BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis sistem ini dilakukan untuk mengidentifikasi permasalahan yang terdapat dalam sistem. Selain itu juga untuk menentukan kebutuhan-kebutuhan dari sistem yang akan dibangun. Analisis sistem ini meliputi analisis masalah, analisis sistem yang sedang berjalan, analisis sistem yang akan dibangun, analisis teknologi, analisis kebutuhan fungsional dan Non-fungsional serta analisis data.

3.1.1 Analisis Masalah

Hasil dari penelitian yang telah dilakukan di Majalengka, terdapat salah satu masalah yang timbul dan dirasakan oleh pengguna kendaraan bermotor di Majalengka. Masalah tersebut mengenai daerah-daerah yang rawan terhadap kecelakaan maupun kejahatan.

Pengguna kendaraan bermotor di Majalengka sebagian besar tidak mengetahui daerah rawan kecelakaan dan kejahatan di daerah Majalengka. Hal tersebut disebabkan oleh kurangnya rambu-rambu peringatan dibeberapa ruas jalan di Majalengka. Selain itu juga kurangnya peneranngan di beberapa wilayah yang membuat pengguna kendaraan bermotor tidak dapat melihat informasi di sekitar jalan yang dilewatinya. Dengan begitu, pengguna kendaraan tidak dapat berhati-hati ketika akan melewati daerah yang rawan kecelakaan maupun mejahatan.

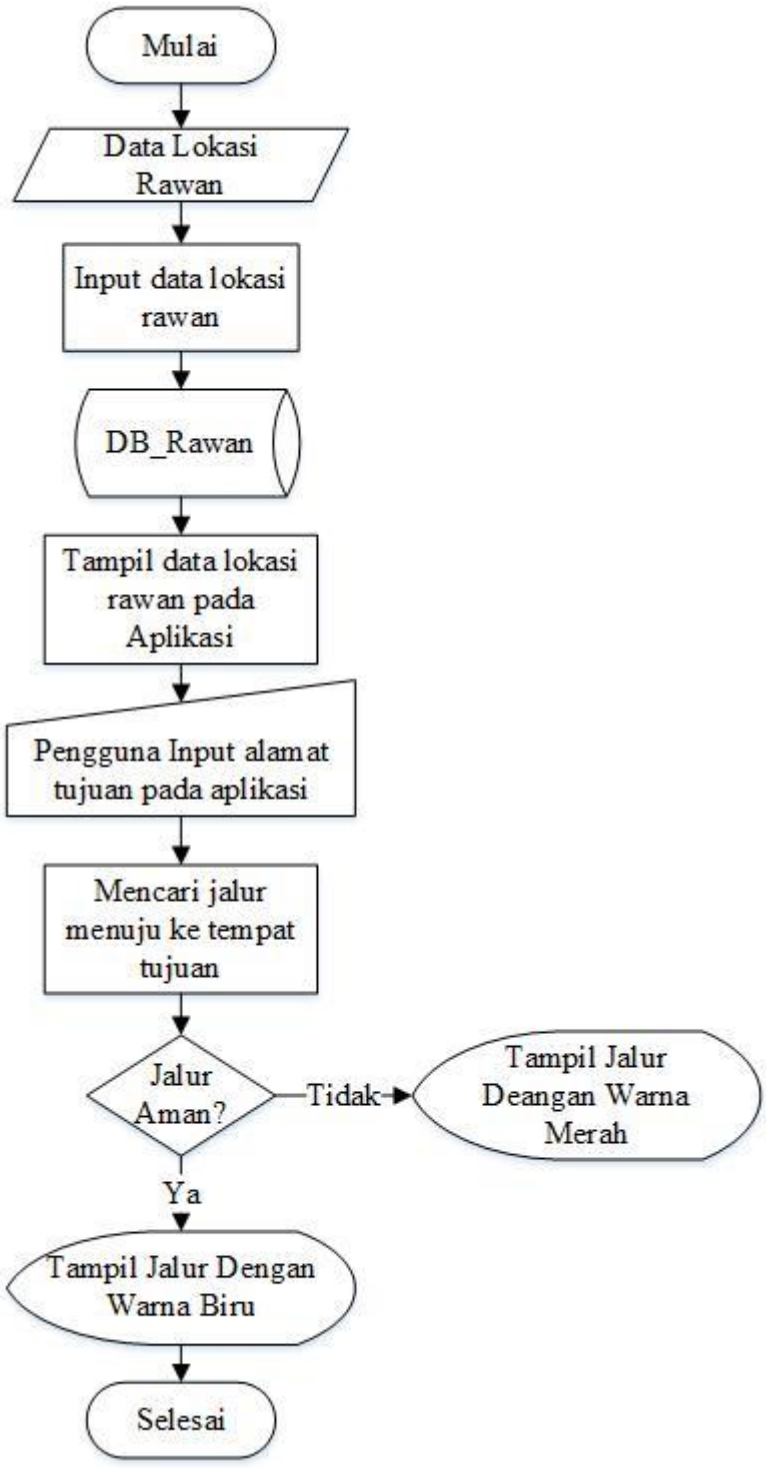
Pengguna kendaraan bermotor di Majalengka juga tidak mengetahui jalur alternatif atau jalur yang aman yang dapat dilalui untuk menghindari daerah yang rawan. Kurangnya informasi dari pihak yang berwajib membuat pengguna kendaraan bermotor tidak mengetahui jalur mana saja yang lebih aman untuk dilewati menuju tempat tujuan. Hasilnya, mereka yang tidak mengetahui tidak dapat menghindari daerah yang rawan.

Dari beberapa masalah yang timbul, maka diperlukan sebuah aplikasi yang dapat memberikan informasi kepada pengguna kendaraan bermotor di Majalengka mengenai daerah-daerah yang rawan. Selain itu juga dapat menampilkan jalur yang aman untuk dilalui dan jalur yang tidak aman jika dilewati. Sehingga, pengguna kendaraan bermotor dapat mengindari daerah yang rawan dan berhati-hati ketika harus melewati darah yang rawan kecelakaan serta kejahatan.

3.1.2 Analisis Sistem Yang Akan Dibangun

Analisis sistem yang dibangun dibuat bertujuan untuk menggambar bagaimana sistem yang akan dibangun. Adapun sistem yang akan dibangun dalam penelitian ini adalah sebagai berikut.

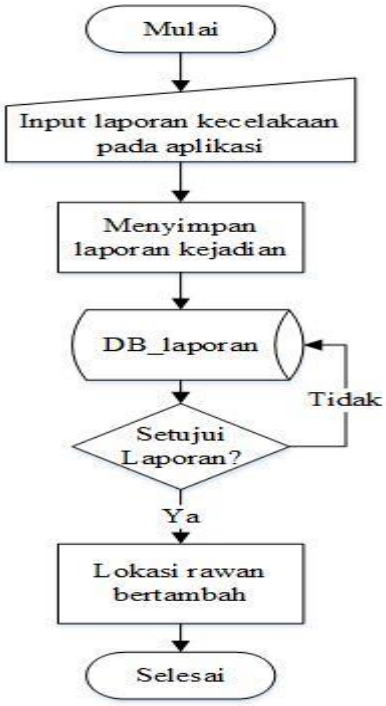
1. Admin dapat memasukan data lokasi rawan pada aplikasi sebagai data awal. Data tersebut akan tersimpad pada database siste.
2. Selanjutnya sistem akan menampilkan data lokasi rawan tersebut pada aplikasi *mobile* dan juga pada web admin. Dengan begitu pengguna bisa mengetahui daerah atau lokasi-lokasi yang rawan di Majalengka.
3. Pengguna bisa melakukan pencarian jalur pada aplikasi dengan cara membuka aplikasi tersebut menggunakan *smartphonenya*. Kemudian masukan alamat tujuan pengguna.
4. Sistem akan mencari jalur menuju ke tempat tujuan pengguna. Setelah jalur tersebut ditemukan, sistem akan memberikan perbedaan warna jalur. Jika jalur tersebut melewati daerah yang rawan maka jalur itu akan diberikan warna merah dan jika jalur tersebut tidak melewati daerah rawan maka akan diberikan warna biru.
5. Dengan begitu pengguna bisa memilih jalur mana yang akan dilewatinya.



Gambar 3.1 Prosedur Sistem Yang Akan di Bangun

Selain pencarian jalur aman, sistem yang akan dibangun juga terdapat fitur pelaporan kejadian kecelakaan dan kejahatan. Adapaun prosedur pelaporan pada sistem yang akan dibangun adalah sebagai berikut.

1. Pengguna dapat melakukan pelaporan melewati aplikasi dengan cara memasukan data kejadian kecelakaan atau kejahatan. Selanjutnya sistem akan menyimpan laporan masyarakat dan menampilkannya pada web admin atau bagian kepolisian.
2. Admin dapat melakukan persetujuan laporan pengguna tersebut, jika admin menyetujuinya lokasi kejadian yang dilaporkan oleh pengguna bisa menjadi lokasi rawan baru pada aplikasi. jika tidak laporan tersebut tetap akan tersimpan namun hanya ditampilkan pada riwayat pelporan.



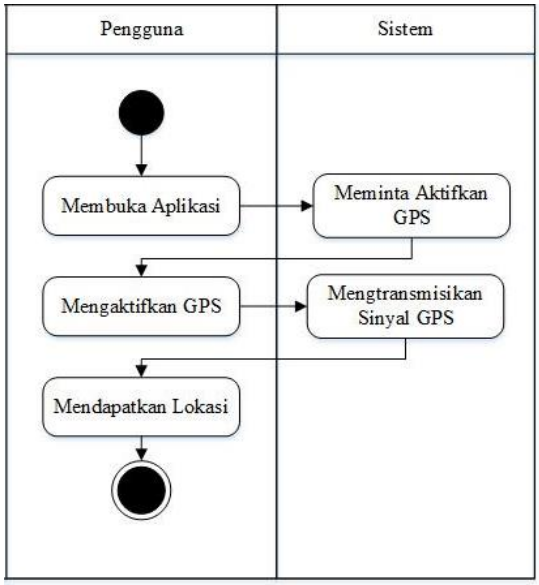
Gambar 3.2 Prosedur Sistem Pelaporan Yang Akan Dibangun

3.1.3 Analisis Teknologi Yang Digunakan

Analisis teknologi ini dilakukan untuk mengetahui teknologi apa saja yang akan digunakan dalam pembangunan aplikasi pencarian jalur aman. Adapun teknologi yang digunakan dalam aplikasi ini adalah sebagai berikut.

3.1.3.1 GPS

GPS atau *Global Position System* merupakan sebuah teknologi yang dapat digunakan untuk memberikan informasi mengenai posisi keberadaan dimana dia berada. Dalam sistem yang akan dibangun ini menggunakan teknologi GPS untuk melihat posisi dari pengguna. Adapun cara kerja GPS pada sistem yang akan dibangun adalah sebagai berikut.



Gambar 3.3 Prosedur Cara Kerja GPS Pada Sistem Yang Akan Dibangun

Adapun penjelasan dari prosedur diatas adalah sebagai berikut.

