Verifikasi Gagal	
Aksi Aktor	**Reaksi sistem**
	1. Menampilkan pesan *"username atau password yang anda masukkan salah"*.
2. Melakukan pengisian field username dan password.	
Kondisi Akhir	Menampilkan halaman menu utama CMS Bandung *Tour Guide*.

b. Class Diagram

Class Diagram menggambarkan keadaan suatu sistem (*atribut*), dan memberikan pelayanan untuk menyelesaikan keadaan tersebut (metoda). Pada gambar 3.6 digambarkan *class* diagram dari sistem Aplikasi Bandung *Tour Guide*.



Gambar 3. 7 Class diagram Bandung tour guide

Tabel 3. 14 Deskripsi class diagram Bandung tour guide

Class	Jenis Kelas	Deskripsi
Main	*interface*	*Class* utama yang menampilkan kategori.
Trans	*Control*	*Class* yang berisikan method yang membentuk informasi transportasi.
DeskripsiTrans	*Interface*	*Class* yang berisikan method-method untuk menampilkan detail transportasi.
MixView	*Control*	Class utama dari aplikasi yang mengontrol *class-class* dengan fungsionalitas yang berbeda.
Marker	*Control*	*Class* yang berisikan method-method pembentuk gambar dan data menjadi *augmented reality*.
POIMarker	*Control*	*Class* yang berisikan method-method fungsionalitas pendukung terbentuknya sebuah *marker augmented reality*.
PaintScreen	*Interface*	Class yang berisikan fungsionalitas pembentuk sebuah *image* menjadi *augmented reality*.
PhsycalPlace	*Control*	*Class* yang berisikan fungsi penentuan lokasi dengan kalklulasi menggunakan metode *great circle distance*.
Camera	*Interface*	*Class* yang berisikan mehod-method fungsionalitas untuk mengaktifkan kamera serta

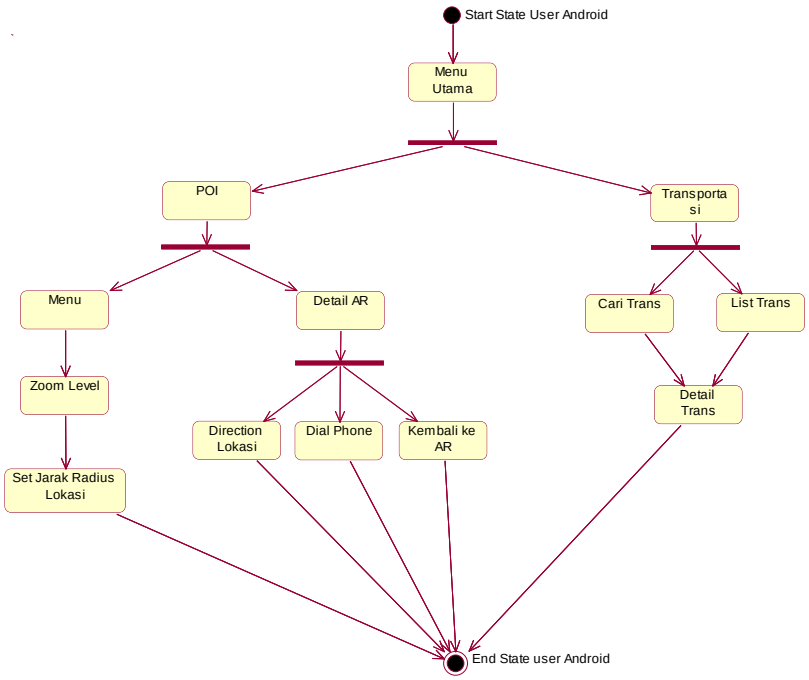
		menampilkan .*augmented reality* ke dalam layar *handphone*.
Matrix	*Control*	*Class* yang mengkalkulasikan matrix 3x3 sebagai kerangka pembentuk objek *augmented reality*.
RadarPoints	*Interface*	*Class* yang berisikan fungsionalitas pembentuk radar ketika berada dalam mode *augmented reality*.
TextObj	*Interface*	*Class* yang berisikan fungsionalitas pembentuk informasi berupa text ketika berada dalam mode *augmented reality*.
ScreeLine	*Interface*	*Class* yang berisikan fungsionalitas pembentuk objek garis ketika berada dalam mode *augmented reality*.
MixUtils	*Control*	*Class* yang mengkalkulasikan jarak lokasi dengan pengguna.
DataHandler	*Control*	*Class* yang berisikan fungsi yang menyediakan data yang telah diterjemahkan.
JSON	*Entity*	*Class* berisikan data-data yang diterjemahkan dari *web server* menuju *client*.
DownloadManager	*Control*	*Class* yang berisikan fungsionalitas koneksi antara *client* dengan *server*.
MyGmap	*Interface*	*Class* yang berisikan method-method untuk menampilkan

		navigasi lokasi melalui map.
GoogleMapsDirection	*Control*	*Class* yang berisikan fungsionalitas pembentuk navigasi antara lokasi *user* dengan lokasi yang ditju.
DirectionOverlay	*Interface*	*Class* yang berisikan fungsionalitas untuk menampilkan jalur navigasi antara lokasi *user* dengan lokasi yang dituju.
Category	*Entity*	*Class* yang berisikan data kategori.
POI	*Entity*	*Class* yang berisikan data POI.
Location	*Entity*	*Class* yang berisikan data lokasi.
Transportasi	*Entity*	*Class* yang berisikan data transportasi.
Template	*Interface*	*Class* yang berisikan method-method yang berfungsi mengontrol tampilan dari web admin

c. State Diagram

State Diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem yang akibat dari stimuli yang diterima. Pada umumnya *State Diagram* menggambarkan suatu *class* tertentu.

1. *State Diagram Android Client Application*(*Frontend*)



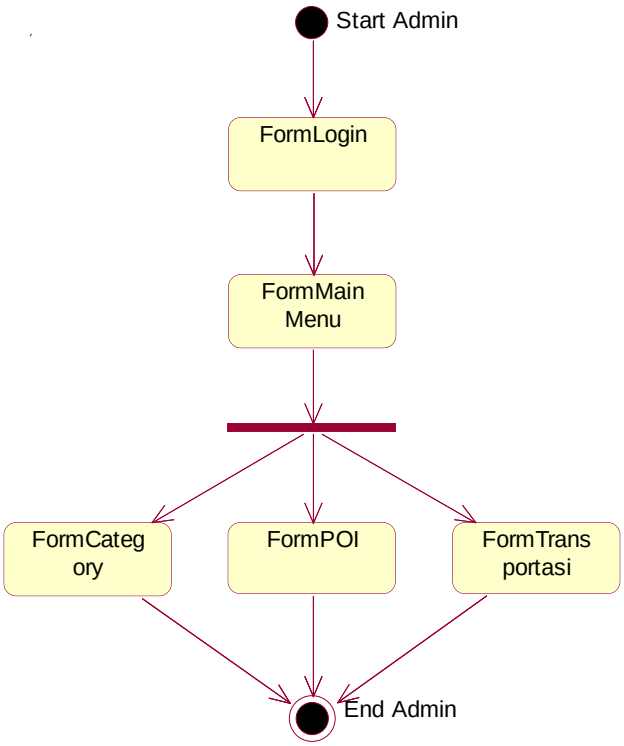
Gambar 3. 8 State diagram android *client application*(frontend)

Pada gambar 3.7 memperlihatkan perubahan keadaan atau *state* yang terjadi untuk menu utama atau keadaan awal dimana aplikasi dijalankan dan pada layar akan tampak menu utama. Pengguna dapat memilih satu dari beberapa kategori pada menu utama. Pada state diagram terlihat hanya terdapat 2 pilihan setelah memilih di menu utama. 2 pilihan tersebut mencakup dari 5 kategori yang ditampilkan dibagi menjadi 2 bagian, dimana POI merupakan penggabungan 4 data kategori yang mencakup data restoran, wisata, penginapan dan resto yang nantinya menampilkan keempat data tersebut dengan tampilan *augmented reality*. Sedangkan Transportasi merupakan data yang terpisah dari POI karena pada transportasi tidak menampilkan *augmented*

reality, melainkan infomasi berupa jurusan transportasi beserta dengan detail dari jurusan transportasi tersebut.

Ketika memilih kategori yang berada dalam gabungan data POI maka akan masuk ke tampilan dimana *marker* AR akan muncul di layar *handphone*. Dibutuhkan interaksi pengguna untuk menampilkan(visualisasi) *marker* AR pada layar handphone. Pengguna dapat memilih salah satu *marker* AR dimana pada setiap masing-masing *marker* telah disisipi detail informasi mengenai lokasi tersebut. Pada tampilan detail informasi selain terdapat 2 fitur yaitu *direction* dan *phone dialer*. Direction berfungsi sebagai penunjuk arah antara lokasi pengguna dengan lokasi yang dipilih. Sedangkan *phone dialer* merupakan fitur yang dapat mengaktifkan *dialer* telepon tersebut. Ketika dalam mode kamera mencari *marker* AR terdapat fitur *zoom level*, yaitu fitur yang dapat mengatur jarak radius lokasi yang ingin ditampilkan dengan maksimum radius 80 km. Ketika pengguna memilih kategori transportasi maka akan menampilkan seuruh jurusan angkot dengan *list* yang dilengkapi fitur pencarian. Jika salah satu jurusan dipilih maka akan muncul tampilan seluruh detail informasi tersebut.

