BAB III
ANALISIS DAN PERANCANGAN SISTEM

Analisis dan perancangan sistem memerlukan tahapan yang sistematis untuk mendapatkan aplikasi yang baik dan bersesuaian dengan kegunaan dan tujuannya. Tahap awal dari analisis adalah menganalisis kebutuhan-kebutuhan sistem mulai dari kubutuhan pengguna, kubutuhan non fungsional, dan kebutuhan fungsional. Sedangkan untuk tahap perancangan aplikasi yaitu perancangan database, dan perancangan antarmuka.

3.1 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat sesuai dengan kebutuhan.

3.1.1 Analisis Masalah

Keberadaan fasilitas umum yang tersebar dibeberapa lokasi masih menjadi kendala dalam memaksimalkan fungsi dari fasilitas umum itu sendiri. Ini semua disebabkan karena minimnya informasi mengenai keberadaan fasilitas umum. Cara yang dapat dilakukan untuk mendapatkan informasi adalah dengan melihat peta yang tersedia atau bertanya kepada orang sekitar mengenai lokasi keberadaan fasilitas umum seperti restoran, ATM, rumah sakit dan SPBU.

Namun cara tersebut kurang efektif, mengingat bahwa orang di sekitar belum tentu mengetahui informasi yang dibutuhkan dan pastinya membutuhkan waktu yang lama. Oleh karena itu dibutuhkan sebuah sistem yang efektif, cepat dan mudah digunakan dalam memberikan informai tersebut yaitu dalam sebuah aplikasi *mobile*. Karena dengan perangkat *mobile* aplikasi ini dapat digunakan kapan saja dan dimana saja asalkan terintegrasi dengan jaringan internet.

3.1.2 Analisis Penentuan Lokasi Dengan GPS

Dalam menentukan posisi suatu lokasi pada penelitian ini metode yang digunakan yaitu *Cell-ID (Cell Identification)*. Yaitu metode yang didasarkan pada daerah geografis yang tercakup oleh sebuah cell yang berhubungan dengan daerah cakupan dari sinyal GPS atau radio. Untuk itu cell yang dikenali dari *base station* merupan sebuah nilai koordinat berupa *Latitude* dan *Longitude*. Untuk mengukur jarak *mobile* dari *base station* tidak dapat diketahui dengan pasti, Oleh karena itu, untuk lebih meningkatkan lagi akurasi hasil pencarian, metode *Cell ID* dapat dikombinasikan dengan metode *Timing Advanced* (*TA*), dengan menggunakan TA ini, metode *Cell ID* akan ditambahkan sebuah fungsionalitas untuk menghitung *Round Trip Time* (*RTT*), yaitu waktu transmisi sebuah frame (dari *base station* ke handphone) dan waktu penerimaan sebuah frame (dari *mobile* ke *base station*). Dengan tambahan metode ini, jarak antara *mobile* dan base station dapat ditentukan dengan keakuratan 50 m.

Berikut ini salah satu rumus yang sering digunakan untuk menentukan jarak lokasi pada metode CELL ID dengan mengabaikan bentuk geografis baik bukit maupun lembah.

Jarak = **Arc Cos (Cos(Lat1) x Cos(Lat2) x Cos(Long2-Long1) +**

Sin(Lat1) x Sin(Lat 2)) x R

Dimana

Lat1 = Nilai Latitude dari Lokasi Awal

Long1 = Nilai Longitude dari Lokasi Awal

Lat2 = Nilai Latitude dari Lokasi Tujuan

Long2 = Nilai Latitude dari Lokasi Tujuan

R = Jari-jari rata-rata Bumi (6378.2 KM)

Catatan : Nilai Lat dan Long dalam satuan radian.

Contoh : Misalkan untuk mencari jarak antara Bandara Sultan Iskandar Muda di Banda Aceh ke Bandara Sentani di Jayapura Jika diketahui :

1. Latitude 1 (Banda Aceh) 5°31'1"N dan Longitude 1 (Banda Aceh) 95°25'12"E
2. Latitude 2 (Jayapura) 2°34'19"S dan Longitude 2 (Jayapura)140°30'41"E

Koordinat diatas kan masih dalam bentuk degree minute secon dirubah ke bentuk degree decimal

1. Latitude 1 (Banda Aceh) 5.516944 dan Longitude 1 (Banda Aceh) 95.42
2. Latitude 2 (Jayapura) -2.571944 dan Longitude 2 (Jayapura) 140.511389

Perhitungan :

Jarak = Arc Cos (Cos(5.516944 * π / 180) *

Cos(-2.571944 * π / 180) *

Cos((140.511389 * π / 180) – (95.42* π / 180)) +

Sin(5.516944 * π / 180) *

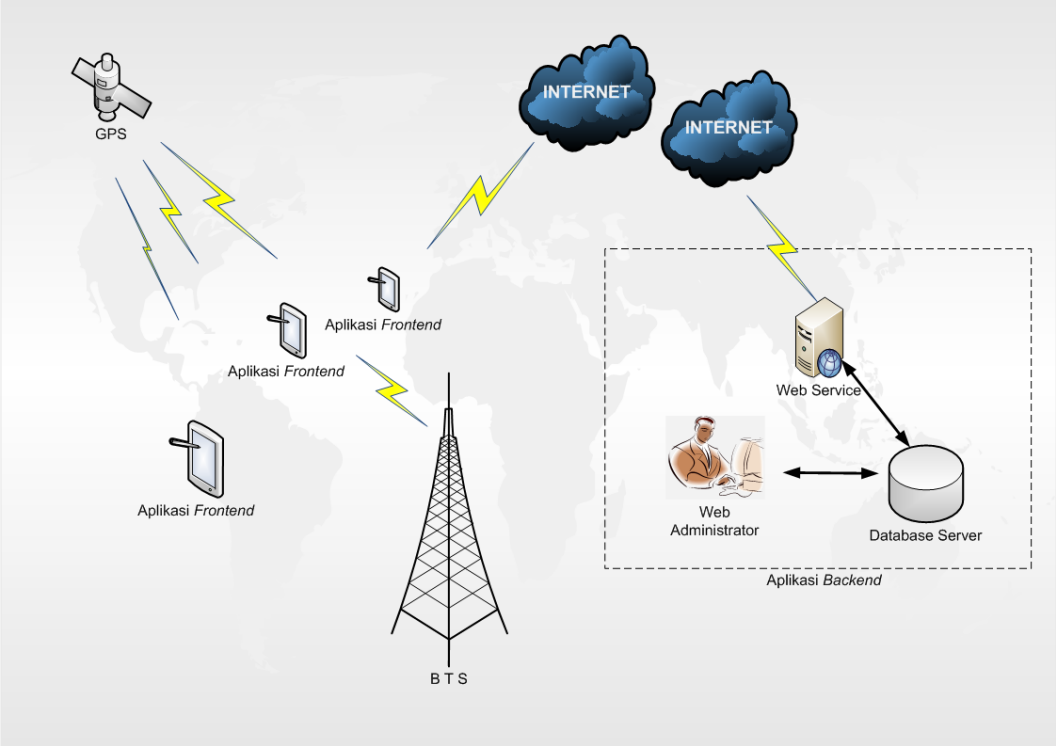
Sin(-2.571944 * π / 180)) * 6378.2

= Arc Cos (0.6976857) * 6.371

= 5093.849 KM

3.1.3 Analisis Arsitektur Sistem

Arsitektur fisik sistem mengacu kepada model arsitektur aplikasi *two tier*. Arsitektur fisik sistem terdiri dari tiga komponen utama yaitu *Client (Frontend)*, application *Server* (Web Admin dan Web Service), dan database *Server*. Arsitektur sistem dalam pembangunan aplikasi Fasilitas Umum ini dtitunjukan oleh gambar 3.1 dibawah ini.



Gambar 3.1. Arsitektur Sistem

GPS pada arsitektur ini merupakan aktor lain yang bergerak didalam stasiun radio luar yang berfungsi untuk memberitahu keberadaan posisi pengguna baik dengan dengan koordinat latitude maupun longitude.

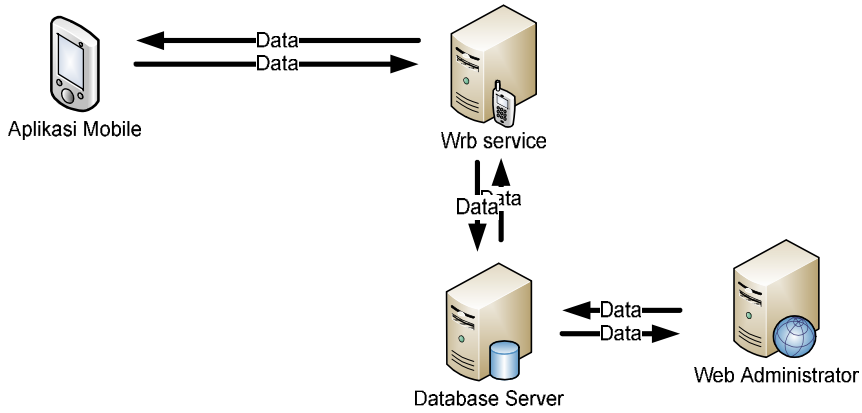
Aplikasi pada *frontend* yaitu Aplikasi Fasilitas Umum yang sudah terinstal pada perangkat *Mobile yang berplatform* Android. Bagian ini berfungsi sebagai

antarmuka antara pengguna dengan sistem. Pengembangan antarmuka sistem dibangun dengan menggunakan IDE Eclipse dan SDK android dengan bahasa pemrogramam java.

Backend merupakan aplikasi menengah (*middle application*) yang berfungsi sebagai penghubung antara aplikasi *frontend* dengan database *server*. Aplikasi ini terdiri dari tiga sub bagian utama yaitu *Web Administrator*, *Webservice* dan Database *server*. *Web Administrator* merupakan web yang berfungsi sebagai pengolah sumber basis data yang terdapat pada database *server* yang dioperasikan oleh admin. *Web Administrator* ini dibangun dengan menggunakan bahasa pemrograman PHP. *Webservice* merupakan web yang bertujuan untuk menguhungkan berbagai platform aplikasi yang dibangun dengan berbagai platform yang berbeda-beda (misalnya Java, C#, VB, Delphi, PHP,Android), sehingga seluruh aplikasi tersebut dapat saling berkomunikasi. Web service pada penelitian ini dibangun diatas pemrograman PHP yang dikombinasikan dengan JSON (*Javascript Object Notation*). Database *Server* merupakan aplikasi yang berfungsi menyimpan data-data yang digunakan oleh layanan (Admin). Pada bagian ini di implementasikan menggunakan MySQL.

Akses koneksi jaringan yang digunakan dari aplikasi *frontend* untuk berintegrasi dengan GPS atau *frontend* ke BTS melalui perangkat *Mobile* dapat menggunakan jaringan *Mobile* GPRS/EDGE/UMTS.

3.1.4 Analisis Alur Data Sistem



Gambar 3.2. Tampilan Cara Kerja Aplikasi

Analisis alur data sistem merupakan analisis yang berfungsi untuk menggambarkan secara rinci bagaimana sistem ini dapat bekerja. Analisis ini meliputi :

1. Aplikasi Mobile

   Aplikasi *mobile* dalam hal ini merupakan analisis fungsionalitas-fungsionalitas aplikasi untuk mengakses data dari database server melaui *web service*. Aplikasi ini dibangun diatas *platform* android dan bekerja dengan cara *memparsing* data dari web service yang bertipe JSON untuk diolah pada platform android.

2. Web service

   Web service pada penelitian ini berfungsi sebagai jembatan antara aplikasi *mobile platform* android dengan database *server*. Cara kerja *web service* ini yaitu dengan mengambil data dari database *server* dan kemudian

mengkonversikan data tersebut kedalam format pertukaran data JSON. Berikut ini gambar yang menunjukan bentuk array dari JSON.

```
{"menu": {
  "id": "file",
  "value": "File",
  "popup": {
    "menuitem": [
      {"value": "New", "onclick": "CreateNewDoc()"},
      {"value": "Open", "onclick": "OpenDoc()"},
      {"value": "Close", "onclick": "CloseDoc()"}
    ]
  }
}}
```

Gambar 3.3. Tampilan Bentuk JSON

3. Protokol HTTP

   Protokol HTTP merupakan protokol jaringan lapisan aplikasi yang digunakan untuk sistem informasi terdistribusi, kolaboratif, dan menggunakan hipermedia. Pada penelitian ini HTTP digunakan aplikasi *mobile* dan web administrator sebagai protokol yang dapat mendistribusikan data yang bersumber dari database *server*.

3.1.5 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional menggambarkan kebutuhan sistem yang menitikberatkan pada properti prilaku 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.

3.1.5.1 Analisis Kebutuhan Perangkat Keras

Berikut ini adalah spesifikasi perangkat keras minimum yang mendukung aplikasi fasilitas umum dengan baik, yaitu :

1. Perangkat Keras Aplikasi *Backend*

   Processor : Dengan Kecepatan Minimum 2.0 GHZ

   Memori : Minimum 512MB

   VGA : Dengan Kecepatan Minimum 32 MB

   Hardisk : Minimum Kapasitas 20 GB

   Network Interface Card

2. Perangkat Keras Aplikasi *Frontend*

   Resolusi : 240 x 320 pixels

   Memori : 100 MB

   CPU : 300 MHz

   WLAN : Wi-fi 802.11 b/g/n

   Jaringan 3G : UMTS

   GPS : A-GPS

3.1.5.2 Analisis Kebutuhan Perangkat Lunak

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

a. Sepesifikasi Minimum Perangkat Lunak

Berikut ini adalah spesifikasi minimum perangkat lunak yang digunakan dalam membangun dan mengimplementasikan aplikasi fasilitas umum ini.

1. Sistem Operasi Windows XP SP1
2. Aplikasi IDE Eclipse Galileo
3. JDK v 1.6 dan Android SDK windows
4. ADT v.8.0.0
5. Platform Android versi. 2.1 keatas.

b. Elemen Google Maps API

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

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).

3.1.5.3 Analisis Pengguna Sistem (*User*)

Analisis pengguna sistem dimaksudkan untuk mengetahui siapa saja aktor yang terlibat dalam menjalankan sistem. Sistem informasi aplikasi fasilitas umum

ini sendiri menggunakan *platform* teknologi bergerak dan web. Pengguna sistem dibagi atas dua bagian, yaitu :

1. Pengguna aplikasi fasilitas umum atau *enduser* yaitu para pengguna *platform* android. Dalam menggunakan sistem ini, pengguna diharuskan memiliki koneksi GRPS/EDGE/UMTS dan fasilitas GPS untuk bisa menggunakan aplikasi ini.
2. Admin sebagai pengelola konten secara keseluruhan memiliki fungsi-fungsi seperti menambah, mengubah dan menghapus konten informasi dalam menjalankan sistem.

Berikut ini karakteristik pengguna sistem yang ditunjukan oleh tabel dibawah ini.

Tabel 3.1 Karakteristik Pengguna Sistem

Pengguna	Usia	Hak Akses	Tingkat Pendidikan	Pengalaman
Enduser	Lebih dari 17 tahun	Mengakses data-data informasi fasilitas umum melaui aplikasi *frontend*	Min. SMA	Min. Membuka Aplikasi pada *mobile* Android
Admin	Lebih dari 21 tahun	Mengakses, mengelola aplikasi *backend*	Min. S1	Min. Mengatur dan memperbaiki data-data fasilitas umum di website

3.1.6 Analisis 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. Analisi kebutuhan fungsional ini meliputi analisi kebutuhan sistem, analisis kebutuhan data, spesifikasi sistem dan pemodelan sistem.

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

3.1.6.1 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. Menampilkan informasi mengenai tempat keberadaan fasilitas umum.
2. Memberikan informasi data fasilitas umum dengan lengkap dan rinci.
3. Menampilkan peta bandung secara keseluruhan dengan menunjukan posisi lokasi fasilitas umum.
4. Dapat melakukan Pencarian fasilitas umum dengan cepat.

3.1.6.2 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 Restoran

   Kriteria data restoran yang akan ditampilkan meliputi nama restoran, alamat, no telpon, fasilitas, perkiraan harga dan aneka jenis makanan yang dijual.

2. Data SPBU

   Kriteria data SPBU yang akan ditampilkan meliputi no registrasi SPBU, alamat, fasilitas, dan barang yang dijual.

3. Rumah Sakit

   Kriteria data rumah sakit yang akan ditampilkan meliputi nama rumah sakit, alamat, no telpon, fasilitas, deskripsinya.

4. Data ATM

   Khusus untuk ATM criteria data yang ditampilkan berupa alamat lokasinya, Pecahan Besaran Uang dan jenis ATM.

