BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem merupakan tahap untuk mempelajari interaksi sistem yang terdiri atas pelaku proses dalam sistem, prosedur, data serta informasi yang terkai0t yang ada di Pusat Penelitian Geoteknologi LIPI.

Tujuan dari analisis sistem yang sedang berjalan adalah sebagai berikut :

1. Menelusuri bagaimana sistem yang berjalan dengan memperhatikan proses aliran data atau informasi dan pelaku sistem.
2. Mengevaluasi sistem sehingga dapat mendukung dan meningkatkan kinerja sistem informasi yang akan dikembangkan.
3. Mendapatkan kemungkinan pengembangan sistem yaitu pengembangan terhadap proses dan subproses yang dapat dimodifikasi kearah yang lebih baik atau akan lebih mudah digunakan dengan sistem yang terotomatisasi.

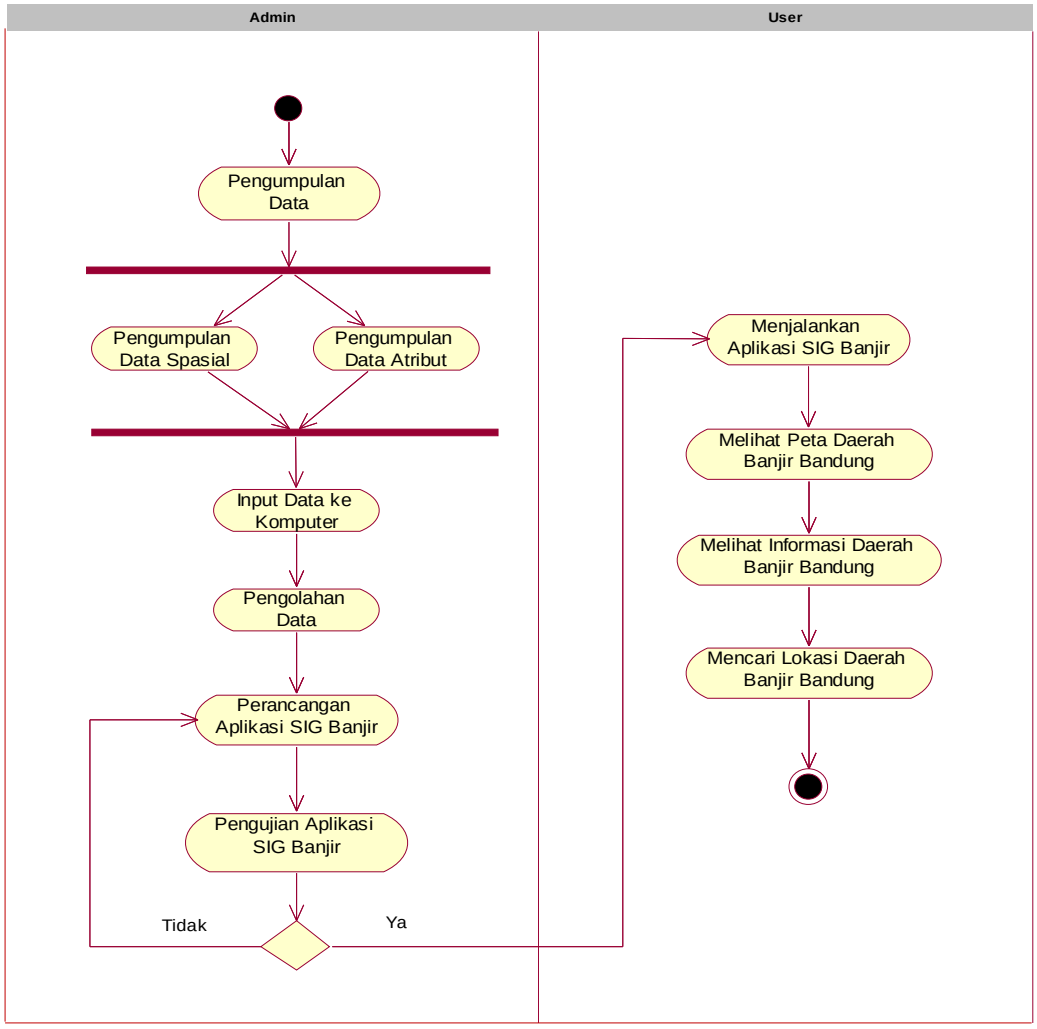
3.1.1 Analisis Masalah

Sesuai dengan hasil penelitian yang dilakukan, terdapat adanya sistem informasi geografis banjir di Bandung berbasis desktop. Dimana sistem informasi geografis banjir berbasis desktop ini merupakan sistem informasi geografis lama yang perlu pembaharuan dan pengembangan. Berdasarkan hal tersebut, maka dibutuhkan suatu sistem yang lebih baik, guna penyampaian informasi mengenai daerah banjir tersebut dapat lebih bermanfaat bagi *user*. Sehingga penulis mencoba untuk membuat suatu sistem informasi geografis berbasis *mobile* agar

dalam mengakses sistem informasi geografis tersebut dapat dilakukan kapan saja dan dimana saja dengan menggunakan telepon seluler.

3.1.2 Analisis Sistem Yang Sedang Berjalan

Analisis sistem yang sedang berjalan memberikan gambaran tentang sistem yang saat ini sedang berjalan di LIPI Bandung. Analisis sistem yang sedang berjalan ini bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem tersebut, sehingga kelebihan dan kekurangan sistem dapat diketahui. Kemudian di dalamnya juga dapat diketahui tahapan-tahapan yang menerangkan mengenai proses apa yang dikerjakan, siapa yang mengerjakan proses tersebut, bagaimana proses tersebut dapat dikerjakan, dan apa saja yang terlibat di dalam proses tersebut. Berikut akan terlihat lebih jelas dalam *Acivity diagram* dibawah ini :



Gambar 3.1 *Activity Diagram* Sistem Yang Sedang Berjalan

Gambar *Activity diagram* diatas menunjukkan alur proses yang dilakukan di Pusat Penelitian Geoteknologi LIPI secara keseluruhan. Penjelasan tentang kegiatan tersebut sebagai berikut :

a) Pengumpulan Data

Data yang dipersiapkan antara lain :

1. Data Spasial

Data spasial berupa peta daerah Bandung yang diperoleh dari hasil penelitian dengan cara membuat peta dengan menggunakan MapInfo 9.0 dan disimpan dalam server.

2. Data Atribut

Data Atribut merupakan data yang berhubungan dengan daerah kebencanaan banjir tahun 2005 - 2008 yang ada di Kabupaten Bandung. Data diperoleh dari data tahunan selama 12 bulan periode tahun 2005 - 2008 yang ada di Pusat Penelitian Geoteknologi LIPI.

b) *Input* Data ke Komputer

Data spasial dan data atribut yang telah dikumpulkan, kemudian di input ke dalam komputer untuk kemudian dapat digunakan dalam bentuk digital.

c) Pengolahan Data

Pengolahan data merupakan suatu proses pemilihan dan pemrosesan data spasial dan atribut, agar data yang telah diolah tersebut dapat bermanfaat dan berguna bagi *user*.

d) Perancangan Aplikasi SIG Banjir

Perancangan aplikasi merupakan suatu proses yang dilakukan setelah data spasial dan data atribut selesai diolah, maka dapat dirancang suatu aplikasi SIG untuk menyampaikan informasi mengenai keadaan geografis.

e) Pengujian Aplikasi SIG Banjir

Pengujian aplikasi merupakan suatu proses yang dilakukan setelah perancangan aplikasi SIG selesai, sehingga dapat diketahui implementasi dari aplikasi tersebut.

f) Menjalankan Aplikasi SIG Banjir

Menjalankan aplikasi SIG banjir merupakan suatu proses yang dilakukan oleh *user* setelah perancangan dan pengujian aplikasi SIG banjir selesai. *User* dapat menjalankan aplikasi tersebut yang sudah terinstal di komputer dan dapat mengaksesnya sesuai kebutuhan.

g) Melihat Peta Daerah Banjir Bandung

Melihat peta daerah banjir Bandung merupakan suatu proses yang dilakukan oleh *user* setelah menjalankan aplikasi SIG tersebut. Dimana *user* dapat melihat peta daerah-daerah banjir di Bandung, sehingga *user* dapat mengetahui dimana saja daerah-daerah di Bandung yang terkena bencana banjir.

h) Melihat Informasi Daerah Banjir Bandung

Melihat informasi daerah banjir Bandung merupakan suatu proses yang dilakukan oleh *user* setelah menjalankan aplikasi SIG tersebut. Dimana selain *user* dapat melihat peta daerah-daerah banjir di Bandung, *user* juga

dapat melihat dan mengetahui informasi apa saja yang ada yang berhubungan dengan daerah banjir tersebut. Misalnya yaitu informasi nama kecamatan di Bandung yang terkena banjir, luas daerah yang terkena banjir, dan sebagainya.

i) Mencari Lokasi Daerah Banjir Bandung

Mencari Lokasi Daerah Banjir Bandung merupakan suatu proses yang dilakukan oleh *user* setelah menjalankan aplikasi SIG tersebut. Dimana *user* dapat mencari lokasi daerah banjir di Bandung, sehingga *user* tidak susah payah dalam mengetahui dimana saja daerah-daerah di Bandung yang terkena bencana banjir.

3.1.3 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem merupakan analisis suatu kebutuhan sistem yang diperlukan dan yang digunakan dalam pembuatan sistem ini, yang terdiri dari beberapa bagian, yaitu : kebutuhan non-fungsional dan kebutuhan fungsional.

3.1.3.1 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukkan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukkan sehingga menghasilkan suatu keluaran yang diinginkan.

3.1.3.1.1 Analisis Perangkat Keras *(Hardware)*

Adapun spesifikasi perangkat keras yang dibutuhkan untuk sistem yang akan dibangun adalah sebagai berikut :

a) *Server*

1) Processor Intel Core 2 Duo 2 Ghz
2) RAM 2 Gb
3) Hardisk 80 Gb
4) LAN Card.
5) *Keyboard* dan *Mouse*
6) Monitor 17“

b) *Client*

1) *Handphone* dapat terkoneksi dengan internet
2) Memiliki *web browser* yang *compatible*

3.1.3.1.2 Analisis Perangkat Lunak *(Software)*

Untuk mendukung dalam penyimpanan informasi, dibutuhkan suatu fasilitas yang memadai, yaitu berupa perangkat lunak *(software)* yang dirancang untuk memudahkan dalam pencarian informasi. Adapun perangkat lunak yang digunakan adalah sebagai berikut :

1) Server

a) Hosting dengan kapasitas penyimpanan *database* sebesar 150 Mb
b) PHP sebagai bahasa pemrograman
c) MySQL untuk membuat Basis data *(database)*

2) *Client*

a) Aplikasi web *mobile* dapat diakses melalui *handphone,* dimana *handphone* tersebut dapat terkoneksi dengan internet

b) *Handphone* memiliki *web browser* yang *compatible*

3.1.3.1.3 Analisis Kebutuhan Pengguna

Analisis kebutuhan pengguna adalah analisis kriteria yang harus dimiliki oleh pengguna agar tidak kesulitan dalam menjalankan aplikasi sistem informasi geografis untuk mengidentifikasi daerah rawan Banjir di Bandung berbasis web *mobile* ini.

Terdapat 2 kategori pengguna dari sistem yang akan dibangun, yaitu admin yang merupakan aktor yang mengelola data daerah banjir di Bandung dan *user* sebagai pengguna aplikasi sistem informasi geografis banjir Bandung berbasis web *mobile*. Spesifikasi pengguna tersebut adalah sebagai berikut :

Tabel 3.1 Karakteristik Pengguna

Pengguna	**Tanggung Jawab**	**Hak akses**	**Tingkat Keterampilan**	**Pengalaman**
Admin	Menangani pengolahan data spasial dan data atribut daerah banjir di Bandung	Menjalankan aplikasi *backend* dan *frontend* sesuai dengan tugasnya	Mengerti serta dapat menggunakan komputer, mengelola *web server* dan teknis perawatannya	Mengoperasikan komputer dan mengelola *web server*

User	-	Melihat halaman utama SIG banjir Bandung berbasis *mobile*, dan dapat mengakses menu-menu lainnya	Mengerti serta dapat mengoperasikan *handphone* yang dapat terkoneksi dengan internet	Mengoperasikan *handphone* yang dapat terkoneksi dengan internet

3.1.3.2 Analisis Google Maps API (*Application Programming Interface)*

Google menyediakan layanan API (Application Programming Interface) memungkinkan aplikasi client untuk melihat, menyimpan dan memperbaharui data peta dalam bentuk Data API Google feed dengan menggunakan model data fitur (letak, garis dan bentuk) dalam peta. Aplikasi ini diberi nama Google Maps API (GMaps API). Peta yang ditampilkan diambil dari layanan Google Maps.

a) Jenis Tampilan Peta Google Maps

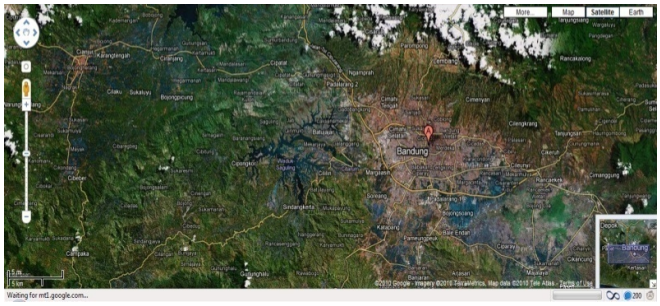
Ada tiga jenis tampilan yang bisa dipilih dari Google Maps, yaitu:

1. *Map* menampilkan peta dalam bentuk peta garis. Seperti pada gambar di bawah ini.



Gambar 3.2 Penggambaran kota Bandung dalam bentuk *map*

2. *Sattelite* menampilkan peta dalam bentuk citra/foto satelit. Seperti pada gambar di bawah ini.



Gambar 3.3 Penggambaran kota Bandung dalam bentuk *Sattelite*

3. *Earth* merupakan gabungan dari *Map* dan *Sattelite*, seperti pada gambar di bawah ini.



Gambar 3.4 Penggambaran kota Bandung dalam bentuk *earth*

b) Elemen-Elemen Yang Terdapat Pada Google Maps API

Elemen-elemen yang terdapat pada Google Maps API adalah:

1. *Marker*

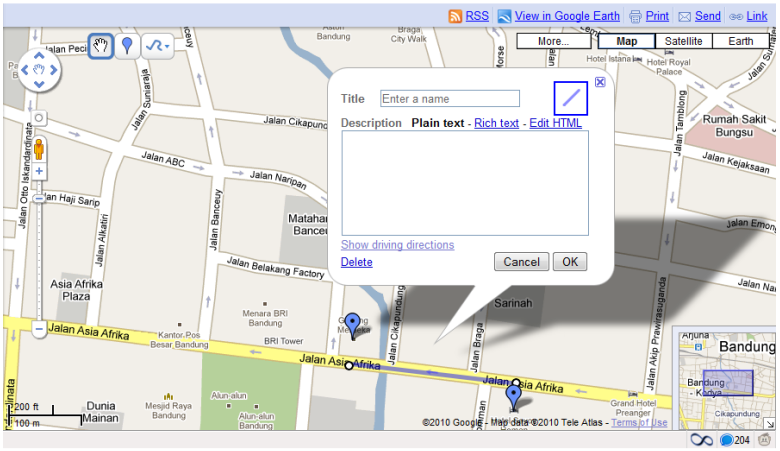
Marker adalah simbol yang menandakan suatu lokasi pada peta yang ditampilkan Google Maps. Contoh penggambaran *marker* dapat dilihat seperti gambar dibawah ini:



Gambar 3.5 Penggunaan *marker*

2. *Polyline*

Polyline adalah *shape* yang digunakan untuk menandakan suatu jalur jalan atau area. *Polyline* itu sendiri yaitu terdiri dari kumpulan titik koordinat. Contoh penggambaran *polyline* dapat dilihat seperti gambar dibawah ini:



Gambar 3.6 Penggunaan *polyline*

c) Objek Model Google Maps API

Objek model yang terdapat pada Google Map API adalah sebagai berikut:

1. Inisialisasi Map

Inisialisasi diproses dengan menggunakan *method setCenter(). Method setCenter()* ini membutuhkan GetLatLng koordinat dan *zoom level* dan *method*

ini harus segera dikirim sebelum ada pengoperasian lain pada peta, termasuk *setting* atribut peta itu sendiri.