2. *State Diagram Server Application*(*Backend*)



Gambar 3. 9 State diagram *server application*(backend)

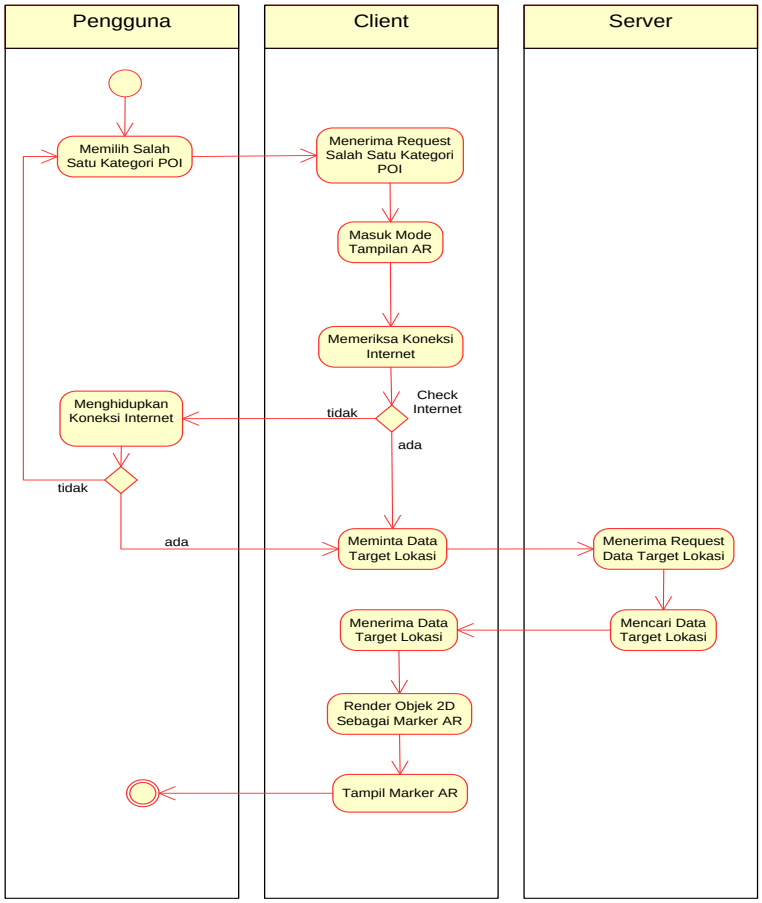
Pada *State Diagram* Aplikasi *Backend* menunjukkan seluruh aktifitas dilakukan oleh admin. Dengan keadaan awal ditampilkan halaman web login, kemudian ketika berhasil maka akan ditampilkan halaman main menu. Selanjutnya admin dapat memilh pilihan menu *category* atau pilihan menu POI informasi pada halaman menu utama atau melalui kolom navigator. Apabila memilih menu *category* maka akan ditampilkan halaman pengolahannya, pada halaman pengolahan tersebut memiliki fitur tambah, ubah, hapus dan cari yang setiap fitur tersebut memiliki peran terbentuknya sebuah data maupun perubahan data. Begitu pula sebaliknya apabila

admin memilih menu POI atau transportasi maka akan ditampilkan *form* pengolahan POI atau transportasi, yang setiap halaman pengolahannya memiliki fitur tambah, ubah, hapus dan cari.

d. 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 paralel yang mungkin terjadi pada beberapa eksekusi. *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. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

1. *Activity Diagram* Lihat POI

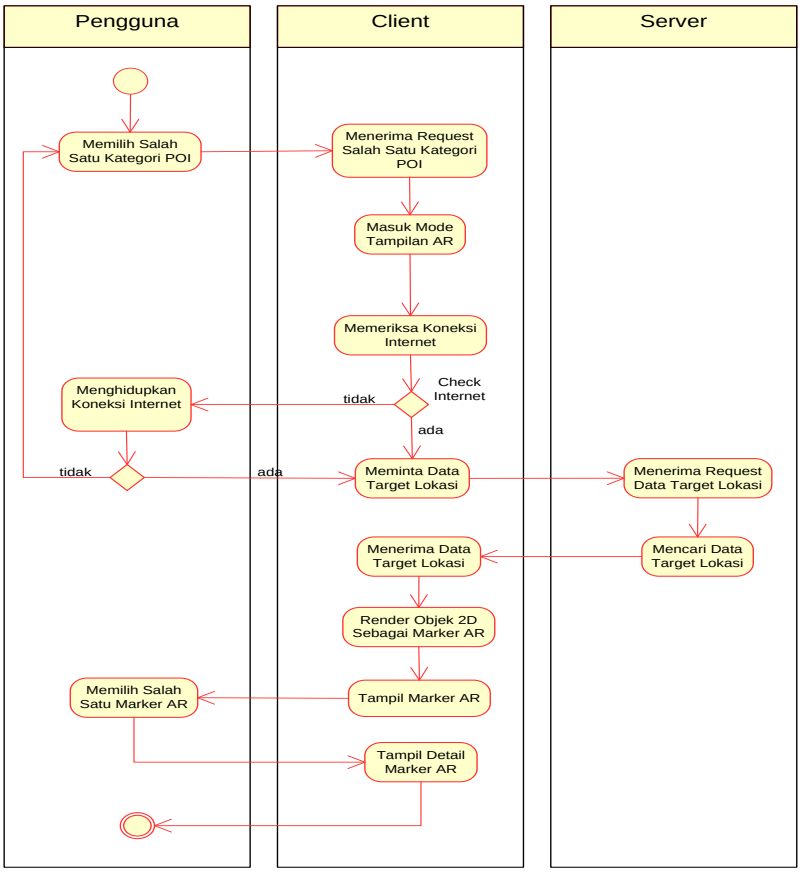


Gambar 3. 10 Activity diagram Lihat POI

Pada gambar 3.10 memperlihatkan aktifitas melihat POI *agumented reality* oleh pengguna. Sebelumnya ada 4 kategori yang termasuk ke dalam visualiasi POI *augmented reality*, yaitu restoran, wisata, penginapan dan perbelanjaan. Jika salah satu dari 4 menu kategori tersebut dipilih maka akan muncul tampilan berupa visualiasi lokasi dengan objek berupa gambar2D(*marker* AR). Sebelum objek tersebut muncul aplikasi akan memeriksa syarat yang dibutuhkan agar aplikasi dapat berjalan dengan baik, syarat tersebut yaitu tersedianya koneksi jaringan internet. Jika

koneksi internet ada maka sistem akan meminta data ke *server* dengan *web service* sebagai jembatan penerjemah datanya. Setelah data tersedia selanjutnya sistem akan membentuk objek gambar 2D(*marker* AR) yang mana memperlihatkan titik lokasi tersebut. Namun apabila tidak memiliki koneksi internet akan muncul pesan untuk menghidupkan koneksi internet, apabila koneksi internet tidak dihidupkan maka terdapat 2 pilihan kembali ke menu utama atau keluar aplikasi.

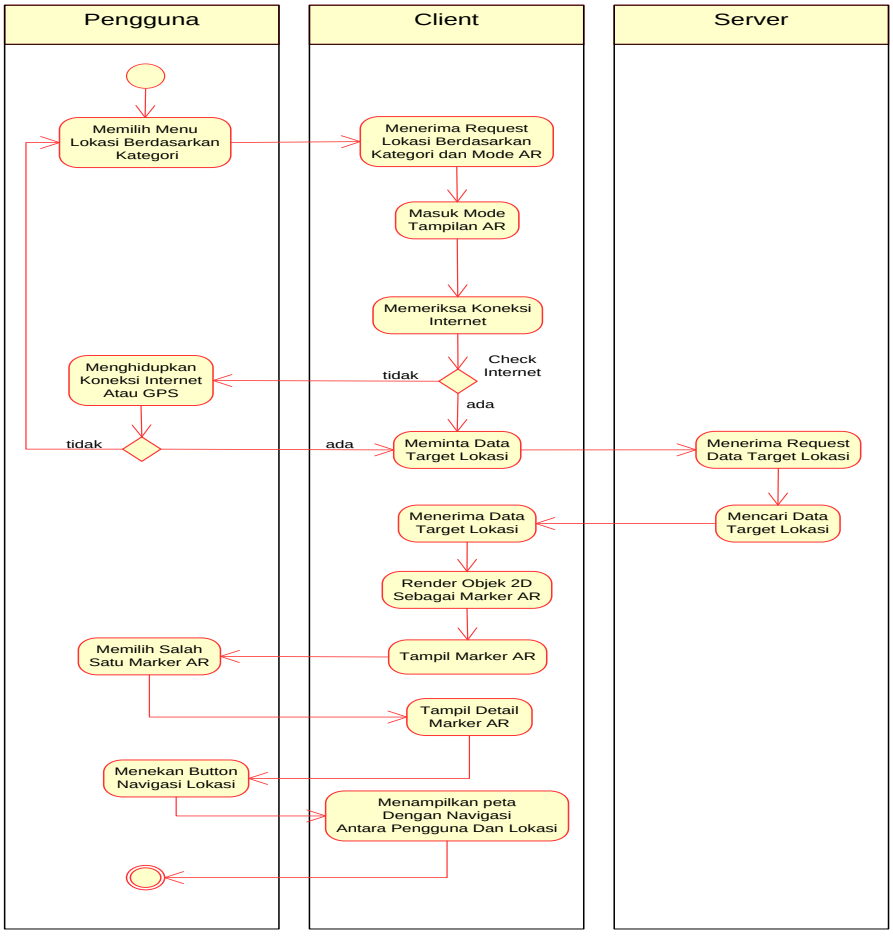
2. *Activity Diagram* Lihat *Detail Marker* AR



Gambar 3. 11 Activity diagram Lihat Detail Marker AR

Pada gambar 3.11 memperlihatkan aktifitas melihat detail marker AR yang sebelumnya merupakan kelanjutan dari *activity diagram* lihat POI. Pada activity diagram lihat POI aktifitas sistem yang terakhir dilakukan yaitu tampil *marker* AR yang kemudian proses tersebut dilanjutkan dengan memilih salah satu *marker* AR oleh pengguna yang kemudian sistem akan merespon dengan menampilkan detail *marker* AR. Detail *marker* AR tersebut berisikan informasi-informasi pendukung lokasi.