3.1.6.3 Spesifikasi Sistem

Adapun spesifikasi sistem yang dibutuhkan antara lain :

1. Sistem dikembangkan dengan model arsitektur *two tier*. Aplikasi Fasilitas Umum dikembangkan disisi *client* atau *frontend* sebagai aplikasi *On*

Device Portal (ODP). Sedangkan disisi *server* atau *backend* akan dikembangkan aplikasi web *Content Managemen System* (CMS).

2. Konektivitas antara aplikasi Fasilitas Umum pada *platform* android ini dengan aplikasi *backend* di s*erver* menggunakan koneksi protocol HTTP dengan memanfaatkan jaringan GPRS/EDGE/UMTS.
3. Sistem informasi yang disediakan dibatasi pada informasi fasilitas umum dan lokasi.
4. Spesifikasi sistem disisi *frontend* yaitu :
    a. Perangkat yang digunakan harus bisa digunakan dimana saja dan kapan saja serta memiliki akses internet.
    b. Aplikasi ini menyediakan informasi tentang lokasi-lokasi yang berhubungan dengan fasilitas umum, alamat lokasi tersebut, lokasi pada peta menggunakan *google maps*, dan lokasi *realtime* pengguna dengan perangkat GPS yang terintegrasi.
    c. Aplikasi ini memiliki ukuran yang minimum sehingga mampu disimpan dalam perangkat seluler yang memiliki keterbatasan.
5. 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.

3.1.6.4 Pemodelan Sistem

Pada tahap ini dilakukan pemodelan sistem untuk menentukan fungsi-fungsi yang dapat dilakukan oleh sistem aplikasi serta menentukan kelas yang

dibutuhkan untuk realisasi fungsi-fungsi sistem yang telah dianalisis sebelumnya dan mendeskripsikannya kedalam bentuk diagram.

a. *Use Case* Diagram

Diagram *Use Case* merupakan bagian tertinggi dari fungsionalitas yang dimiliki sistem yang akan menggambarkan bagaimana seseorang atau aktor akan menggunakan dan memanfaatkan sistem. Diagram ini juga mendeskripsikan apa yang akan dilakukan oleh sistem. *Use Case* terdiri dari tiga bagian yaitu identifikasi aktor, identifikasi *Use Case* dan skenario *Use Case*.

1. Identifikasi Aktor

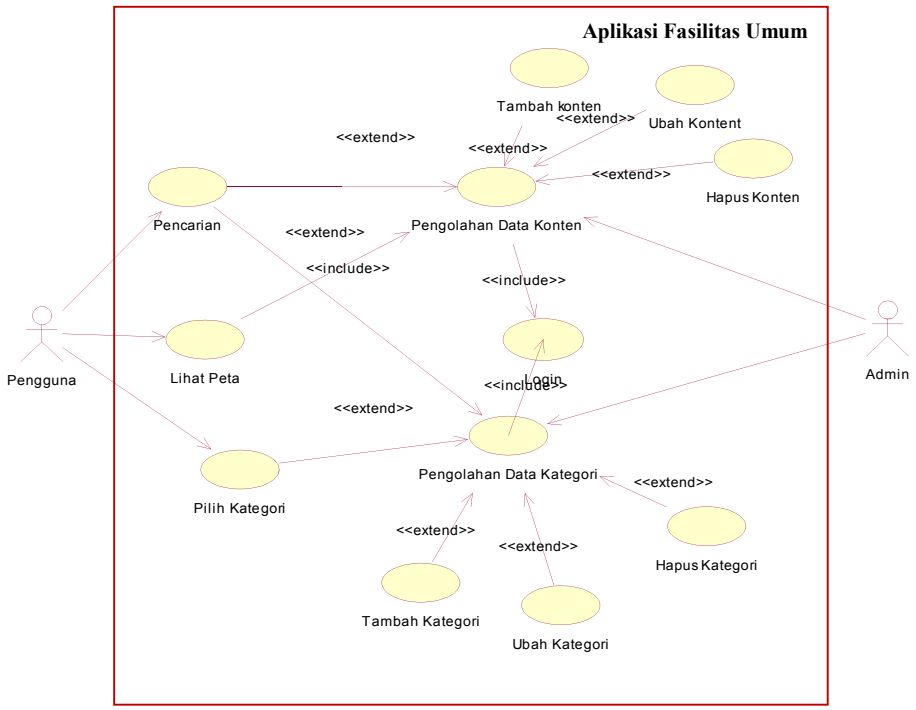
Aktor yang berperan dalam menjalankan sistem dapat dilihat pada tabel dibawah ini.

Tabel 3.2 Aktor Sistem

No	Aktor	Deskripsi
P-01	Pengguna	Merupakan aktor yang menggunakan aplikasi Fasilitas Umum untuk informasi mengenai suatu lokasi dan letaknya
P-02	Admin	Merupakan aktor yang akan mengelola konten informasi. Admin akan melakukan administrasi melalui CMS berbasi web.

Gambar 3.4 dibawah ini menunjukan interaksi antara aktor dan sistem. Diman aktor terdiri dari dua pengguna dan sedangkan sistem terdiri dari aplikasi *frontend* dan *backend*. Aktor pertama yang berperan yaitu pengguna aplikasi *frontend* (*Android Client*), yang berinteraksi dengan aplikasi *frontend* yang bergerak pada

mobile. Sedangkan aktor kedua yang berperan adalah admin yang berinteraksi pada aplikasi *backend* pada CMS berbasis web.



Gambar 3.4. *Use Case* Aplikasi Fasilitas Umum

2. Identifikasi *Use Case*

Identifikasi *use case* yang terdapat dalam sistem dapat dilihat pada tabel 3.3 dibawah ini.

Tabel 3.3 Deskripsi *Use Case* Pengguna

No	*Use Case*	Deskripsi
UC-01	Pilih Kategori	Fungsionalitas untuk memilih dan melakukan penjelasan untuk kategori-kategori yang terdiri dari Restoran, ATM, SPBU dan Rumah Sakit
UC-02	Lihat Peta	Fungsionalitas untuk melihat posisi dan lokasi tempat-tempat fasilitas umum pada

		peta Google Map.
UC-03	Pencarian	Fungsionalitas untuk melakukan Pencarian suatu lokasi tertentu berdasarkan kata kunci.
UC-04	Login	Fungsionalitas untuk proses otentifikasi pengguna web CMS
UC-05	Pengolahan Data Kategori	Fungsionalitas untuk melihat manipulasi data kategori
UC-06	Tambah Kategori	Fungsionalitas untuk menambah kategori
UC-07	Ubah Kategori	Fungsionalitas untuk mengubah kategori
UC-08	Hapus Kategori	Fungsionalitas untuk menghapus kategori
UC-09	Pengolahan Data Konten	Fungsionalitas untuk melihat manipulasi data konten
UC-10	Tambah Konten	Fungsionalitas untuk menambah konten
UC-11	Ubah Konten	Fungsionalitas untuk mengubah konten
UC-12	Hapus Konten	Fungsionalitas untuk menghapus konten

3. Skenario *Use Case*

Skenario proses-proses yang terdapat didalam *Use Case* dapat dilihat pada tabel-tabel dibawah ini.

Tabel 3.4 Skenario *Use Case* Pilih Kategori

Identifikasi	
Nama *Use Case*	Pilih Kategori
Aktor	Pengguna
Tujuan	Memilih pilihan menu
Keadaan Awal	Sistem menampilkan menu pilihan
Skenario Utama	
Aksi Aktor	**Reaksi sistem**
1. Memilih menu (Restoran/ ATM/Rumah Sakit/SPBU) yang ditampilkan oleh sistem.	

	2. Jika memilih menu restoran maka menampilkan menu list restoran/ jika memilih menu ATM maka menampilkan menu list ATM/ jika memilih menu rumah sakit maka menampilkan menu list rumah sakit/ jika memilih menu SPBU maka menampilkan menu list SPBU.
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi sistem**
	1. Menampilkan pesan bahwa proses gagal dilakukan (”Cek Koneksi Jaringan”) atau (”Nyalakan GPS”)
2. Menekan tombol ”OK”	
	3. Sistem akan *terminate*.
Kondisi Akhir	Menampilkan informasi kategori sesuai yang dipilih

Tabel 3.5 Skenario *Use Case* Lihat Peta

Identifikasi	
Nama *Use Case*	Lihat Peta
Aktor	Pengguna
Tujuan	Melihat peta lokasi
Keadaan Awal	Sistem menampilkan halaman detail_content
Skenario Utama	
Aksi Aktor	**Reaksi sistem**
1. Menekan tombol view_map yang ada pada halaman detail_conten	
	2.Menampilkan map lokasi fasilitas umum sesuai dengan konten yang dipilih.
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi sistem**
	1. Menampilkan pesan bahwa proses gagal dilakukan (”Cek Koneksi Jaringan”) atau (”Nyalakan GPS”)
2. Menekan tombol ”OK”	
	3. Sistem akan *terminate*.
Kondisi Akhir	Menampilkan map yang sesuai diminta.

Tabel 3.6 Skenario *Use Case* Pencarian

Identifikasi	
Nama *Use Case*	Pencarian
Aktor	Pengguna
Tujuan	Mencari informasi fasilitas umum
Keadaan Awal	Sistem menampilkan halaman utama

Skenario Utama	
Aksi Aktor	**Reaksi sistem**
1.Memasukan kata kunci pada pada komponen text	
	2. Menampilkan konten informasi sesuai dengan kata kunci yang diminta
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi sistem**
	1. Menampilkan pesan bahwa proses gagal dilakukan (”Data tidak ditemukan”)
Kondisi Akhir	Menampilkan konten informasi sesuai kata kunci

Tabel 3.7 Skenario *Use Case* Login

Identifikasi	
Nama *Use Case*	Login
Aktor	Admin
Tujuan	Masuk kedalam sistem
Keadaan Awal	Sistem menampilkan halaman login admin
Skenario Utama	
Aksi Aktor	**Reaksi sistem**
1.Mengisi username dan password pada halaman web login admin	
	2. mengautentikasi username dan password yang diinputkan pada basis data
	3. bila cocok menampilkan halaman web menu utama
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi sistem**
	1.Menampilkan pesan bahwa username dan password tidak benar
2. Mengisi kembali username dan password pada halaman web login admin	
	2. mengautentikasi username dan password yang diinputkan pada basis data
	3. bila cocok menampilkan halaman web menu utama
Kondisi Akhir	Halaman menu utama ditampilkan

Tabel 3.8 Skenario *Use Case* Pengolahan Kategori

Identifikasi	
Nama *Use Case*	Tambah Kategori
Aktor	Admin
Tujuan	Melihat Fungsionalitas Data Kategori
Keadaan Awal	Sistem menampilkan halaman Menu Utama
Skenario Utama	
Aksi Aktor	**Reaksi sistem**
1.Memilih link Pengolahan Kategori pada pilihan menu	
	2.Menampilkan halaman form pengolahan kategori
Kondisi Akhir	Data kategori bertambah sesuai yang diinginkan

Tabel 3.9 Skenario *Use Case* Tambah Kategori

Identifikasi	
Nama *Use Case*	Tambah Kategori
Aktor	Admin
Tujuan	Menambah data kategori
Keadaan Awal	Sistem menampilkan halaman pengolahan kategori
Skenario Utama	
Aksi Aktor	**Reaksi sistem**
1.Memilih link *Add* pada pilihan menu	
	2.Menampilkan halaman form pengisian tambah kategori
3.Mengisi field-field pada form pengisian tambah kategori	
	4.Melakukan proses tambah data pada basis data
	5.Menyimpan hasil tambah data
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi sistem**
	1.Menampilkan pesan bahwa belum diisi dengan lengkap
2.Melengkapi field-field pada form pengisian tambah kategori	
	3.Melakukan proses tambah data pada basis data
	4.Menyimpan hasil tambah data
Kondisi Akhir	Data kategori bertambah sesuai yang diinginkan

Tabel 3.10 Skenario *Use Case* Ubah Kategori

Identifikasi	
Nama *Use Case*	Ubah Kategori
Aktor	Admin
Tujuan	Mengubah data kategori
Keadaan Awal	Sistem menampilkan halaman pengolahan kategori
Skenario Utama	
Aksi Aktor	**Reaksi sistem**
1.Memilih link *Edit* pada pilihan menu	
	2.Menampilkan halaman form ubah kategori
3.Mengisi field-field pada form ubah kategori	
	4.Melakukan proses ubah data pada basis data
	5.Menyimpan hasil perubahan data
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi sistem**
-	-
Kondisi Akhir	Perubahan data sesuai yang diinginkan

Tabel 3.11 Skenario *Use Case* Hapus Kategori

Identifikasi	
Nama *Use Case*	Hapus Kategori
Aktor	Admin
Tujuan	Menghapus data kategori
Keadaan Awal	Sistem menampilkan halaman pengolahan kategori fasilitas umum
Skenario Utama	
Aksi Aktor	**Reaksi sistem**
1.Memilih link *Delete* pada pilihan menu	
	2.Menampilkan halaman form hapus kategori
	3. Menampilkan pesan konfirmasi hapus kategori
4.Menjawab konfirmasi dengan menekan tombol ”OK”	
	5.Menghapus kategori
	6.Menyimpan hasil perubahan data
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi sistem**
-	-
Kondisi Akhir	

Tabel 3.12 Skenario *Use Case* Pengolahan Konten

Identifikasi	
Nama *Use Case*	Pengolahan Konten
Aktor	Admin
Tujuan	Melihat Fungsionalitas Data Konten
Keadaan Awal	Sistem menampilkan halaman Menu Utama
Skenario Utama	
Aksi Aktor	**Reaksi sistem**
1.Memilih link Pengolahan konten pada pilihan menu	
	2.Menampilkan halaman form pengolahan konten
Kondisi Akhir	Data konten bertambah sesuai yang diinginkan

Tabel 3.13 Skenario *Use Case* Tambah Konten

Identifikasi	
Nama *Use Case*	Tambah Konten
Aktor	Admin
Tujuan	Menambah data konten fasilitas umum
Keadaan Awal	Sistem menampilkan halaman pengolahan konten fasilitas umum
Skenario Utama	
Aksi Aktor	**Reaksi sistem**
1.Memilih link *Add* pada pilihan menu	
	2.Menampilkan halaman form isian tambah konten
3.Mengisi field-field pada form tambah konten	
	4.Melakukan proses tambah data pada basis data
	5.Menyimpan hasil tambah data
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi sistem**
	1.Menampilkan pesan bahwa belum diisi dengan lengkap
3.Melengkapi field-field pada form tambah konten	
	4.Melakukan proses tambah data pada basis data
	5.Menyimpan hasil tambah data
Kondisi Akhir	Data konten bertambah sesuai yang diinginkan

Tabel 3.14 Skenario *Use Case* Ubah Konten

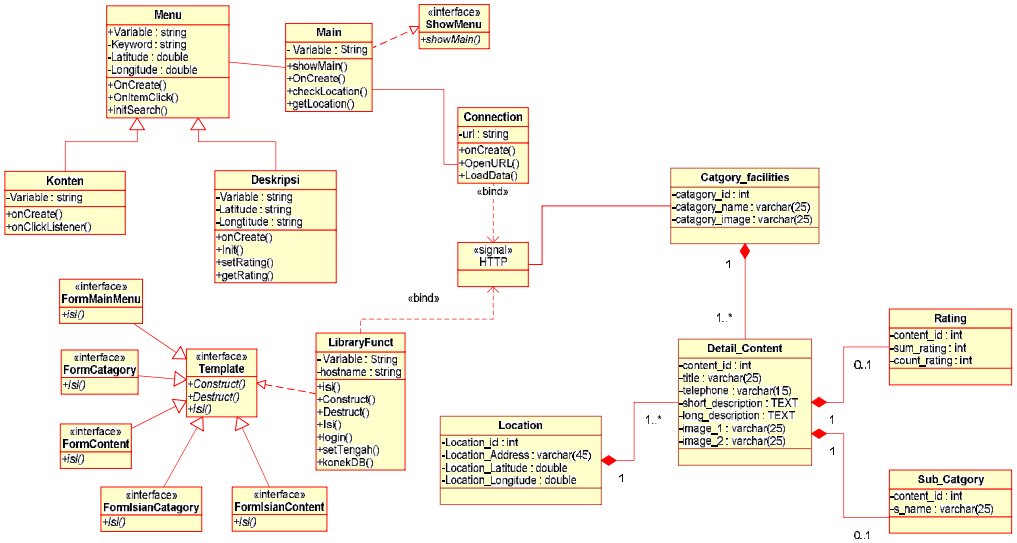
Identifikasi	
Nama *Use Case*	Ubah Konten
Aktor	Admin
Tujuan	Mengubah data konten fasilitas umum
Keadaan Awal	Sistem menampilkan halaman pengolahan konten fasilitas umum
Skenario Utama	
Aksi Aktor	**Reaksi sistem**
1.Memilih link *Edit* pada pilihan menu	
	2.Menampilkan halaman form ubah konten
3.Mengisi field-field pada form ubah konten	
	4.Melakukan proses ubah data pada basis data
	5.Menyimpan hasil perubahan data
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi sistem**
-	-
Kondisi Akhir	Perubahan data sesuai yang diinginkan

Tabel 3.15 Skenario *Use Case* Hapus Konten

Identifikasi	
Nama *Use Case*	Hapus Konten
Aktor	Admin
Tujuan	Menghapus data konten
Keadaan Awal	Sistem menampilkan halaman konten fasilitas umum
Skenario Utama	
Aksi Aktor	**Reaksi sistem**
1.Memilih link *Delete* pada pilihan menu	
	2.Menampilkan halaman form hapus konten
	3. Menampilkan pesan konfirmasi hapus konten
4.Menjawab konfirmasi dengan menekan tombol ”OK”	
	5.Menghapus konten
	6.Menyimpan hasil perubahan data
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi sistem**
-	-
Kondisi Akhir	

b. *Class Diagram*

Class Diagram adalah sebuah spesifikasi dari fungsionalitas yang menghasilkan objek dan merupakan inti dari pengembangan aplikasi ini. Diagram ini juga merupakan gambaran keadaan atribut atau properti dari sistem yang melakukan manipulasi fungsi atau metode. Berikut ini gambaran *class* diagram dari sistem Aplikasi Fasilitas Umum.