2. *Loading* Google Maps API

   Yaitu suatu koneksi *script* yang dibuat dalam server Google Maps API dengan menggunakan *key* yang didapatkan pada saat mendaftar ke Google Maps API.

3. GMap2 - *Elementary Object*

   Adalah suatu *class javascript* yang membuat peta itu ada yaitu suatu *class* GMap2, *Object* dari class ini akan menyediakan sebuah peta di halaman web, selanjutnya *variable* map akan memberikan nilai sebuah object dari class GMap2. Fungsi GMap2 adalah sebagai constructor dan definisinya.

4. Loading Map

   Ketika halaman HTML di *render*, *document object model* (DOM) sudah bisa digunakan dan semua gambar *external* dan *script* diterima oleh *object document*. Untuk memastikan peta kita dimuat sesudah halaman selesai dimuat oleh *browser*.

5. *Latitude and Longitude*

   Yaitu suatu objek GLatLng yang menentukan titik koordinat peta yang akan ditampilkan, parameternya terdiri dari lintang/*latitude* dan bujur/*longitude*.

3.1.3.3 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dilakukan untuk memberikan gambaran aliran data yang ada pada program aplikasi yang akan dibangun. Tahapan-tahapan yang ada yaitu dengan mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem. Adapun sebagai alat bantu yang

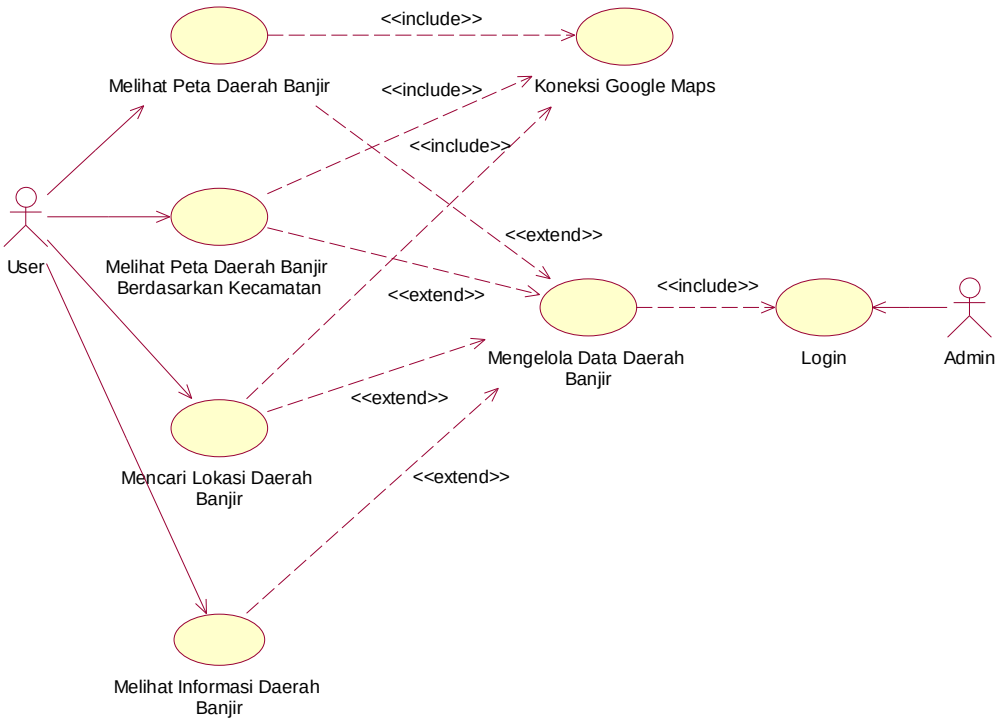
digunakan untuk menggambarkan atau memodelkan sistem secara umum yang akan dibangun yaitu dengan menggunakan *Unified Modeling Language* (UML).

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

3.2. Perancangan *Unified Modeling Language* (UML)

3.2.1 *Use Case Diagram*

Perangkat lunak yang dibuat akan menangani tujuh fungsi utama, seperti terlihat dalam use case berikut :



Gambar 3.7 *Use Case Diagram* Sistem Informasi Geografis Banjir Bandung Berbasis *Mobile*

Sistem ini terdiri dari dua aktor, yaitu *user* dan admin. Kegiatan yang dapat dilakukan oleh *user* adalah melihat peta daerah banjir, melihat peta daerah banjir berdasarkan Kecamatan, mencari lokasi daerah banjir, dan melihat informasi daerah banjir. Sedangkan kegiatan yang dapat dilakukan oleh admin adalah login dan mengelola data daerah banjir.

3.2.1.1 Definisi Aktor

Tabel 3.2 Definisi Aktor

No	Aktor	Definisi
1	*User*	Aktor yang berperan dalam meggunakan aplikasi SIG berbasis *mobile*, dengan syarat memiliki *handphone* yang dapat terkoneksi dengan internet
2	Admin	Aktor yang berperan dalam mengelola data daerah banjir Bandung di server dan memiliki kewenangan untuk tambah data, edit data, dan hapus data

3.2.1.2 Definisi *Use Case*

Tabel 3.3 Definisi *Use Case*

No	*Use Case*	Deskripsi
1.	Melihat Peta Daerah Banjir	Berisi proses melihat peta daerah banjir kota bandung secara keseluruhan
2.	Melihat Peta Daerah Banjir Berdasarkan Kecamatan	Berisi proses melihat peta kota bandung beserta informasinya berdasarkan Kecamatan
3.	Mencari Lokasi Daerah Banjir	Berisi proses untuk mencari lokasi daerah-daerah banjir di kota Bandung
4.	Melihat Informasi Daerah Banjir	Berisi proses untuk melihat informasi daerah banjir di Bandung, berdasarkan daerah banjir yang di pilih
5.	Koneksi Google Maps	Berisi proses untuk melakukan koneksi antara client dengan server, dimana server telah terkoneksi dengan google maps
6.	Login	Berisi fungsionalitas yang berguna hanya untuk admin, agar admin dapat mengelola data SIG banjir Bandung

7.	Mengelola Data Daerah Banjir	Berisi fungsionalitas yang berguna hanya untuk admin, agar admin dapat mengolah data daerah banjir di Bandung

3.2.1.3 Skenario *Use Case*

Skenario *use case* adalah alur cerita atau proses-proses yang terjadi pada sistem antara aktor dengan *use case*. Skenario Use Case ditunjukkan pada tabel-tabel dibawah ini :

3.2.1.3.1 Melihat Peta Daerah Banjir

Tabel 3.4. Skenario *Use Case* Melihat Peta Daerah Banjir

Identifikasi		
Nomor	1	
Nama *Use Case*	Melihat Peta Daerah Banjir	
Tujuan	Melihat peta daerah banjir di Bandung secara keseluruhan	
Deskripsi	*User* dapat menjalankan web SIG banjir di *handphone* dan dapat melihat peta daerah banjir di Bandung	
Tipe		
Aktor	*User*	
Skenario Utama		
Kondisi awal	Web SIG Banjir Bandung sudah dijalankan di *handphone*	
Aksi Aktor		**Reaksi Sistem**
1. *User* menjalankan web SIG banjir Bandung di *handphone*		2. Menampilkan peta daerah banjir di Bandung
Skenario Alternatif – Autentikasi Gagal		
Aksi Aktor		**Reaksi Sistem**
User merefresh halaman web SIG banjir Bandung		
Kondisi akhir	Menampilkan peta daerah banjir di Bandung secara keseluruhan	

3.2.1.3.2 Melihat Peta Daerah Banjir Berdasarkan Kecamatan

Tabel 3.5. Skenario *Use Case* Melihat Peta Daerah Banjir Berdasarkan Kecamatan

Identifikasi		
Nomor	2	
Nama *Use Case*	Melihat Peta Daerah Banjir Berdasarkan Kecamatan	
Tujuan	Melihat peta daerah banjir di Bandung berdasarkan Kecamatan beserta informasinya	
Deskripsi	*User* dapat menjalankan web SIG banjir di *handphone* dan dapat melihat peta daerah banjir di Bandung berdasarkan Kecamatan yang di pilih	
Tipe		
Aktor	*User*	
Skenario Utama		
Kondisi awal	Web SIG Banjir Bandung sudah dijalankan di *handphone*	
Aksi Aktor		**Reaksi Sistem**
1. *User* menjalankan web SIG banjir Bandung di *handphone*		2. Menampilkan peta daerah banjir di Bandung berdasarkan Kecamatan yang di pilih
Skenario Alternatif – Autentikasi Gagal		
Aksi Aktor		**Reaksi Sistem**
User merefresh halaman web SIG banjir Bandung		
Kondisi akhir	Menampilkan peta daerah banjir di Bandung berdasarkan Kecamatan yang di pilih	

3.2.1.3.3 Mencari Lokasi Daerah Banjir

Tabel 3.6. Skenario *Use Case* Mencari Lokasi Daerah Banjir

Identifikasi	
Nomor	3
Nama *Use Case*	Mencari Lokasi Daerah Banjir
Tujuan	Mencari lokasi daerah banjir di Bandung
Deskripsi	*User* dapat menjalankan web SIG banjir di *handphone* dan dapat mencari lokasi daerah banjir di Bandung
Tipe	
Aktor	*User*
Skenario Utama	
Kondisi awal	Web SIG Banjir Bandung sudah dijalankan di *handphone*

Aksi Aktor	Reaksi Sistem
1. *User* menjalankan web SIG banjir Bandung di *handphone*	2. Menampilkan hasil pencarian lokasi daerah banjir di Bandung berupa peta daerah banjir beserta informasinya
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
User merefresh halaman web SIG banjir Bandung	
Kondisi akhir	Menampilkan hasil pencarian lokasi daerah banjir di Bandung berupa peta daerah banjir beserta informasinya

3.2.1.3.4 Melihat Informasi Daerah Banjir

Tabel 3.7. Skenario *Use Case* Melihat Informasi Daerah Banjir

Identifikasi	
Nomor	4
Nama *Use Case*	Melihat Informasi Daerah Banjir
Tujuan	Melihat informasi saerah banjir sesuai dengan daerah banjir yang di pilih di peta
Deskripsi	*User* dapat menjalankan web SIG banjir di *handphone* dan dapat melihat peta daerah banjir di Bandung beserta dengan informasinya
Tipe	
Aktor	*User*
Skenario Utama	
Kondisi awal	Web SIG Banjir Bandung sudah dijalankan di *handphone*
Aksi Aktor	**Reaksi Sistem**
1. *User* melihat informasi daerah banjir Bandung berdasarkan daerah yang di pilih di peta	2. Menampilkan informasi daerah banjir di Bandung sesuai dengan daerah banjir yang ada di peta
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
Admin memelihara *server*	
Kondisi akhir	Informasi daerah banjir di Bandung dapat di lihat oleh *user*

3.2.1.3.5 Koneksi Google Maps

Tabel 3.8. Skenario *Use Case* Koneksi Google Maps

Identifikasi		
Nomor	5	
Nama *Use Case*	Koneksi Google Maps	
Tujuan	Melakukan koneksi ke google maps	
Deskripsi	*User* melakukan koneksi *web server* ke google maps, agar dapat menampilkan peta di web SIG banjir Bandung pada *handphone*, sehingga nantinya dapat di akses oleh *user*	
Tipe		
Aktor	*User*	
Skenario Utama		
Kondisi awal	Menjalankan web sever SIG banjir Bandung	
Aksi Aktor		**Reaksi Sistem**
1. User menjalankan menu admin dan *web server*		2. Menampilkan koneksi *web server* dengan google maps server
Skenario Alternatif – Autentikasi Gagal		
Aksi Aktor		**Reaksi Sistem**
1. Merefresh halaman *web server*		2. Mengecek koneksi database *web server* dengan google server
Kondisi akhir	Menampilkan peta SIG banjir Bandung di *web server*, sehingga dapat ditampilakan juga di web *mobile*	

3.2.1.3.6 Login

Tabel 3.9. Skenario *Use Case* Login

Identifikasi	
Nomor	6
Nama *Use Case*	Login
Tujuan	Mengelola Data
Deskripsi	Admin harus melakukan login terlebih dahulu dengan memasukkan *username* dan *password* untuk dapat mengelola data SIG banjir di Bandung
Tipe	
Aktor	Admin
Skenario Utama	
Kondisi awal	Menampilkan halaman login

Aksi Aktor	Reaksi Sistem
1. Admin mengakses halaman login 2. Admin memasukkan *username* dan *password*	3.Memeriksa kesamaan antara *username* dan *password* 4.Jika *username* dan *password* benar maka akan menampilkan menu admin
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
2. Menekan tombol ”OK”	1. Jika *username* dan *password* salah maka menampilkan Pesan ”*username* dan *password* salah” 3. Sistem akan *terminate*
Kondisi akhir	Menampilkan menu admin

3.2.1.3.7 Mengelola Data Daerah Banjir

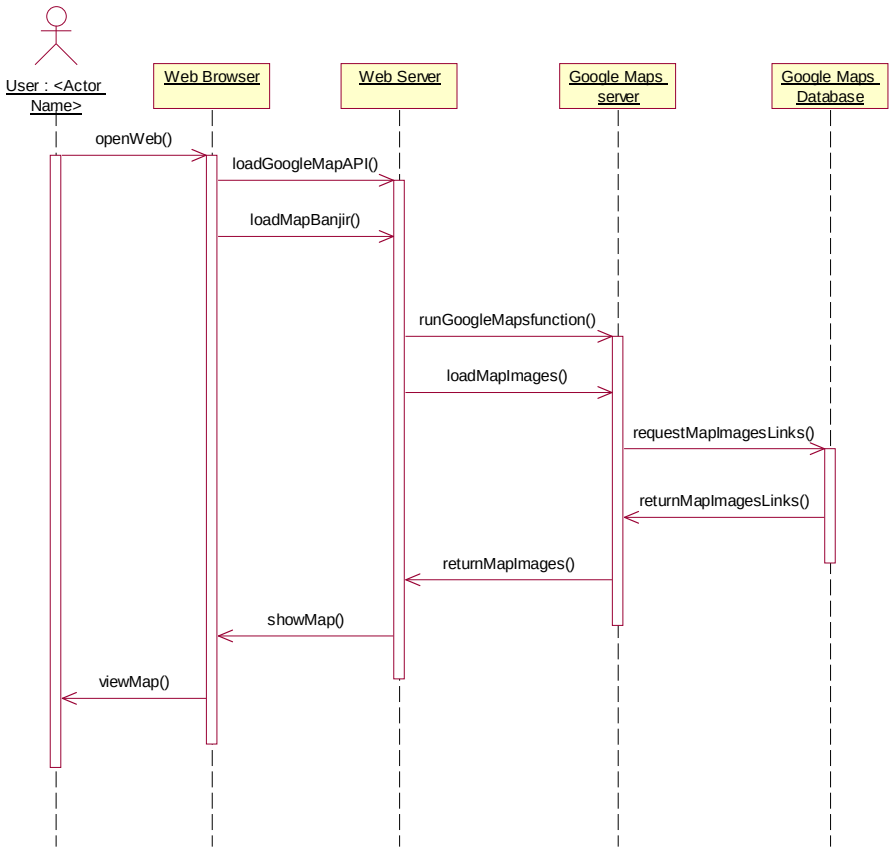
Tabel 3.10. Skenario *Use Case* Mengelola Data Daerah Banjir

Identifikasi	
Nomor	7
Nama *Use Case*	Mengelola Data Daerah Banjir
Tujuan	Mengelola data daerah banjir di Bandung, yang terdiri dari data atribut dan data spasial
Deskripsi	Admin dapat mengelola data atribut dan data spasial daerah banjir di Bandung, tetapi hanya data atribut saja yang bisa di *update*, sehingga nantinya dapat ditampilkan di web SIG banjir Bandung pada *handphone*
Tipe	
Aktor	Admin
Skenario Utama	
Kondisi awal	Menampilkan halaman menu admin mengelola data daerah banjir
Aksi Aktor	**Reaksi Sistem**
1. Admin dapat tambah, edit, dan hapus data atribut	2. Menerima data atribut baru yang dikelola oleh admin dan database *terupdate*
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
Admin memelihara *server*	
Kondisi akhir	Data atribut telah berhasil di *update*

3.2.2 *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh objek dan *message* yang diletakkan diantara objek-objek ini didalam *use case*. Biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai *respon* dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari sebuah aktivitas tertentu, kemudian berproses mengikuti urutan tertentu yang bisa terlihat melalui *message* antar objeknya.