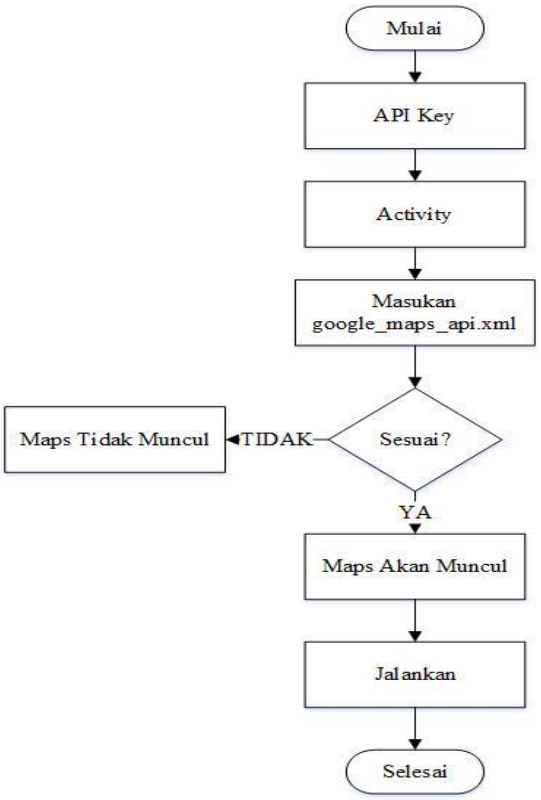
1. Ketika pengguna membuka aplikasi, sistem akan meminta pengguna untuk mengaktifkan GPS.
2. Setelah pengguna mengaktifkan GPS, sistem akan mengtransmisikan sinyal dari satelit ke perangkat GPS berupa posisi pengguna

3. Pengguna dapat melihat informasi berupa posisi dan ditampilkan pada peta elektronik.

Setelah mengetahui posisi keberadaan pengguna, selanjutnya pengguna bisa langsung untuk mencari jalur menuju ke tempat tujuan. Selanjutnya, sistem akan menampilkan rekomendasi rute atau jalur yang aman menuju tujuan pengguna.

3.1.3.2 Google Maps API

Google Maps API ini digunakan dalam pembangunan aplikasi pencarian jalur aman di kota majalengka. Tujuan dari menggunakan *Google Maps API* ini untuk mendapatkan layanan peta digital yang diperlukan oleh aplikasi. Salah satu layanan yang digunakan dari *Google Maps API* pada aplikasi yang akan dibangun ini adalah *marker*. Adapun tahapan dari penggunaan Google Maps API adalah sebagai berikut.



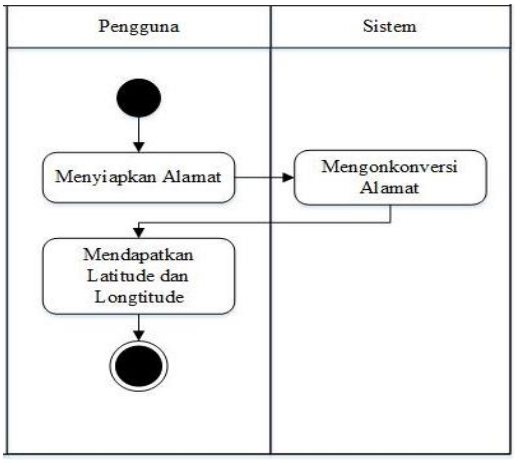
Gambar 3.4 Alur Menggunakan *Google Maps* API

Adapun penjelasan dari alur diatas adalah sebagai berikut.

1. Dapatkan API Key terlebih dahulu. Key ini berfungsi untuk izin akses terhadap penggunaan layanan *Google Maps API*.
2. Siapkan *activity* atau template google *maps*. Karena aplikasi yang akan dibangun menggunakan *tools* Android Studio, jadi secara *default* Android Studio sudah memiliki template untuk *activity Google Maps* dan dapat ditambahkan langsung pada *project* yang akan dibangun.
3. Menyematkan *Google Maps API Key* yang sudah didapat pada file google_maps_api.xml dalam *project*. Jika *key* tidak sesuai atau tidak disematkan maka maps tidak akan muncul pada aplikasi.
4. Tes atau me-*runing* aplikasi untuk memastikan bahwa *setting* dan tampilan peta berhasil dibuat pada aplikasi.

3.1.3.3 *Geocoding*

Pada pembangunan aplikasi ini menggunakan teknologi *geocoding*. Teknologi tersebut digunakan untuk mengubah suatu alamat menjadi suatu letak geografis untuk dapat menempatkan *marker*. Penempatan *marker* tersebut diperlukan sebagai penanda dari titik-titik yang rawan kecelakaan serta kejahatan. Dengan begitu pengguna aplikasi bisa dengan mudah melihat titik-titik tersebut pada peta yang ada di *smartphonenya*. Teknologi *geocoding* dalam sistem yang akan dibangun ini digunakan dengan cara sebagi berikut.



Gambar 3.5 Prosedur *Geocoding* Pada Sistem Yang Akan Dibangun

Berikut ini penejelasan dari prosedur menggunakan *geocoding* pada sistem yang akan dibangun.

1. Memasukan alamat-alamat daerah yang rawan akan kecelakaan dan kejahatan di Majalengka
2. Setelah itu akan didapat latitude dan longtitude atau koordinat geografis dari alamat tersebut
3. Koordinat geografis tersebut akan di tempatkan sebuah *marker* yang menandakan bahwa daerah tersebut rawan. Selain untuk menempatkan marker, *latitude* dan *longtitude* yang didapatkan juga untuk di *geofencingkan*.

Dari tahapan diatas, adapun *source code* yang digunakan untuk mengkonversi *latitude* dan *longtitude* pada sistem yang akan dibangun adalah sebagai berikut.

Tabel 3.1 Analisis *Geocoding*

```
private String getCompleteAddressString(double LATITUDE, double
LONGITUDE) {
        String strAdd = "";
        Geocoder geocoder = new Geocoder(this,
Locale.getDefault());
        try {
            List<Address> addresses =
geocoder.getFromLocation(LATITUDE, LONGITUDE, 1);
 if (addresses != null) {
    Address returnedAddress = addresses.get(0);
StringBuilder strReturnedAddress = new StringBuilder("");
for (int i = 0; i <= returnedAddress.getMaxAddressLineIndex();
i++) {
strReturnedAddress.append(returnedAddress.getAddressLine(i)).app
end("\n");
      }
strAdd = strReturnedAddress.toString();
Log.d("My Current address", strReturnedAddress.toString());
Log.w("", strReturnedAddress.toString());
} else {
Log.w("My Current  address", "No Address returned!");
       }
} catch (Exception e) {
  e.printStackTrace();
  Log.w("My Current address", "Canont get Address!");
}
 return strAdd;
}
```

3.1.3.4 *Geofencing*

Pada pembangunan aplikasi pencarian jalur aman ini juga menggunakan teknologi *geofencing*. Teknologi ini digunakan untuk mendeteksi keberadaan pengguna yang sedang melakukan penjalanan. Kegunanaan dari teknologi ini adalah jika pengguna yang berada dalam wilayah yang sudah dipagari memasuki daerah rawan yang sudah ditandai, maka aplikasi akan memberikan peringatan. Jadi dengan *geofencing* ini pengguna aplikasi akan dipantau keberadaannya selama perjalanan. Adapun cara kerja *geofencing* dalam sistem yang akan dibangun adalah sebagai berikut.

Adapun tahapan dalam menentukan wilayah yang akan di *geofencingkan* dalam sistem yang akan dibangun adalah sebagai berikut.

1. Menentukan titik koordinat

   Tahap ini adalah menentukan *latitude* dan *longtitude* yang akan di *geofencingkan*. *Latitude* dan *longtitude* di dapatkan dari proses *geocoding*.
2. Menentukan Aksi

   Setelah titik rawan tersebut di tentukan dan telah di *geofencingkan*, selanjutnya yaitu menentukan aksi yang akan terjadi jika derah tersebut dilewati oleh sesuatu. Pada sistem yang akan dibangun ini, jika terdapat pengguna yang melewati daerah rawan akan di beri pesan peringatan.

Berikut ini *source code* untuk teknologi geofencing dalam sistem yang dibangun adalah sebagai berikut.

Tabel 3.2 Analisis *Geofencing*

```
private void startGeofenceTitik(int i) {
Log.i(TAG, "startGeofence()");
if( geoFenceMarkerTitik != null ) {
Geofence geofence = createGeofenceTitik(
geoFenceMarkerTitik.get(i).getPosition(), GEOFENCE_RADIUS );
GeofencingRequest geofenceRequest =
createGeofenceRequestTitik(geofence);
addGeofenceTitik(geofenceRequest,i);
} else {
```

```
Log.e(TAG, "Geofence marker is null");
}
}
private static final long GEO_DURATION = 60 * 60 * 1000;
private static final String GEOFENCE_REQ_ID = "My Geofence";
private static final float GEOFENCE_RADIUS = 500.0f;
```

3.1.3.5 Geotagging

Geotagging merupakan proses memberikan suatu identitas metadata pada video, foto, gambar maupun website. Dalam sistem yang akan dibangun ini teknologi geotagging digunakan untuk mengetahui apakah foto kejadian kecelakaan maupun kejahatan tersebut sesuai dengan lokasi yang di kirimkan oleh masyarakat atau tidak. Berikut ini cara kerja dari geotagging pada aplikasi yang akan dibangun adalah:

1. Pengguna di haruskan mengaktifkan GPS terlebih dahulu sebelum menggunakan aplikasi.
2. Setelah GPS pengguna aktif, sistem akan mentransmisikan sinyal berupa koordinat sesuai dengan posisi pengguna saat itu
3. Pengguna melakukan pelaporan dan mengambil foto dengan kamera yang ada pada smartphonenya. Karema tersebut berfungsi untuk menangkap foto atau merekam video, sedangkan GPS berfungsi untuk mengetahui titik koordinat dari foto.

Adapun sistem yang akan dibangun pada penelitian ini menggunakan *source code* berikut ini untuk teknologi *geotagging*.

Tabel 3.3 Analisis *Geotagging*

```
String []tempAlamatGps= alamatHasilGps.split(",");
String []tempAlamatMaps= jalan.split(",");
String [] alamatMaps=new String[2];
String [] alamatGps = new String[2];
if (tempAlamatGps[0].contains(" No.")) {
alamatGps = tempAlamatGps[0].split(" No.");
}else {
alamatGps[0]=tempAlamatGps[0];
}
if (tempAlamatMaps[0].contains(" No.")) {
alamatMaps= tempAlamatMaps[0].split(" No.");
}else {
```

```
alamatMaps[0]=tempAlamatMaps[0];
}
System.out.println("HASIL ALAMAT GPS FINAL "+alamatGps[0]);
System.out.println("HASIL ALAMAT MAPS FINAL "+alamatMaps[0]);
if (!alamatGps[0].equalsIgnoreCase(alamatMaps[0])){
showAlert("lokasi yang anda tandai tidak sama dengan lokasi anda
merekam bukti");
validasi=false;
}else {
validasi=true;
}
```

3.1.4 Analisis Level Rawan

Dalam penelitian ini, aplikasi yang akan dibangun memiliki 3 kategori titik rawan, yaitu titik rawan dengan level rendah, level sedang dan level tinggi. Menurut penelitian sebelumnya, untuk menentukan sebuah titik rawan tersebut masuk ke dalam kategori mana saja, maka dilakukan perhitungan berdasarkan rentang waktu dan jumlah kejadian. Dalam penelitian ini data kejadian kecelakaan maupun kejahatan diambil dari pelaporan masyarakat. Adapun ketiga kategori titik rawan dalam penelitian ini adalah sebagai berikut.

1. Titik Rawan Level Rendah

   Pada titik rawan ini jumlah kejadian kecelakaan dan kejahatan minimal terjadi 1 kali dalam waktu sekitar 2 bulan.
2. Titik Rawan Level Sedang

   Pada titik rawan ini jumlah kejadian kecelakaan dan kejahatan minimal terjadi 1 kali dalam waktu 1 bulan.
3. Titik Rawan Level Tinggi

   Pada titik rawan ini jumlah kejadian kecelakaan dan kejahatan minimal terjadi 1 kali dalam waktu 1 sampai 2 minggu.