Gambar 3.5. *Class Diagram* Aplikasi Fasilitas Umum

Tabel 3.16 Deskripsi *Class Diagram*

Class	Jenis Kelas	Deskripsi
Main	Control	*Class* utama yang berisikan method-method yang berhubungan dengan *class* lain dan berfungsi sebagai *controller*
Menu	Control	*Class* yang berisikan method-method yang berfungsi mengatur dan mengontrol *variable* data

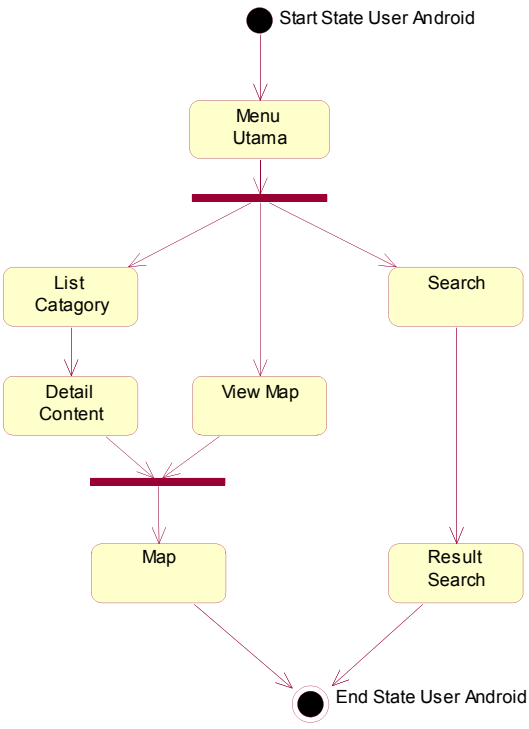
ShowMenu	Interface	*Class* menu yang berisikan tampilan menu dari aplikasi
Konten	Control	*Class* yang berisikan method-method yang berhubungan dengan data jenis fasilitas umum
Deskripsi	Control	*Class* yang berisikan method-method yang berhubungan dengan data rinci fasilitas umum
Connection	Control	*Class* yang berisikan method-method yang menangani koneksi ke *Server*
HTTP	Control	Merupakan *class* yang berfungsi sebagai jembatan data antara kelas satu dengan kelas lainnya
FormMainMenu	Interface	*Class* yang berisikan tampilan pilihan menu admin.
FormContent	Interface	*Class* yang berisikan tampilan data detail konten fasilitas
FormCatagory	Interface	*Class* yang berisikan tampilan data fasilitas
FormIsianContent	Interface	*Class* yang berisikan tampilan data field detail konten fasilitas
FormIsianCatagory	Interface	*Class* yang berisikan tampilan data field detail kategori fasilitas
Template	Interface	*Class* yang berisikan method-method yang berfungsi mengontrol tampilan dari web admin
LibraryFunct	Control	*Class* yang berisikan method-method untuk mengontrol *class* interface dan *class* entity.

Catagory_facilities	Entity	*Class* yang berisikan data kategori.
Detail_Content	Entity	*Class* yang berisikan data konten.
Location	Entity	*Class* yang berisikan data lokasi.
Rating	Entity	*Class* yang berisikan data rating.
Sub_category	Entity	*Class* yang berisikan data sub_catagory.

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.

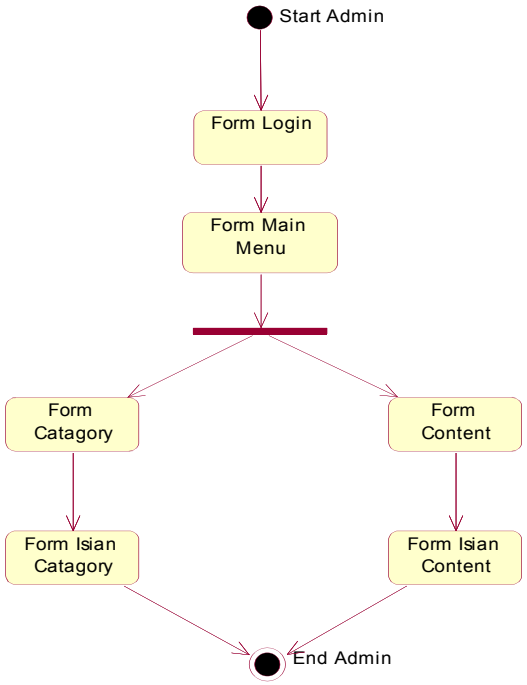
1. *State Diagram* Aplikasi *Frontend*



Gambar 3.6. *State Diagram* Aplikasi *Frontend*

Gambar diatas 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 bersesuaian, melakukan Pencarian, dan melihat tampilan map. Selanjutkan apabila pengguna memilih menu dari daftar kategori maka akan muncul halaman deskripsi konten rinci dari kategori pilihannya. Pengguna dapat memilih konten tertentu atau kembali ke menu utama. Apabila pengguna memilih salah satu konten tertentu, maka keadaan berikutnya adalah halaman deskripsi konten akan ditampilkan pada layar. Pengguna dapat memilih map atau kembali ke halaman sebelumnya. Apabila pengguna memilih map maka keadan selanjutnya adalah tampilan peta mengenai lokasi konten tersebut. Apabila pengguna memilih lihat peta secara langsung, maka selanjutnya pada layar akan ditampilkan map dengan seluruh *marker* lokasi, dan apabila pengguna memilih menu Pencarian maka akn muncul tampilan hasil Pencarian.

2. *State Diagram* Aplikasi *Backend*



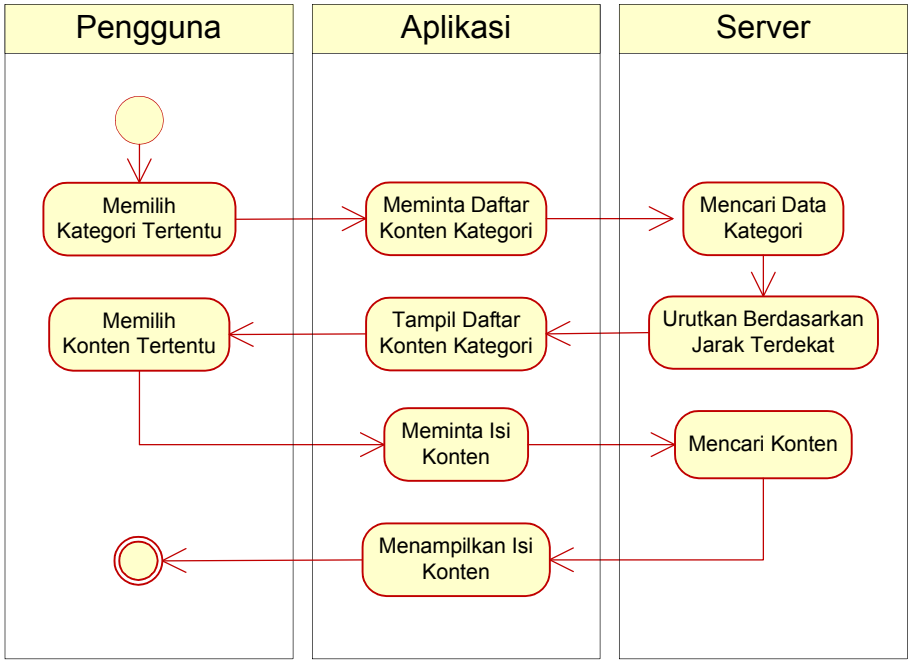
Gambar 3.7. *State Diagram* Aplikasi *Backend*

Berbeda dengan state sebelumnya, Gambar 3.5 diatas seluruh aktifitas yang dilakukan oleh admin. Dengan keadaan awal ditampilkan halaman web login, kemudian ketika berhasil maka akan ditampilkan halaman main menu. Kemudian admin dapat memilh pilihan menu *catagory* atau pilihan menu *conten* informasi. Apabila memilih menu catagory maka akan ditampilkan halaman pengolahannya, lalu apabila memili ADD/EDIT maka akan ditampilkan halaman isian form catagory. Begitu pula sebaliknya apabila admin memilih menu content maka akan ditampilkan form oengolahan conten dan ketika memilih ADD/EDIT/DELETE maka akan ditampilkan halaman isian conten informasinnya.

d. *Activity Diagram*

Diagram activity merupakan bagian dari penggambaran sistem secara fungsional menjelaskan proses-proses logika atau fungsi yang terimplementasi oleh kode program. *Activity Diagram* memodelkan event-event yang terjadi didalam suatu *Use Case* dan digunakan untuk pemodelan aspek dinamis dari sistem.

1. *Activity Diagram* pilih kategori

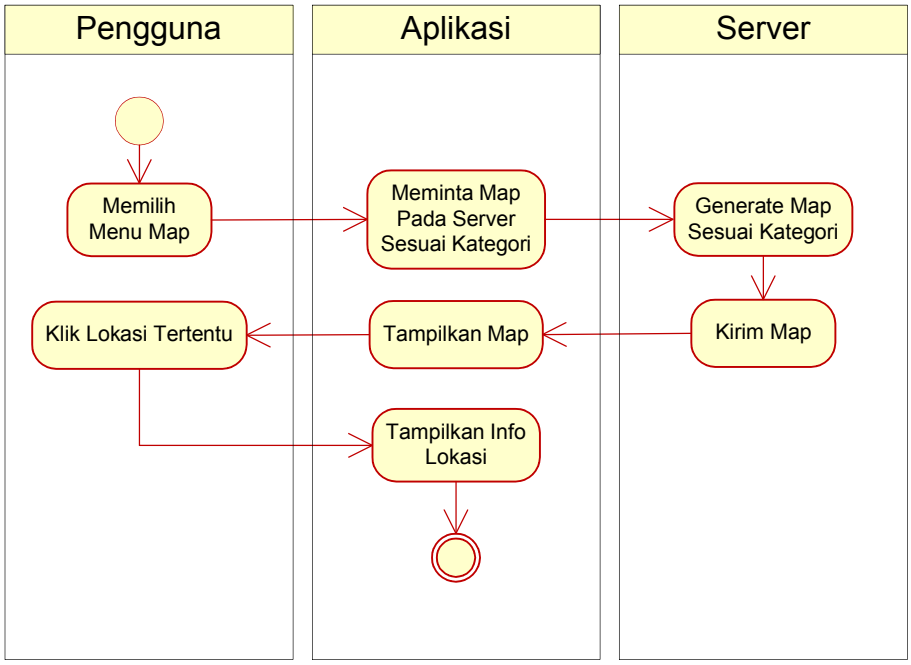


Gambar 3.8. *Activity Diagram* Pilih Kategori

Gambar diatas memperlihatkan aktivitas memilih kategori oleh pengguna. Pengguna memilih kategori tertentu, selanjutnya aplikasi akan memiinta pada *Server*, daftar konten dari kategori yang bersesuaian. Kemudian *Server* akan mencari kategori yang bersesuaian pada basis data yang ada, menyortirnya berdasarkan jarak terdekat dan mengirimkan daftar konten yang telah disortir ke aplikasi dan kemudian aplikasi akan

menampilkan pada pengguna. Setelah itu pengguna dapat memilih konten tertentu untuk mendapatkan hasil informasi. Aplikasi akan meminta data konten yang terpilih pada *Server*, kemudian *Server* akan mengirimkan kembali ke aplikasi untuk ditampilkan kepada pengguna.

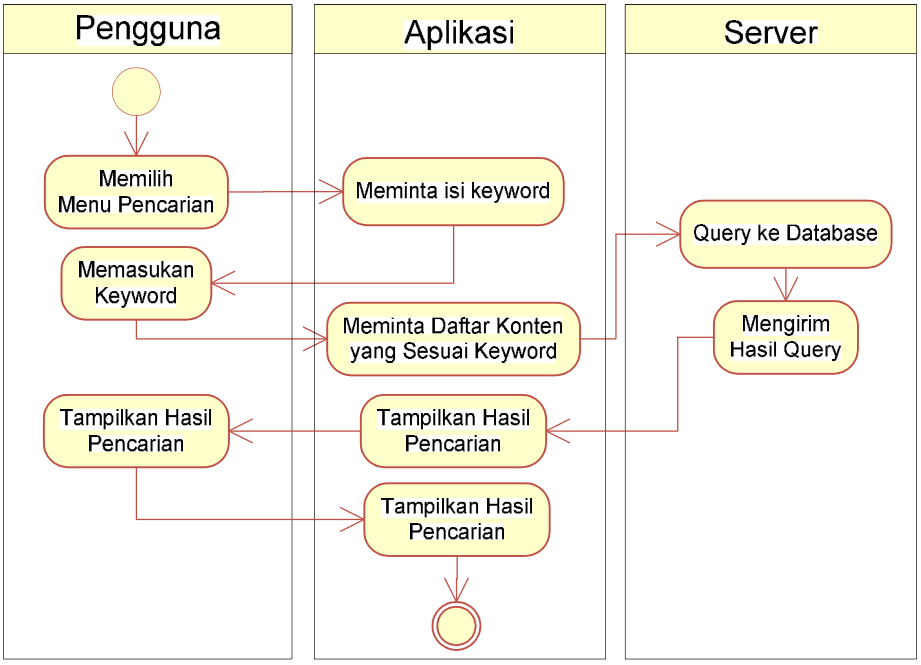
2. *Activity Diagram* Lihat Peta



Gambar 3.9. *Activity Diagram* Lihat Peta

Gambar diatas memperlihatkan aktivitas melihat peta oleh pengguna setelah pengguna memilih menu map, aplikasi akan meminta map pada *Server* sesuai kategori yang sedang aktif. Kemudian, *Server* akan meminta data koordinat pada basis data, meng*enerate* map sesuai dengan data-data koordinat kategori tersebut dengan menggunakan layanan Google Map. Setelah itu, map tersebut akan ditampilkan oleh aplikasi kepada pengguna. Kemudian apabila pengguna mengetik satu lokasi tertentu, maka aplikasi akan menampilkan informasi lokasi tersebut.

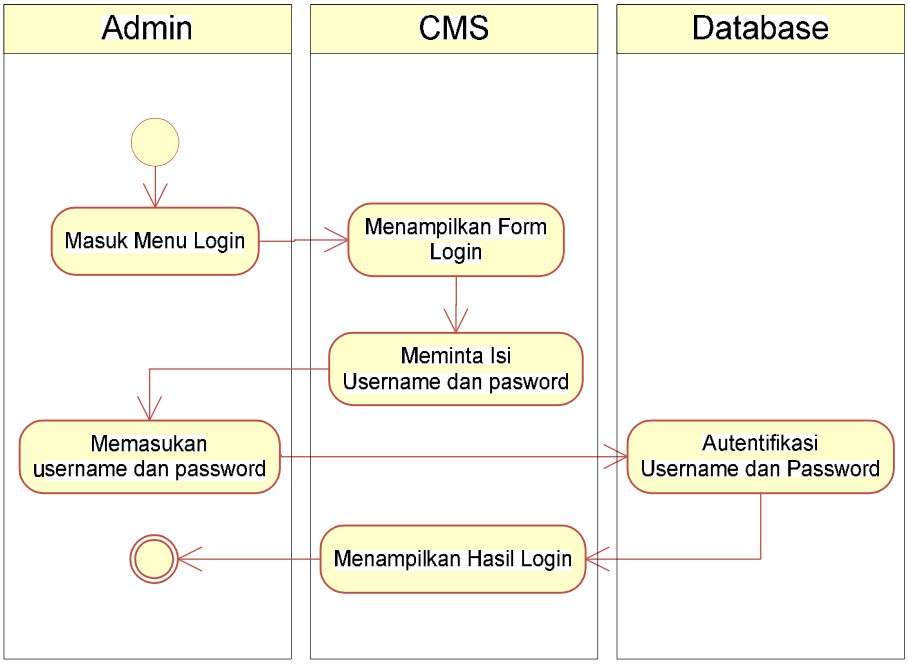
3. *Activity Diagram* Pencarian



Gambar 3.10. ***Activity Diagram Pencarian***

Gambar diatas memperlihatkan aktivitas Pencarian lokasi. Pengguna memilih menu Pencarian, kemudian pengguna memasukkkan kata kunci, aplikasi meminta daftar konten yang membuat kata kunci, *Server* akan meminta daftar konten pada database yang memuat kata kunci, mengirimkan hasil Pencarian, dan aplikasi akan menampilkan daftar hasil Pencarian. Setelah itu pengguna dapat memilih salah satu konten dari daftar hasil Pencarian untuk ditampilkan informasinya oleh aplikasi.