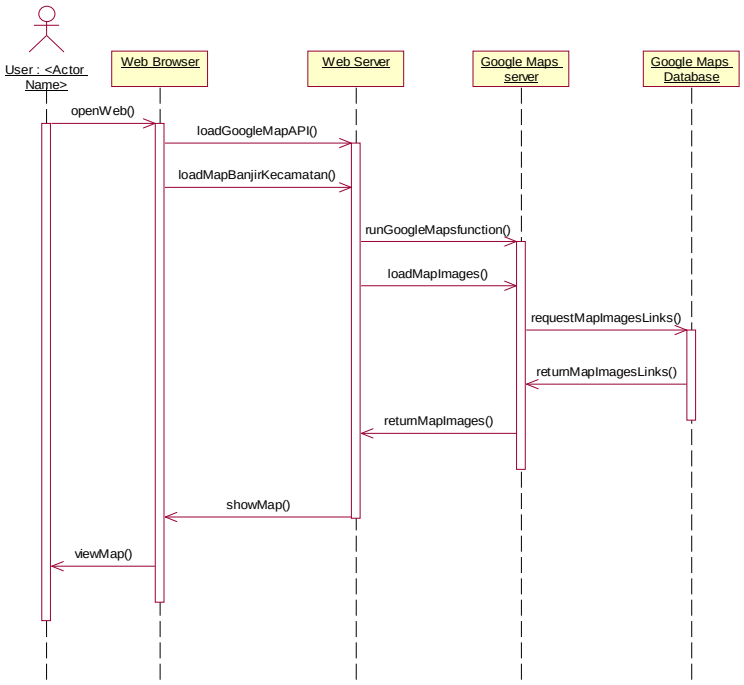
3.2.2.1 *Sequence Diagram* Melihat Peta Daerah Banjir



Gambar 3.8 *Sequence Diagram* Melihat Peta Daerah Banjir

Gambar *sequence diagram* diatas terdiri dari satu aktor yaitu *user* dan empat objek yaitu *web browser*, *web server*, google maps server, dan google maps database. Proses pertama yang terjadi adalah *user* membuka halaman situs melalui *web browser*, dan bentuk awal peta akan langsung di *load*. Proses ini akan dijalankan oleh *web server* yang akan mengirimkan *request* ke Google Maps server. Google Maps server akan mengembalikan data dan link yang diminta. *Web server* akan kembali memproses link tersebut untuk selanjutnya di kirimkan ke *web browser* untuk diproses selanjutnya yang terjadi di saat *user* memilih objek yang ingin dilihat. Saat pengguna menentukan pilihannya, *web browser* akan memproses pilihan tersebut ke *web server* dimana terjadi hubungan dengan database. Data dari databse tersebut antara lain berisi lokasi koordinat yang diperlukan akan dikirim ke Google Maps server. Setelah mendapat data yang diperlukan dari databasenya, data tersebut dikembalikan ke *web server*, untuk selanjutnya bersama data lain yang terdapat pada database ditampilkan pada *web browser*.

3.2.2.2 *Sequence Diagram* Melihat Peta Daerah Banjir Berdasarkan Kecamatan

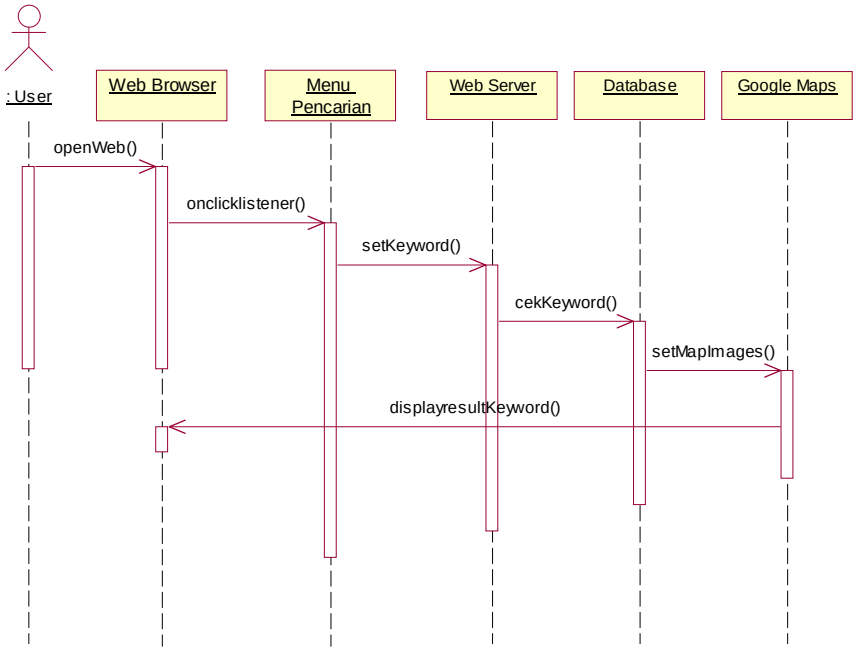


Gambar 3.9 *Sequence Diagram* Melihat Peta Daerah Banjir Berdasarkan Kecamatan

Gambar *sequence diagram* diatas terdiri dari satu aktor yaitu *user* dan empat objek yaitu *web browser*, *web server*, google maps server, dan google maps database. Proses pertama yang terjadi adalah *user* membuka halaman situs melalui *web browser*, dan bentuk awal peta akan langsung di load. Proses ini akan dijalankan oleh *web server* yang akan mengirimkan request ke Google Maps server . Google Maps server akan mengembalikan data dan link yang diminta. *Web server* akan kembali memproses link tersebut untuk selanjutnya di kirimkan ke *web browser* untuk di proses selanjutnya yang terjadi di saat *user* memilih peta daerah banjir berdasarkan Kecamatan di Bandung yang ingin dilihat. Saat pengguna menentukan pilihannya, *web browser* akan memproses pilihan tersebut

ke *web server* dimana terjadi hubungan dengan database. Data dari databse tersebut antara lain berisi lokasi koordinat berdasarkan Kecamatan di Bandung yang diperlukan yang selanjutnya akan dikirim ke Google Maps server. Setelah mendapat data yang diperlukan dari databasenya, data tersebut dikembalikan ke *web server*, untuk selanjutnya bersama data lain yang terdapat pada database ditampilkan pada *web browser*.

3.2.2.3 *Sequence Diagram* Mencari Lokasi Daerah Banjir

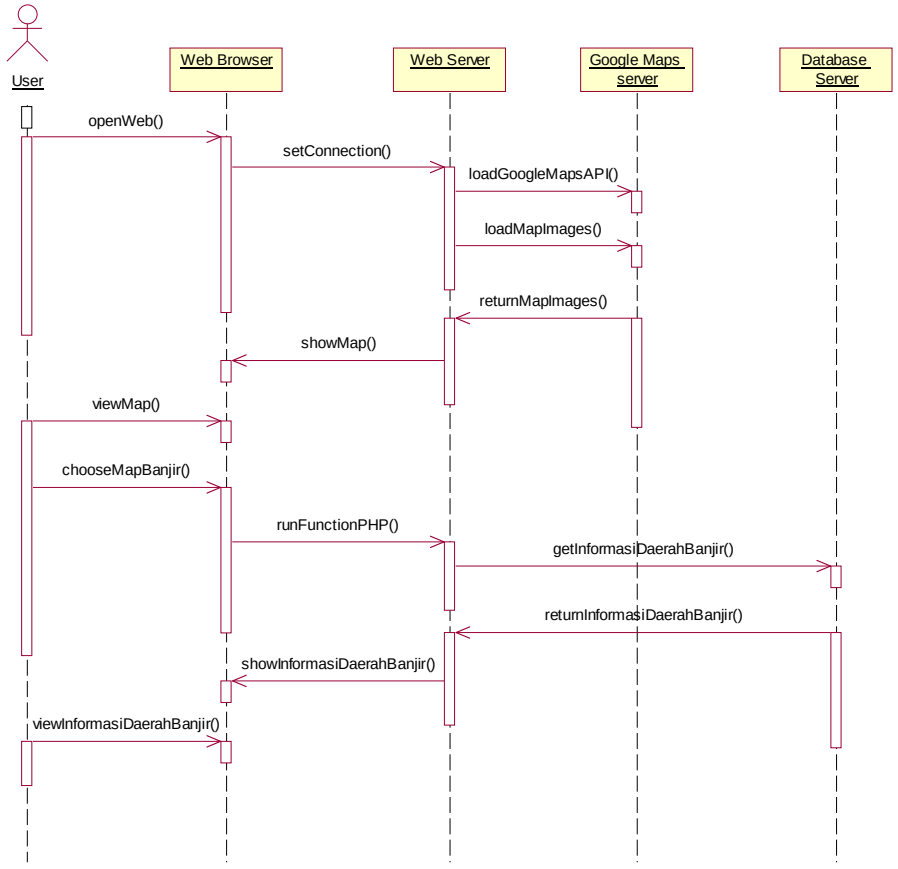


Gambar 3.10 Kecamatan Mencari Lokasi Daerah Banjir

Gambar *sequence diagram* diatas terdiri dari satu aktor yaitu *user* dan lima objek yaitu *web browser*, pencarian, *web server*, database, dan google map. Proses pertama yang terjadi adalah *user* membuka halaman situs melalui *web browser*,

yang akan menampilkan halaman utama web SIG banjir Bandung. Selanjutnya *user* memilih menu pencarian lokasi daerah banjir dan memasukkan nama daerah banjir yang di set oleh *web server*. Selanjutnya keyword nama daerah banjir tersebut akan di cek oleh database. Setelah pengecekan keyword oleh database, kemudian database akan meload peta ke Google Maps yang telah terkoneksi antara database dengan google maps tersebut. Selanjutnya setelah peta daerah banjir hasil dari pencarian tersedia, maka akan dikirim dan ditampilkan di *web browser*.

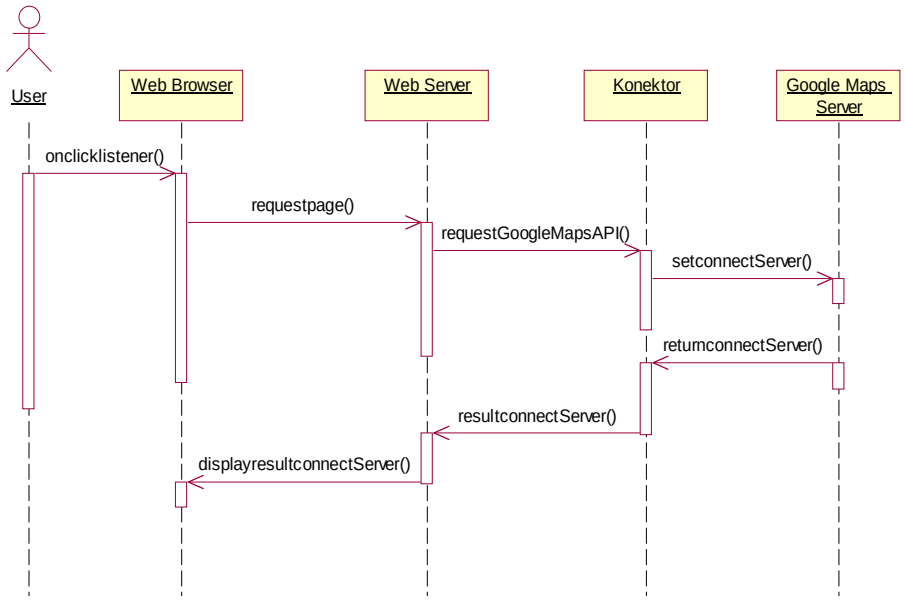
3.2.2.4 *Sequence Diagram* Melihat Informasi Daerah Banjir



Gambar 3.11 *Sequence Diagram* Melihat Informasi Daerah Banjir

Gambar *sequence diagram* diatas terdiri dari satu aktor yaitu *user* dan empat objek yaitu *web browser*, *web server*, google maps server dan google maps server.

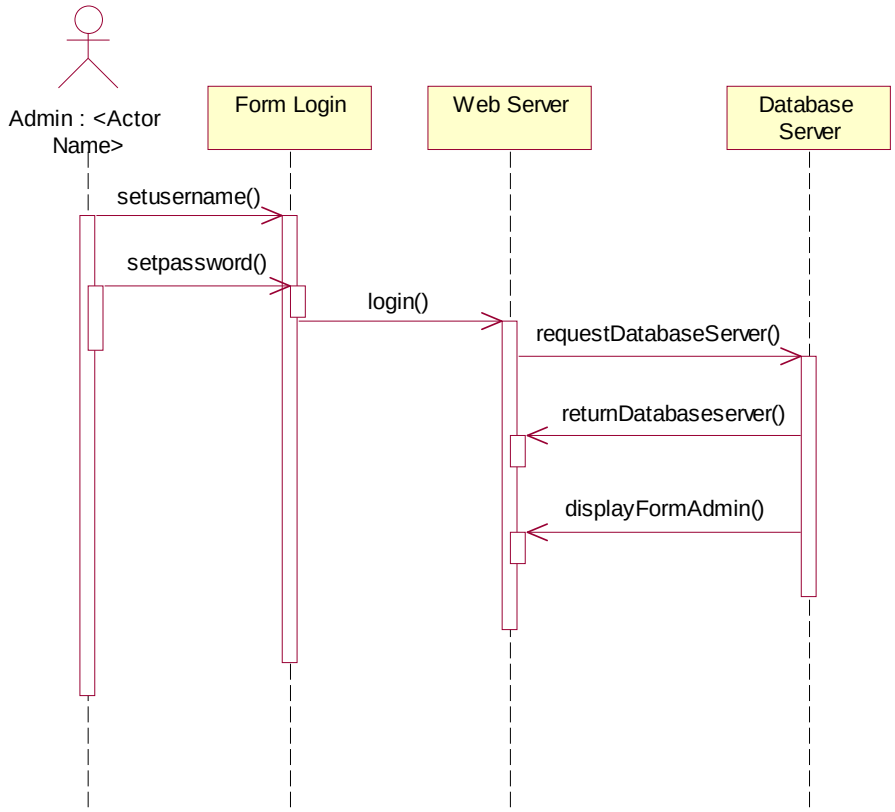
3.2.2.5 *Sequence Diagram* Koneksi Google Maps



Gambar 3.12 *Sequence Diagram* Koneksi Google Maps

Gambar *sequence diagram* diatas terdiri dari satu aktor yaitu *user* dan empat objek yaitu *web browser*, *web server*, konektor, dan google maps server. Proses pertama yang terjadi adalah *user* membuka halaman situs melalui *web browser*, yang akan mengirimkan request halaman ke *web server*. Selanjutnya *web server* mengirimkan request ke konektor untuk meminta google maps API key. Konektor disini yaitu koneksi internet berupa http. Kemudian google maps server akan menerima koneksi tersebut yang kemudian akan mengembalikan kembali hasil request koneksi tersebut ke konektor yang kemudian akan diteruskan ke *web browser*.