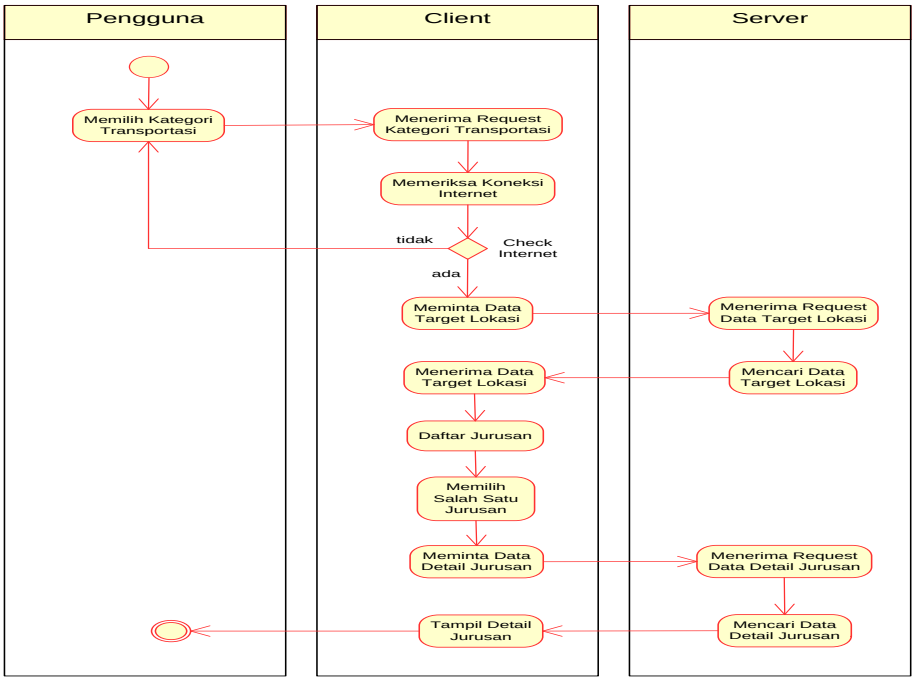
3. *Activity Diagram* Lihat Peta



Gambar 3. 12 Activity diagram Lihat Peta

Pada gambar 3.12 memperlihatkan aktifitas melihat detail marker AR yang sebelumnya merupakan kelanjutan dari *activity diagram* lihat detail *marker* AR. Pada activity diagram lihat detail marker AR aktifitas sistem yang terakhir dilakukan yaitu menampilkan detail marker AR. Aktifitas selanjutnya yang dilakukan setelah menampilkan detail marker adalah menekan button navigasi lokasi(gambar icon map) pada detail marker AR, kemudian sistem akan merespon dengan cara menampilkan navigasi lokasi pengguna dengan lokasi yang dituju. Sistem akan menampilkan lokasi pengguna dan lokasi tujuan melalui map yang telah terintegrasi dengan *handphone*, fitur ini akan tampil apabila pengguna menekan button navigasi lokasi(gambar icon map) pada detail marker AR.

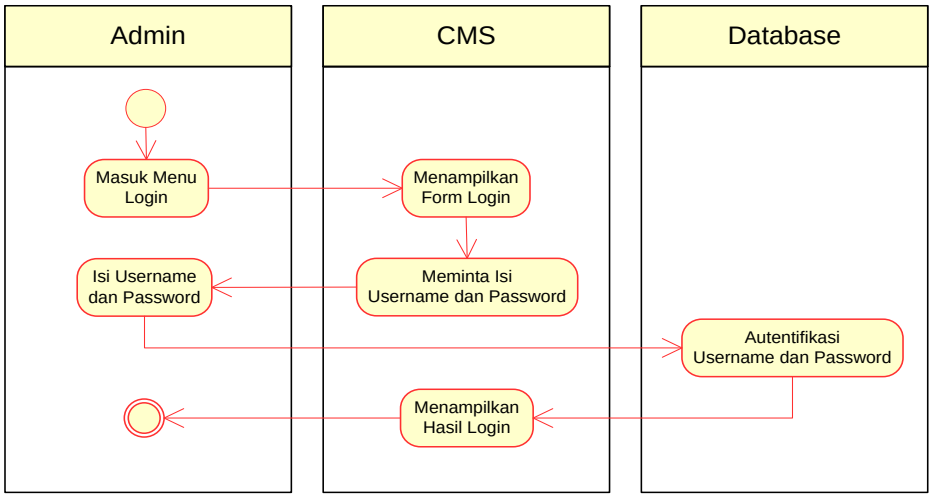
4. *Activity Diagram* Lihat Transportasi



Gambar 3. 13 Activity diagram lihat transportasi

Pada Gambar 3.13 memperlihatkan aktivitas ketika kategori transportasi dipilih oleh pengguna. Ketika kategori tersebut dipilih maka aplikasi akan menampilkan seluruh daftar jurusan angkot dengan *list* yang dilengkapi juga dengan fitur pencarian. Sebelum masuk ke tampilan list daftar seluruh jurusan angkot, aplikasi akan memeriksa koneksi jaringan internet yang merupakan syarat sistem tersebut agar data yang berada dalam server dapat ditampilkan. Ketika syarat tersebut terpenuhi maka langkah pengguna selanjutnya yaitu memilih salah satu jurusan tersebut. Jika salah satu tersebut dipilih maka sistem akan meminta data detail informasi yang berada dalam *server* dengan web *service* sebagai jembatan penerjemah datanya. Setelah data diterima sistem maka detail informasi jurusan tersebut akan tampil.

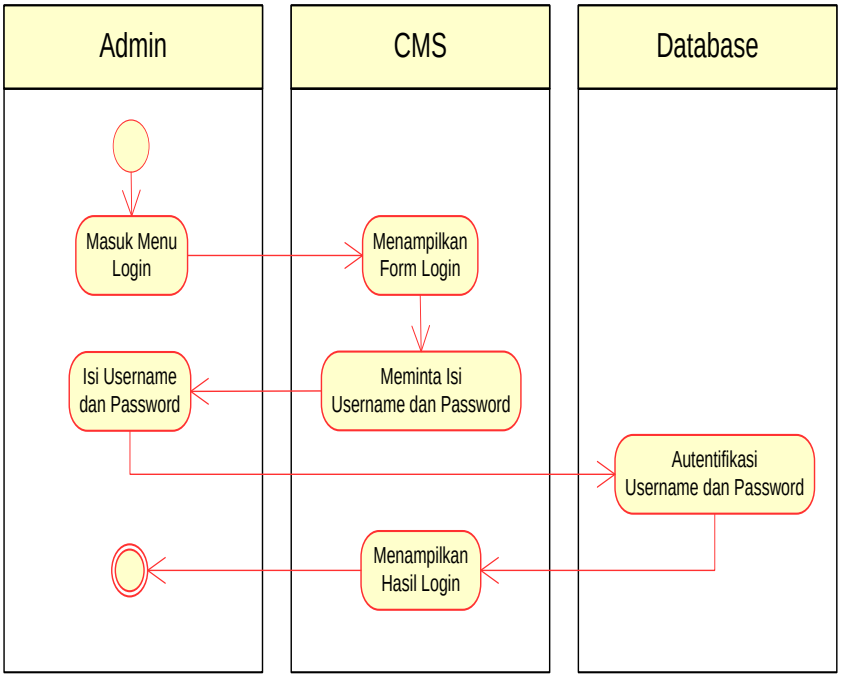
5. *Activity Diagram* Login



Gambar 3. 14 Activity diagram login

Gambar 3.14 memperlihatkan aktivitas login. Admin membuka halaman web login, kemudian memasukan username dan password. *Server* akan mengautentifikasi data yang dimasukan hasilnya diberitahukan ke aplikasi.

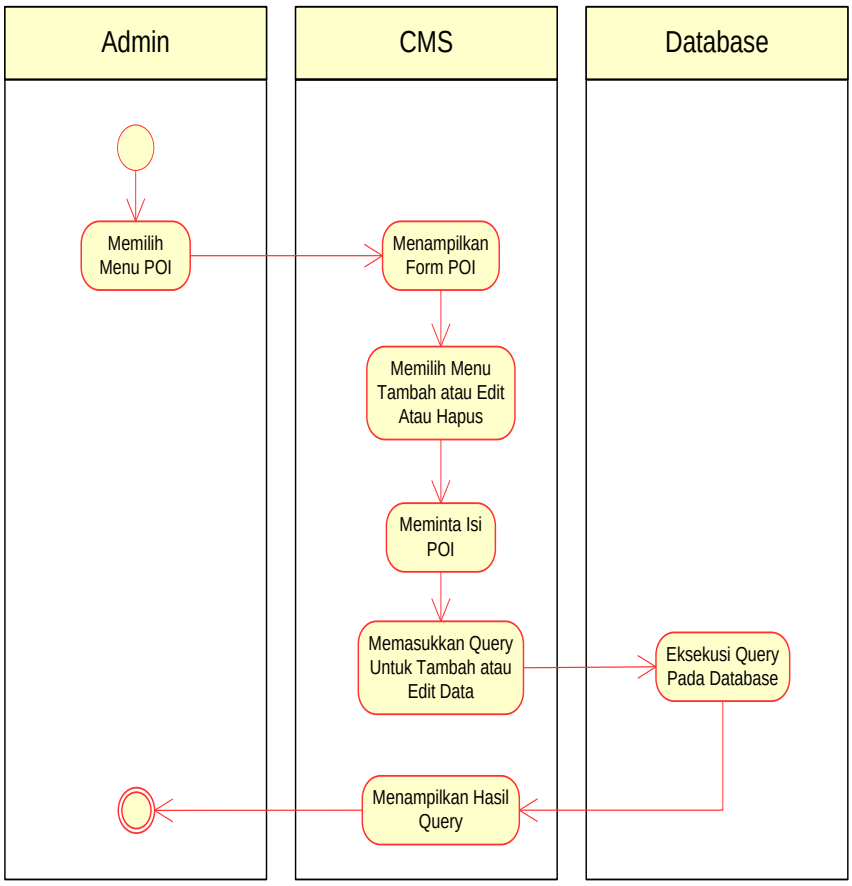
6. *Activity Diagram* Mengelola kategori