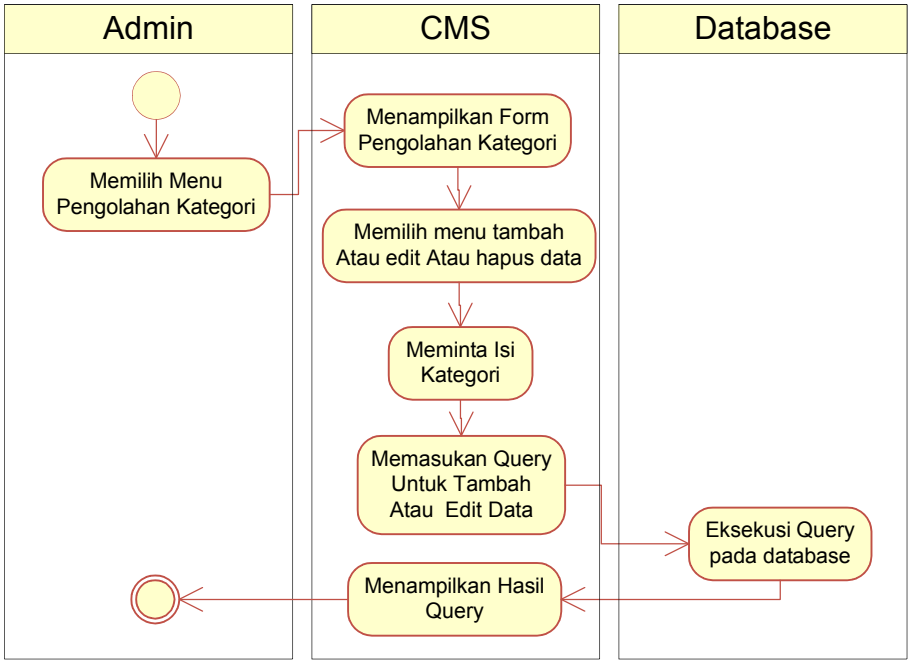
4. *Activity Diagram* login



Gambar 3.11. *Activity Diagram* login

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

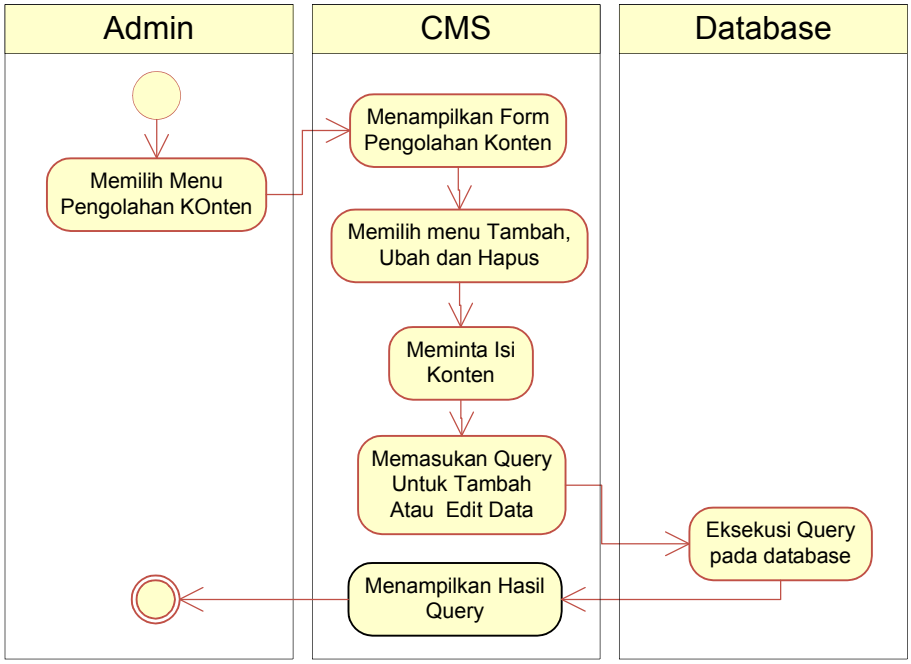
5. *Activity Diagram* pengolahan kategori



Gambar 3.12. *Activity Diagram* pengolahan kategori

Gambar diatas memperlihatkan aktivitas Pencarian pengolahan kategori. Admin memilih daftar menu pengolahan kategori, kemudian CMS akan menampilkan form pengolahan 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.

6. *Activity Diagram* pengolahan konten



Gambar 3.13. ***Activity Diagram Pencarian***

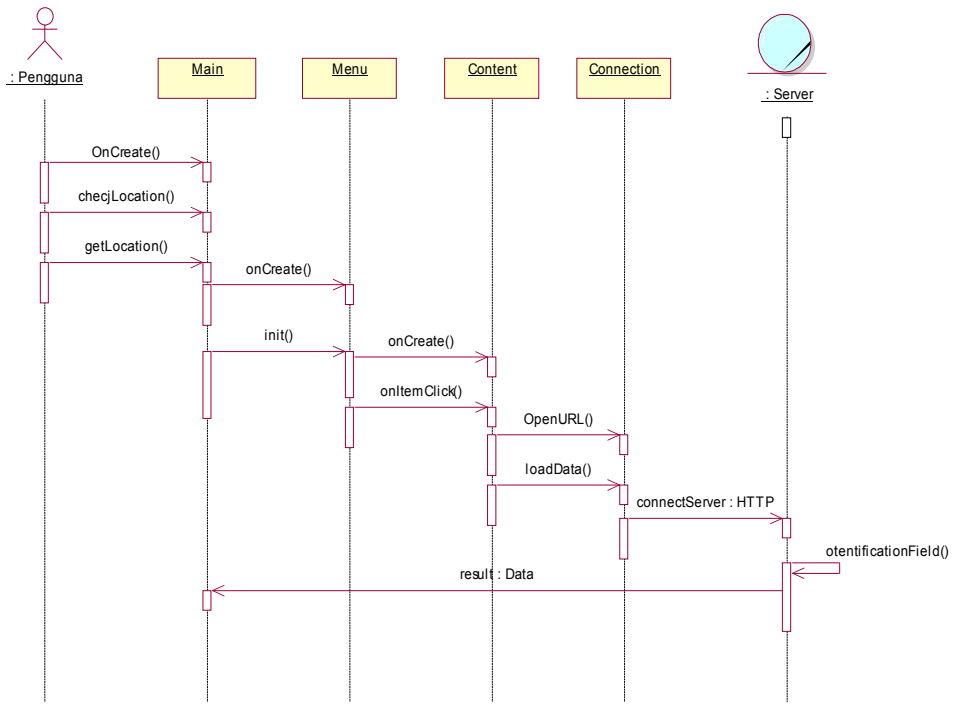
Gambar diatas memperlihatkan aktivitas Pencarian pengolahan Konten. Admin memilih daftar menu pengolahan konten, kemudian CMS akan menampilkan form pengolahan konten. Lalu admin dapat memilih sublink untuk tambah atau ubah atau hapus kategori lalu CMS akan menampilkan Form isian data apabila tambah atau hapus kategori 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*

Pada tahap ini *Sequence Diagram* menjelaskan secara detil urutan proses yang dilakukan sistem untuk mencapai tujuan dari *Use Case*. interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang

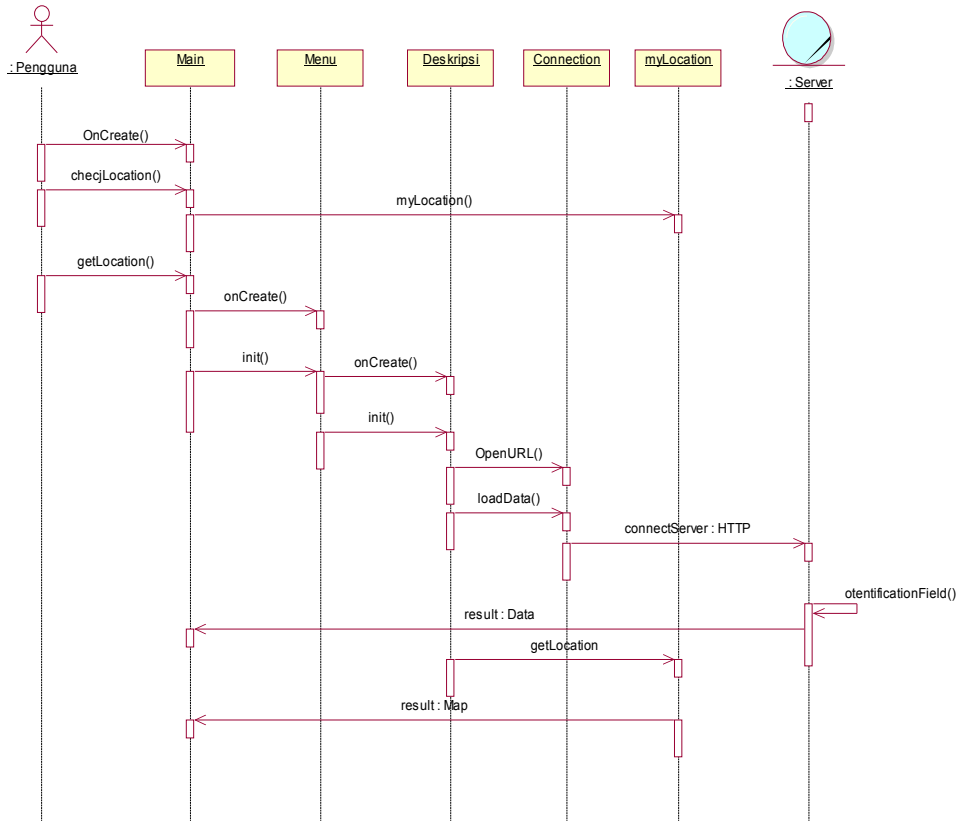
diperlukan oleh masing-masing operasi.

1. *Sequence* Pilih Kategori



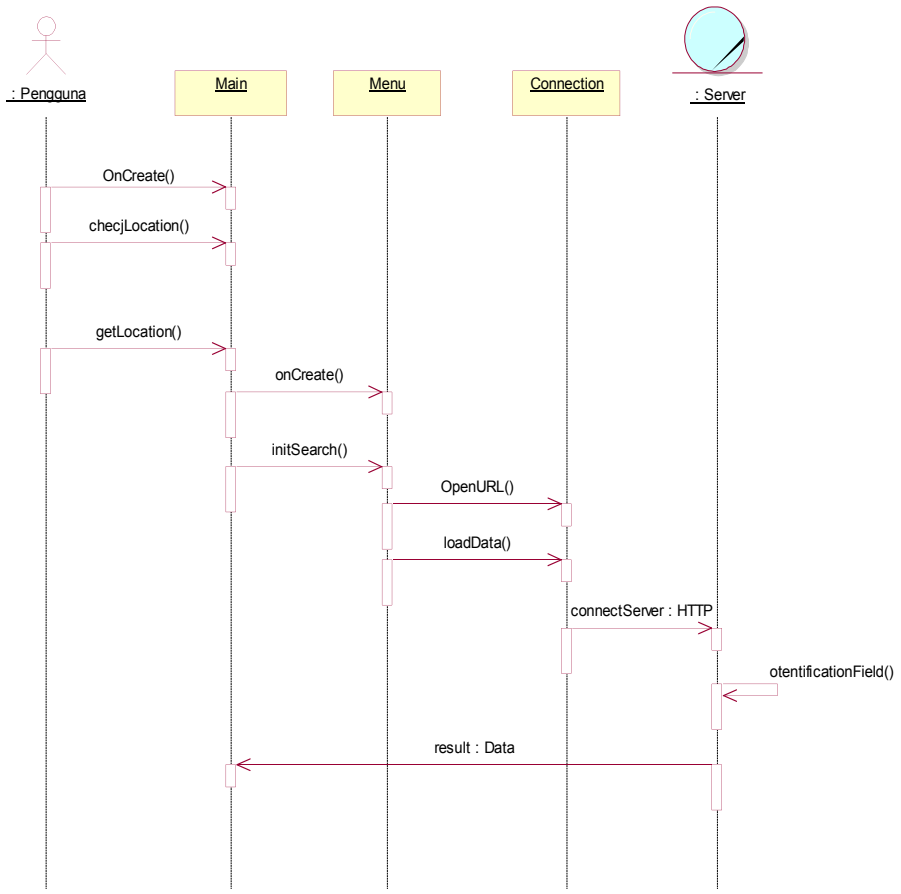
Gambar 3.14. *Sequence Diagram Chooce Catagory*

2. *Sequence* Lihat Peta



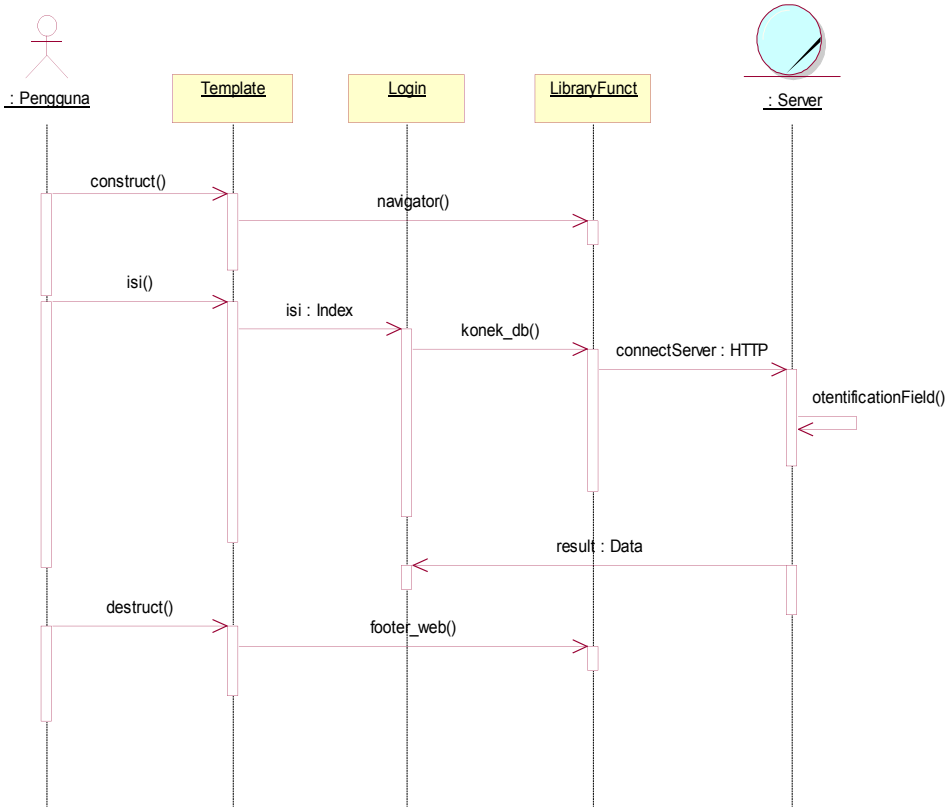
Gambar 3.15. *Sequence Diagram* Lihat Peta