3.1.5 Analisis Contoh Kasus

Analisis kasus pada penelitian ini untuk memberikan sebuah contoh dari sistem yang akan dibangun dalam menentukan jalur menuju tempat tujuan. Aplikasi yang akan dibangun ini akan menampilkan jalur mana saja yang dapat dilalui meunuju ke tujuannya. Jalur-jalur tersebut dibedakan berdasarkan warnanya.

Pengguna dapat memilih sendiri jalur mana yang akan pengguna lalui. Adapun contoh kasus dari rekomendasi jalur pada penelitian ini adalah sebagai berikut.



Gambar 3.6 Contoh Kasus Jalur Aman

Dari gambar 3.6 di atas, hasil dari pencarian jalur menuju ke Jl.Kebon Bolo, Majalengka atau dari titik yang berada di atas menuju ke titik yang berada di bawah. Aplikasi yang akan dibangun ini pertama akan melihat tujuan dari pengguna, selanjutnya mencari jalur-jalur yang dapat dilalui oleh pengguna menuju ke tujuannya. Jalur yang di rekomendasikan dari aplikasi adalah jalur yang tidak terdapat titik rawannya, namun untuk jalur yang terdapat titik rawannya aplikasi juga akan tetap menampilkan jalur tersebut. Sehingga pengguna sendiri yang akan memilih jalurnya, dan masih bisa melewati jalur yang tidak aman tersebut.

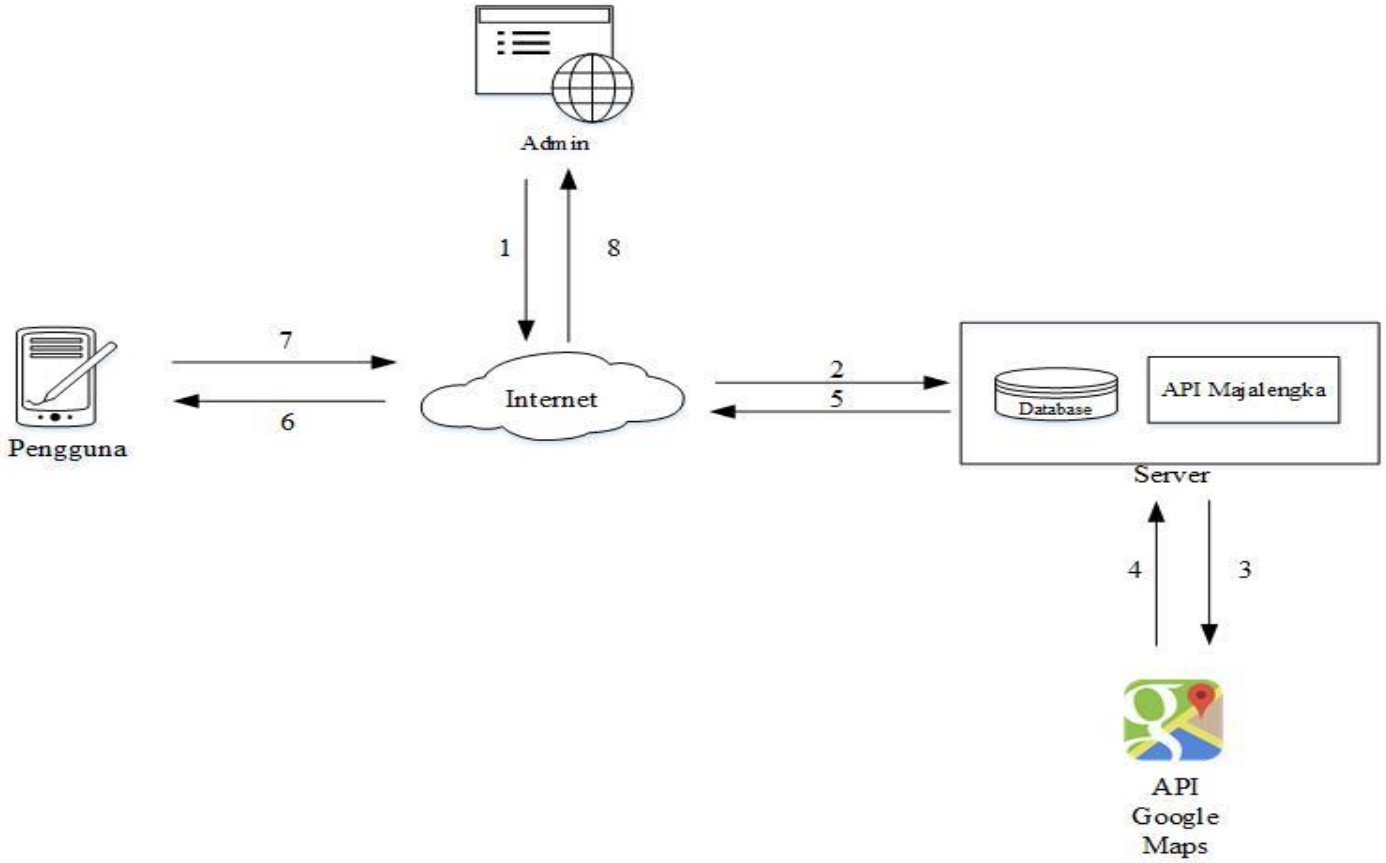
Terlihat seperti pada gambar 3.6 di atas, aplikasi menampilkan 2 buah jalur yang dapat dilalui oleh pengguna. Jalur tersebut merupakan jalur rekomendasi dan juga jalur biasa yang bisa di lalui, meskipun jalur tersebut terdapat beberapa titik rawan.

Jika pengguna memilih jalur yang terdapat titik rawannya dan pengguna tersebut melewati daerah rawannya, pengguna akan mendapatkan sebuah pemberitahuan dari aplikasi bahwa pengguna tersebut memasuki daerah rawan. dengan begitu, pengguna masih tetap bisa berhati-hati meskipun melewati daerah yang rawan.

3.1.6 Analisis Arsitektur Sistem

Analisis arsitektur sistem dibuat untuk mengidentifikasi arsitektur yang akan dibangun berdasarkan *flatform mobile* dan web. Flatform Web pada aplikasi ini dibangun untuk administrator. Administrator ini bertugas untuk mengelola data pengguna serta data-data yang terkait dengan daerah-daerah rawan kecelakaan dan kejahatan.

Sedangkan *flatform mobile* dibuat untuk 2 jenis pengguna, yaitu pengguna sebagai polisi dan pengguna sebagai masyarakat. Polisi ini berugas untuk menerima pesan atau informasi dari masyarakat yang mengirimkan pelaporan kejadian kecelakaan maupun kejahatan serta dapat menyetujui laporan tersebut jika laporan tersebut benar. Adapun sistem yang akan dibangun memiliki arsitektur sebagai berikut.



Gambar 3.7 Arsitektur Sistem

Berikut ini penjelasan dari arsitektur sistem yang akan dibangun adalah:

1. Pertama, admin menambahkan data awal kecelakaan berupa titik-titiknya dan menyimpannya pada database. Admin dapat terhubung dengan server dan menyimpan data pada database tersebut menggunakan sebuah jaringan internet.
2. Server akan mengambil data yang dibutuhkan atau terhubung dengan *Google Maps* API, yang selanjutnya *Google Maps* API akan mengirimkan kembali data sesuai dengan apa yang dibutuhkan oleh server.
3. Selanjutnya server akan menampilkan data-data yang telah di tambahkan oleh admin kepada pengguna melalui aplikasi *mobilenya*. Dengan begitu pengguna dapat melihat titik atau daerah-daerah rawan mana saja di Majalengka.

3.1.7 Analisis Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak pada sistem yang akan dibangun terbagi menjadi dua, yaitu kebutuhan fungsional dan kebutuhan non-fungsional. Spesifikasi kebutuhan fungsional dibuat untuk menjelaskan bagaimana sistem yang harus dibangun. Berikut ini spesifikasi kebutuhan fungsional.

Tabel 3.4 Spesifikasi Kebutuhan Fungsional *Mobile*

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-F-001	Sistem *mobile* dapat menyediakan fasilitas bagi masyarakat untuk daftar
SKPL-F-002	Sistem *mobile* dapat menyediakan fasilitas bagi masyarakat untuk login
SKPL-F-003	Sistem *mobile* dapat menyediakan fasilitas bagi polisi untuk login
SKPL-F-004	Sistem *mobile* dapat menyediakan fasilitas bagi masyarakat dan polisi untuk melihat titik rawan kecelakaan dan kejahatan

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-F-005	Sistem *mobile* dapat menyediakan fasilitas bagi masyarakat dan polisi untuk mencari jalur aman menuju tempat tujuan
SKPL-F-006	Sistem *mobile* dapat menyediakan fasilitas bagi masyarakat dan polisi untuk melihat jalur aman menuju tempat tujuan
SKPL-F-007	Sistem *mobile* dapat menyediakan fasilitas bagi masyarakat dan polisi untuk melihat pemberitahuan atau peringatan melewati jalur rawan
SKPL-F-008	Sistem *mobile* dapat menyediakan fasilitas bagi masyarakat untuk melakukan pelaporan kejadian kecelakaan dan kejahatan
SKPL-F-009	Sistem *mobile* dapat menyediakan fasilitas bagi polisi untuk menyetujui pelaporan dari masyarakat
SKPL-F-010	Sistem *mobile* dapat menyediakan fasilitas bagi polisi untuk melihat pelaporan masyarakat

Tabel 3.5 Spesifikasi Kebutuhan Fungsional Web

Kode SKPL	Spesifikasi kebutuhan perangkat lunak
SKPL-F-011	Sistem web dapat menyediakan fasilitas bagi admin untuk melakukan *login*
SKPL-F-012	Sistem web dapat menyediakan fasilitas bagi admin untuk melakukan ubah *password*
SKPL-F-013	Sistem web menyediakan fasilitas bagi admin untuk dapat mengelola data titik rawan
SKPL-F-014	Sistem web dapat menyediakan fasilitas bagi admin untuk mengelola data pengguna yaitu masyarakat dan polisi
SKPL-F-015	Sistem web dapat menyediakan fasilitas bagi admin untuk mengelola data pelaporan masyarakat

Selain spesifikasi kebutuhan fungsional, berikut ini spesifikasi kebutuhan non-fungsional. Kebutuhan non-fungsional ini merupakan batasan layanan yang ditawarkan sistem. Adapun spesifikasi kebutuhan non-fungsional adalah sebagai berikut.

Tabel 3.6 Spesifikasi Kebutuhan Non Fungsional *Mobile*

Kode SKPL	Spesifikasi kebutuhan perangkat lunak
SKPL-NF-001	Subsistem *mobile* yang dibangun minimal menggunakan sistem operasi Versi 5.0
SKPL-NF-002	Subsistem *mobile* yang dibangun menggunakan teknologi GPS.
SKPL-NF-003	Sistem dibangun dengan spesifikasi perangkat keras yang memenuhi standar minimum kebutuhan

Tabel 3.7 Spesifikasi Kebutuhan Non Fungsional Web

Kode SKPL	Spesifikasi kebutuhan perangkat lunak
SKPL-NF-004	Subsistem web dibangun dengan menggunakan *framework* CI.
SKPL-NF-005	Sistem dibangun dengan spesifikasi perangkat keras yang memenuhi standar minimum kebutuhan

3.1.8 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional ini dibuat untuk menentukan kebutuhan dari sistem yang akan dibangun. Spesifikasi ini juga meliputi komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan apa saja yang diperlukan, proses yang dibutuhkan oleh sistem sehingga dapat menjadi keluaran berupa informasi untuk pengguna.analisis kebutuhan non-fungsional ini meliputi analisis kebutuhan perangkat lunak dan kebutuhan perangkat keras.