Gambar 3. 15 Activity diagram olah kategori

Gambar 3.15 memperlihatkan aktivitas olah data kategori. Admin memilih daftar menu kategori, kemudian CMS akan menampilkan form kategori. Lalu admin dapat memilih sublink untuk tambah atau ubah kategori lalu CMS akan menampilkan Form isian data. *Server* mengeksekusi hasil query kemudian mengirimkannya ke aplikasi CMS.

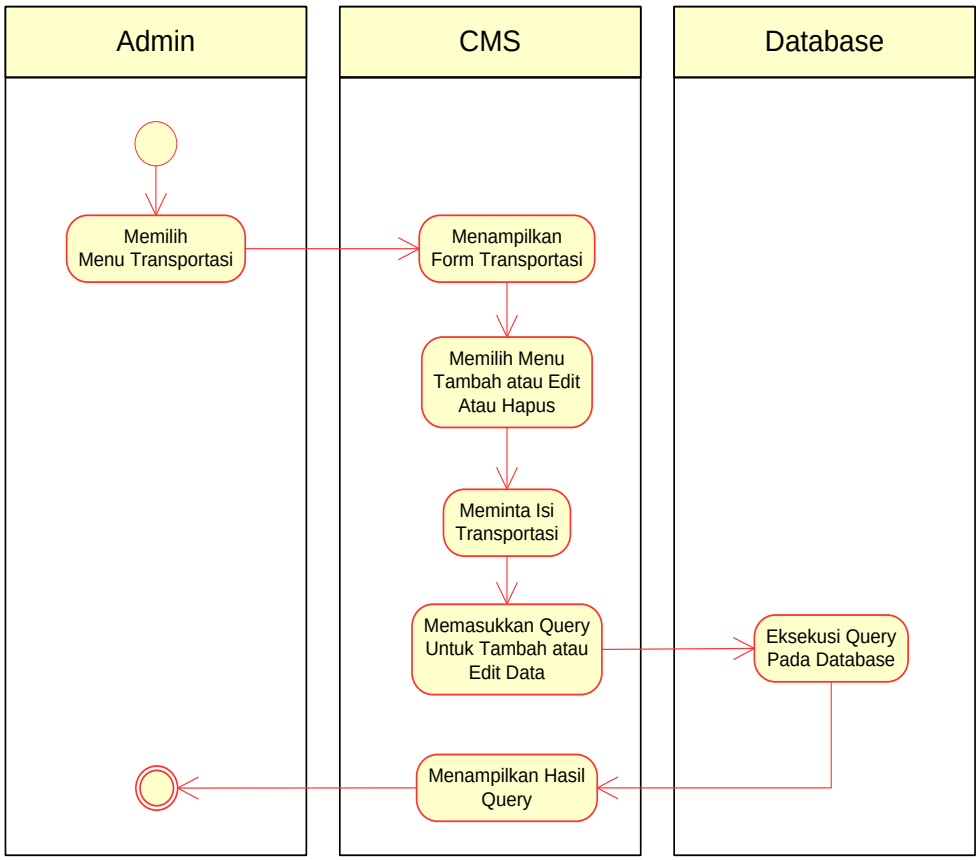
7. *Activity Diagram* Mengelola POI



Gambar 3. 16 Activity diagram olah poi

Gambar 3.16 memperlihatkan aktivitas olah poi. Admin memilih daftar menu olah poi, kemudian CMS akan menampilkan form poi. Lalu admin dapat memilih sublink untuk tambah atau ubah atau hapus poi lalu CMS akan menampilkan Form isian data apabila tambah atau hapus poi yang dipilih namun apabila hapus yang dipilih maka akan menampilkan pesan konfirmasi penghapusan. *Server* mengeksekusi hasil query kemudian mengirimkannya ke aplikasi CMS.

8. *Activity Diagram* Mengelola Transportasi



Gambar 3. 17 Activity diagram olah transportasi

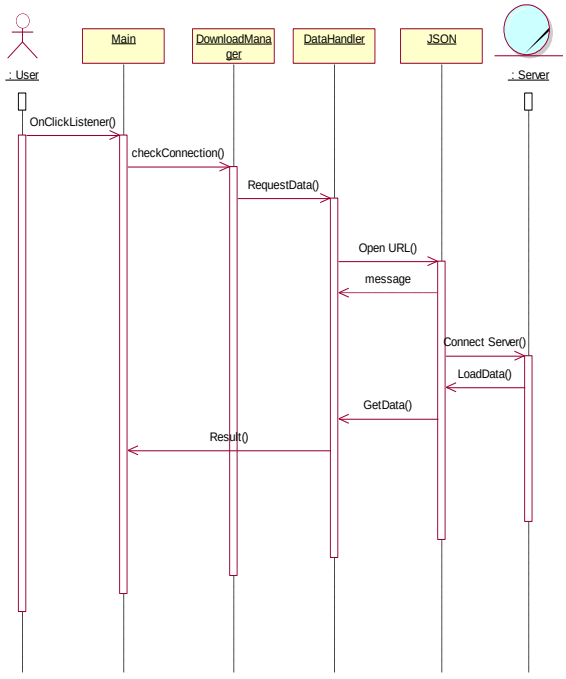
Gambar 3.17 memperlihatkan aktivitas olah transportasi. Admin memilih daftar menu olah poi, kemudian CMS akan menampilkan form transportasi. Lalu admin dapat memilih sublink untuk tambah atau ubah atau hapus transportasi lalu CMS akan menampilkan Form isian data apabila tambah atau hapus transportasi yang dipilih namun apabila hapus yang dipilih maka akan menampilkan pesan konfirmasi penghapusan. *Server* mengeksekusi hasil query kemudian mengirimkannya ke aplikasi CMS.

e. 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).

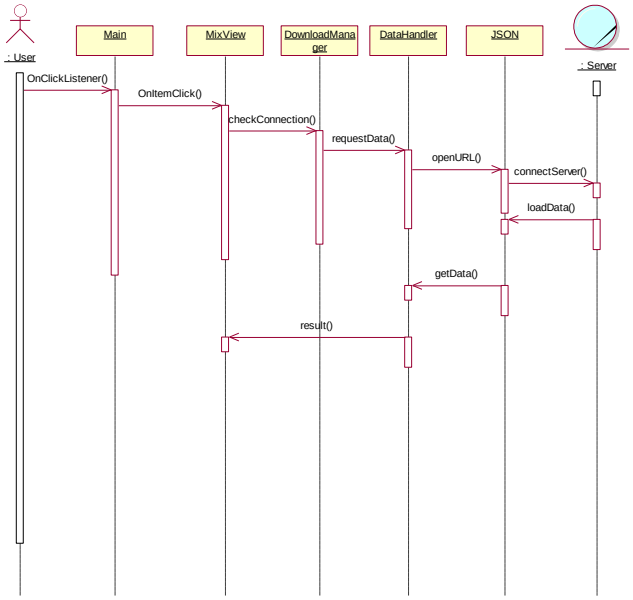
Pada *Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan pada sistem sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu.

1) *Sequence Diagram* Pilih Kategori



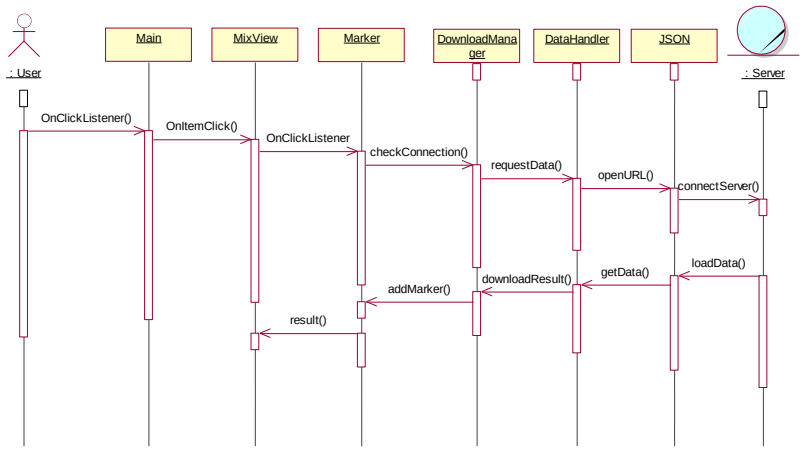
Gambar 3. 18 Sequence diagram lihat kategori