3. *Sequence* Pencarian



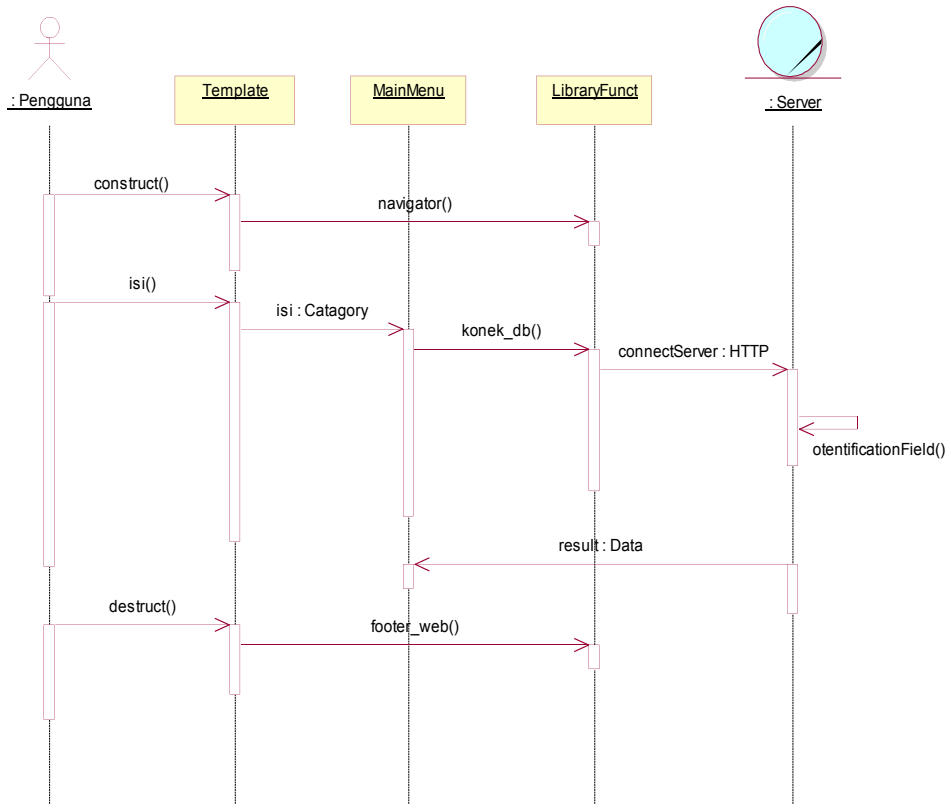
Gambar 3.16. *Sequence Diagram* Pencarian

4. *Sequence* Login



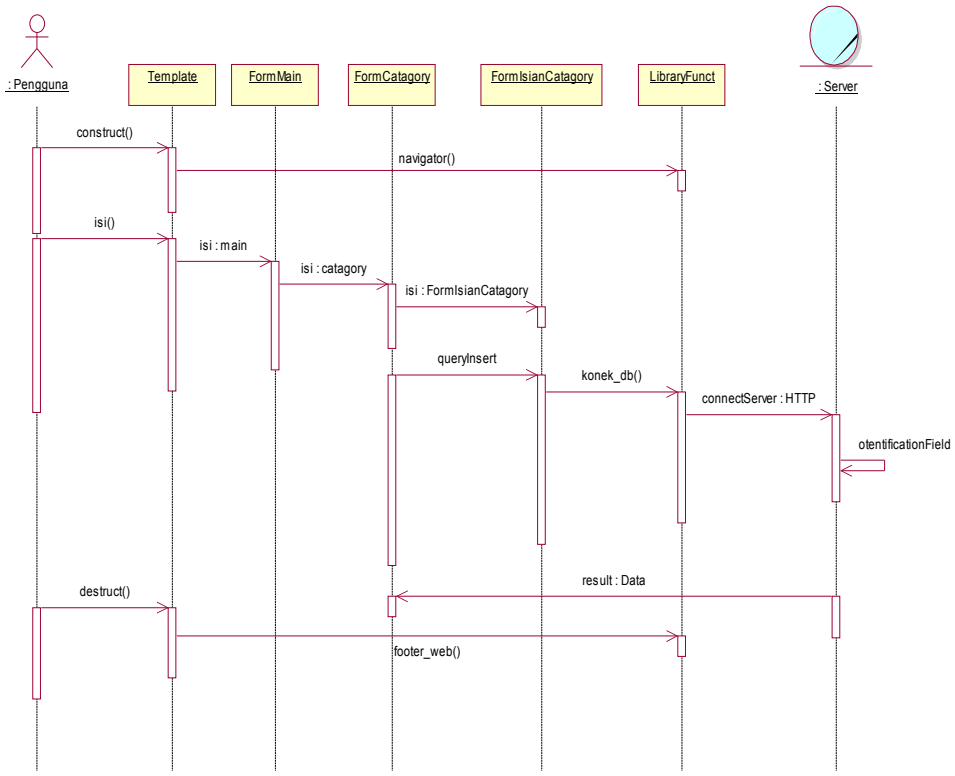
Gambar 3.17. *Sequence Diagram Login*

5. *Sequence* Pengolahan data Kategori



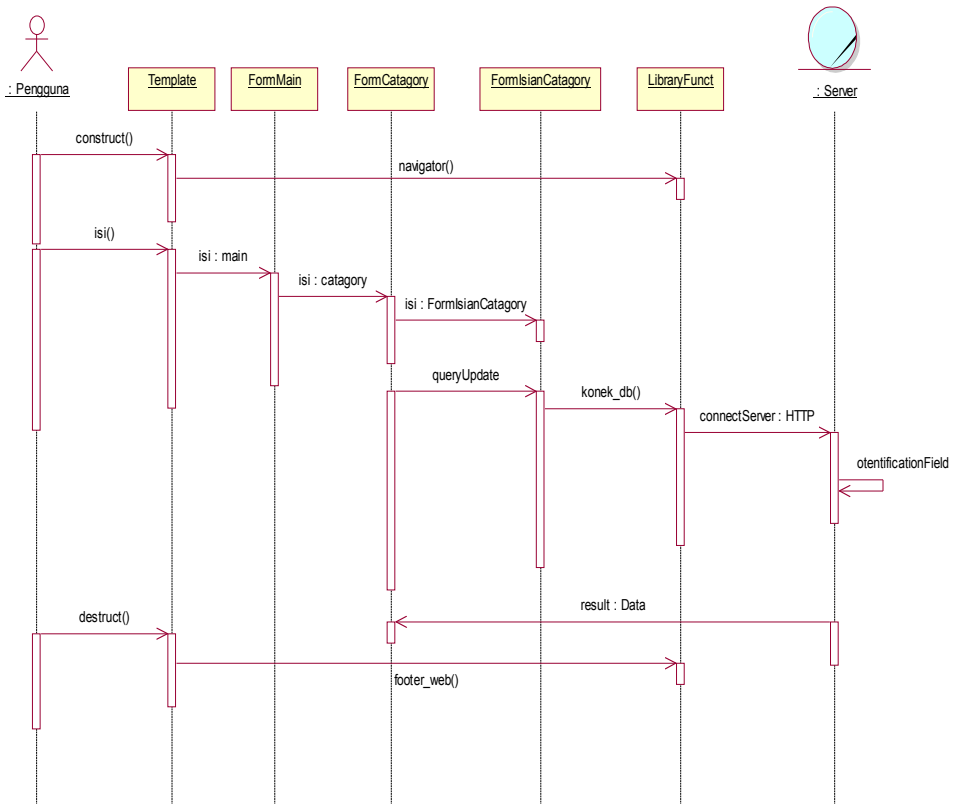
Gambar 3.18. ***Sequence Diagram Pengolahan Data Kategori***

6. *Sequence* Tambah Kategori



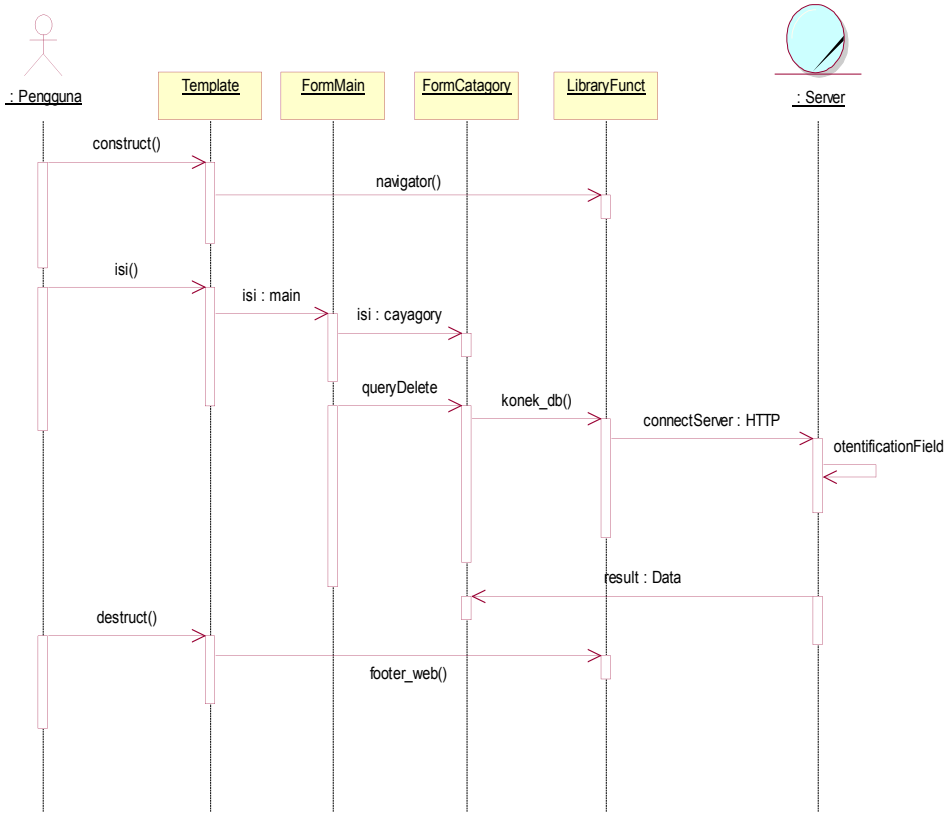
Gambar 3.19. *Sequence Diagram* Tambah Kategori

7. *Sequence* Ubah Kategori



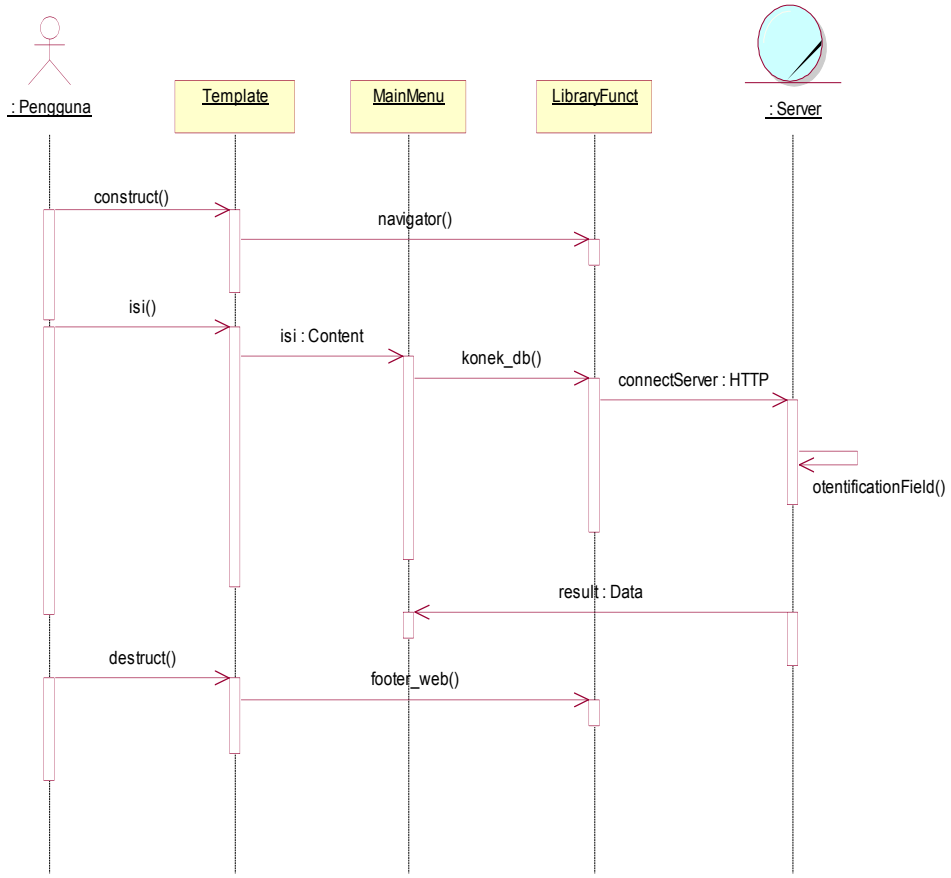
Gambar 3.20. *Sequence Diagram* Ubah Kategori

8. *Sequence* Hapus Kategori



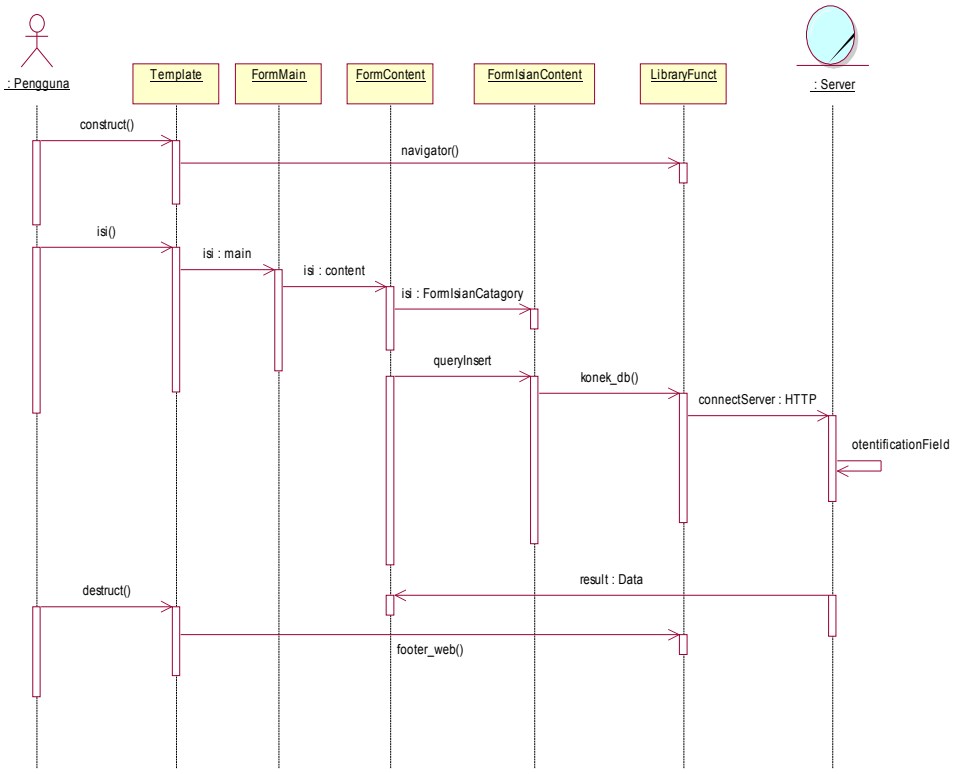
Gambar 3.21. *Sequence Diagram* Hapus Kategori

9. *Sequence* Pengolahan Data Konten



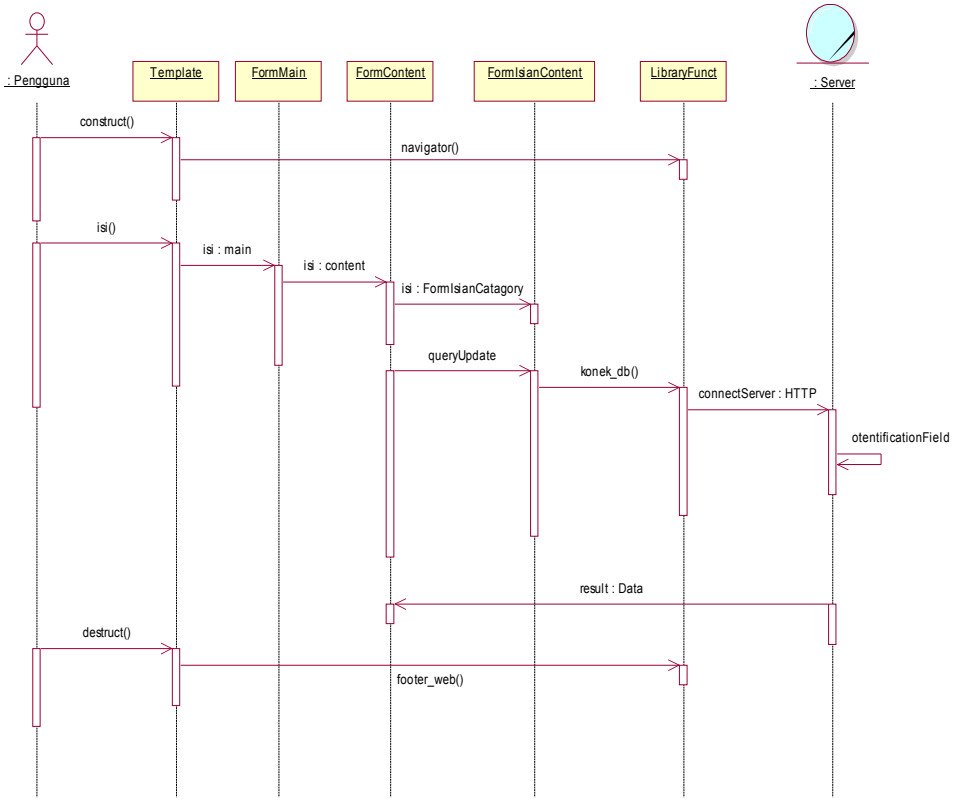
Gambar 3.22. *Sequence Diagram* Pengolahan Data Konten