3.2.2.6 *Sequence Diagram* Login

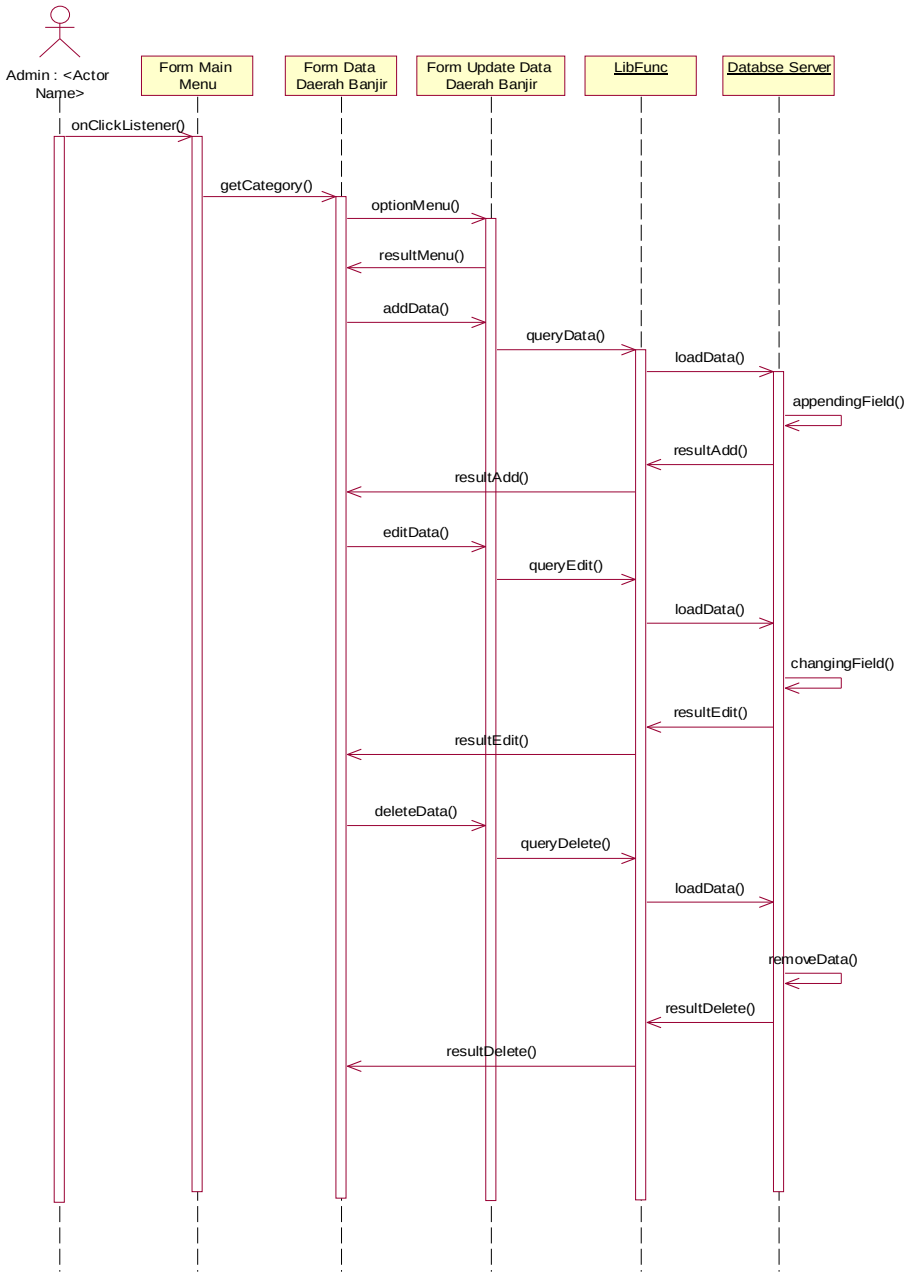


Gambar 3.13 *Sequence Diagram* Login

Gambar *sequence diagram* diatas terdiri dari satu aktor yaitu admin dan tiga objek yaitu form login, *web server*, database server. Proses pertama yang terjadi adalah admin membuka halaman login yaitu form login. Kemudian admin memasukkan *user* name dan password untuk dapat masuk ke halaman admin. Kemudian *web server* akan menerima *user*name dan password admin sebagai login admin. Selanjutnya *web server* akan mengirimkan request halaman admin ke database

server dan kemudian database server akan mengembalikan hasil request tadi dalam bentuk form admin, serta akan menampilkannya pada *web server*.

3.2.2.7 *Sequence Diagram* Mengelola Data Daerah Banjir



Gambar 3.14 *Sequence Diagram* Mengelola Data Daerah Banjir

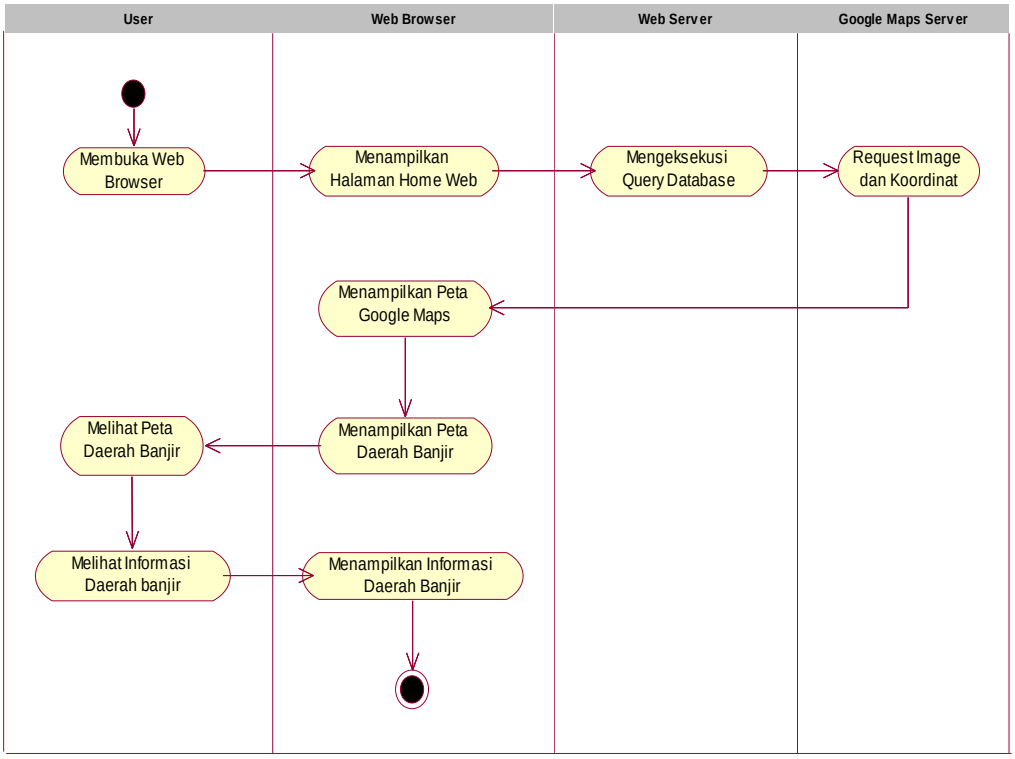
Gambar sequence diagram diatas terdiri dari satu aktor yaitu admin dan lima objek yaitu form main menu, form data daerah banjir, form *update* data daerah banjir, *lib function* dan database server.

3.2.3 *Activity Diagram*

Activity diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis, aliran kerja dalam banyak kasus, menggambarkan rangkaian aliran dari aktivitas, dan digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

Activity diagram mempunyai peran seperti halnya *flowchart*, akan tetapi perbedannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku *parallel* sedangkan *flowchart* tidak bisa.

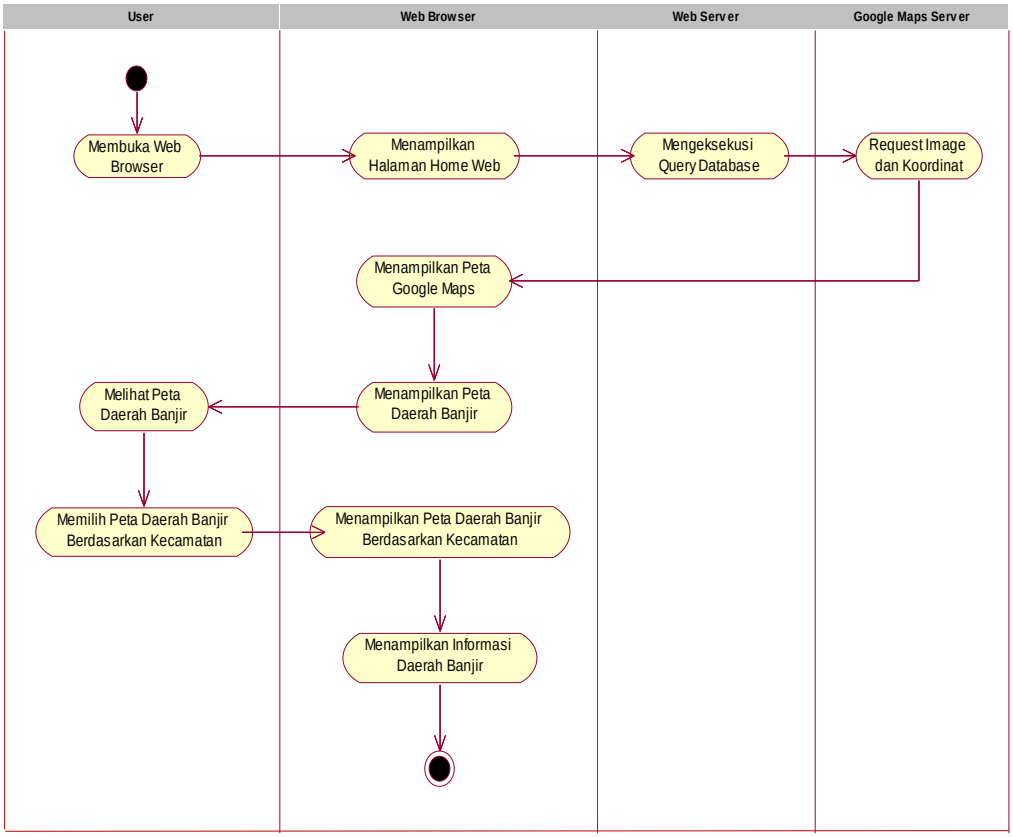
3.2.3.1 *Activity Diagram* Melihat Peta Daerah Banjir



Gambar 3.15 *Activity Diagram* Melihat Peta Daerah Banjir

Gambar *activity diagram* diatas menjelaskan tentang aktivitas yang terjadi pada halaman melihat peta daerah banjir Bandung yang dilakukan oleh *user*. Pada awalnya *user* membuka *web browser* pada *handphone,* kemudian sistem menampilkan halaman home yang selanjutnya akan menampilkan halaman peta daerah banjir Bandung. Selanjutnya *user* dapat melihat legenda daerah banjir disertai dengan informasi-informasi mengenai daerah banjir Bandung tersebut.

3.2.3.2 *Activity Diagram* Melihat Peta Daerah Banjir Berdasarkan Kecamatan

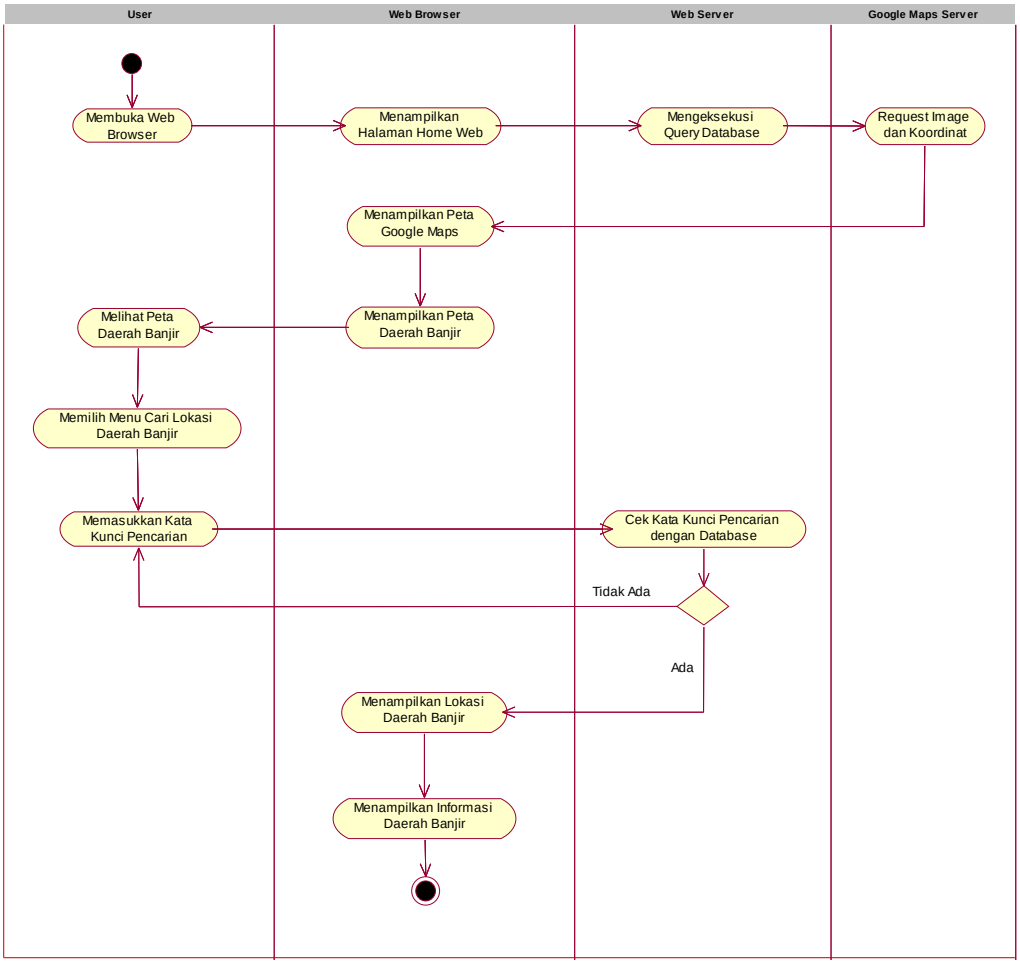


Gambar 3.16 Activity Diagram Melihat Peta Daerah Banjir Berdasarkan Kecamatan

Gambar *activity diagram* diatas menjelaskan tentang aktivitas yang terjadi pada halaman melihat peta daerah banjir berdasarkan Kecamatan di Bandung yang dilakukan oleh *user*. Pada awalnya *user* membuka *web browser* pada *handphone*, kemudian sistem menampilkan halaman *home* yang selanjutnya akan menampilkan halaman peta daerah banjir Bandung. Selanjutnya *user* dapat memilih menu melihat peta daerah banjir berdasarkan Kecamatan di Bandung

yang akan menampilkan legenda daerah banjir berdasarkan Kecamatan disertai dengan informasi-informasi mengenai daerah banjir Bandung tersebut.