2) *Sequence Diagram* Lihat POI



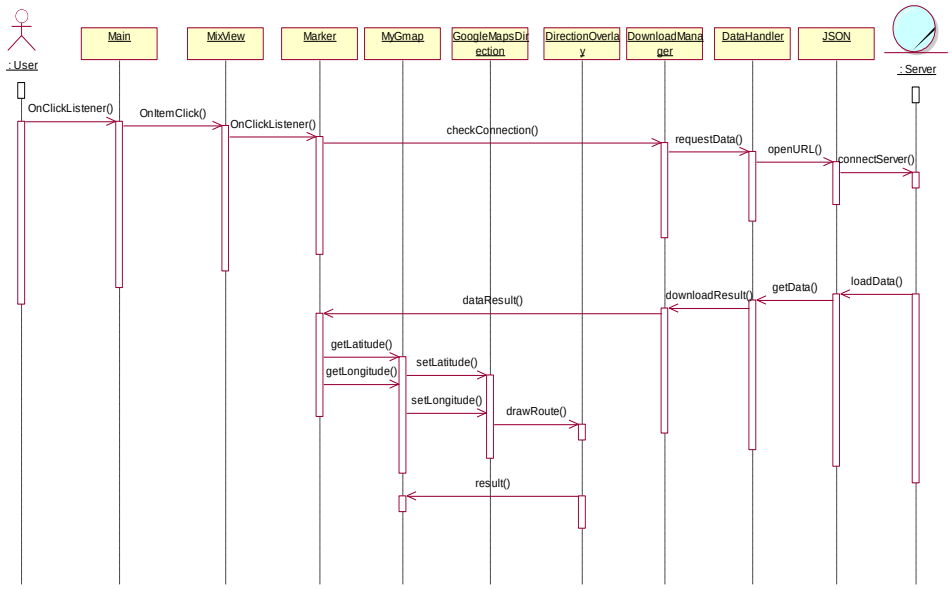
Gambar 3. 19 Sequence diagram lihat POI

3) *Sequence Diagram* Lihat Detail Marker AR



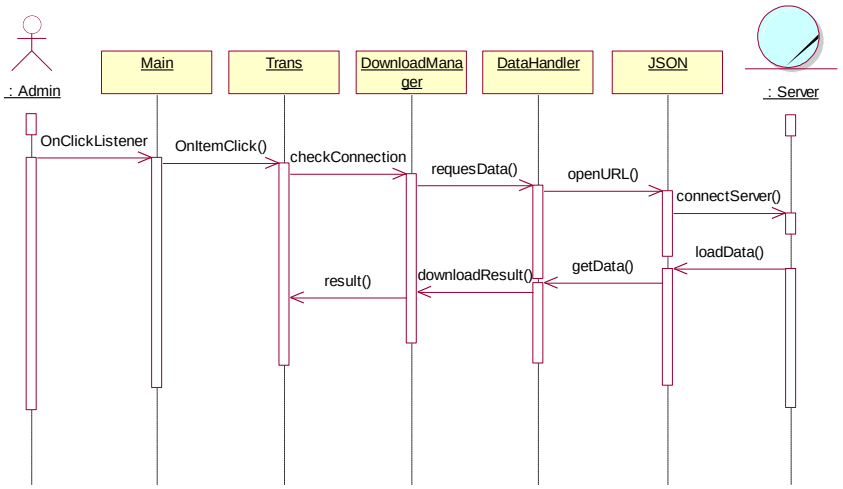
Gambar 3. 20 Sequence diagram lihat detail marker AR

4) *Sequence Diagram* Lihat Peta



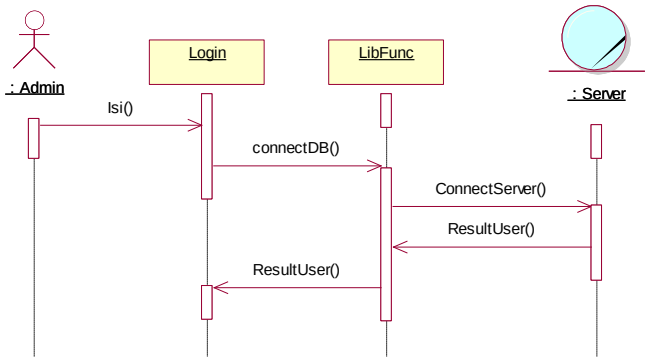
Gambar 3. 21 Sequence diagram lihat peta

5) *Sequence Diagram* Lihat Transportasi



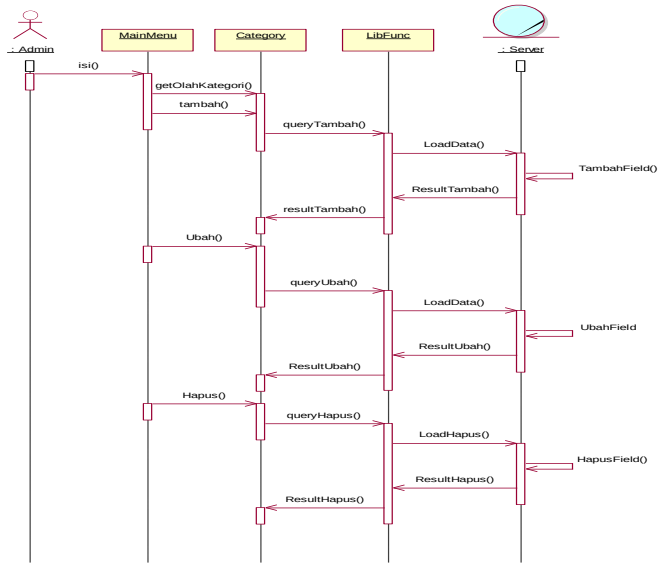
Gambar 3. 22 Sequence Diagram lihat transportasi

6) *Sequence Diagram Login*



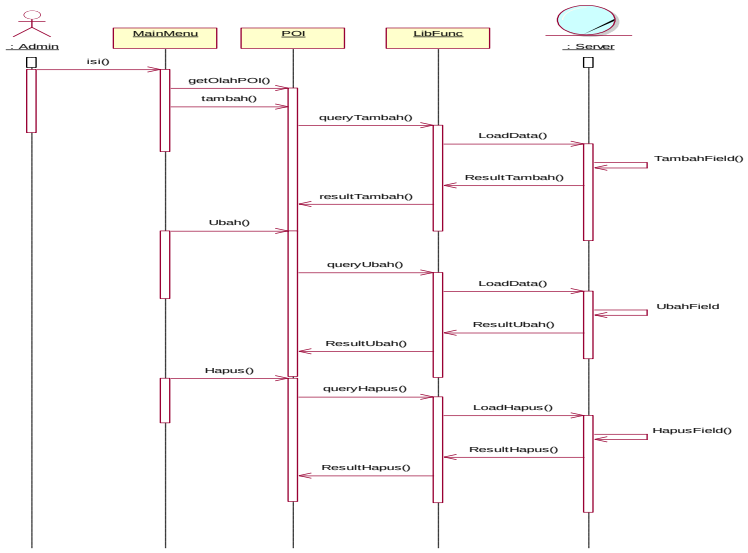
Gambar 3. 23 Sequence diagram login

7) *Sequence Diagram* Mengelola Kategori



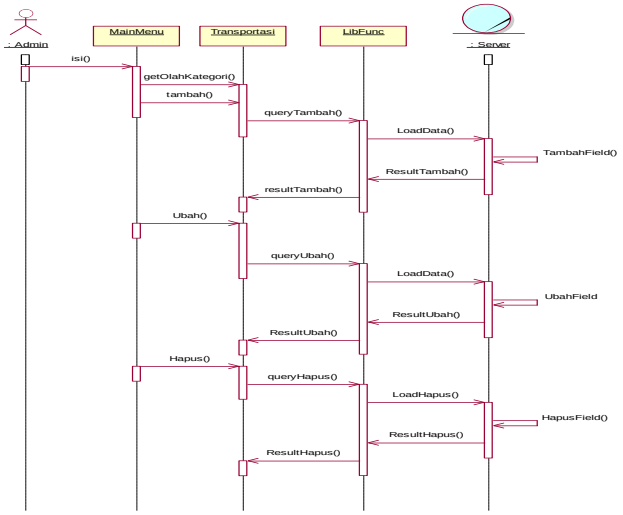
Gambar 3. 24 Sequence diagram mengelola kategori

8) *Sequence Diagram* Mengelola POI



Gambar 3. 25 Sequence diagram mengelola POI

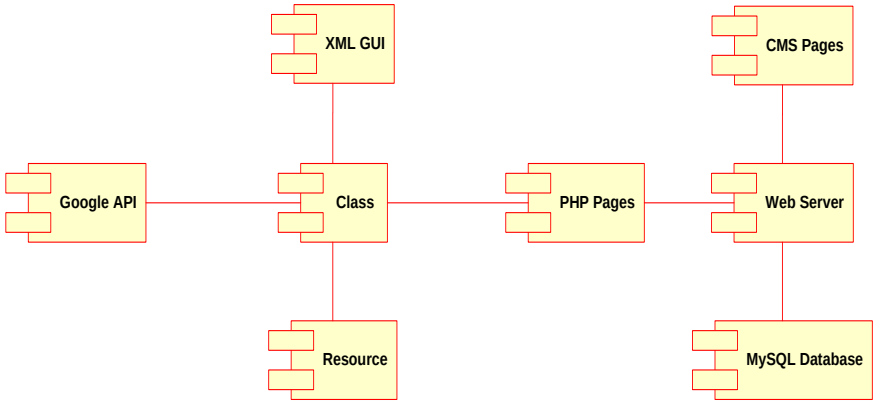
9) *Sequence Diagram* Mengelola Transportasi