10. *Sequence* Tambah Konten



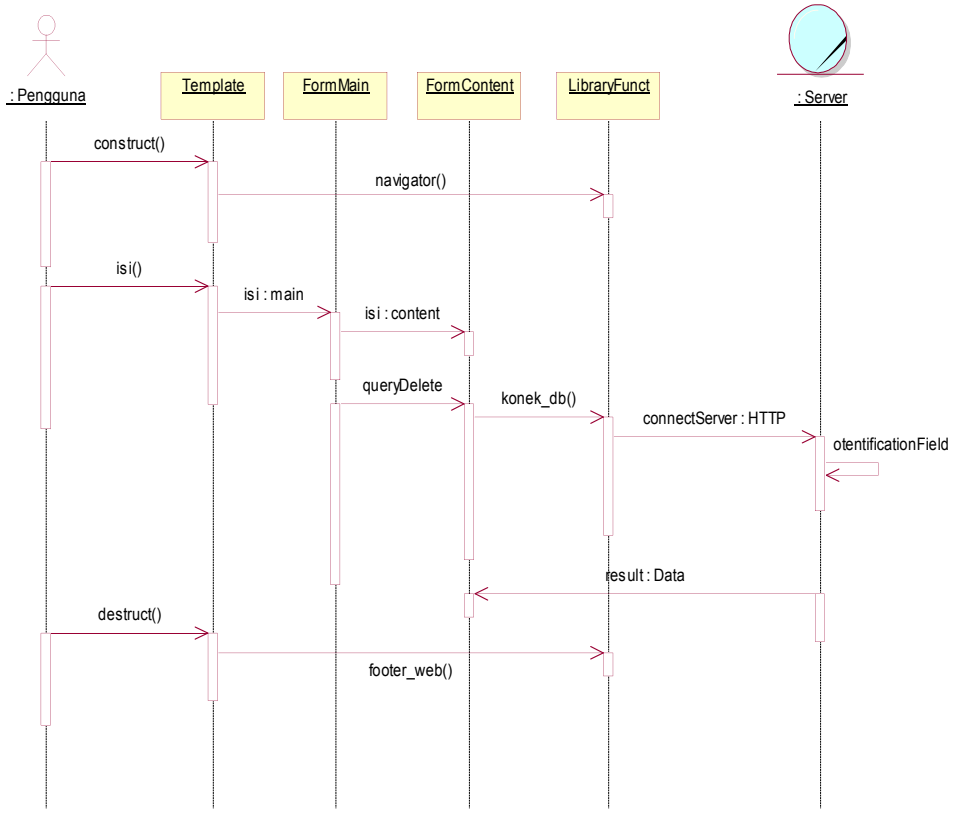
Gambar 3.23. *Sequence Diagram* Tambah Konten

11. *Sequence* Ubah Konten



Gambar 3.24. *Sequence Diagram* Ubah Konten

12. *Sequence* Hapus Konten

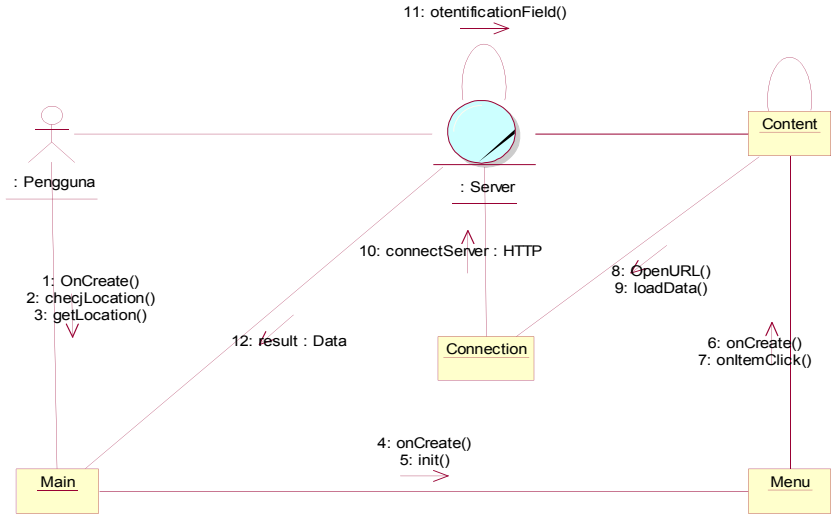


Gambar 3.25. *Sequence Diagram* Hapus Konten

f. *Collaboration Diagram*

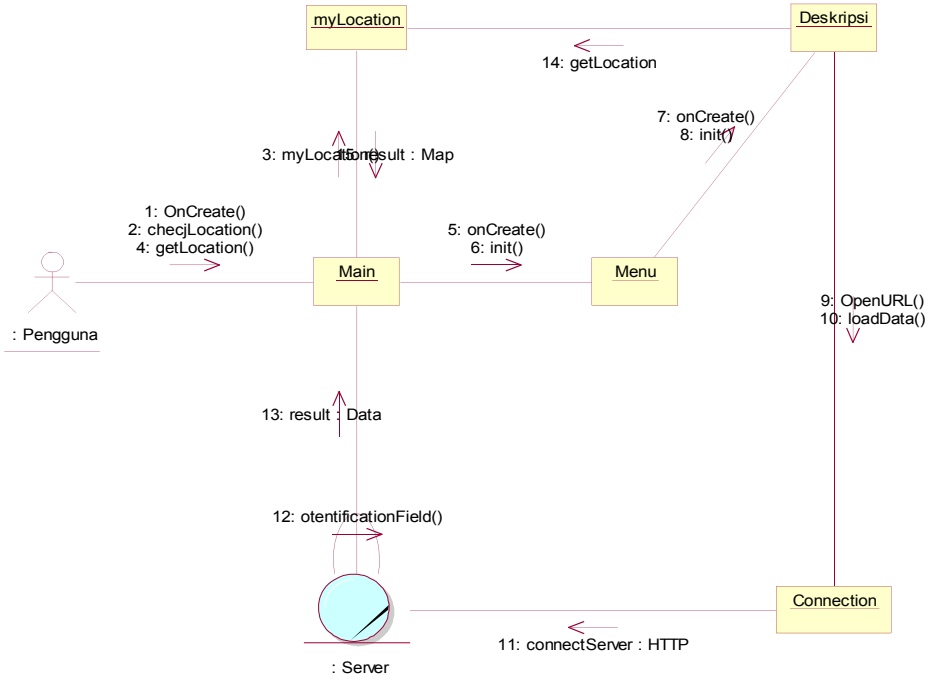
Pada tahap ini *Collaboration Diagram* dapat menggambarkan interaksi antar objek seperti *Sequence Diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian message. Setiap message memiliki *Sequence* number, di mana message dari level tertinggi memiliki nomor satu. Messages dari level yang sama memiliki prefiks yang sama.

1. *Collaboration* Pilih Kategori



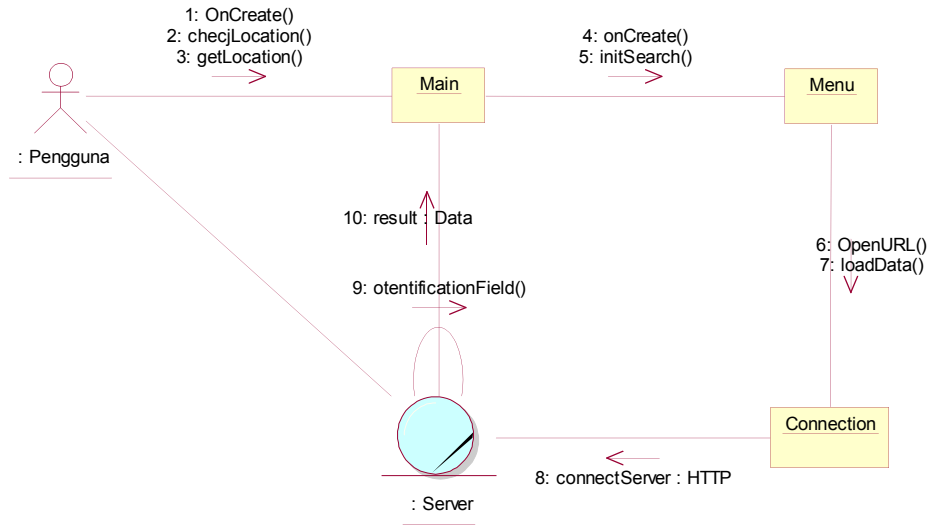
Gambar 3.26. *Collaboration Diagram* Pilih Kategori

2. *Collaboration* Lihat Peta



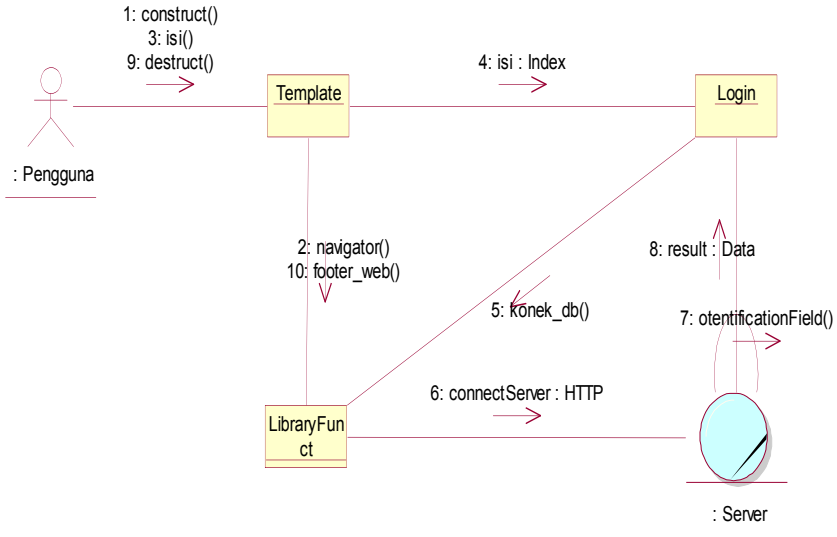
Gambar 3.27. *Collaboration Diagram* Lihat Peta

3. *Collaboration* Pencarian



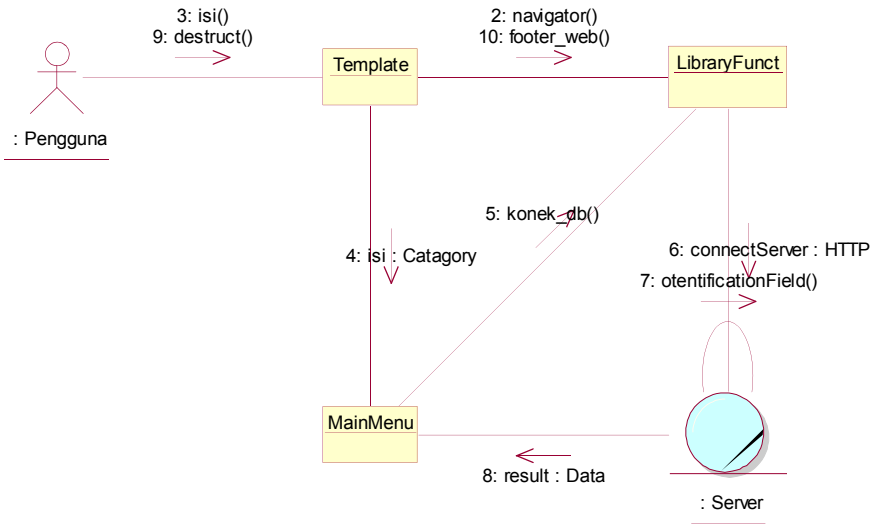
Gambar 3.28. *Collaboration Diagram* Pencarian

4. *Collaboration* Login



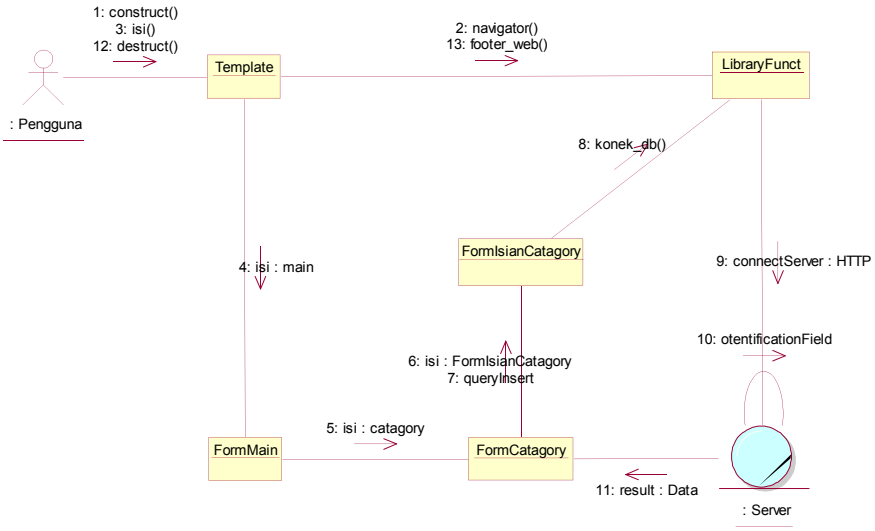
Gambar 3.29. *Collaboration Diagram* Login

5. *Collaboration* Pengolahan Data Kategori



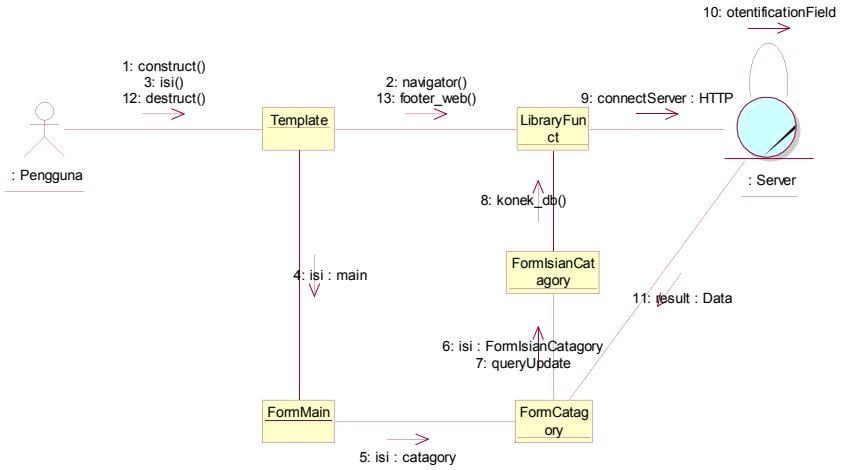
Gambar 3.30. *Collaboration Diagram* Pengolahan Kategori

6. *Collaboration* Tambah Kategori



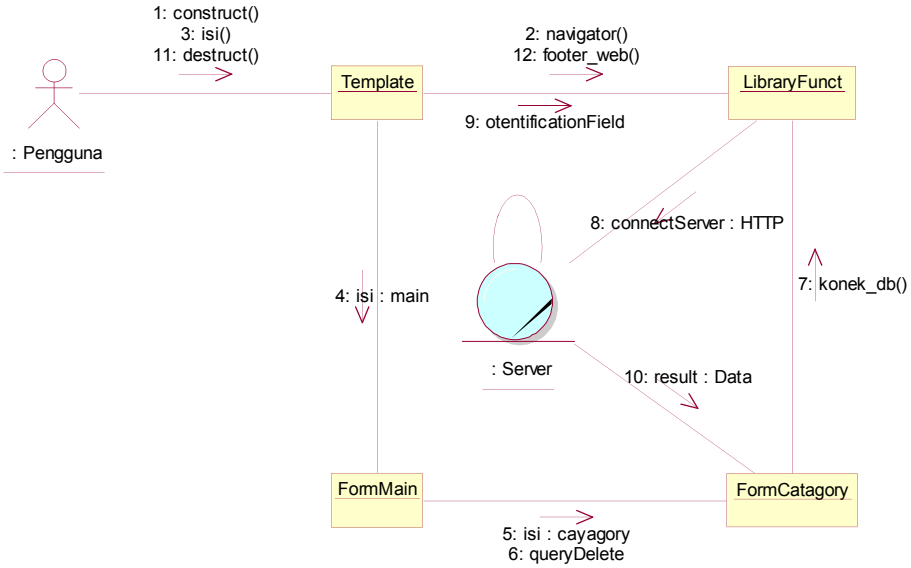
Gambar 3.31. *Collaboration Diagram* Tambah Kategori