1. *Product Reqruitment*

   Berikut ini kebutuhan non fungsional berdasarkan *product reqriutment*.

   a. SKPL-NF-001 Subsistem *Mobile* yang dibangun minimal menggunakan sistem operasi Versi 5.0.

Tabel 3.8 Kebutuhan Perangkat Lunak Dalam Membangun Sistem

NO	Kebutuhan Minimum Perangkat Lunak
1.	Sistem Operasi Windows 8 dan 10
2.	Web server XAMPP versi 3.2 ke atas sebagai web server
3.	Kode editor berupa Sublime atau Notepad++
4.	IDE Android Studio

b. SKPL-NF-002 Subsistem yang dibangun menggunakan teknologi GPS.

c. SKPL-NF-003 Sistem dibangun dengan spesifikasi hardware yang memenuhi standart minimum kebutuhan. Adapun mininum kebutuhan perangkat keras yang digunakan untuk menjalankan aplikasi adalah sebagai berikut.

Tabel 3.9 Kebutuhan Perangkat Keras *Mobile*

NO	Kebutuhan Minimum Perangkat Keras
1.	Minimum RAM 1GB
2.	Memiliki sensor GPS

Selain kebutuhan perangkat keras *Mobile*, berikut ini kebutuhan perangkat keras minimum yang digunakan untuk menjalankan aplikasi yang akan dibangun dalam flatform Web.

Tabel 3.10 Kebutuhan Perangkat Keras Web

NO	Kebutuhan Minimum Perangkat Keras
1.	Minimum Processor Intel Dual Core
2.	Minimum RAM 512MB
3.	Minimum hardisk 500MB

d. Subsistem web dibangun dengan menggunakan *framework* CI.

2. Analisis Pengguna

Analisis pengguna sistem dimaksudkan untuk mengetahui siapa saja pengguna yang terlibat dalam menjalankan sistem, sehingga dapat diketahui tingkat dan pemahaman pengguna. Adapun pengguna yang terlibat dalam sistem adalah sebagai berikut.

Tabel 3.11 Analisis Pengguna

No	Pengguna	Karakteristik
1.	Admin : Asep Kusjaya	a. Dapat mengoperasikan komputer b. Memiliki pengalaman menggunakan internet c. Dapat mengoperasikan web browser
2.	Polisi : Bagian Lakalantas dan Reskrim	a. Dapat mengoperasikan smartphone b. Memiliki pengalaman menggunakan internet
3.	Masyarakat : Pengguna kendaraan bermotor	a. Dapat mengoperasikan smartphone b. Memiliki pengalaman menggunakan internet

Setiap pengguna yang terlibat diatas memiliki tanggung jawab yang berbeda. Berikut ini tanggung jawab dari setiap pengguna, yaitu.

Tabel 3.12 Tanggung Jawab Pengguna

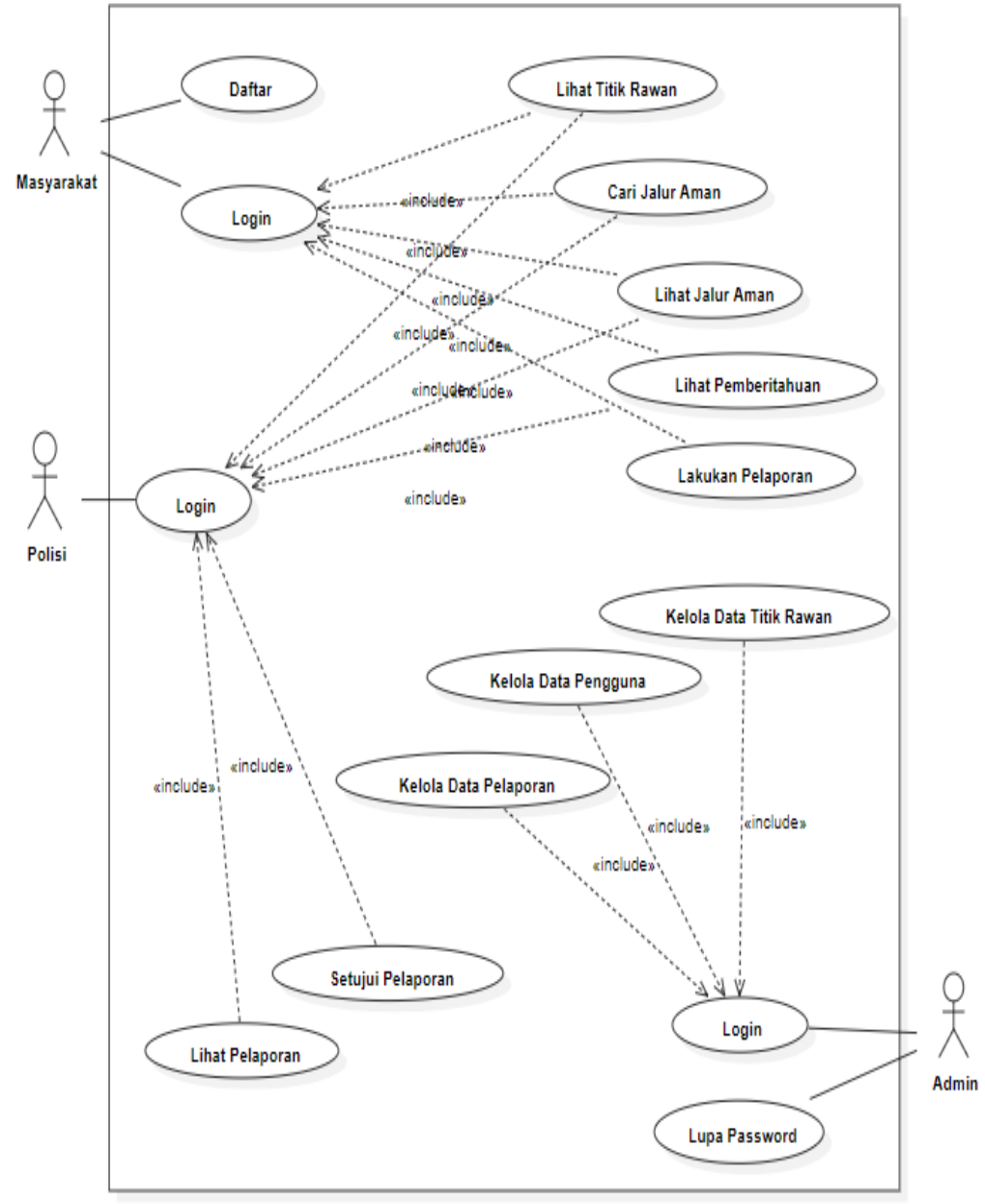
Pengguna	Tanggung Jawab
Admin	a. Mengelola data pengguna aplikasi b. Mengelola daerah rawan kecelakaan maupun kejahatan c. Menyetujui pelporan dari masyarakat d. Mengelola data pelaporan masyarakat
Polisi	a. Menyetujui laporan dari masyarakat
Masyarakat	a. Melakukan pelaporan kejadian kecelakaan atau kejahatan

3.1.9 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dibuat untuk menggambarkan proses kegiatan yang akan diterapkan dalam sistem yang akan dibangun dan menjelaskan kebutuhan yang diperlukan agar sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan. Analisis yang dilakukan dimodelkan dengan UML (*Unifield Modeling Language*). Tahap-tahap pemodelan UML meliputi *use case, scenario use case, activity diagram, class diagram* dan diagram *sequence*.

3.1.9.1 Use Case

Use case diagram digambarkan untuk melihat interaksi antara aktor atau pengguna dengan sistem yang akan dibangun. Berikut ini adalah diagram use case dari sistem yang akan dibangun.



Gambar 3.8 Diagram Use Case

Adapun aktor dan deskripsi peran untuk aktor dapat dilihat pada tabel 3.13 dibawah ini.

Tabel 3.13 Deskripsi Aktor

Aktor	Deskripsi
Masyarakat	Aktor pada peran ini memiliki wewenang untuk menggunakan fungsionalitas pada level *interface*.
Polisi	Aktor pada peran ini memiliki wewenang untuk menggunakan fungsionalitas pada level *interface*.
Admin	Aktor pada peran ini memiliki wewenang sebagai penyedia API atau sebagai sumber.

Berikut ini definisi *use case* dalam sistem yang akan dibangun dapat dilihat pada tabel 3.14.

Tabel 3.14 Deskripsi *Use Case*

Use Case	Deskripsi
Daftar	Sistem menyediakan fasilitas daftar untuk pengguna baru yang akan menggunakan aplikasi
Login	Sistem menyediakan fasilitas *login* untuk masyarakat bisa masuk kedalam sistem
Lihat Titik Rawan	Sistem menyediakan fasilitas untuk masyarakat dan polisi bisa melihat titik-titik rawan dalam bentuk peta digital
Cari Jalur Aman	Sistem menyediakan fasilitas untuk masyarakat dan polisi bisa mencari jalur alternatif menghindari daerah rawan
Lihat Jalur Aman	Sistem menyediakan fasilitas untuk masyarakat dan polisi bisa melihat rute atau jalur rekomendasi
Lihat Pemberitahuan	Sistem menyediakan fasilitas untuk masyarakat dan polisi bisa melihat pesan peringatan dari sistem ketika melewati titik rawan

Use Case	**Deskripsi**
Lakukan Pelaporan	Sistem menyediakan fasilitas untuk masyarakat dan polisi bisa melakukan pelaporan kecelakaan atau kejahatan
Login	Sistem menyediakan fasilitas *login* untuk bagian polisi bisa masuk kedalam sistem
Lihat Pelaporan	Sistem menyediakan fasilitas untuk polisi bisa melihat pelaporan masyarakat
Setujui Pelaporan	Sistem menyediakan fasilitas untuk bagian polisi bisa menyetujui atau meng-accept laporan dari masyarakat
Login	Sistem menyediakan fasilitas untuk admin bisa masuk ke dalam sistem
Lupa *Password*	Sistem menyediakan fasilitas untuk admin melakukan lupa *password* atau mereset *password*
Kelola Data Titik Rawan	Sistem menyediakan fasilitas untuk bagian admin bisa melakukan tambah, hapus, ubah dan lihat data titk rawan
Kelola Data Pengguna	Sistem menyediakan fasilitas untuk bagian admin bisa menyetujui dan menolak pelaporan masyarakat
Kelola Data Pelaporan	Sistem menyediakan fasilitas untuk bagian admin bisa melakukan hapus, dan lihat data pelaporan masyarakat

3.1.9.2 Scenario Use Case

Skenario yang dibuat ditujukan untuk beberapa use case utama, pada skenario use case ini dapat dijabarkan bagaimana alur dari setiap proses pada setiap use case berdasarkan interaksi antara aktor dan sistem. Adapun use case scenario untuk setiap use case sebagai berikut.

1. Use Case Daftar

   Berikut ini adalah skenario yang menggambarkan use case Daftar.

Tabel 3.15 Skenario Use Case Daftar

Use Case Name	Daftar	
Related Requirements	-	
Goal Context	Pengguna dapat melakukan pendaftaran *account* baru pada aplikasi	
Preconditions	Pengguna belum mempunyai *account*	
Successful End Condition	Pengguna berhasil melakukan pendaftaran *account*	
Failed End Condition	Pengguna gagal dalam mendaftarkan *accountnya* pada aplikasi	
Primary Actor	Masyarakat	
Trigger	Pengguna memilih menu daftar	
Main Flow	*Step*	*Action*
	1.	Pengguna membuka halaman daftar
	2.	Sistem menampilkan form daftar
	3.	Pengguna mengisi form daftar dengan benar
	4.	Sistem melakukan validasi data
	5.	Sistem menampilkan pesan pendaftran berhasil dilakukan
Extension	4.1	Sistem menyimpan data pendaftaran pengguna
	4.2	Sistem menampilkan pesan pendaftraan gagal dilakukan

2. Use Case *Login*

Berikut ini adalah skenario yang menggambarkan use case *Login*.