Gambar 3. 26 Sequence diagram mengelola Transportasi

f. Component Diagram

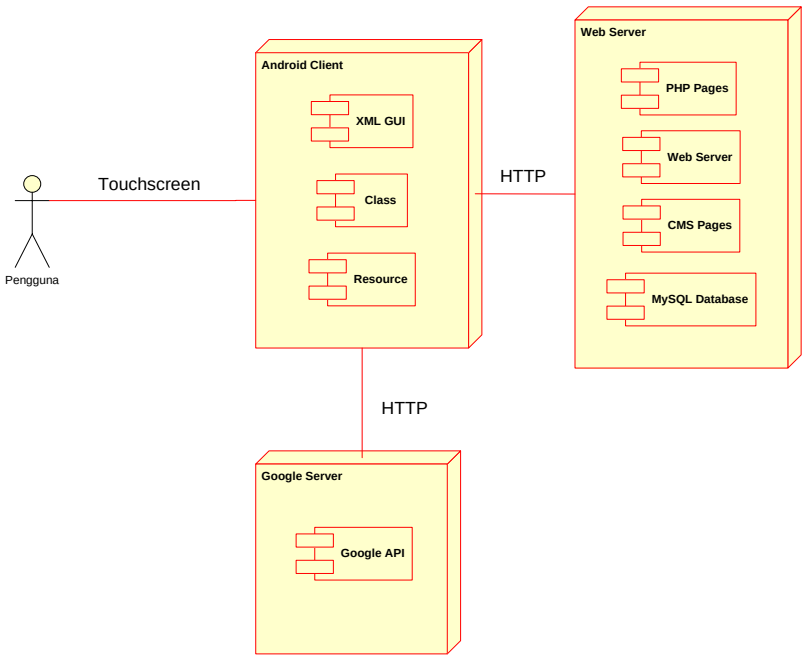
Pada tahap ini, *Component Diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak termasuk ketergantungan (*dependency*). Komponen piranti lunak yang yang dimaksud adalah modul yang berisi *source code* atau *binary code*, yang ada *library* atau *excutetable* yang muncul pada *compile time*, *link time* ataupun pada *runtime*. *Component Diagram* ditunjukan pada gambar 3.26.



Gambar 3. 27 Component diagram

g. Deployment Diagram

Diagram Deployment menggambarkan rinci bagaimana komponen di-deploy dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin *Server* atau perangkat keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *Server* dan hal-hal yang lain bersifat fisikal.



Gambar 3. 28 Deployment diagram

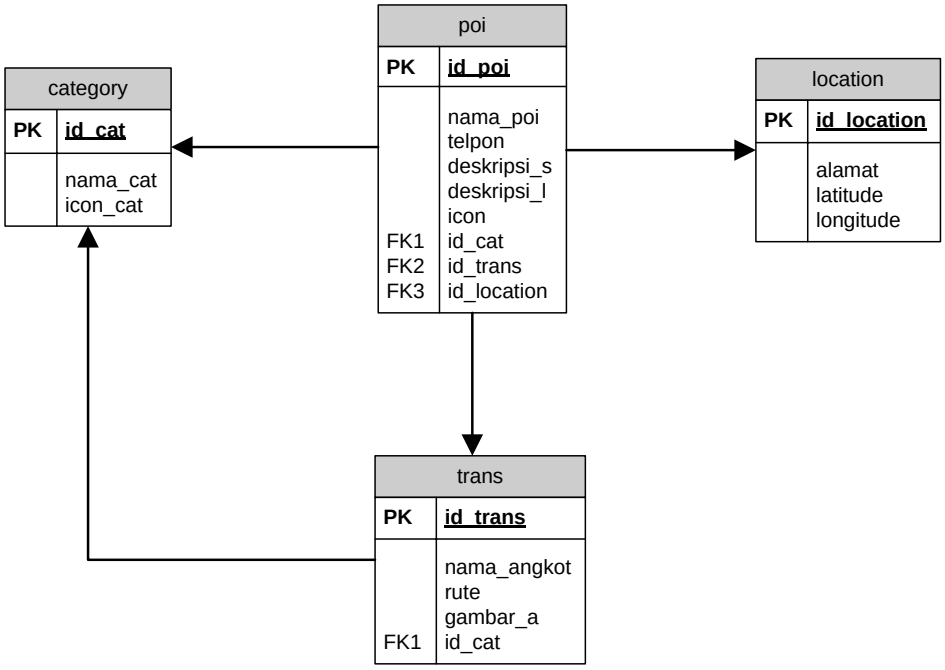
3.5 Perancangan Sistem

Perancangan sistem adalah suatu proses yang menggambarkan bagaimana suatu sistem dibangun untuk memenuhi kebutuhan pada fase analisis. Adapun tahapan yang dilakukan dalam perancangan sistem ini membahas mengenai tujuan perancangan database, dan perancangan antar muka.

3.5.1 Skema Relasi

Relasi antar tabel merupakan gabungan antar *file* yang mempunyai kunci utama yang sama, sehingga *file-file* tersebut menjadi satu kesatuan yang dihubungkan oleh *field* kunci (*Primary Key*). Pada proses ini elemen-elemen data

dikelompokkan menjadi satu *file database* beserta *entitas* dan hubungannya. Skema relasi ini dapat dilihat pada gambar dibawah ini.



Gambar 3. 29 Skema relasi aplikasi bandung tour guide

Tabel 3. 15 Struktur tabel category

Nama Field	Tipe Data	Panjang	Keterangan
id_cat	Integer	11	PK (Primary Key)
nama_cat	Varchar	20	
icon_category	Varchar	25	

Tabel 3. 16 Struktur tabel POI

Nama Field	Tipe Data	Panjang	Keterangan
id_poi	Integer	11	PK (Primary Key)
Nama_poi	Varchar	50	
telpon	Varchar	20	
deskripsi_s	Text		
deksripsi_l	Text		
icon	Varchar	10	
id_cat	Integer	11	FK(Foreign Key) ref category(id_cat)
id_location	Integer	11	FK(Foreign Key) ref location(id_location)
id_trans	Integer	11	FK(Foreign Key) ref trans(id_trans)

Tabel 3. 17 Struktur tabel location

Nama Field	Tipe Data	Panjang	Keterangan
id_location	Integer	11	PK (Primary Key)
alamat	Varchar	50	
latitude	Double		
longitude	Double		

Tabel 3. 18 Struktur tabel trans

Nama Field	Tipe Data	Panjang	Keterangan
id_trans	Integer	11	PK (Primary Key)
Nama_angkot	Varchar	50	
warna	Varchar	30	
rute	Text		
Gambar_a	Varchar	10	
id_category	Integer	11	FK(Foreign Key) ref category(id_cat)

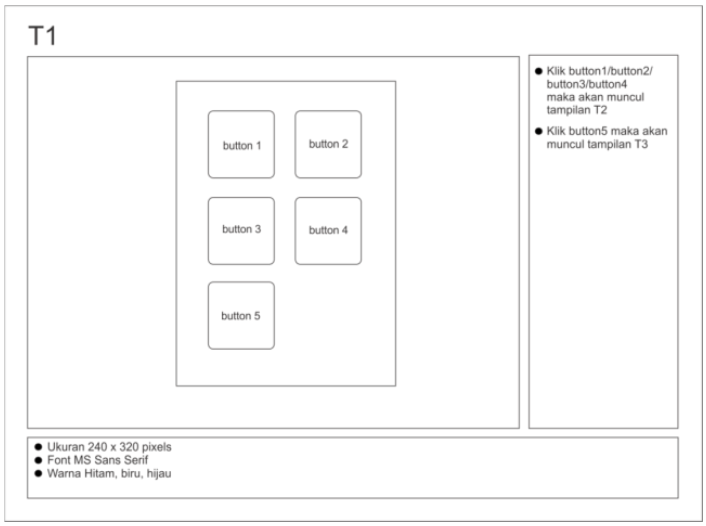
3.5.2 Perancangan Antar Muka

Perancangan antarmuka dibutuhkan untuk mewakili keadaan sebenarnya dari aplikasi yang akan dibangun, berikut akan disajikan perancangan antarmuka dari aplikasi yang akan dibangun yang terdiri dari 2 bagian yaitu :

3.5.2.1 Antar Muka Client Application

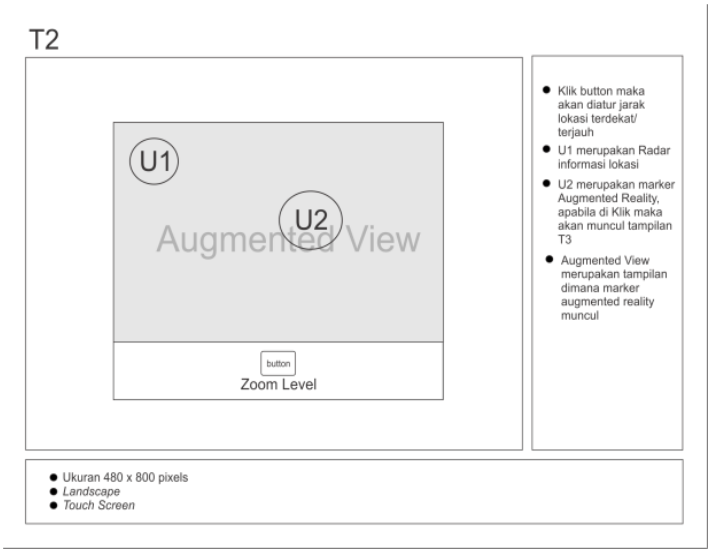
Agar dapat memudahkan proses pembuatan aplikasi maka terlebih dahulu membuat rancangan design tampilannya. Pada perancangan antar muka *client application* ini meliputi tampilan antar muka pilih kategori, POI *augmented reality* dan transportasi.