7. *Collaboration* Ubah Kategori



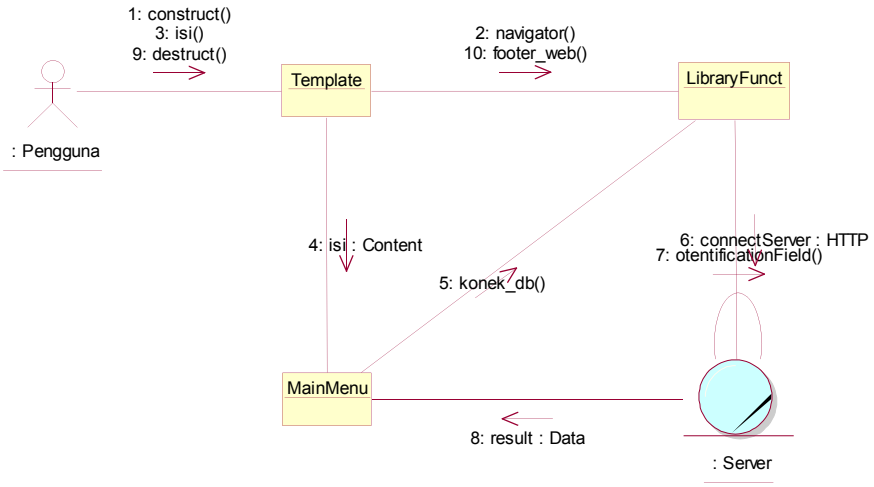
Gambar 3.32. *Collaboration Diagram* Ubah Kategori

8. *Collaboration* Pengolahan Hapus Kategori



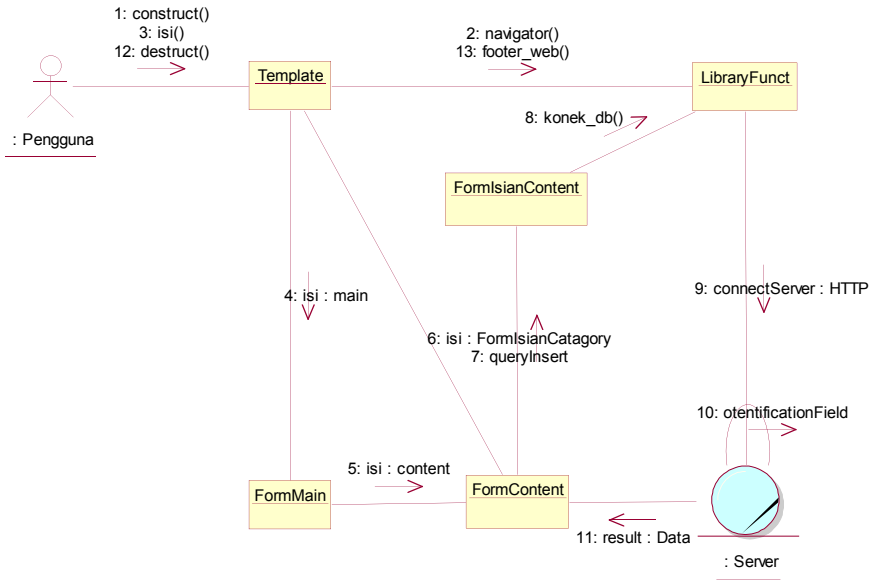
Gambar 3.33. *Collaboration Diagram* Hapus Kategori

9. *Collaboration* Pengolahan Data Konten



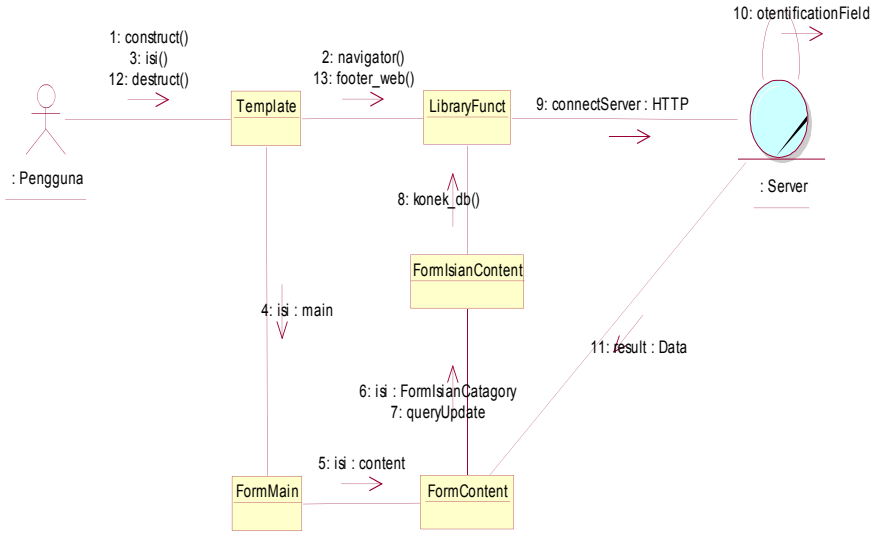
Gambar 3.34. *Collaboration Diagram* Pengolahan Konten

10. *Collaboration* Tambah Konten



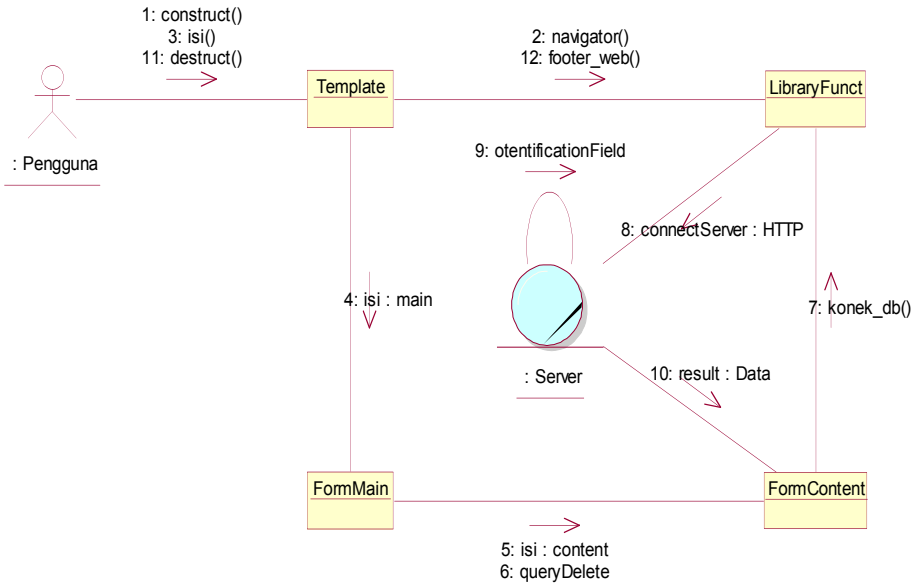
Gambar 3.35. *Collaboration Diagram* Tambah Konten

11. *Collaboration* Ubah Konten



Gambar 3.36. *Collaboration Diagram* Ubah Konten

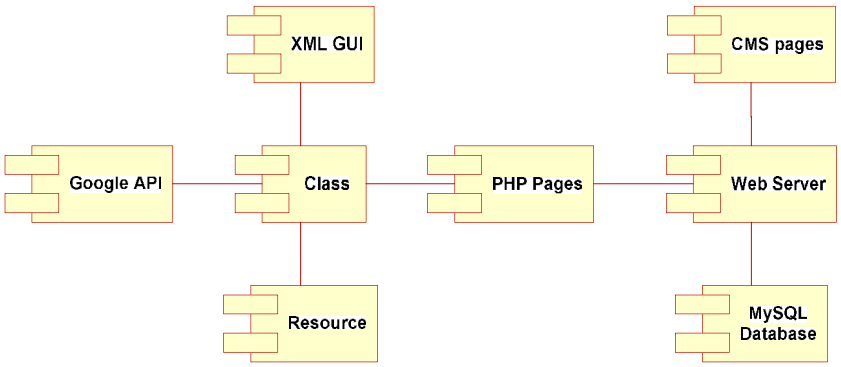
12. *Collaboration* Hapus Konten



Gambar 3.37. *Collaboration Diagram* Pengolahan Kategori

g. *Component Diagram*

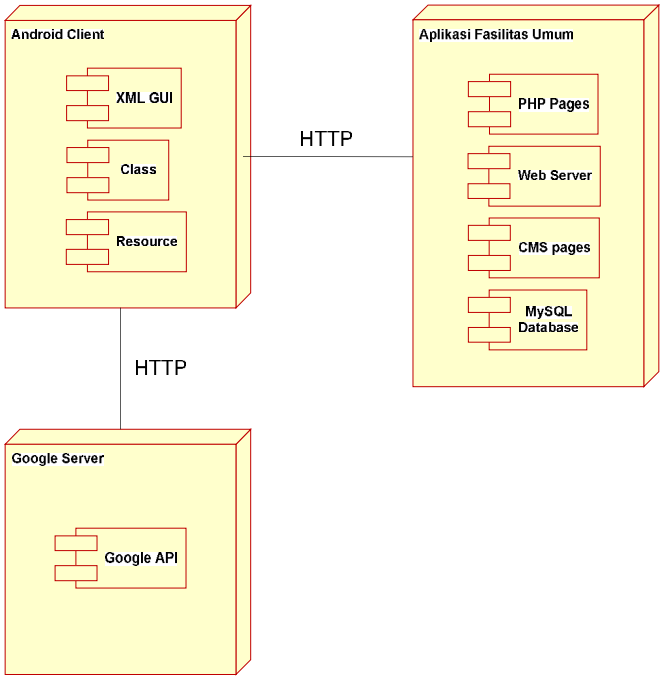
Pada tahap ini, *Component Diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak termasuk ketergantungan (*dependency*). Komponen piranti lunak yang yang dimaksud adalah modul yang berisi *source code* atau *binary code*, yang ada *library* atau *excutetable* yang muncul pada *compile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



Gambar 3.38. *Component Diagram*

h. *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. Gambaran arsitektur fisik dari perangkat keras dan perangkat lunak dari aplikasi fasilitas umum seperti ditunjuk dalam gambar dibawah ini :



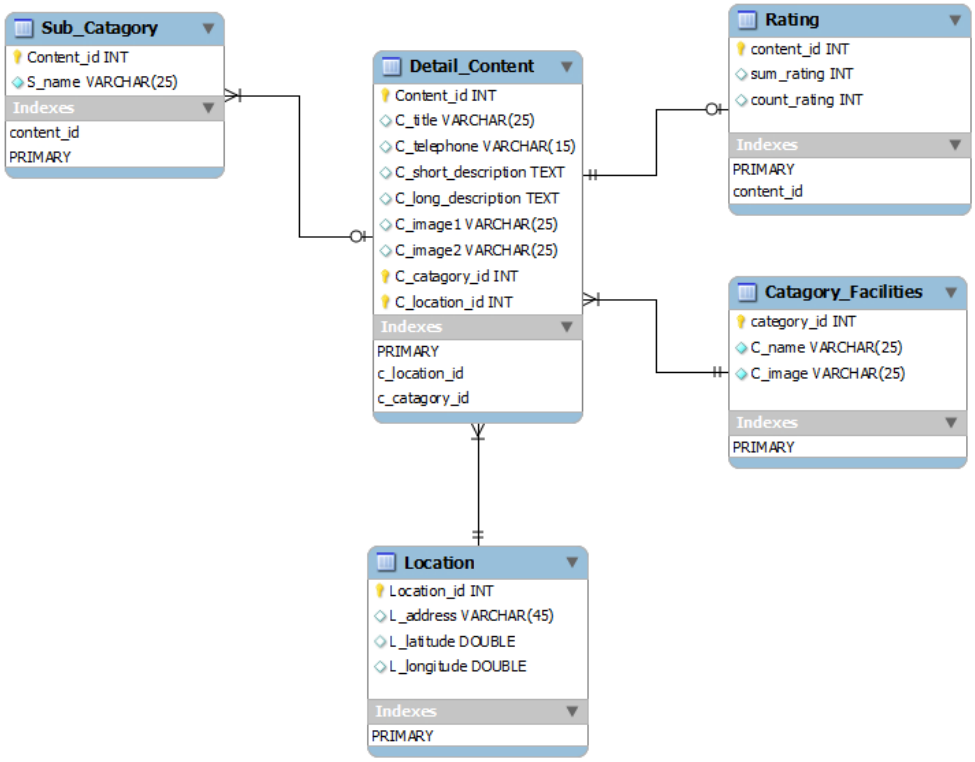
Gambar 3.39. *Deployment Diagram*

3.2 Perancangan Sistem

Perancangan sistem bertujuan untuk menspesifikasikan aspek-aspek teknik yang menjadi solusi dalam perencanaan. Pada tahap ini perancangan akan didefinisikan secara detail untuk mengatasi masalah-masalah yang lebih teknis, berkaitan dengan kegiatan implementasi seperti perancangan database, dan perancangan antarmuka.

3.2.1 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 3.40. Skema Relasi Aplikasi Fasilitas Umum

3.2.2 Perancangan Grafis dan Antar Muka

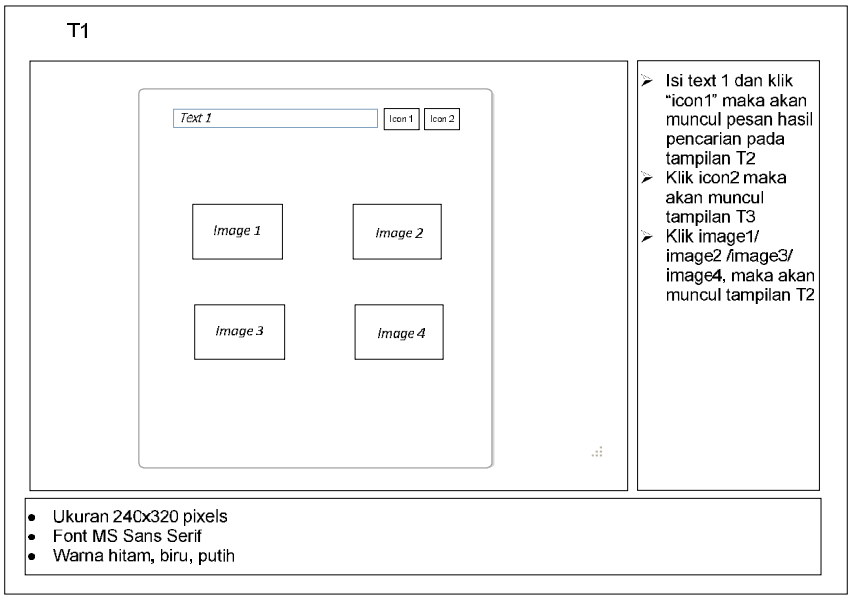
Tahap perancangan desain bertujuan untuk mencari bentuk yang optimal dari aplikasi yang akan dibangun dengan pertimbangan factor-faktor permasalahan dan kebutuhan yang ada pada sistem seperti yang telah ditetapkan pada tahap analisis. Dalam tahap ini upaya yang dilakukan yaitu dengan cara mengkombinasikan penggunaan teknologi perangkat keras dan perangkat lunak yang tepat sehingga diperoleh yang optimal dan mudah diimplementasikan.

Perancangan grafis dan antarmuka terdiri dari dua bagian utama yaitu disain grafis dan antarmuka *frontend application* dan *backend application*.

3.2.2.1 Perancanagan Antar Muka *Frontend Application*

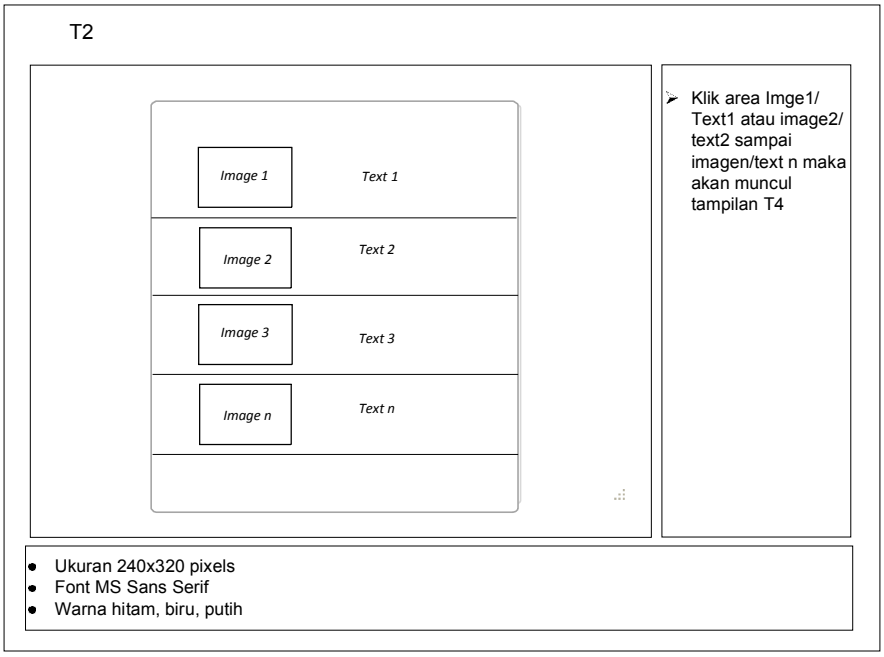
Untuk memudahkan proses pembuatan aplikasi fasilitas umum pada *Mobile* ini maka terlebih dahulu membuat rancangan desain tampilannya. Perancangan ini meliputi rancanagan antar muka menu utama, list content, detail conten, dan map.