3.2.3.3 Activity Diagram Mencari Lokasi Daerah Banjir

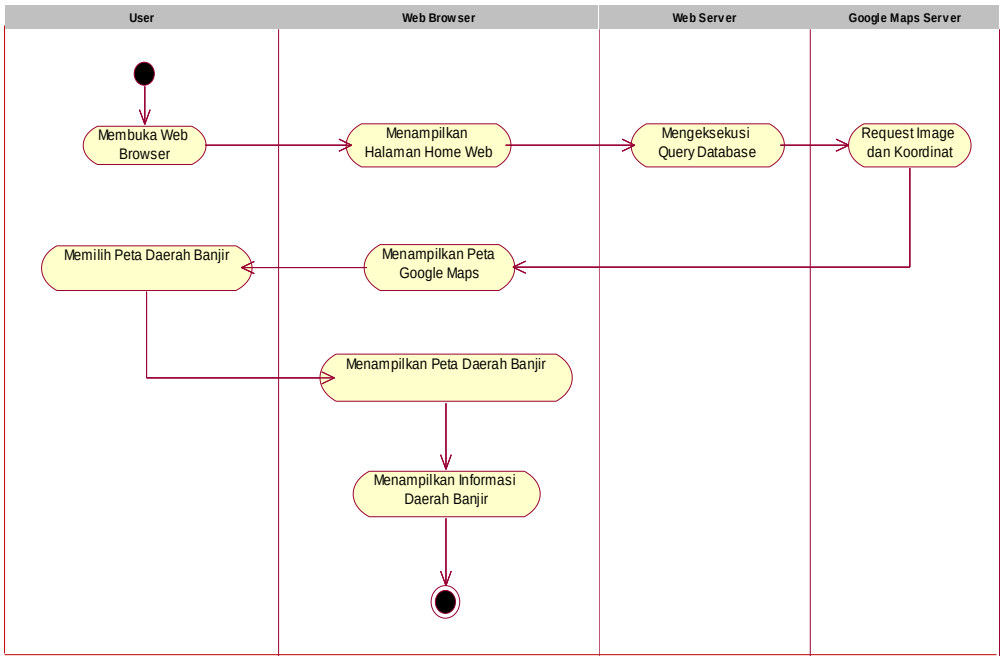


Gambar 3.17 Activity Diagram Mencari Lokasi Daerah Banjir

Gambar *activity diagram* diatas menjelaskan tentang aktivitas yang terjadi pada halaman mencari lokasi daerah banjir Bandung yang dilakukan oleh *user*. Pada awalnya *user* membuka *web browser* pada *handphone,* kemudian sistem menampilkan halaman home yang selanjutnya akan menampilkan halaman peta daerah banjir Bandung. Selanjutnya *user* dapat memilih menu pecarian lokasi

daerah banjir di Bandung yang selanjutnya akan menampilkan peta daerah banjir sesuai dari hasil pencarian yang dimasukkan oleh *user*, yang dilengkapi dengan informasi-informasi mengenai daerah banjir Bandung tersebut.

3.2.3.4 *Activity Diagram* Melihat Informasi Daerah Banjir

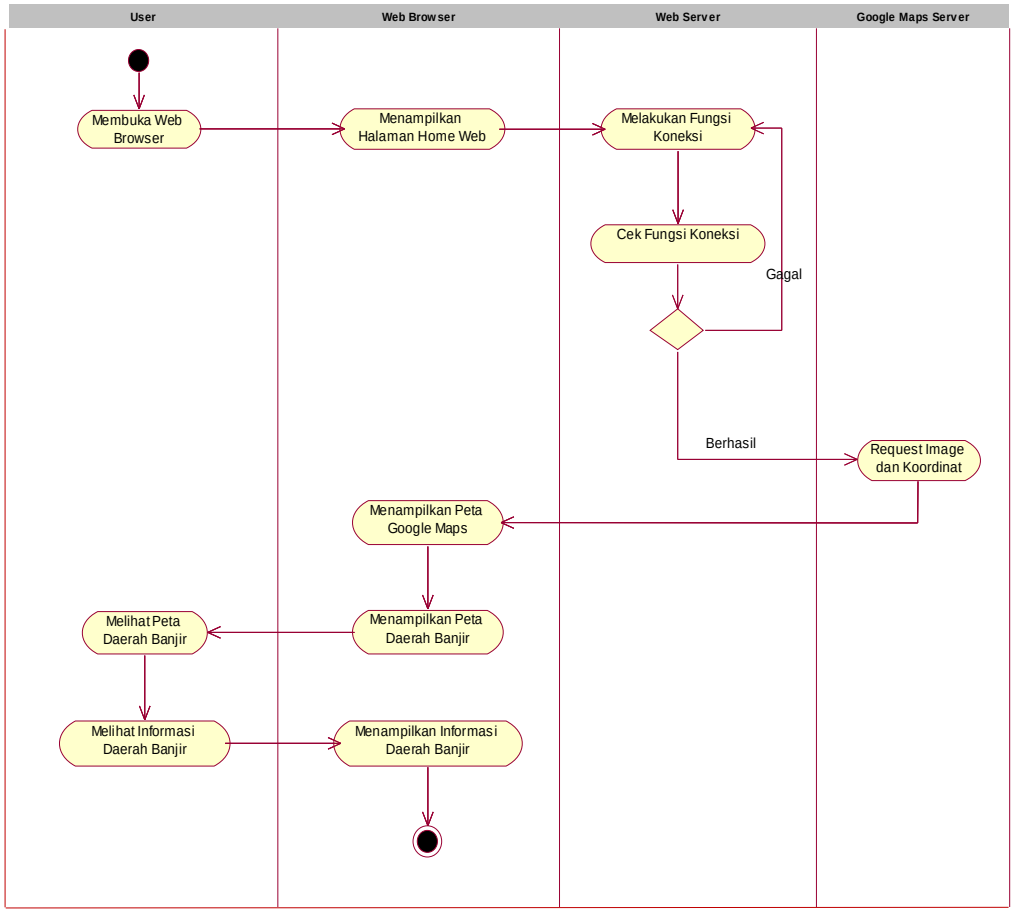


Gambar 3.18 *Activity Diagram* Melihat Informasi Daerah Banjir

Gambar *activity diagram* diatas menjelaskan tentang aktivitas yang terjadi pada halaman melihat informasi daerah banjir yang dilakukan oleh *user*. Pada awalnya *user* membuka *web browser* pada *handphone*, kemudian sistem menampilkan halaman web yang selanjutnya akan menampilkan halaman peta daerah banjir Bandung. Selanjutnya *user* dapat memilih menu melihat peta daerah banjir berdasarkan Kecamatan di Bandung yang akan menampilkan legenda daerah

banjir berdasarkan Kecamatan disertai dengan informasi-informasi mengenai daerah banjir Bandung tersebut.

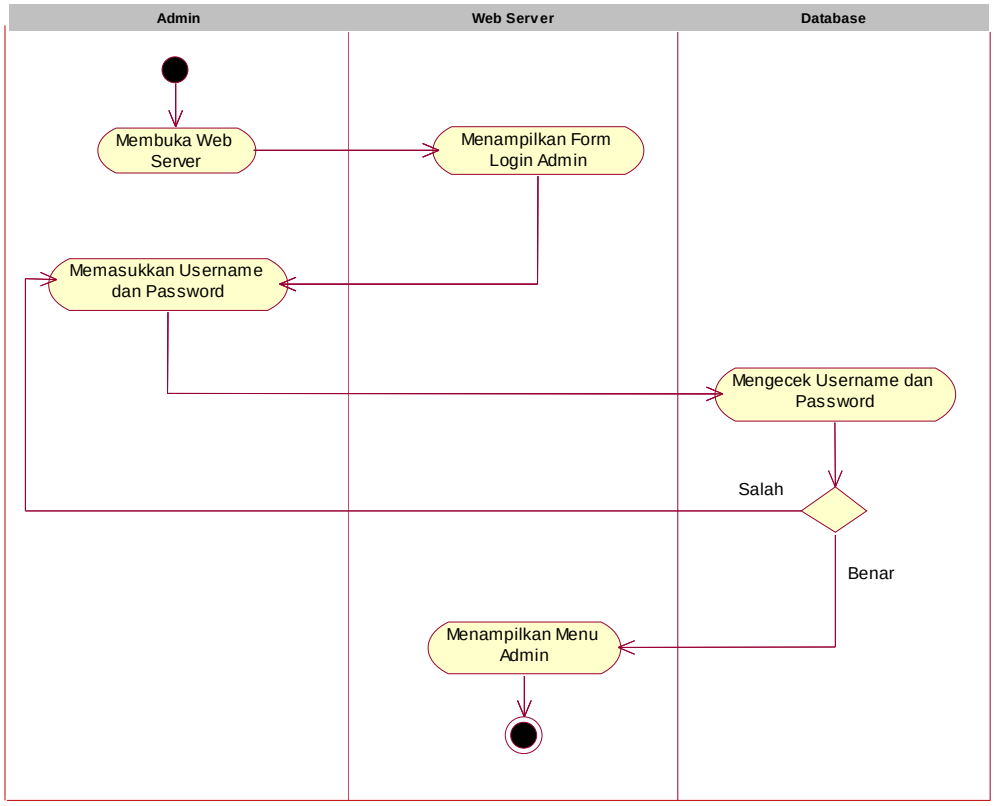
3.2.3.5 *Activity Diagram* Koneksi Google Maps



Gambar 3.19 *Activity Diagram* Koneksi Google Maps

Gambar *activity diagram* diatas menjelaskan tentang aktivitas yang terjadi pada koneksi google maps yang dilakukan oleh *user*. Pada awalnya *user* membuka *web browser* pada *handphone,* kemudian sistem akan melakukan koneksi ke google maps, dimana untuk melakukan koneksi ke google maps ini harus terlebih dahulu melakukan koneksi internet.

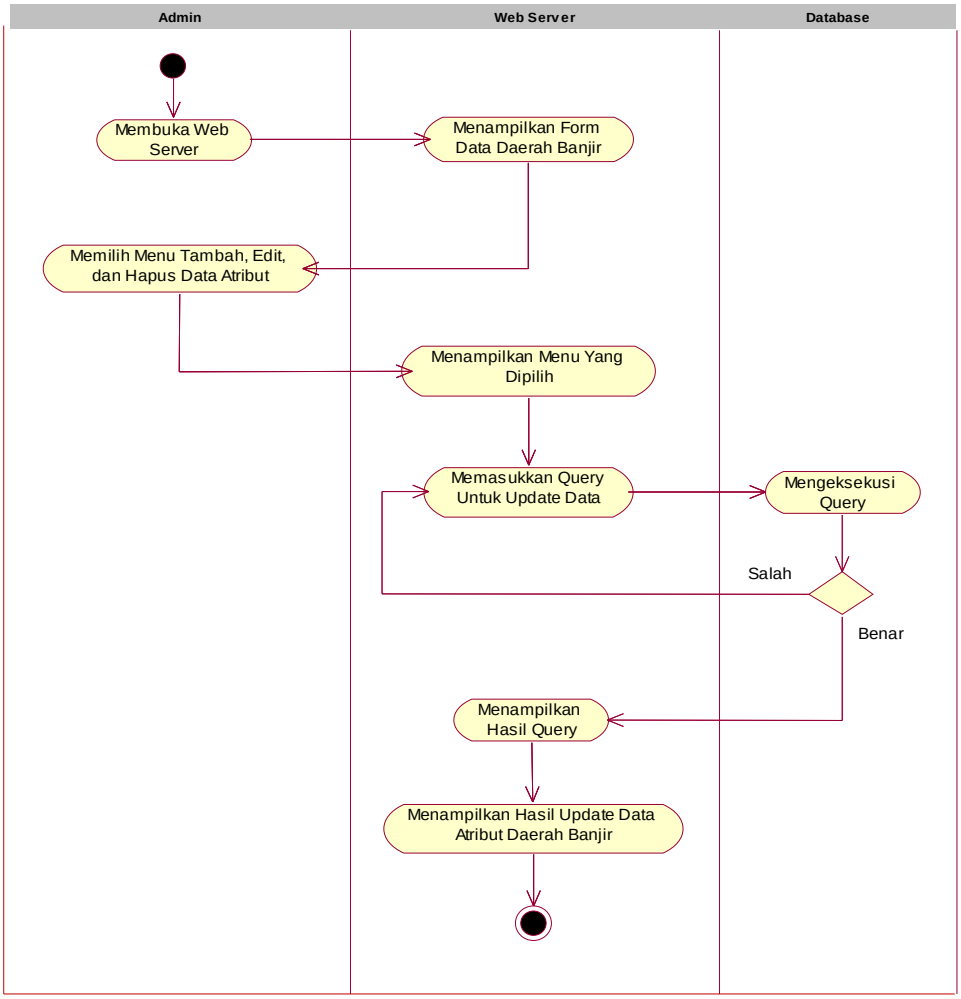
3.2.3.6 *Activity Diagram* Login



Gambar 3.20 *Activity Diagram Login*

Gambar *activity diagram* diatas menjelaskan tentang aktivitas yang terjadi pada halaman login yang dilakukan oleh admin. Pada awalnya admin membuka *web browser* pada halaman login, kemudian sistem menampilkan halaman login. Selanjutnya admin memasukkan *user*name dan password untuk dapat masuk ke halaman admin, kemudian sistem akan menampilkan halaman admin. Dimana dalam halaman admin ini, admin dapat melakukan kegiatan yang berhubungan dengan mengolah data atribut dan data spasial daerah banjir di Bandung serta data admin.

3.2.3.7 *Activity Diagram* Mengelola Data Daerah Banjir



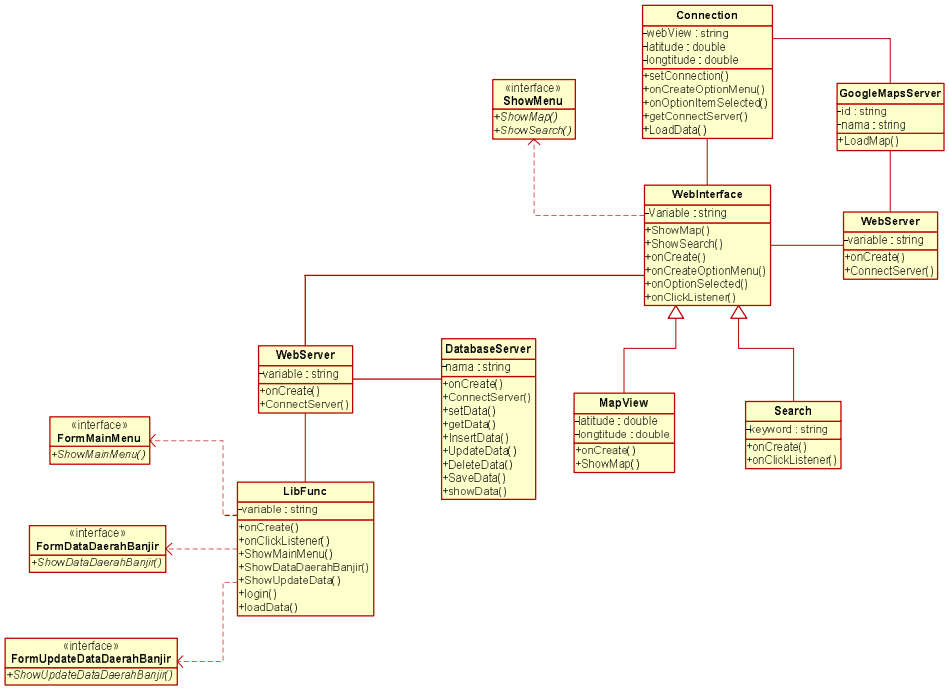
Gambar 3.21 *Activity Diagram* Mengelola Data Daerah Banjir

Gambar *activity diagram* diatas menjelaskan tentang aktivitas yang terjadi pada halaman mengelola data daerah banjir yang dilakukan oleh admin. Pada awalnya admin membuka *web browser* pada halaman login, kemudian sistem menampilkan halaman login. Selanjutnya admin memasukkan *user*name dan password untuk dapat masuk ke halaman admin, kemudian sistem akan

menampilkan halaman admin. Dimana dalam halaman admin ini, admin dapat melakukan kegiatan yang berhubungan dengan mengelola data atribut dan data spasial daerah banjir di Bandung serta data admin. Selanjutnya admin dapat melakukan tambah data, edit data, dan hapus data.