1. Antar Muka Menu Tampilan Pilih Kategori



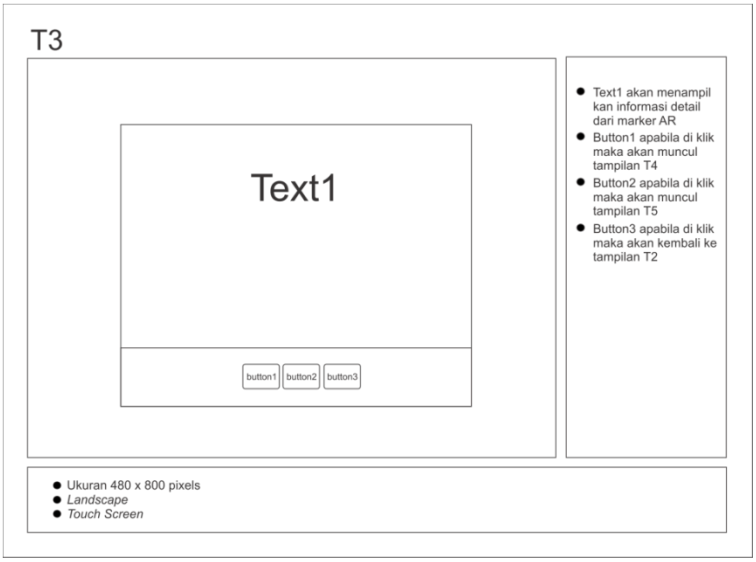
Gambar 3. 30 Antar muka menu tampilan pilih kategori

2. Antar Muka Lihat *Marker* AR



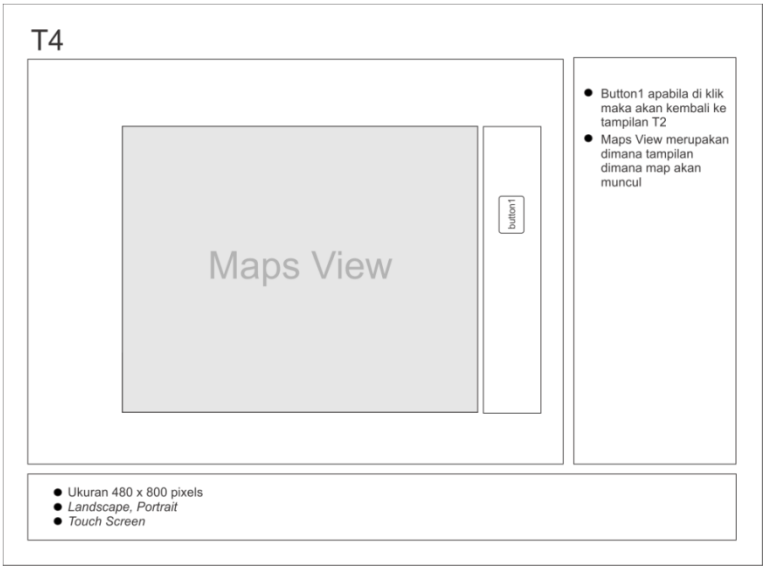
Gambar 3. 31 Antar muka lihat marker AR

3. Antar Muka Detail *Marker* AR



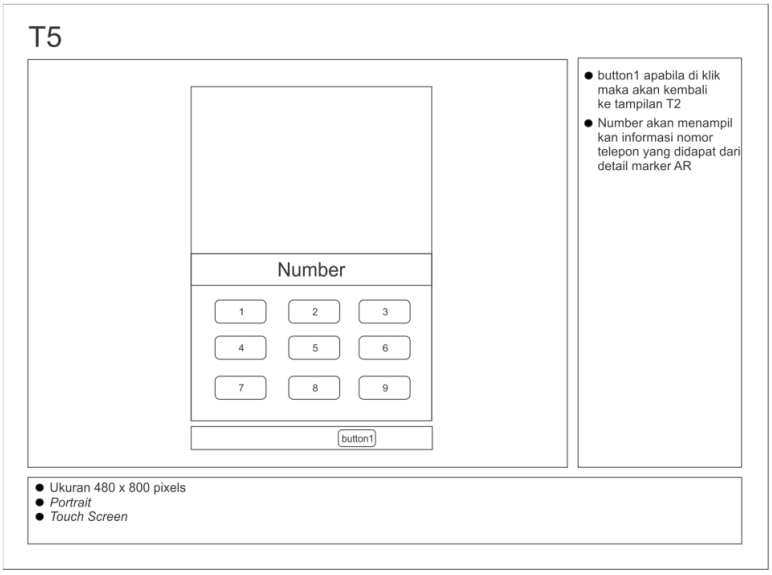
Gambar 3. 32 Antar muka detail marker AR

4. Antar Muka Navigasi Map Lokasi



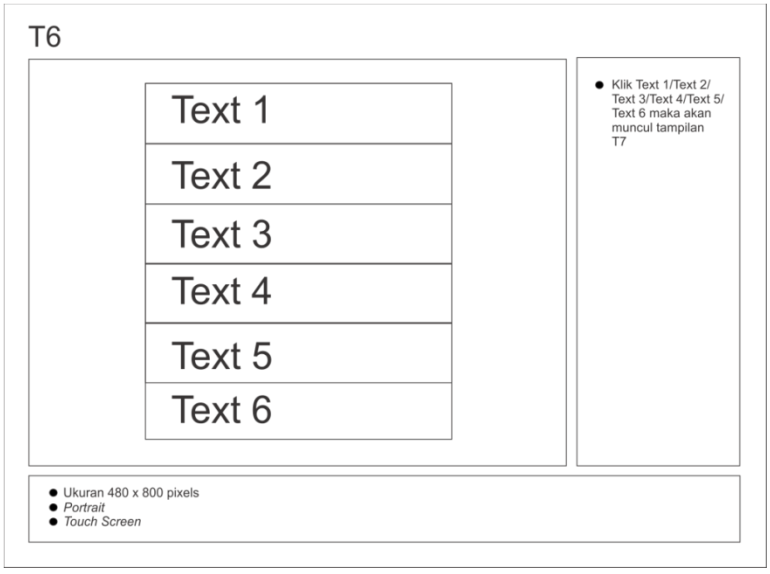
Gambar 3. 33 Antar muka navigasi map lokasi

5. Antar Muka hubungi Lokasi



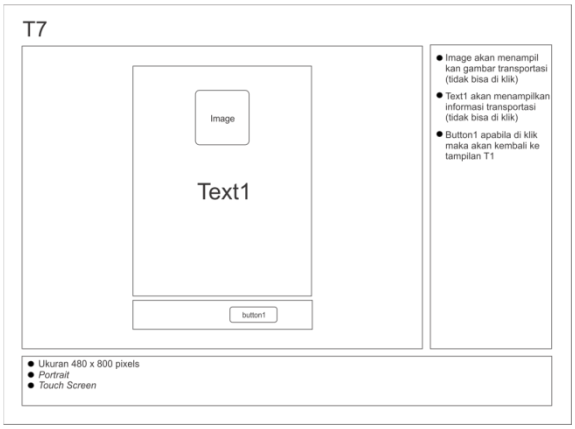
Gambar 3. 34 Antar muka hubungi lokasi

6. Antar Muka Lihat List Transportasi



Gambar 3. 35 Antar muka lihat list transportasi

7. Antar Muka Menu Lihat Detail Transportasi

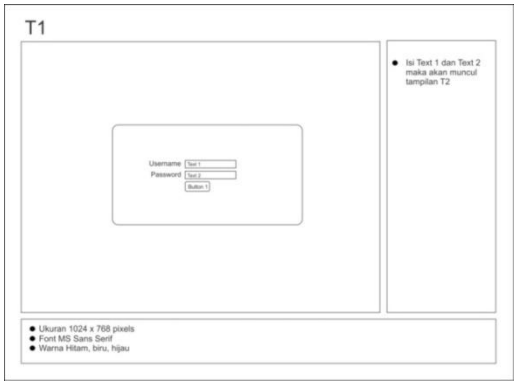


Gambar 3. 36 Antar muka lihat detail transportasi

3.5.2.2 Antar Muka Server Application

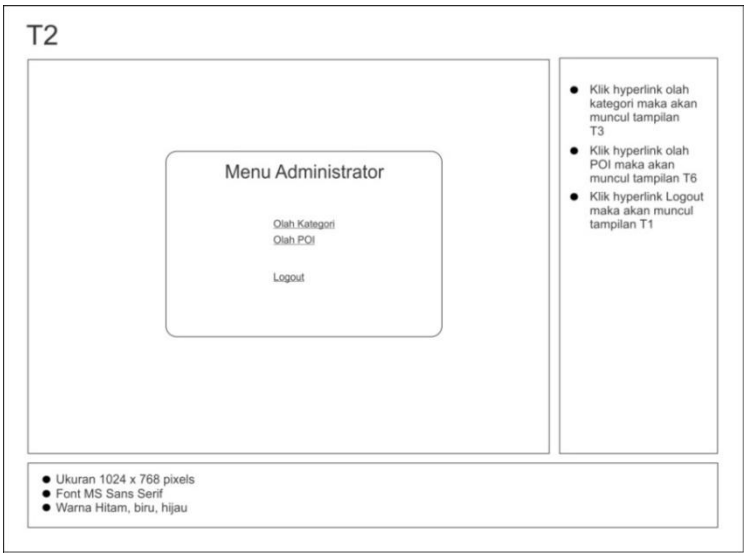
Berbeda dengan *frontend application* yang memiliki rancangan yang begitu kompleks, *backend application* hanya meliputi rancanagan antar muka saja yaitu terdiri dari halaman login, menu administrator dan halaman tambah dan ubah data.