1. Perancangan antarmuka menu utama aplikasi *frontend*



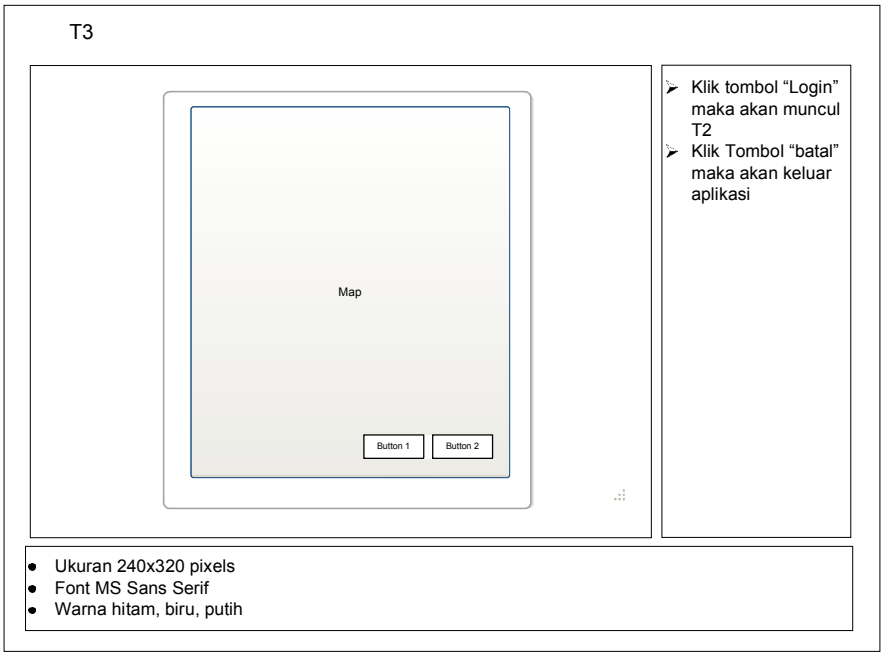
Gambar 3.41. Tampilan Menu Utama Aplikasi *Frontend*

2. Perancangan antarmuka List Content



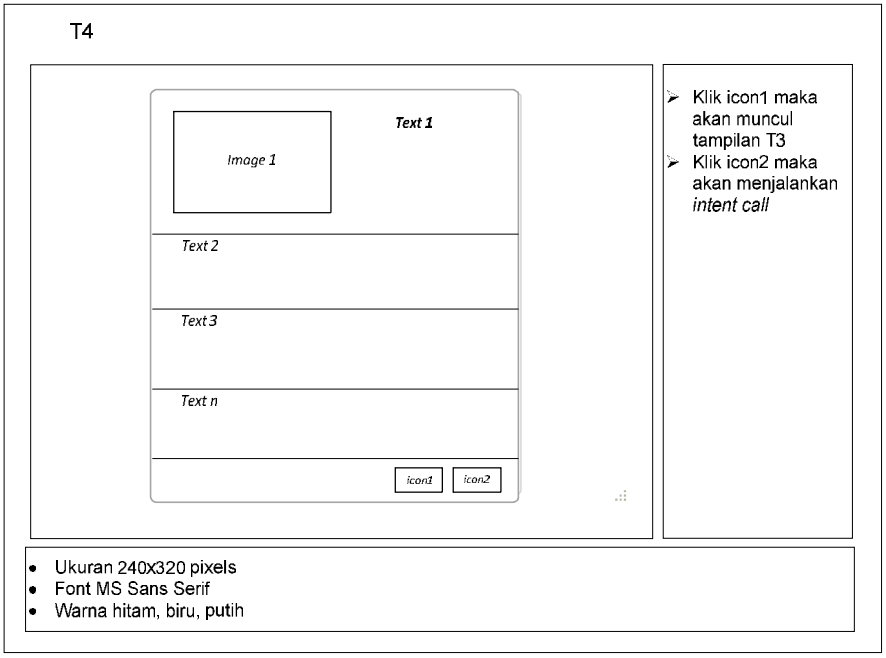
Gambar 3.42. Tampilan List Conten Aplikasi *Frontend*

3. Perancangan antarmuka Map



Gambar 3.43. Tampilan Map Aplikasi *Frontend*

4. Perancangan antarmuka Detail Content

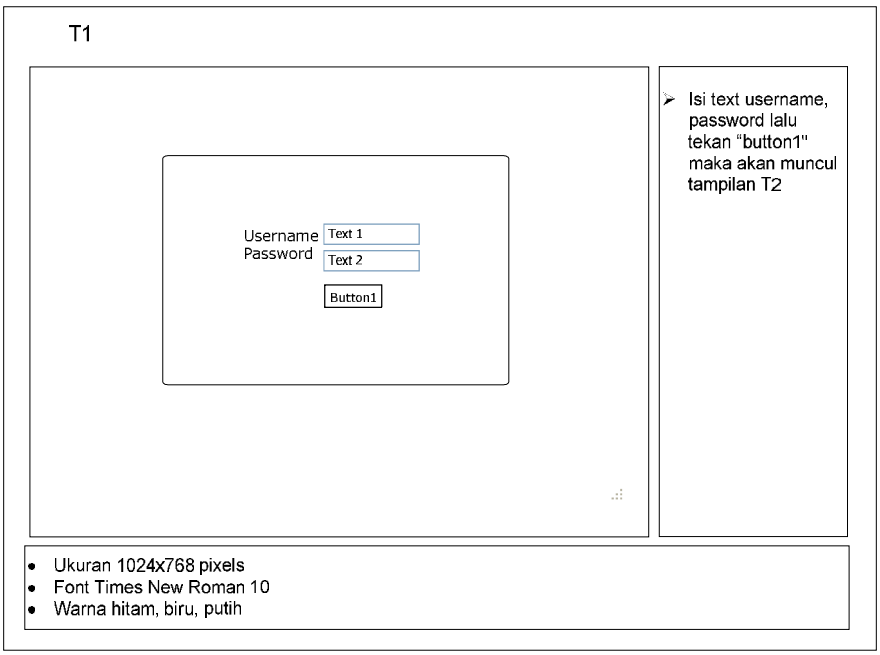


Gambar 3.44. Tampilan Detail Conten Aplikasi *Frontend*

3.2.4.2 Perancangan Grafis dan Antar Muka *Backend Application*

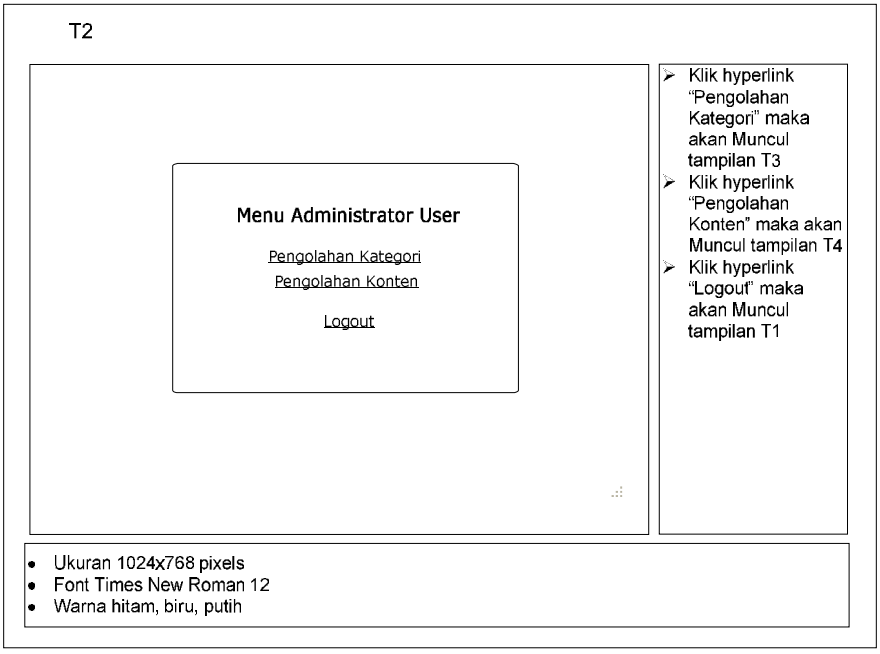
Berbeda dengan *frontend application* yang memiliki rancangan yang begitu kompleks, *backend application* hanya meliputi rancanagan antar muka saja yaitu terdiri dari halaman login, menu administrator dan halaman edit/delete konten.

1. Perancangan Halaman Login Menu Admin



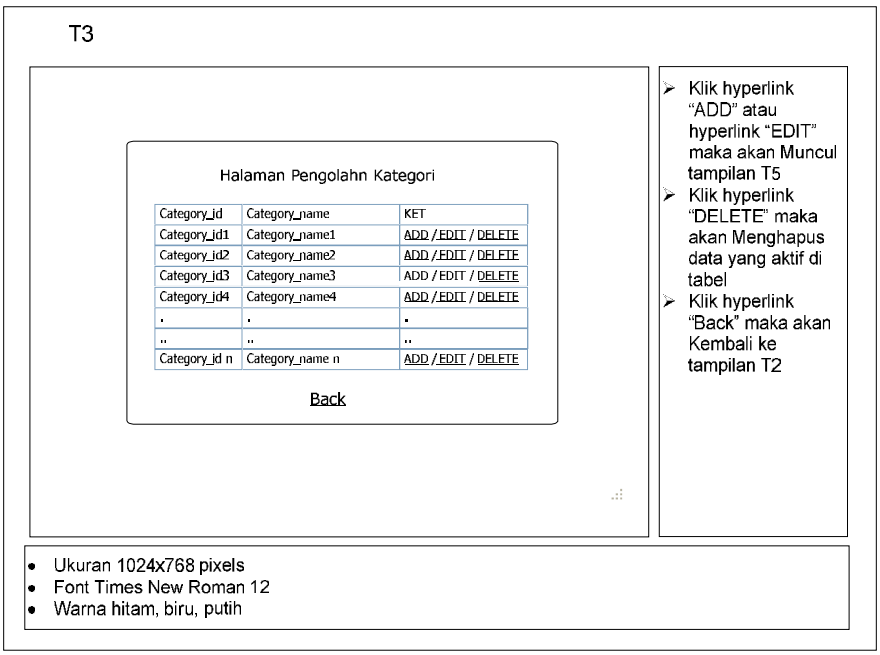
Gambar 3.45. Tampilan Halaman Login

2. Disain Halaman Utama Web Admin



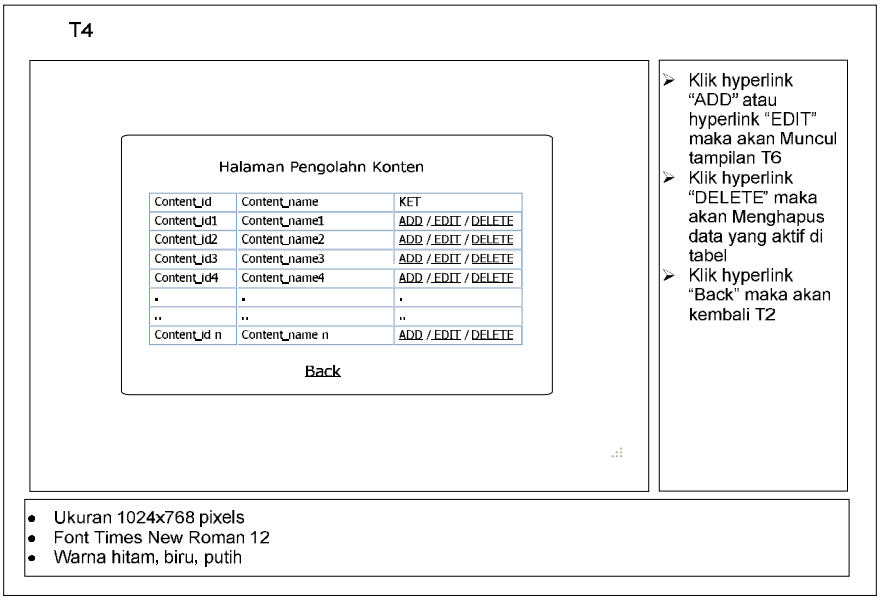
Gambar 3.46. Tampilan Halaman Menu Utama *Backend*

3. Disain Halaman Pengolahan Kategori



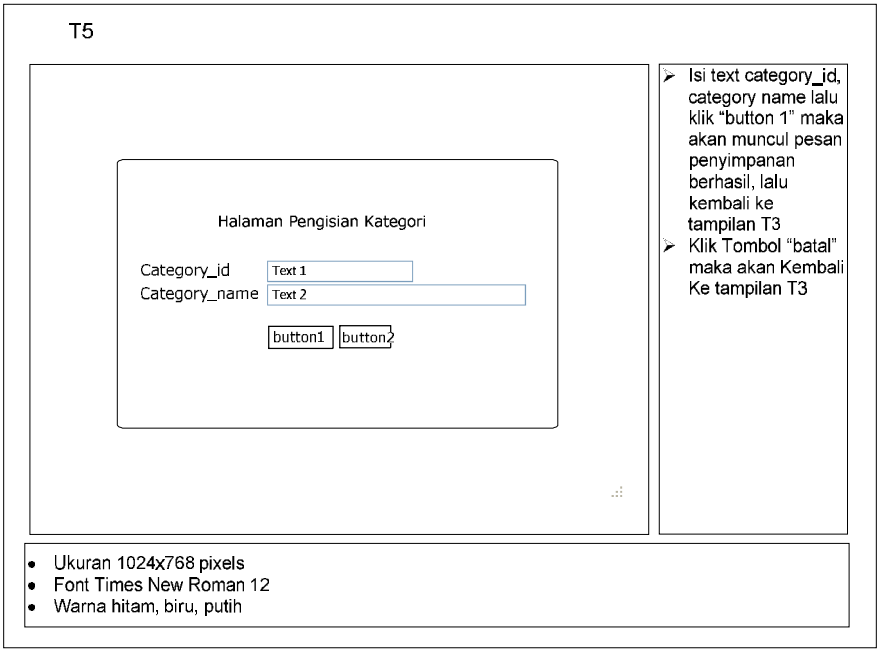
Gambar 3.47. Tampilan Halaman Pengolahan Kategori

4. Disain Halaman Pengolahan Konten



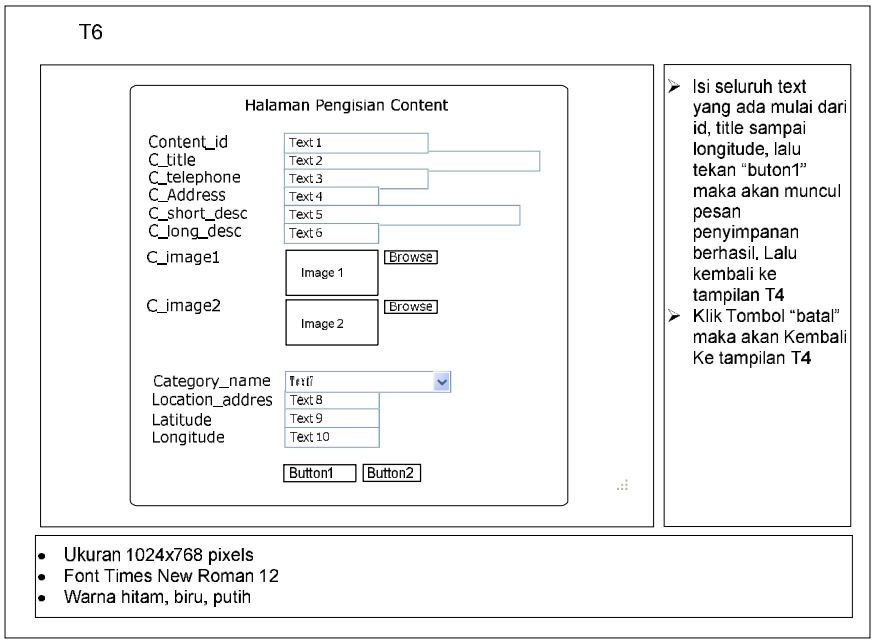
Gambar 3.48. Tampilan Halaman Pengolahan Konten

5. Desain Halaman Pengisian Kategori



Gambar 3.49. Tampilan Halaman Isian Kategori

6. Desain Halaman Pengisian Konten



Gambar 3.50. Tampilan Halaman Isian Konten