Tabel 3.16 Skenario Use Case Login

Use Case Name	*Login*	
Related Requirements	-	
Goal Context	Masyarakat dapat melakukan *login* dan masuk pada aplikasi	
Preconditions	Masyarakat telah daftar dan mempunyai *account*	
Successful End Condition	Masyarakat berhasil *login*	
Failed End Condition	Masyarakat gagal melakukan *login*	
Primary Actor	Masyarakat	
Trigger	Masyarakat memilih menu *login*	
Main Flow	*Step*	*Action*
	1.	Pengguna membuka halaman login
	2.	Sistem menampilkan form login
	3.	Pengguna mengisi form login dengan benar
	4.	Sistem melakukan validasi data login pengguna
	5.	Sistem menampilkan halaman awal aplikasi
Extension	5.1	Sistem menampilkan *error* gagal login

3. Use Case Lihat Titik Rawan

   Berikut ini adalah skenario yang menggambarkan use case Lihat Titik Rawan.

Tabel 3.17 Skenario Use Case Lihat Titik Rawan

Use Case Name	Lihat Titik Rawan	
Related Requirements	-	
Goal Context	Pengguna dapat melihat titik rawan kecelakaan dan kejahatan	
Preconditions	Pengguna berhasil *Login* dan masuk pada sistem	
Successful End Condition	Pengguna dapat melihat titik rawan kecelakaan dan kejahatan	
Failed End Condition	Pengguna gagal melakukan *login*	
Primary Actor	Masyarakat dan Polisi	
Trigger	Pengguna telah *login*	
Main Flow	*Step*	*Action*
	1.	Pengguna berhasil login pada aplikasi
	2.	Sistem menampilkan titik-titik rawan kecelakaan dan kejahatan pada aplikasi
Extension	1.1	Sistem menampilkan *error* gagal login dan tidak menampilkan titik rawan

4. Use Case Cari Jalur Aman

Berikut ini adalah skenario yang menggambarkan use case Cari Jalur Aman.

Tabel 3.18 Skenario Use Case Cari Jalur Aman

Use Case Name	Cari Jalur Aman	
Related Requirements	-	
Goal Context	Pengguna dapat melakukan pencarian jalur aman pada aplikasi	
Preconditions	Pengguna berhasil *Login* dan masuk pada sistem	
Successful End Condition	Pengguna berhasil melakukan pencarian jalur aman	
Failed End Condition	Pengguna gagal melakukan pencarian jalur aman	
Primary Actor	Masyarakat dan Polisi	
Trigger	Pengguna memilih menu pencarian	
Main Flow	*Step*	*Action*
	1.	Pengguna berhasil masuk pada sistem
	2.	Sistem menampilkan titik rawan kecelakaan dan kejahatan
	3.	Pengguna mengisi form pencarian jalur aman menuju tempat tujuna
	4.	Sistem mencari jalur aman menuju tempat tujuan pengguna
	5.	Sistem menampilkan rekomendasi jalur aman
Extension	5.1	Sistem menampilkan error dan gagal menmpilkan rekomendasi jalur aman

5. Use Case Lihat Jalur Aman

   Berikut ini adalah sekenario yang menggambarkan use case Lihat Jalur Aman.

Tabel 3.19 Skenario Use Case Lihat Jalur Aman

Use Case Name	Lihat Jalur Aman	
Related Requirements	-	
Goal Context	Pengguna dapat melihat rekomendasi jalur aman	
Preconditions	Pengguna telah mengisi form tempat tujuan	
Successful End Condition	Pengguna berhasil mendapatkan jalur aman menuju tujuannya	
Failed End Condition	Pengguna tidak mendapat rekomendasi jalur aman	
Primary Actor	Masyarakat dan Polisi	
Trigger	Pengguna memilih menu pencarian jalur aman	
Main Flow	*Step*	*Action*
	1.	Pengguna mengisi form cari jalur aman menuju tempat tujuan
	2.	Sistem menampilkan beberapa rekomendasi jalur aman
	3.	Pengguna dapat melihat rekomendasi jalur aman dari aplikasi
Extension	2.1	Sistem gagal menampilkan rekomendasi jalur aman pada masyarakat

6. Use Case Lihat Pemberitahuan

   Berikut ini adalah skenario yang menggambarkan use case Lihat Pemberitahuan.

Tabel 3.20 Skenario Use Case Lihat Pemberitahuan

Use Case Name	Lihat Pemberitahuan	
Related Requirements	-	
Goal Context	Pengguna dapat melihat pemberitahuan atau pesan dari aplikasi ketika melewati titik rawan	
Preconditions	Pengguna telah melakukan pencarian jalur aman pada aplikasi	
Successful End Condition	Pengguna berhasil mendapat pesan peringatan	
Failed End Condition	Pengguna tidak mendapat pesan peringatan	
Primary Actor	Masyarakat dan Polisi	
Trigger	Pengguna memilih melakukan perjalanan sesuai rekomendasi aplikasi	
Main Flow	*Step*	*Action*
	1.	Pengguna mendapatkan jalur rekomendasi dan melakukan perjalanan
	2.	Sistem mendeteksi perjalanan pengguna
	3.	Pengguna mendapatkan pesan ketika melewati titik rawan
Extension	2.1	Sistem tidak memberikan pesan peringatan kepada pengguna

7. Use Case Lakukan Pelaporan

   Berikut ini adalah skenario yang menggambarkan use case Lakukan Pelaporan.

Tabel 3.21 Skenario Use Case Lakukan Pelaporan

Use Case Name	Lakukan Pelaporan	
Related Requirements	-	
Goal Context	Pengguna dapat melakukan pelaporan kejadian kecelakaan maupun kejahatan	
Preconditions	Pengguna berhasil *Login* dan masuk pada sistem	
Successful End Condition	Pengguna berhasil melakukan pelaporan dan mnegirimnya menggunakan aplikasi	
Failed End Condition	Pengguna gagal melakukan pelaporan kejadian kecelakaan dan kejahatan	
Primary Actor	Masyarakat	
Trigger	Pengguna memilih menu pelaporan	
Main Flow	*Step*	*Action*
	1.	Pengguna memilih menu pelaporan
	2.	Sistem menampilkan form pelaporan kejadian kecelakaan dan kejahatan
	3.	Pengguna mengisi form pelaporan
	4.	Sistem melakukan validasi masukan dari pelapor
	5.	Pengguna berhasil melakukan pelaporan
Extension	4.1	Sistem menampilkan error dan gagal mengirim laporan

8. Use *Login*

Berikut ini adalah skenario yang menggambarkan use case Login untuk polisi

Tabel 3.22 Skenario Use Case Login Polisi

Use Case Name	*Login*	
Related Requirements	-	
Goal Context	Polisi dapat melakukan *login* dan masuk pada aplikasi	
Preconditions	Polisi telah daftar dan mempunyai *account*	
Successful End Condition	Polisi berhasil *login*	
Failed End Condition	Polisi gagal melakukan *login*	
Primary Actor	Polisi	
Trigger	Masyarakat memilih menu *login*	
Main Flow	*Step*	*Action*
	1.	Pengguna membuka halaman login
	2.	Sistem menampilkan form login
	3.	Pengguna mengisi form login dengan benar
	4.	Sistem melakukan validasi data login pengguna
	5.	Sistem menampilkan halaman awal aplikasi
Extension	5.1	Sistem menampilkan *error* gagal login

9. Use Case Lihat Pelaporan

Berikut ini adalah skenario yang menggambarkan use case Lihat Pelaporan.

Tabel 3.23 Skenario Use Case Lihat Pelaporan

Use Case Name	Lihat Pelaporan	
Related Requirements	-	
Goal Context	Pengguna dapat melihat laporan dari masyarakat	
Preconditions	Pengguna berhasil *Login* dan masuk pada sistem	
Successful End Condition	Pengguna berhasil melihat laporan dari masyarakat	
Failed End Condition	Pengguna gagal melihat laporan dari masyarakat	
Primary Actor	Polisi	
Trigger	Pengguna berhasil login dan berada dalam sistem	
Main Flow	*Step*	*Action*
	1.	Pengguna berhasil masuk pada sistem
	2.	Sistem menampilkan pesan pelaporan dari masyarakat
	3.	Pengguna dapat meihat pesan pelaporan dari masyarakat
Extension	2.1	Sistem gagal menampilkan pesan pelaporan dari masyarakat

10. Use Case Setujui Pelaporan

Berikut ini adalah skenario yang menggambarkan use case Setujui Pelaporan.

Tabel 3.24 Skenario Use Case Setujui Pelaporan

Use Case Name	Setujui Pelaporan	
Related Requirements	-	
Goal Context	Pengguna dapat melakukan acc atau menyetujui pelaporan masyarakat	
Preconditions	Pengguna berhasil *Login* dan masuk pada sistem	
Successful End Condition	Laporan masyarakat berhasil di setujui dan tersimpan	
Failed End Condition	Laporan masyarakat tidak disetujui	
Primary Actor	Polisi	
Trigger	Pengguna berhasil login dan berada pada sistem	
Main Flow	*Step*	*Action*
	1.	Pengguna berhasil masuk pada sistem
	2.	Sistem menampilkan pesan pelaporan dari masyarakat
	3.	Pengguna meng-acc atau menyetujui pelaporan masyarakat
	4.	Sistem menyimpan laporan masyarakat

11. Use Case *Login*

Berikut ini adalah skenario yang menggambarkan use case *login* untuk admin

Tabel 3.25 Skenario Use Case *Login* Admin

Use Case Name	*Login*	
Related Requirements	-	
Goal Context	Admin dapat melakukan *login* dan masuk pada aplikasi	
Preconditions	Admin telah daftar dan mempunyai *account*	
	Admin berhasil *login*	
Successful End Condition	Admin gagal melakukan *login*	
Failed End Condition	Admin	
Primary Actor	Admin memilih menu *login*	
Trigger		
Main Flow	*Step*	*Action*
	1.	Pengguna membuka halaman login
	2.	Sistem menampilkan form login
	3.	Pengguna mengisi form login dengan benar
	4.	Sistem melakukan validasi data login pengguna
	5.	Sistem menampilkan halaman awal aplikasi
Extension	5.1	Sistem menampilkan *error* gagal login

12. Use Case Lupa *Password*

Berikut ini adalah skenario yang menggambarkan use case lupa *password* untuk admin

Tabel 3.26 Skenario Use Case Lupa Password

Use Case Name	Lupa *Password*	
Related Requirements	-	
Goal Context	Admin melakukan reset atau mengubah *password*	
Preconditions	Admin berada pada halaman login aplikasi	
Successful End Condition	Admin berhasil mengubah *password*	
Failed End Condition	Admin gagal mengubah *password*	
Primary Actor	Admin	
Trigger	Admin memilih menu menu lupa *password*	
Main Flow	*Step*	*Action*
	1.	Pengguna membuka halaman lupa *password*
	2.	Sistem menampilkan form lupa *password*
	3.	Pengguna melakukan isi form lupa *password* berupa email
	4.	Sistem mengirim *password* baru melalui email pengguna

13. Use Case Kelola Data Titik Rawan

Berikut ini adalah skenario yang menggambarkan use case Kelola Data Titik Rawan.