1. Antar Muka Menu Halaman Login Admin



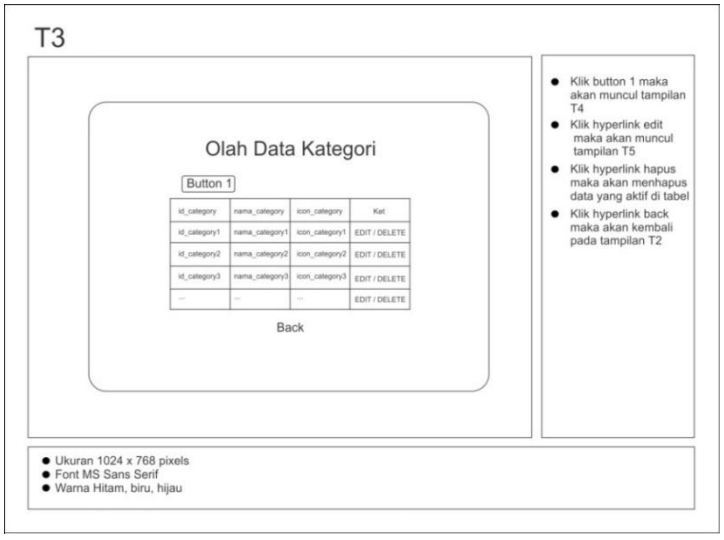
Gambar 3. 37 Antar muka halaman login admin

2. Antar Muka Halaman Utama Web Admin



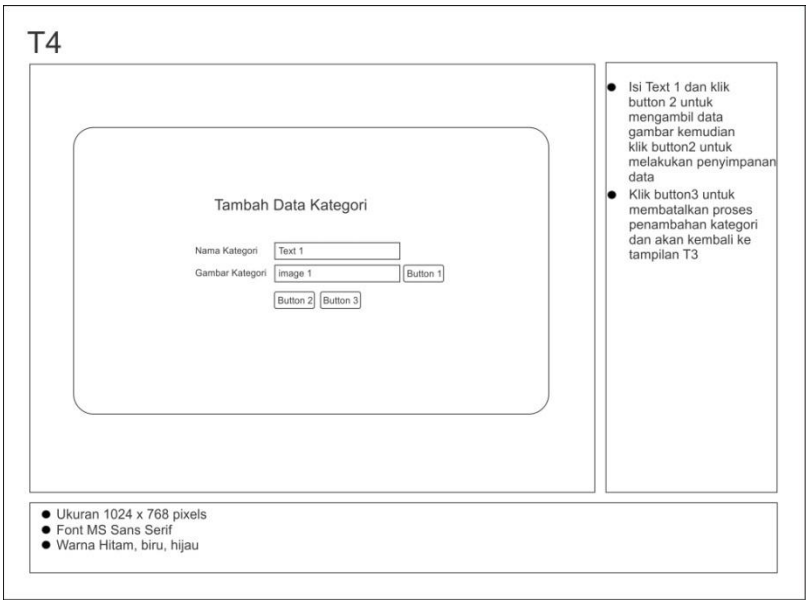
Gambar 3. 38 Antar muka halaman utama web admin

3. Antar Muka Halaman Olah Data Kategori



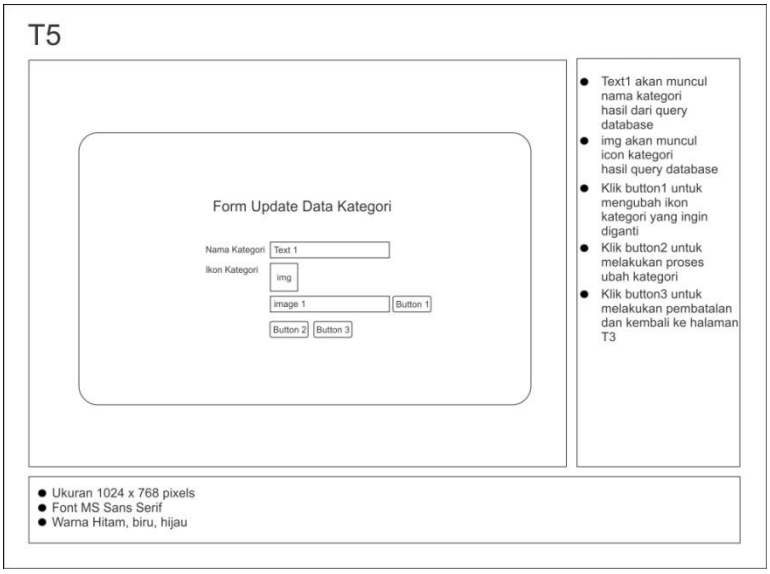
Gambar 3. 39 Antar muka halaman olah data kategori

4. Antar Muka Halaman Tambah Data Kategori



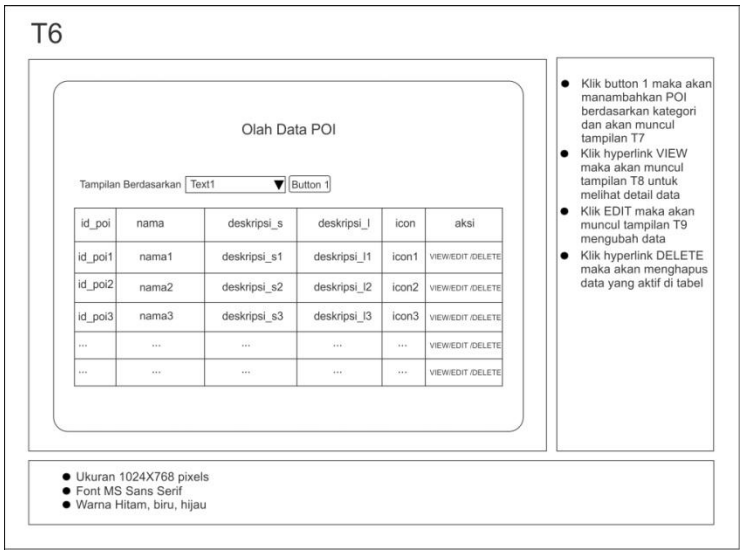
Gambar 3. 40 Antar muka halaman tambah data kategori

5. Antar Muka Halaman *Update* Data Kategori



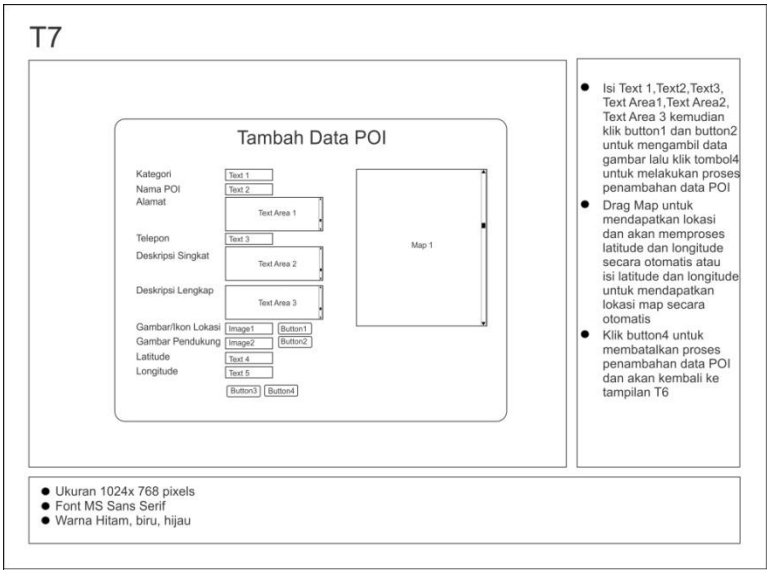
Gambar 3. 41 Antar muka halaman update data kategori

6. Antar Muka Halaman Olah Data POI



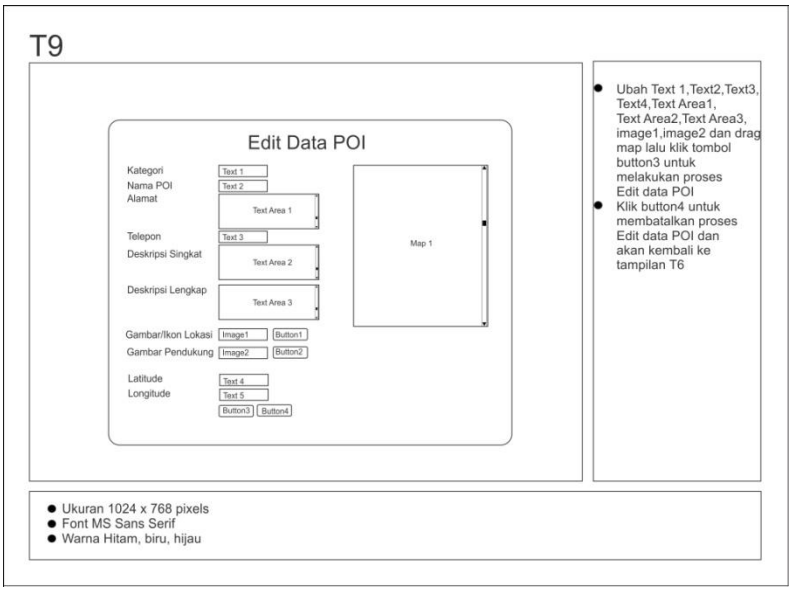
Gambar 3. 42 Antar muka halaman olah data POI

7. Antar Muka Halaman Tambah Data POI



Gambar 3. 43 Antar muka halaman tambah data POI

8. Antar Muka Halaman *Edit* Data POI



Gambar 3. 44 Antar muka halaman edit data POI