3.2.4 *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. Dibawah ini adalah *class* diagram dari aplikasi sistem informasi geografis untuk mengidentifikasi daerah rawan banjir di Bandung berbasis *mobile*:



Gambar 3.22 *Class Diagram* Aplikasi SIG Banjir Bandung Berbasis *Mobile*

Tabel 3.11 Deskripsi *Class Diagram*

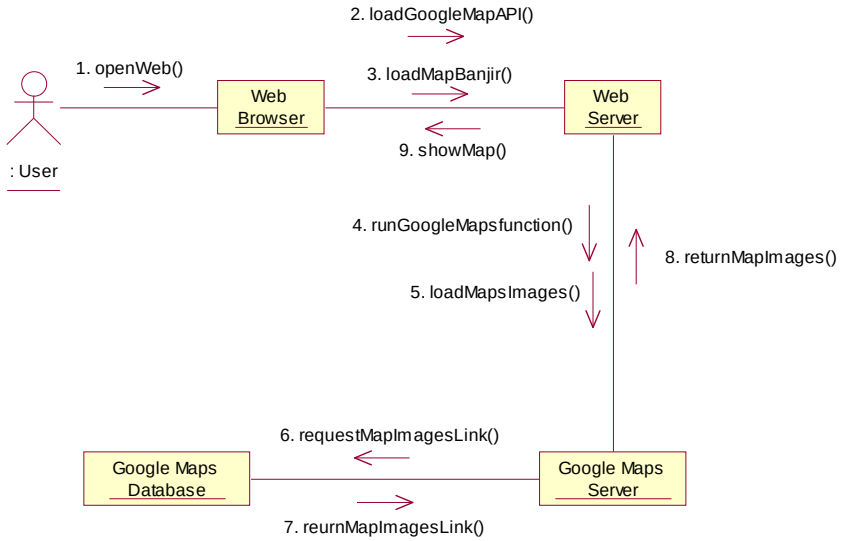
Class	Jenis *Class*	Deskripsi
WebInterface	*Entity*	*Class* yang berisikan method-method yang berhubungan dengan variabel-variabel yang akan digunakan dalam query ke Server
ShowMenu	*Interface*	*Class* menu yang berisikan method-method menu tambahan
MapView	*Control*	*Class* yang berisikan method-method yang berhubungan dengan menampilkan peta
Search	*Control*	*Class* yang berisikan method-method yang berhubungan dengna Pencarian suatu lokasi tertentu
Connection	*Control*	*Class* yang berisikan method-method yang menangani koneksi ke Server
FormMainMenu	*Interface*	*Class* utama yang berisikan method-method pilihan menu admin.
FormDataDaerahBanjir	*Interface*	*Class* yang berisikan method-method untuk menampilkan data daerah banjir.
FormUpdateDataDaerah Banjir	*Interface*	*Class* yang berisikan form update data daerah banjir.
WebInterface	*Entity*	*Class* yang berisikan method-

		method yang berhubungan dengan variabel-variabel yang akan digunakan dalam query ke Server
GoogleMapsServer	*Entity*	*Class* yang berisikan method-method terkoneksi dengan WebServer, untuk dapat menampilkan petagooglemaps
LibraryFunc	*Control*	*Class* yang berisikan method-method untuk mengontrol class interface dan class entity.
DatabaseServer	*Entity*	*Class* yang menyimpan data-data daerah banjir

3.2.5 *Collaboration Diagram*

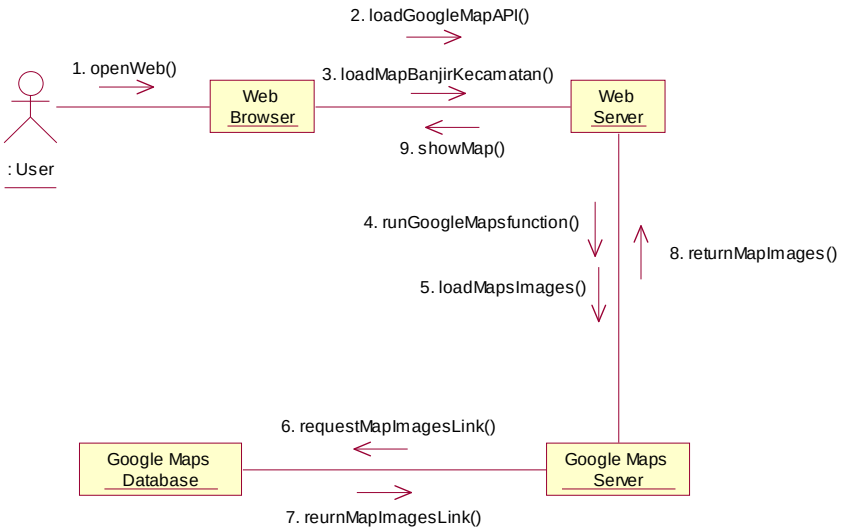
Collaboration diagram adalah diagram yang mengelompokkan pesan pada kumpulan *sequence diagram* menjadi sebuah diagram. Dalam *collaboration diagram* terdapat method yang dijalankan antara objek yang satu dan objek yang lainnya. Di *collaboration diagram* ini, objek harus melakukan sinkronisasi pesan dengan serangkaian pesan-pesan lainnya.

3.2.5.1 *Collaboration Diagram* Melihat Peta Daerah Banjir Bandung



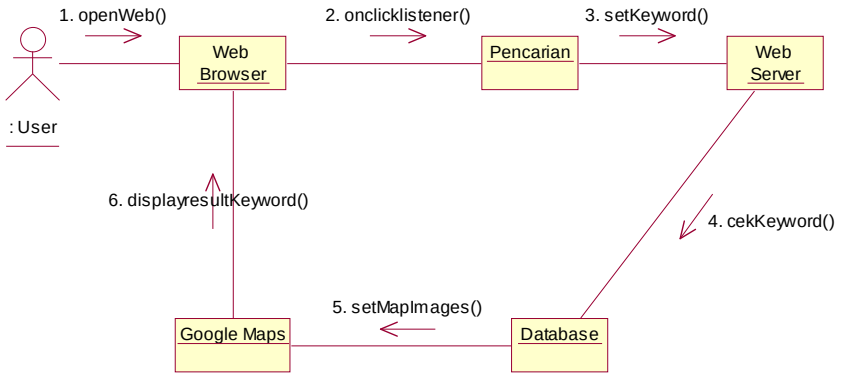
Gambar 3.23 *Collaboration Diagram* Melihat Peta Daerah Banjir Bandung

3.2.5.2 *Collaboration Diagram* Melihat Peta Daerah Banjir Bandung Berdasarkan Kecamatan



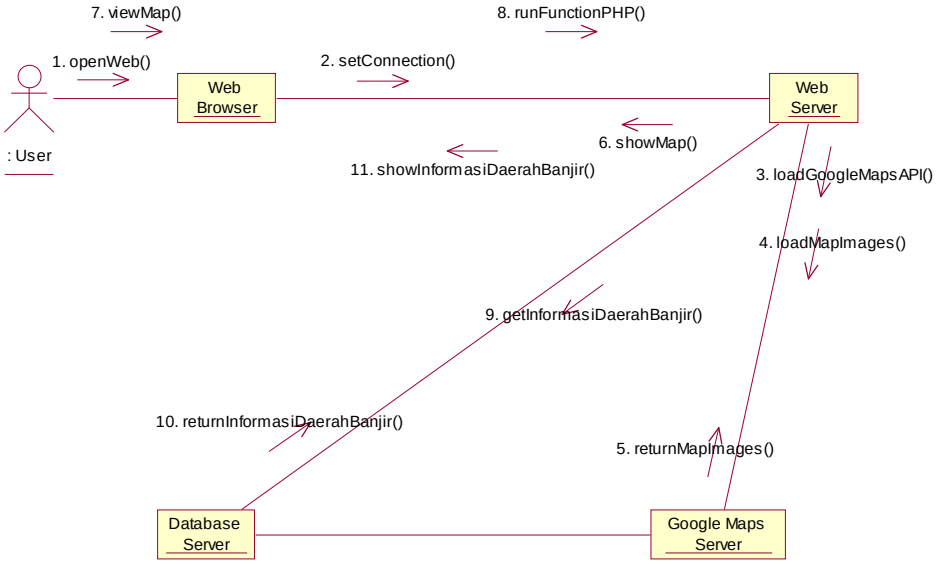
Gambar 3.24 *Collaboration Diagram* Melihat Peta Daerah Banjir Bandung Berdasarkan Kecamatan

3.2.5.3 *Collaboration Diagram* Mencari Lokasi Daerah Banjir



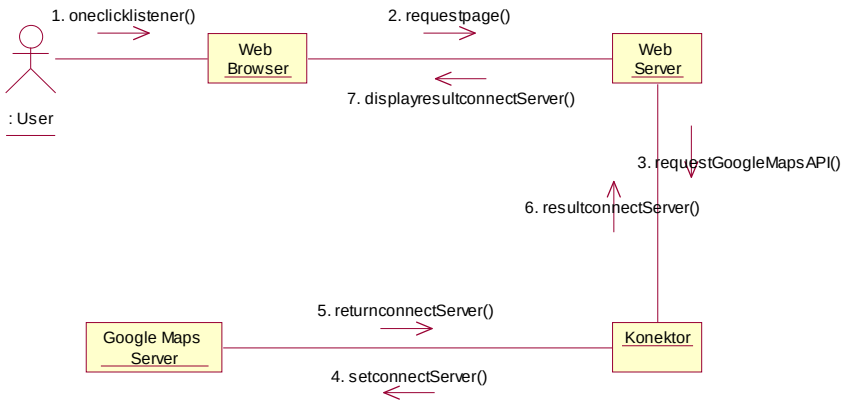
Gambar 3.25 *Collaboration Diagram* Mencari Lokasi Daerah Banjir

3.2.5.4 *Collaboration Diagram* Melihat Informasi Daerah Banjir



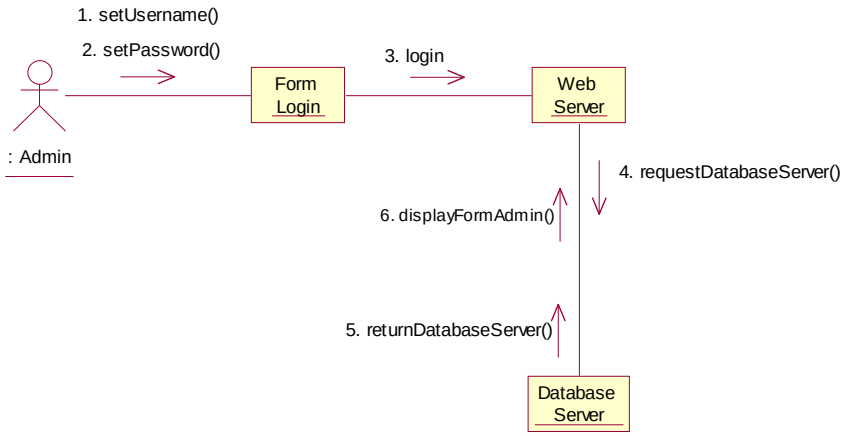
Gambar 3.26 *Collaboration Diagram* Menampilkan Informasi Daerah Banjir

3.2.5.5 *Collaboration Diagram* Koneksi Google Maps



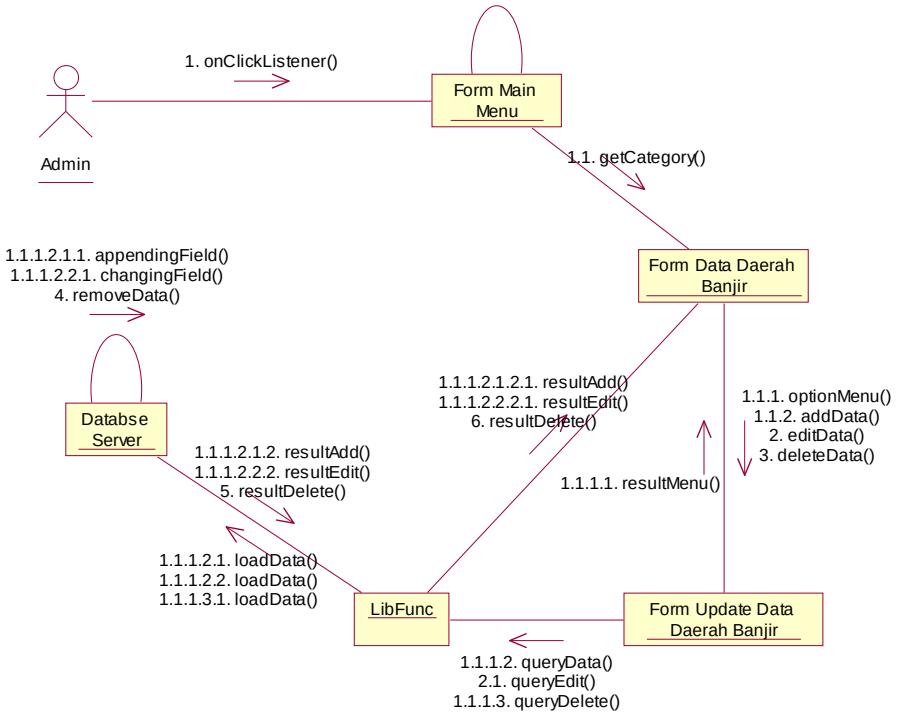
Gambar 3.27 *Collaboration Diagram* Koneksi Google Maps

3.2.5.6 *Collaboration Diagram* Login



Gambar 3.28 *Collaboration Diagram* Login

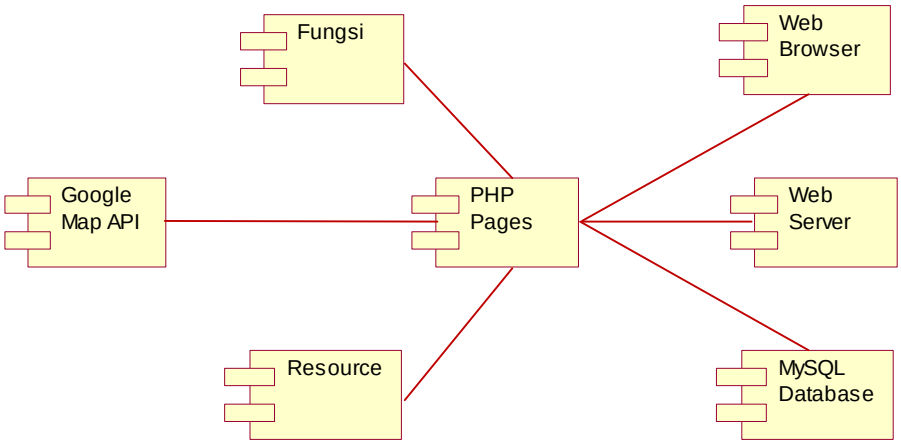
3.2.5.7 *Collaboration Diagram* Mengelola Data Daerah Banjir



Gambar 3.29 *Collaboration Diagram* Mengelola Data Daerah Banjir

3.2.6 *Component Diagram*

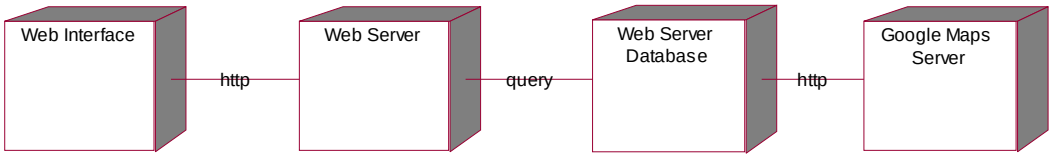
Component diagram adalah bagian fisik dari sebuah sistem, karena menetap di komputer, bukan di benak para analis. Komponen bisa berupa tabel, file data, file exe, dan lain-lain. Diagram ini menggambarkan struktur dan hubungan antara komponent perangkat lunak, termasuk ketergantungan (*dependency*). Diataranya modul berisi kode, baik berisi source kode, *binary*, *library*, *executable*. *Compoenent diagram* ditunjukkan pada gambar berikut:



Gambar 3.30 *Component Diagram*

3.2.7 *Deployment Diagram*

Deployment diagram menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian software yang berjalan pada bagian-bagian hardware. Deployment diagram menggambarkan detail bagaimana komponen dibentuk dan didistribusikan *(deploy)* dalam infrastruktur sistem. Dimana komponen akan terletak pada mesin, server atau peranti keras. Bagaimana jaringan pada lokasi tersebut, misalnya server, client dan hal-hal lain yang bersifat fisik.