Tabel 3.27 Skenario Use Case Kelola Data Titik Rawan

Use Case Name	Kelola Data Titik Rawan	
Related Requirements	-	
Goal Context	Admin dapat melakukan tambah, hapus, ubah dan lihat data titik rawan	
Preconditions	Admin telah berada dalam sistem	
Successful End Condition	Admin berhasil melakukan tambah, hapus, ubah dan lihat data titik rawan	
Failed End Condition	Admin gagal melakukan tambah, hapus, ubah dan lihat data titik rawan	
Primary Actor	Admin	
Trigger	Admin memilih menu data titik rawan	
Main Flow	*Step*	*Action*
	5.	Pengguna membuka halaman data titik rawan
	6.	Sistem menampilkan data titik rawan
	7.	Pengguna melakukan tambah, hapus, ubah dan lihat titik rawan
	8.	Sistem menyimpan apa yang pengguna lakukan
Extension	4.1	Sistem gagal menyimpan perubahan data titik rawan

14. Use Case Kelola Data Pengguna

Berikut ini adalah skenario yang menggambarkan use case Kelola Data Pengguna.

Tabel 3.28 Skenario Use Case Kelola Data Pengguna

Use Case Name	Kelola Data Pengguna	
Related Requirements	-	
Goal Context	Admin dapat melakukan tambah, hapus dan lihat data pengguna	
Preconditions	Admin telah berada dalam sistem	
Successful End Condition	Admin berhasil melakukan tambah, hapus dan lihat data pengguna	
Failed End Condition	Admin gagal melakukan tambah, hapus dan lihat data pengguna	
Primary Actor	Admin	
Trigger	Admin memilih menu data pengguna	
Main Flow	*Step*	*Action*
	1.	Pengguna membuka halaman data pengguna pada sistem
	2.	Sistem menampilkan data pengguna
	3.	Pengguna melakukan tambah, hapus lihat data pengguna
	4.	Sistem menyimpan apa yang pengguna lakukan
Extension	4.1	Sistem gagal menyimpan perubahan data pengguna

15. Use Case Kelola Data Pelaporan

Berikut ini adalah skenario yang menggambarkan use case Kelola Data Pelaporan.

Tabel 3.29 Skenario Use Case Kelola Data Pelaporan

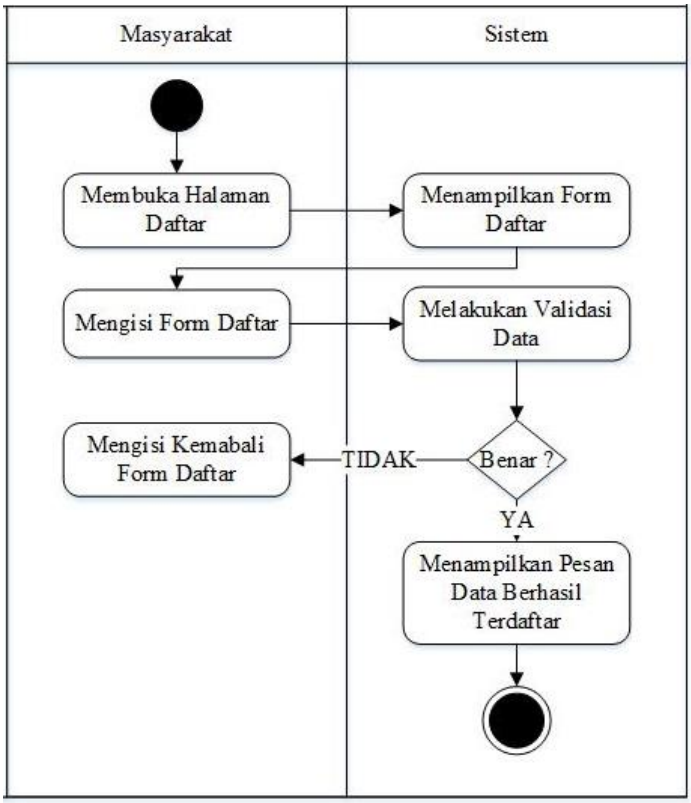
Use Case Name	Kelola Data Pelaporan	
Related Requirements	-	
Goal Context	Admin dapat menyujutui dan menolak pelaporan masyarakat	
Preconditions	Admin telah berada dalam sistem	
Successful End Condition	Admin berhasil melakukan aksi setujui atau tolak pelaporan masyarakat	
Failed End Condition	Admin gagal melakukan aksi pada paleporan masyarakat	
Primary Actor	Admin	
Trigger	Admin memilih menu data pelaporan	
Main Flow	*Step*	*Action*
	1.	Pengguna membuka halaman data pelaporan masyarakat
	2.	Sistem menampilkan data pelaporan
	3.	Pengguna menyetujui ataumenolak pelpaoran masyarakat
	4.	Sistem menyimpan apa yang pengguna lakukan

3.1.9.3 Activity Diagram

Diagram *Activity* ini dibuat untuk menggambarkan urutan proses dari setiap *use case*. *Activity diagram* sangat bermanfaat untuk memahami proses dari sistem secara keseluruhan. Adapun *Activity diagram* dari setiap *use case* adalah sebagai berikut.

1. *Activity Diagram* Daftar

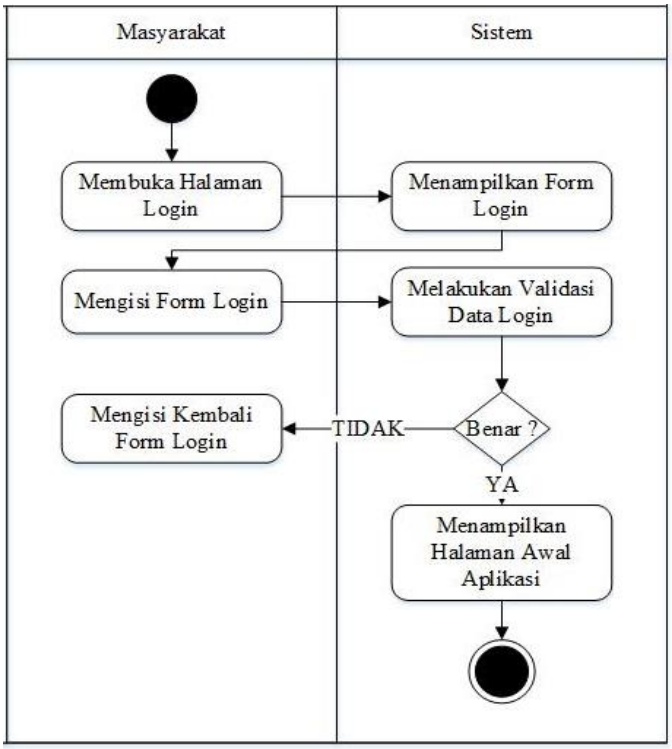
   Berikut ini urutan proses dari *use case* Daftar.



Gambar 3.9 *Activity Diagram* Daftar

2. *Activity Diagram Login*

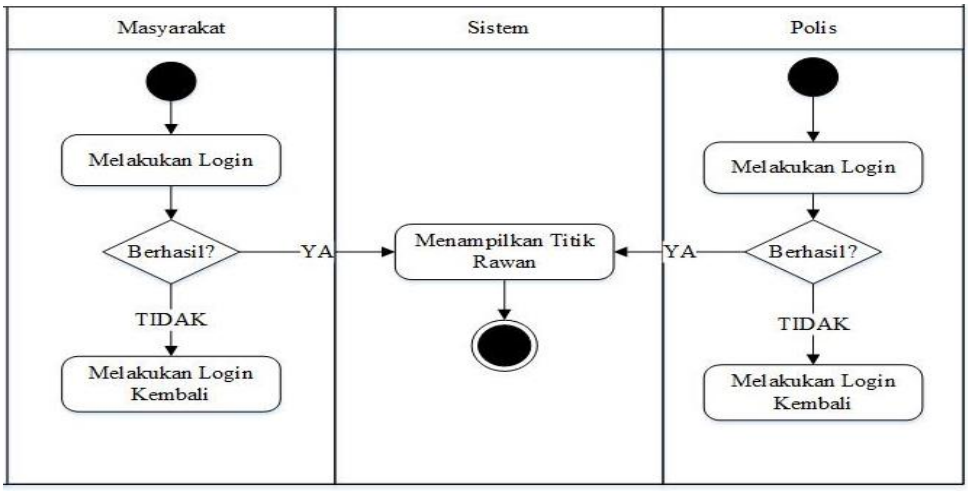
   Berikut ini urutan proses dari *use case Login*.



Gambar 3.10 *Activity Diagram Login* Masyarakat

3. *Activity Diagram* Lihat Titik Rawan

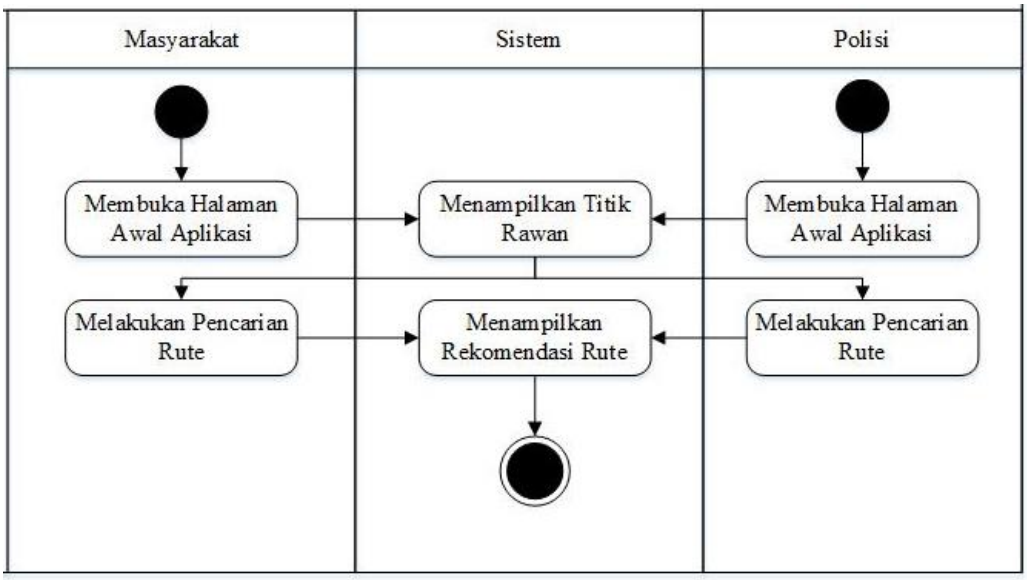
   Berikut ini urutan proses dari *use case* Lihat Titik Rawan.



Gambar 3.11 *Activity Diagram* Lihat Titik Rawan

4. *Activity Diagram* Cari Jalur Aman

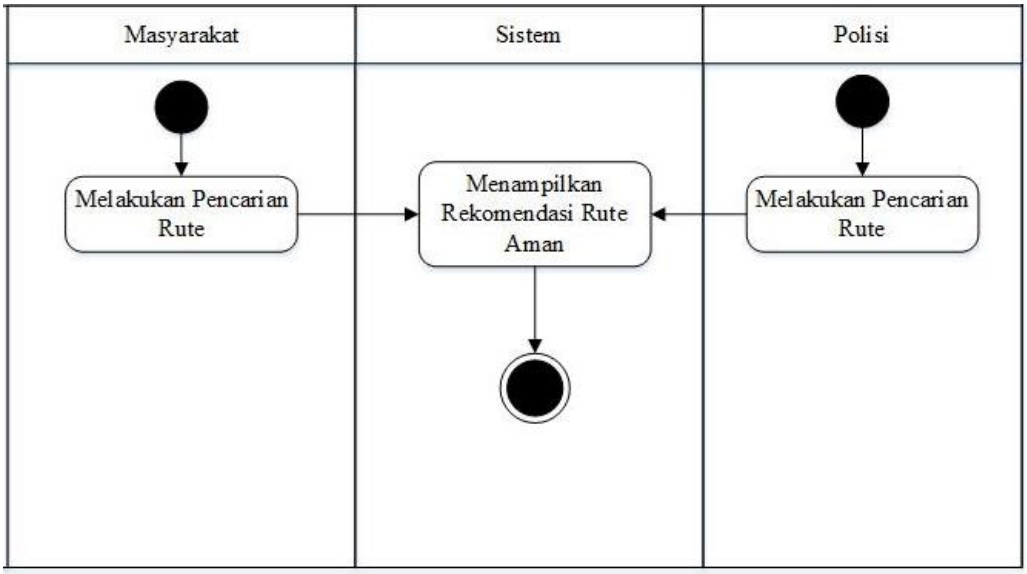
   Berikut ini urutan proses dari *use case* Cari Jalur Aman.



Gambar 3.12 *Activity Diagram* Cari Jalur Aman

5. *Activity Diagram* Lihat Jalur Aman

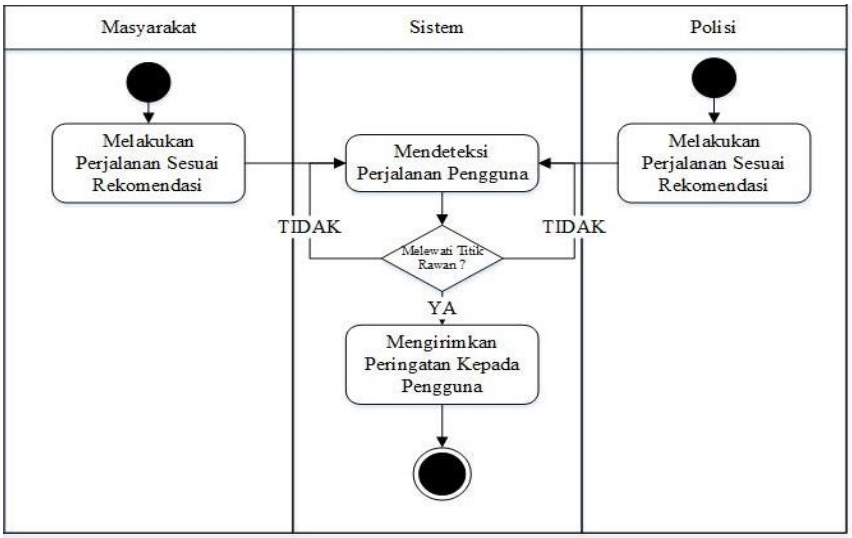
   Berikut ini urutan proses dari *use case* Lihat Jalur Aman.



Gambar 3.13 *Activity Diagram* Lihat Jalur Aman

6. *Activity Diagram* Lihat Pemberitahuan

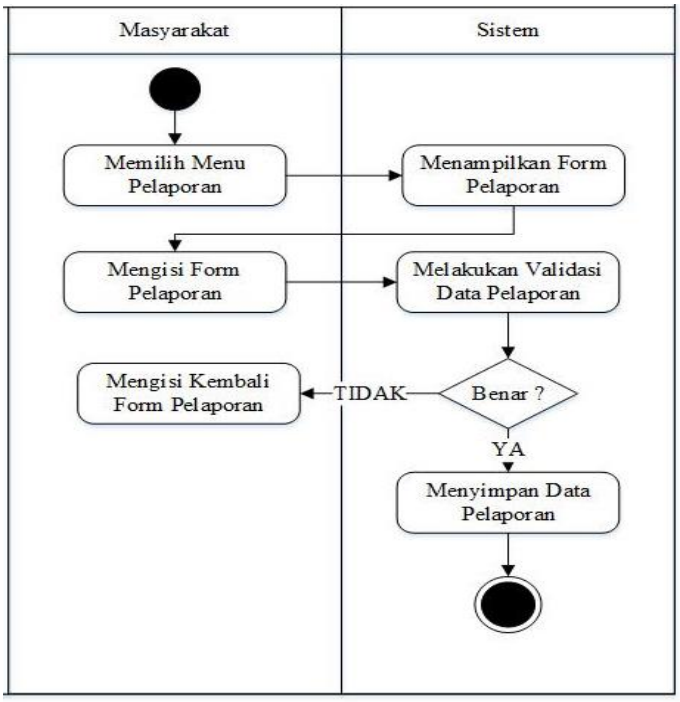
   Berikut ini urutan proses dari *use case* Lihat Pemberitahuan.



Gambar 3.14 *Activity Diagram* Lihat Pemberitahuan

7. *Activity Diagram* Lakukan Pelaporan

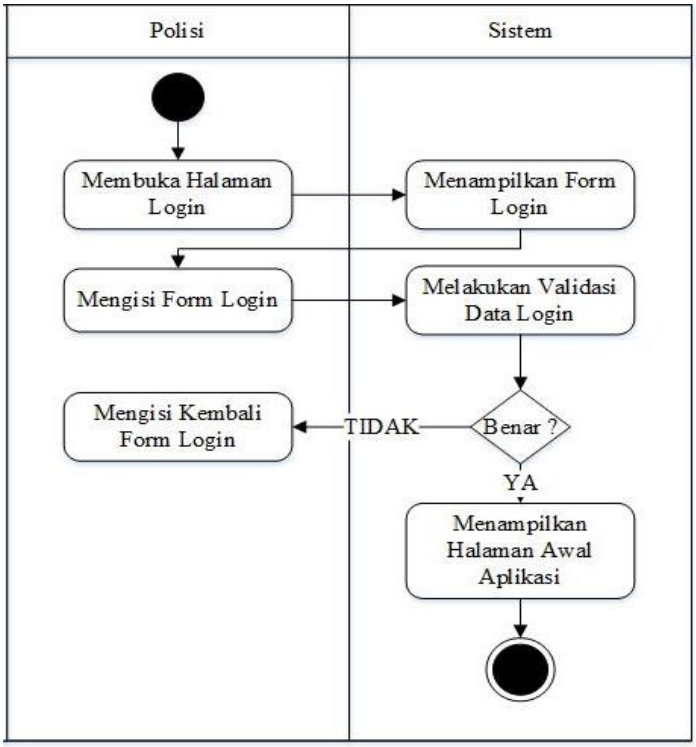
   Berikut ini urutan proses dari *use case* Lakukan Pelaporan.



Gambar 3.15 *Activity Diagram* Lakukan Pelaporan

8. *Activity Diagram Login* Polisi

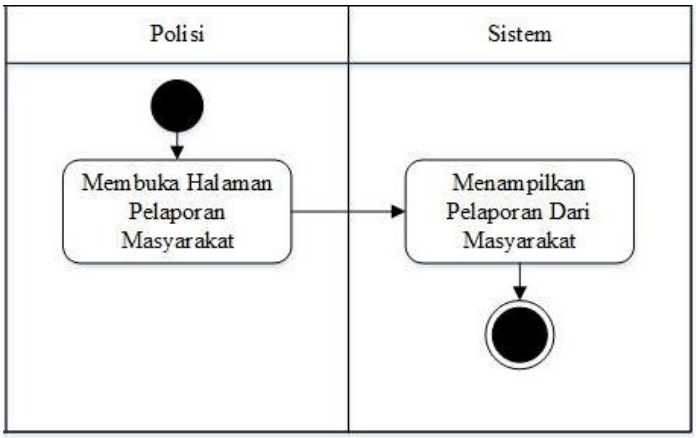
   Berikut ini urutan proses dari *use case Login* Polisi.



Gambar 3.16 *Activity Diagram Login* Polisi

9. *Activity Diagram* Lihat Pelaporan

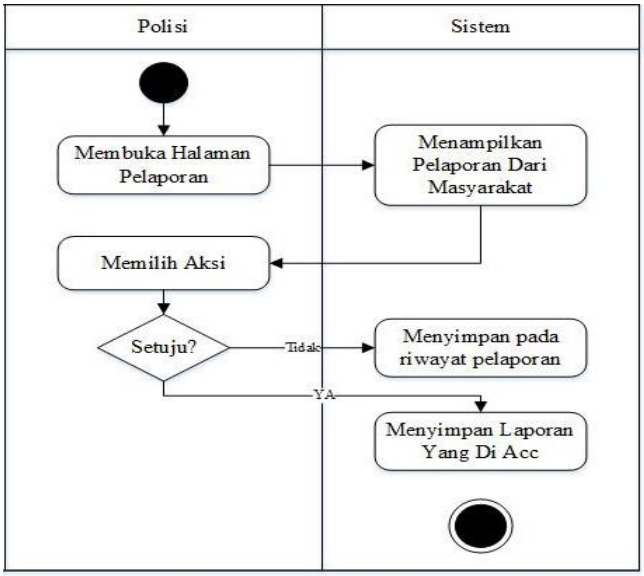
   Berikut ini urutan proses dari *use case* Lihat Pelaporan.



Gambar 3.17 *Activity Diagram* Lihat Pelaporan

10. *Activity Diagram* Setujui Pelaporan

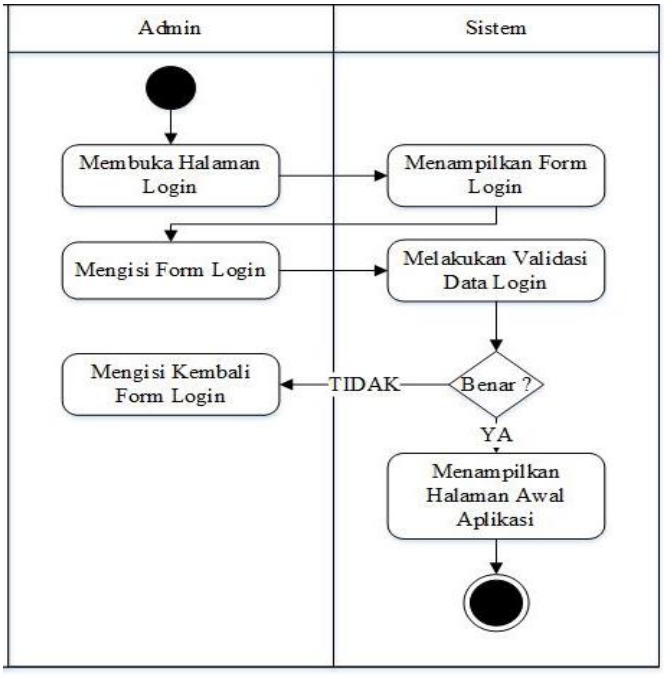
Berikut ini urutan proses dari *use case* Setujui Pelaporan.



Gambar 3.18 *Activity Diagram* Setujui Pelaporan

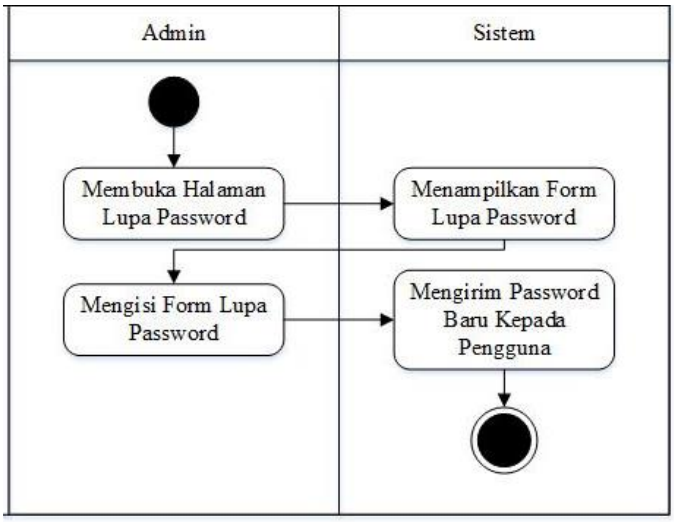
11. *Activity Diagram Login*

Berikut ini urutan proses dari *use case Login*.