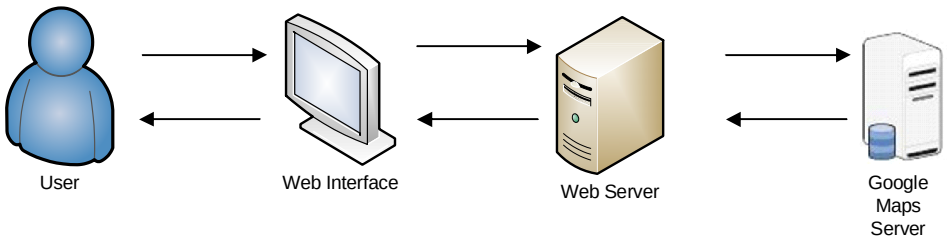
Gambar 3.31 *Deployment Diagram*

3.2.8 Perancangan Sistem

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

3.2.8.1 Perancangan Arsitektur Sistem

Arsitektur fisik sistem terdiri dari empat bagian utama yaitu *user*, web interface, *web server* dan google maps server. Arsitektur sistem dalam pembangunan aplikasi sistem informasi geografis banjir Bandung berbasis *mobile* ini ditunjukan oleh gambar 3.26 dibawah ini.



Gambar 3.32 Arsitektur Sistem

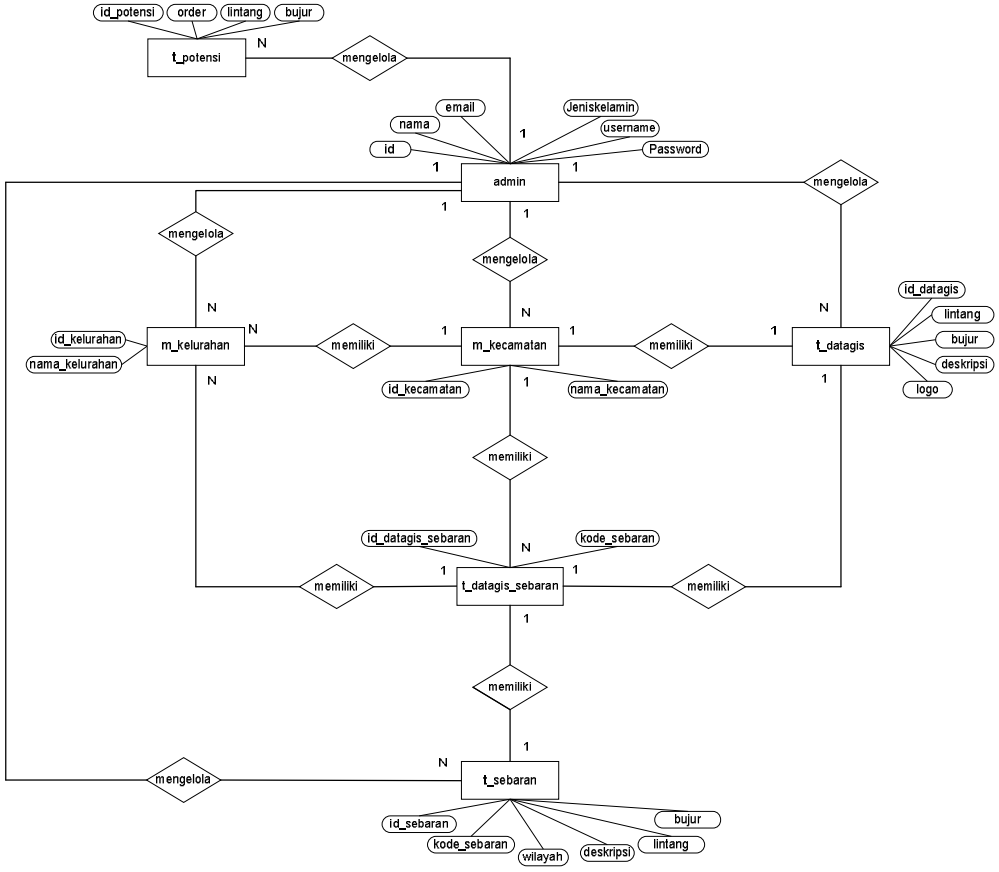
Aplikasi ini akan dijalankan oleh pengguna yang berinteraksi dengan *web interface*. Pengguna dapat menggunakan *web browser* pada *handphone* seperti Opera. Apabila situs peta ini dibuka, maka browser akan mengunduh tampilan situs dengan program berbasis Google Maps API yang terdapat pada *web server*. Program dengan tampilan web inilah yang berinteraksi secara interaktif dengan pengguna.

Apabila pengguna melakukan suatu perintah, eksekusinya akan diproses di *web server*. Sedangkan bila terdapat permintaan untuk mengakses database, maka database tersebut akan dipanggil ke dalam program. Lalu dilakukan request data yang diminta ke *web server* Google Maps. Hasilnya adalah berupa gambar peta, serta objek-objek yang dimiliki oleh Google Maps. Hasil ini dikembalikan ke *web server*, untuk selanjutnya dikembalikan ke *web browser*.

3.2.8.2 Perancangan Basis Data

Perancangan basis data dibangun berdasarkan kebutuhan akan informasi dalam suatu sistem. Oleh sebab itu pada umumnya perancangan basis data dimulai dari pengamatan kebutuhan dalam suatu sistem yang akan dibangun.

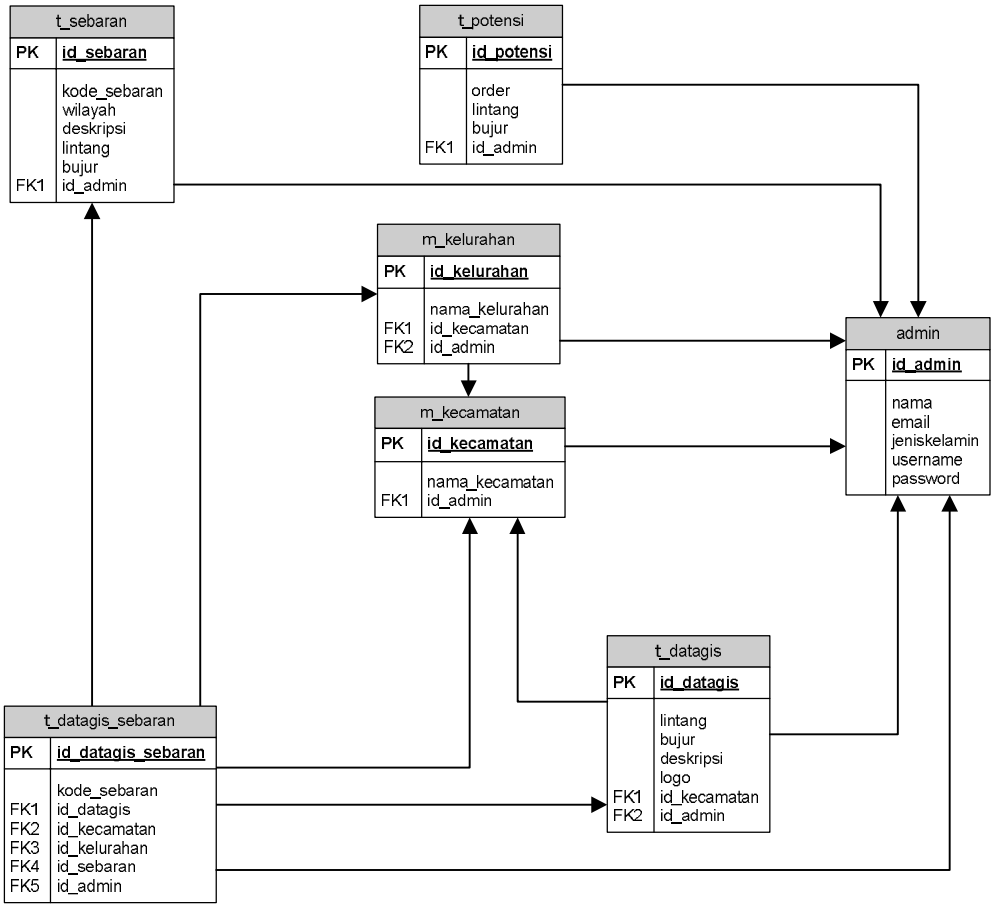
Dari hasil analisis, diperoleh data yang akan digunakan dalam proses pembangunan aplikasi SIG banjir Bandung berbasis web *mobile*, yang kemudian dari data yang telah diperoleh, dibangun sebuah perancangan basis data dengan menggunakan *Entity Relationship Diagram* (ERD) yang ditunjukan pada gambar dibawah ini :



Gambar 3.33 *Entity Relationship Diagram* (ERD) SIG Banjir Bandung

3.2.8.3 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 3.34 Skema Relasi SIG Banjir Bandung

3.2.8.4 Struktur Tabel

Dari hasil analisis yang telah dilakukan dan dari hasil perancangan *Entity Relationship Diagram* (ERD) maka tabel yang terbentuk adalah :

3.2.8.4.1 Tabel admin

Tabel admin digunakan untuk mencatat data penting yang merupakan identitas admin. Secara umum, struktur tabel yang akan digunakan untuk mencatat data admin dapat dilihat pada tabel 3.12

Tabel 3.12 Tabel admin

Field	Tipe Data	Kunci	Keterangan
id_admin	int(3)	*Primary key,Auto increment*	*Not Null*
nama	varchar(50)	-	*Not Null*
email	varchar (25)	-	*Not Null*
jeniskelamin	varchar(10)	-	*Not Null*
username	varchar(25)	-	*Not Null*
password	varchar(50)	-	*Not Null*

3.2.8.4.2 Tabel m_kecamatan

Tabel m_kecamatan digunakan untuk mencatat data penting yang merupakan identitas kecamatan. Secara umum, struktur tabel yang akan digunakan untuk mencatat data kecamatan dapat dilihat pada tabel 3.13

Tabel 3.13 Tabel m_kecamatan

Field	Tipe Data	Kunci	Keterangan
id_kecamatan	int(3)	*Primary key,Auto increment*	*Not Null*
nama_kecamatan	varchar(50)	-	*Not Null*

3.2.8.4.3 Tabel m_kelurahan

Tabel kelurahan digunakan untuk mencatat data penting yang merupakan identitas dari kelurahan. Secara umum, struktur tabel yang akan digunakan untuk mencatat data kelurahan dapat dilihat pada tabel 3.14

Tabel 3.14 Tabel m_kelurahan

Field	Tipe Data	Kunci	Keterangan
id_kelurahan	int(3)	*Primary key,Auto increment*	*Not Null*
id_kecamatan	int(3)	*Foreign Key to* m_kecamatan (‘id_kecamatan’)	*Not Null*
nama_kelurahan	varchar(50)		*Null*

3.2.8.4.4 Tabel t_datagis

Tabel datagis digunakan untuk mencatat data penting yang merupakan identitas datagis. Secara umum, struktur tabel yang akan digunakan untuk mencatat data datagis dapat dilihat pada tabel 3.15

Tabel 3.15 Tabel t_datagis

Field	Tipe Data	Kunci	Keterangan
id_datagis	int(3)	*Primary key,Auto increment*	*Not Null*
id_kecamatan	int(3)	*Foreign Key to* m_kecamatan (‘id_kecamatan’)	*Not Null*
lintang	double(8,5)	-	*Not Null*
bujur	double(8,5)	-	*Not Null*
deskripsi	varchar(400)	-	*Not Null*
logo	varchar(100)	-	*Not Null*

3.2.8.4.5 Tabel t_datagis_sebaran

Tabel datagis sebaran digunakan untuk mencatat data penting yang merupakan identitas sebaran. Secara umum, struktur tabel yang akan digunakan untuk mencatat datagis sebaran dapat dilihat pada tabel 3.16

Tabel 3.16 Tabel t_datagis_sebaran

Field	Tipe Data	Kunci	Keterangan
id_datagis_sebaran	int(5)	*Primary key,Auto increment*	*Not Null*
id_datagis	int(3)	*Foreign Key to* t_datagis (‘id_datagis’)	*Not Null*
id_kecamatan	int(3)	*Foreign Key to* m_kecamatan (‘id_kecamatan’)	*Not Null*
id_kelurahan	int(3)	*Foreign Key to* m_kelurahan (‘id_kelurahan)	*Not Null*
id_sebaran	int(5)	*Foreign Key to* t_sebaran (‘id_sebaran)	
kode_sebaran	varchar(5)		*Not Null*

3.2.8.4.6 Tabel t_potensi

Tabel potensi digunakan untuk mencatat data penting yang merupakan potensi banjir. Secara umum, struktur tabel yang akan digunakan untuk mencatat potensi banjir dapat dilihat pada tabel 3.17

Tabel 3.17 Tabel t_potensi

Field	Tipe Data	Kunci	Keterangan
id_potensi	int(5)	*Primary key,Auto increment*	*Not Null*
order	int(2)	-	*Not Null*
lintang	double(8,5)	-	*Not Null*
bujur	double(8,5)	-	*Not Null*

3.2.8.4.7 Tabel t_sebaran

Tabel sebaran digunakan untuk mencatat data penting yang merupakan sebaran banjir. Secara umum, struktur tabel yang akan digunakan untuk mencatat sebaran banjir dapat dilihat pada tabel 3.18

Tabel 3.18 Tabel t_sebaran

Field	Tipe Data	Kunci	Keterangan
id_sebaran	int(5)	*Primary key,Auto increment*	*Not Null*
kode_sebaran	varchar(5)	-	*Not Null*
wilayah	text	-	*Not Null*
deskripsi	varchar(400)	-	*Not Null*
lintang	double(8,5)	-	*Not Null*
bujur	double(8,5)	-	

3.2.9 Perancangan Kode

Berikut ini adalah perancangan kode yang di buat untuk semua kode yang ada dalam sistem yang akan di bangun :

3.2.9.1 Kode Untuk Datagis Sebaran

Kode yang digunakan pada pengkodean datagis sebaran yaitu hanya menggunakan maksimal 2 digit yang terdiri dari huruf dan angka. Untuk kode datagis sebaran ditentukan berdasarkan data yang diperoleh dari Puslit Geoteknologi Lembaga Ilmu Pengetahuan Indonesia (LIPI) Bandung. Dimana kode legenda sebaran dimulai dari huruf abjad pertama. Berikut adalah beberapa contoh kode untuk legenda sebaran:

a



Menunjukan kode sebaran yang menunjukkan cakupan wilayah banjir

k2

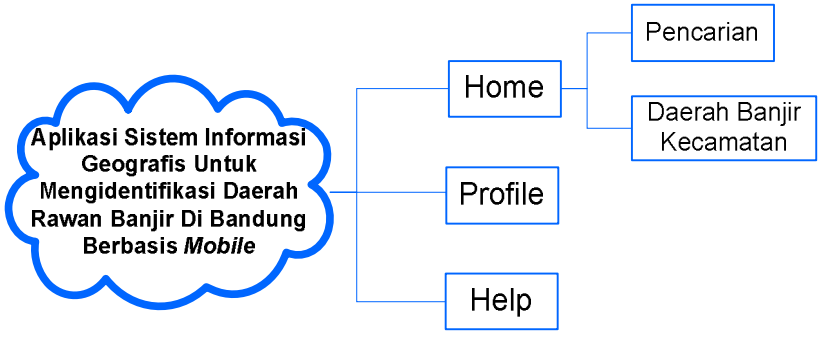
Menunjukan kode sebaran pelengkap

Menunjukan kode sebaran yang menunjukkan cakupan wilayah banjir

3.2.10 Perancangan Struktur Menu

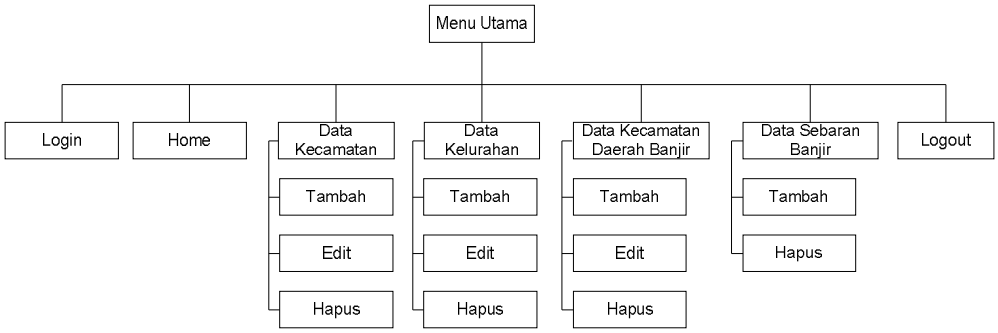
Perancangan struktur menu dari aplikasi sistem informasi geografis untuk mengidentifikasi daerah rawan banjir di Bandung berbasis *mobile* adalah sebagai berikut:

3.2.10.1 Perancangan Struktur Menu Sistem Informasi Geografis Banjir Bandung Berbasis *Mobile*



Gambar 3.35 Perancangan Struktur Menu Sistem Informasi Geografis Banjir Bandung Berbasis *Mobile*

3.2.10.2 Perancangan Struktur Menu Admin



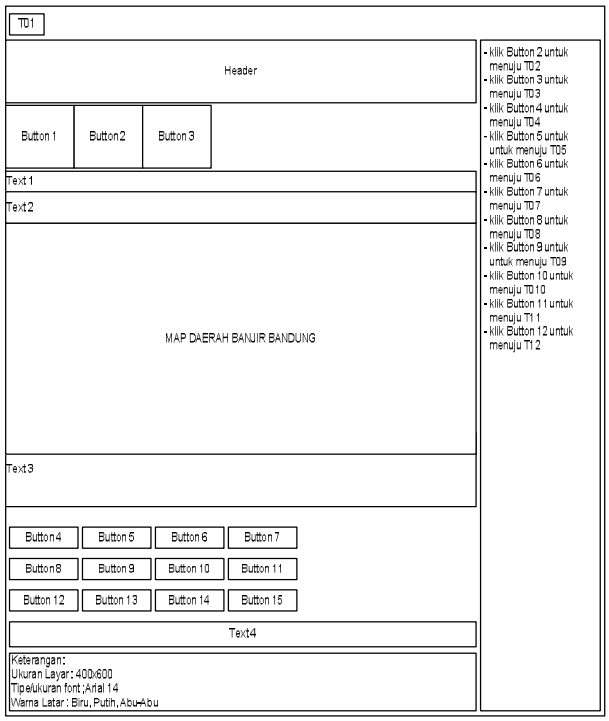
Gambar 3.36 Perancangan Struktur Menu Admin

3.2.11 Perancangan Antarmuka

Perancangan antarmuka merupakan perancangan yang dibuat untuk memeberikan gambaran sebelum aplikasi dibangun. Perancangan antar muka *input/output*, struktur dan tampilan dirancang sedemikian rupa agar mudah dioperasikan oleh *user*.

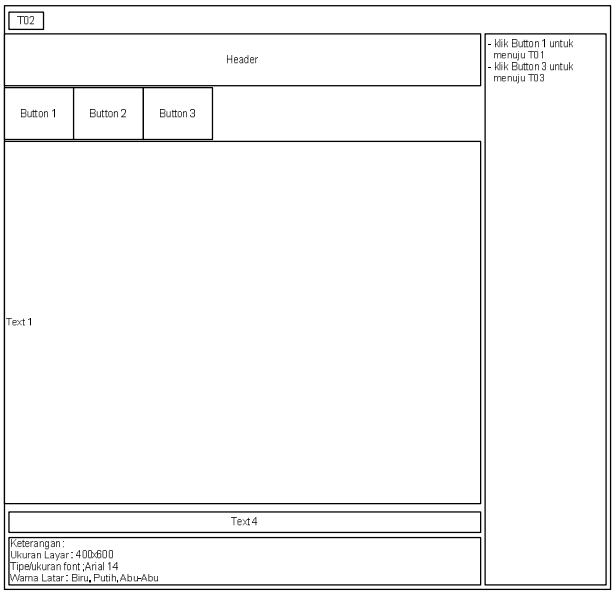
3.2.11.1 Perancangan Antar Muka *Frontend*

3.2.11.1.1 Perancangan Antar Muka Menu *Home*



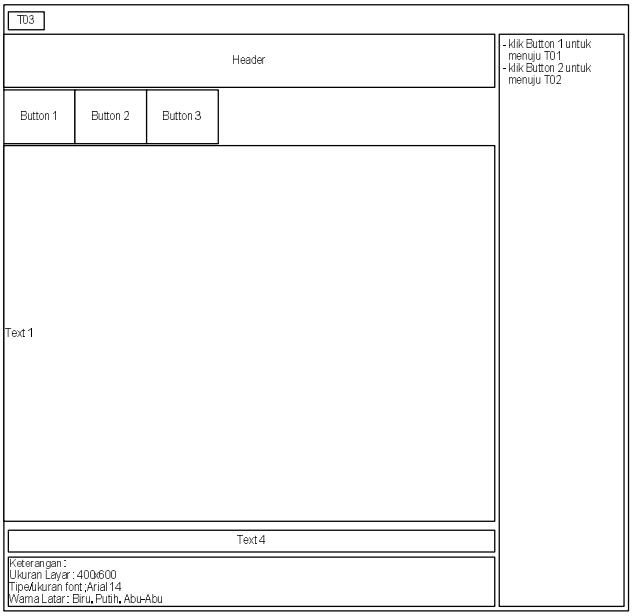
Gambar 3.37 Perancangan Antar Muka Menu *Home*

3.2.11.1.2 Perancangan Antar Muka Menu *Profile*



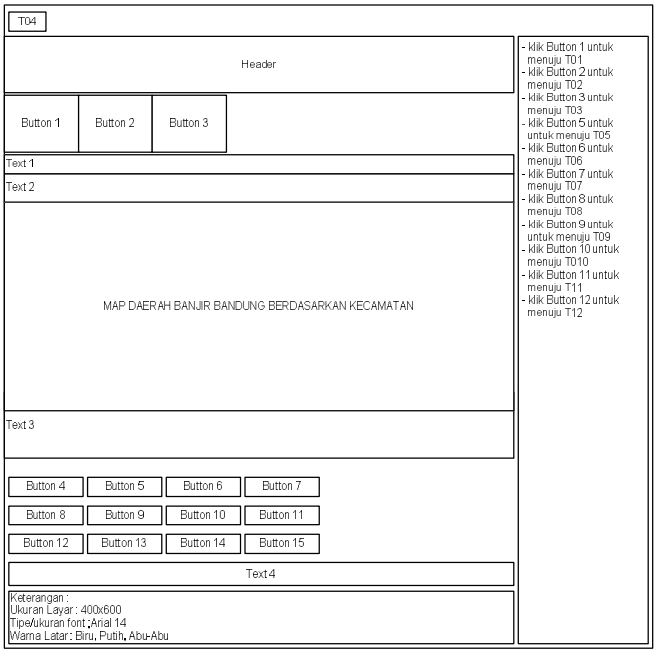
Gambar 3.38 Perancangan Antar Muka Menu *Profile*

3.2.11.1.3 Perancangan Antar Muka Menu *Help*



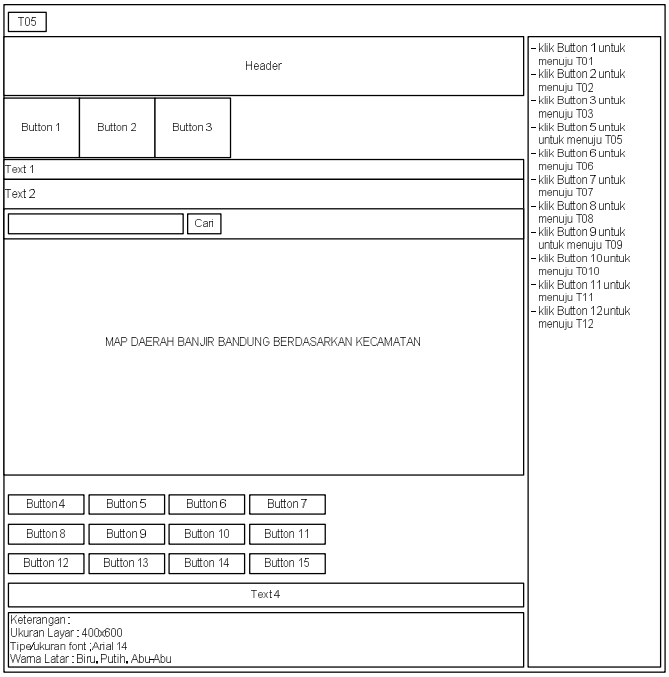
Gambar 3.39 Perancangan Antar Muka Menu *Help*

3.2.11.1.4 Perancangan Antar Muka Menu Daerah Banjir Berdasarkan Kecamatan



Gambar 3.40 Perancangan Antar Muka Menu Daerah Banjir Berdasarkan Kecamatan

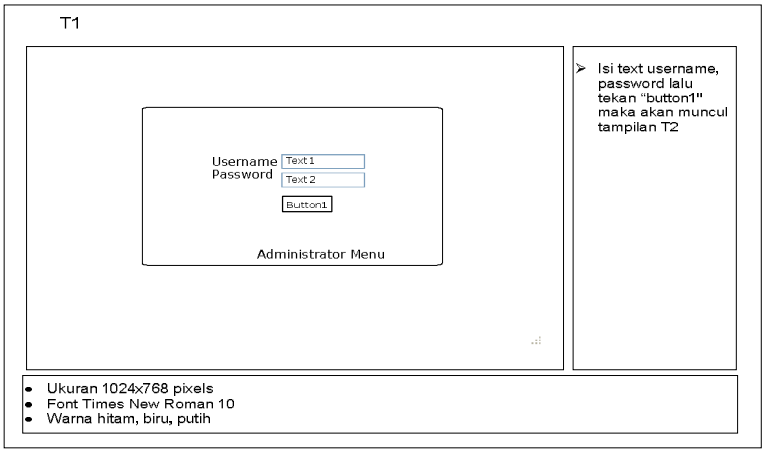
3.2.11.1.5 Perancangan Antar Muka Menu Pencarian Lokasi Daerah Banjir



Gambar 3.41 Perancangan Antar Muka Menu Pencarian Lokasi Daerah Banjir

3.2.11.2 Perancangan Antar Muka *Backend*

3.2.11.2.1 Perancangan Antar Muka Halaman *Login* Admin



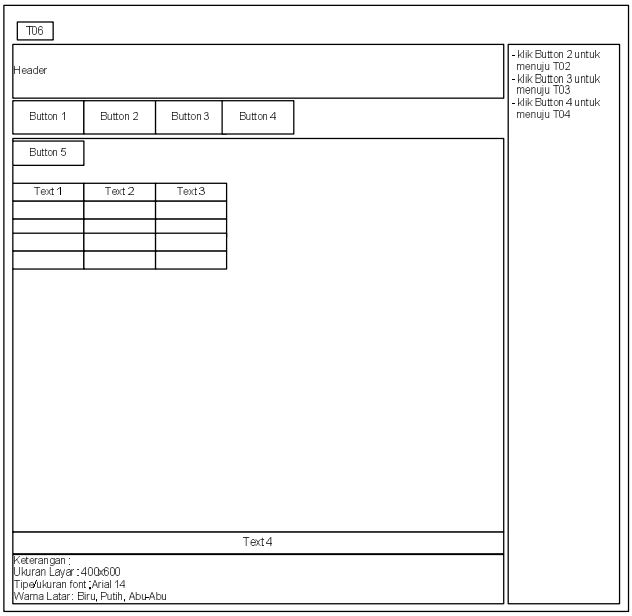
Gambar 3.42 Perancangan Antar Muka Halaman *Login* Admin

3.2.11.2.2 Perancangan Antar Muka Halaman Depan Admin



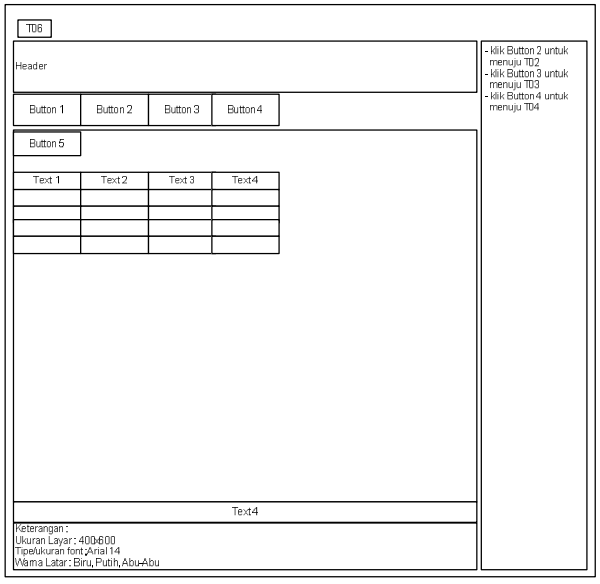
Gambar 3.43 Perancangan Antar Muka Halaman Depan Admin

3.2.11.2.3 Perancangan Antar Muka Halaman Data Kecamatan



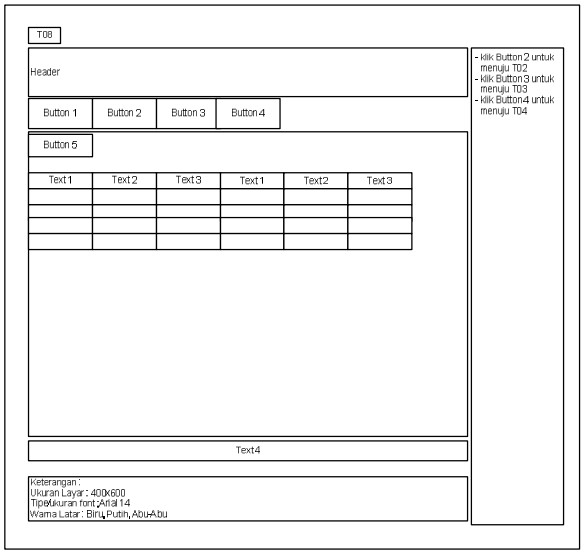
Gambar 3.44 Perancangan Antar Muka Halaman Data Kecamatan

3.2.11.2.4 Perancangan Antar Muka Halaman Data Kelurahan



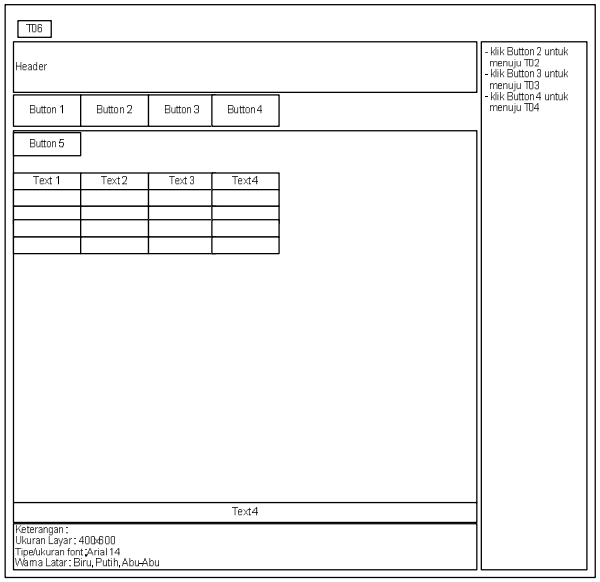
Gambar 3.45 Perancangan Antar Muka Halaman Data Kelurahan

3.2.11.2.5 Perancangan Antar Muka Halaman Data Kecamatan Daerah Banjir



Gambar 3.46 Perancangan Antar Muka Halaman Data Kecamatan Daerah Banjir

3.2.11.2.6 Perancangan Antar Muka Halaman Data Sebaran Banjir



Gambar 3.47 Perancangan Antar Muka Halaman Data Sebaran Banjir