Gambar 3.19 *Activity Diagram Login* Admin

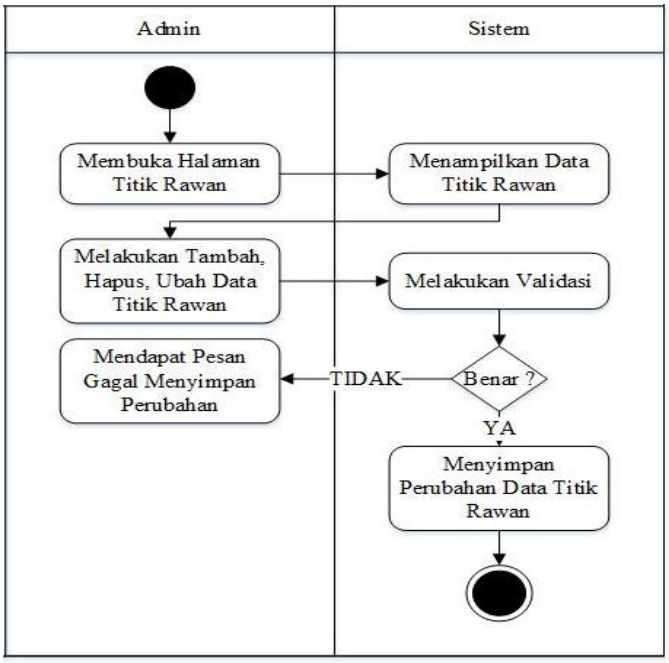
12. *Activity Diagram* Lupa *Password*



Gambar 3.20 *Activity Diagram* Lupa *Password*

13. *Activity Diagram* Kelola Data Titik Rawan

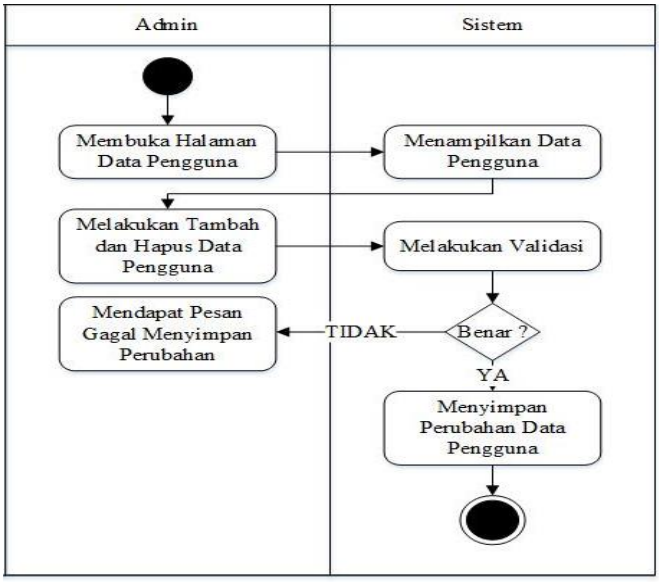
Berikut ini urutan proses dari *use case* Kelola Data Titik Rawan.



Gambar 3.21 *Activity Diagram* Kelola Data Titik Rawan

14. *Activity Diagram* Kelola Data Pengguna

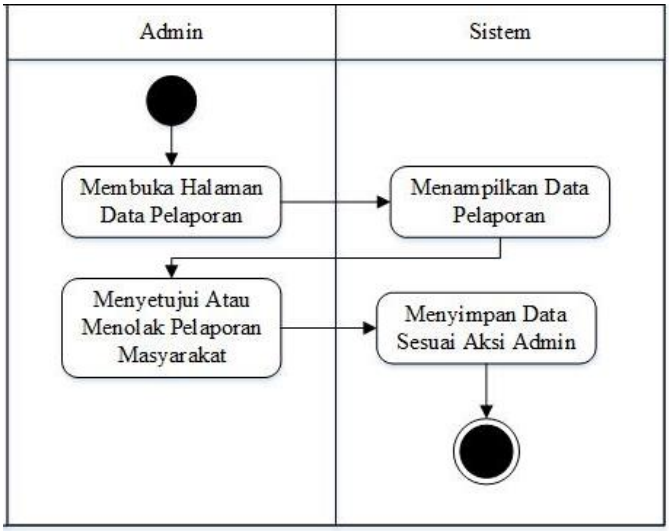
Berikut ini urutan proses dari *use case* Kelola Data Pengguna.



Gambar 3.22 *Activity Diagram* Kelola Data Pengguna

15. *Activity Diagram* Kelola Data Pelaporan

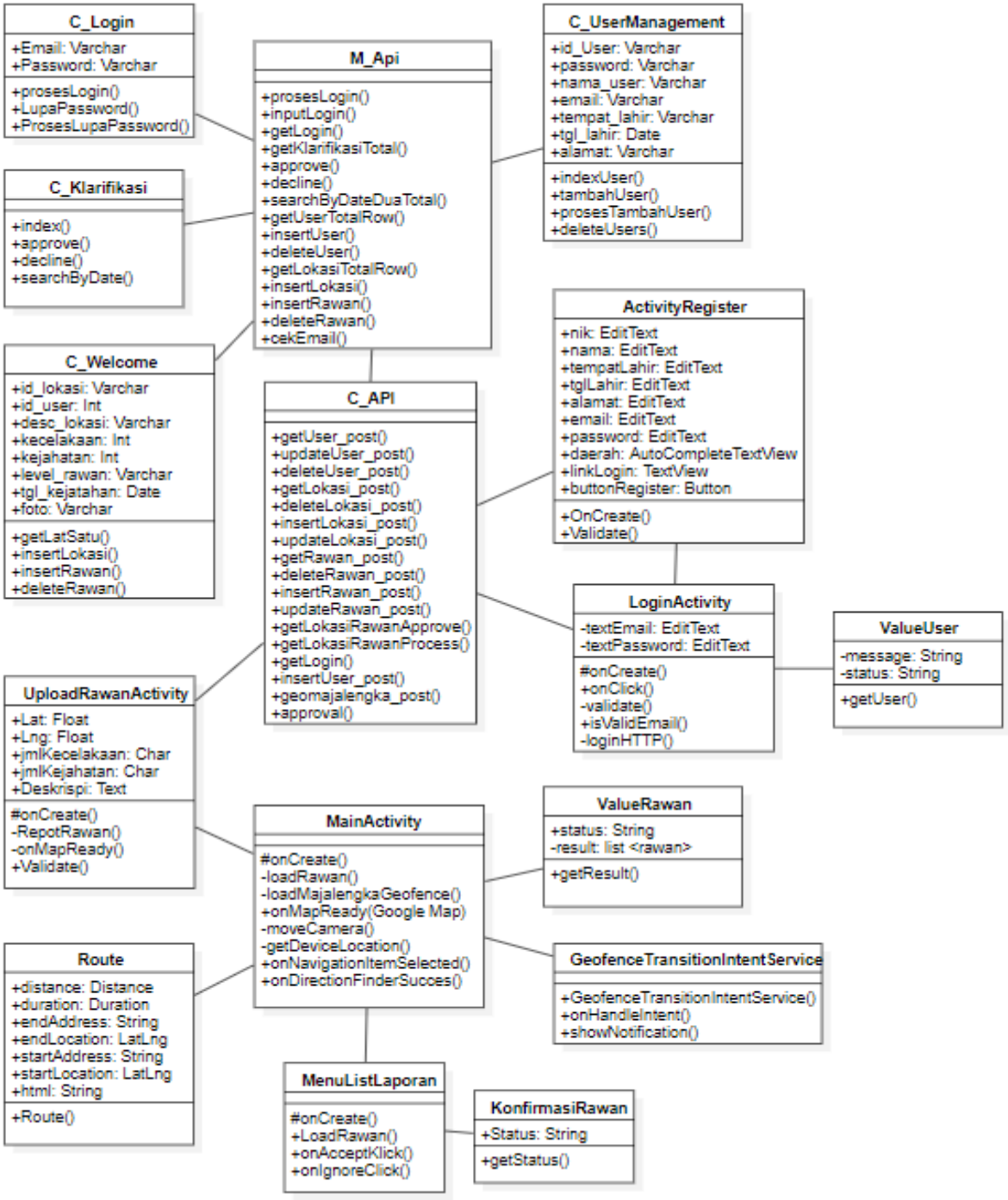
Berikut ini urutan proses dari *use case* Kelola Data Pelporan.



Gambar 3.23 *Activity Diagram* Kelola Data Pelaporan

3.1.9.4 Class Diagram

Class diagram dibuat untuk menggambarkan class yang terdapat pada sistem yang akan dibangun. Adapun *class diagram* pada sistem yang akan dibangun adalah sebagai berikut.



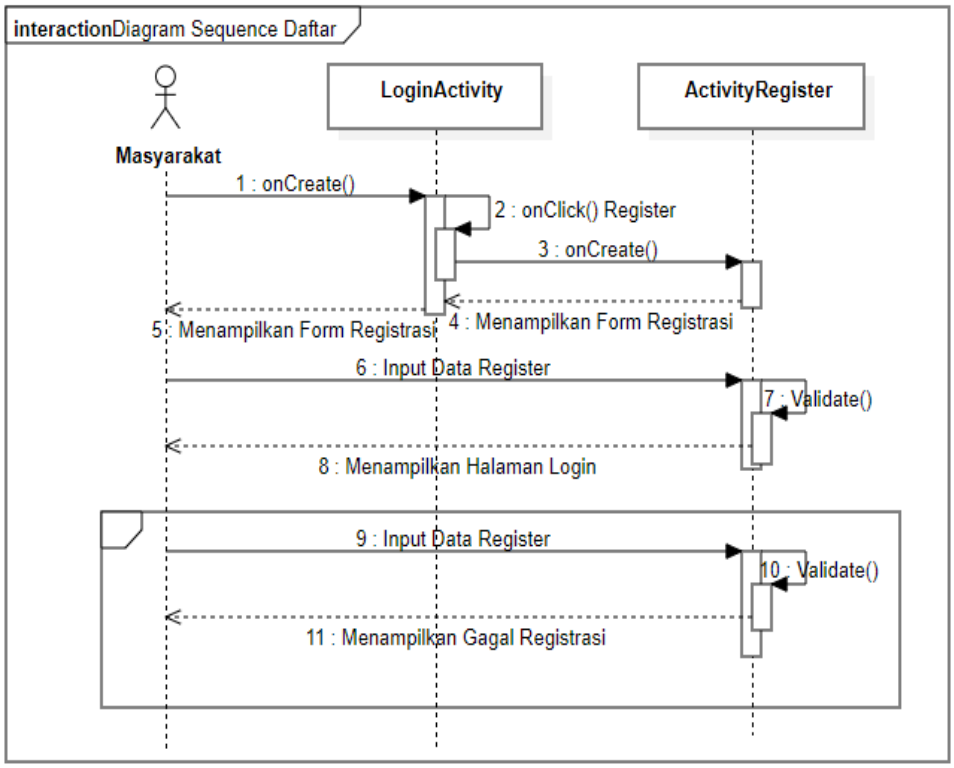
Gambar 3.24 *Class Diagram*

3.1.9.5 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa message yang digambarkan terhadap waktu. Berikut ini *sequence diagram* dari sistem yang akan dibangun.

1. *Sequence Diagram* Daftar

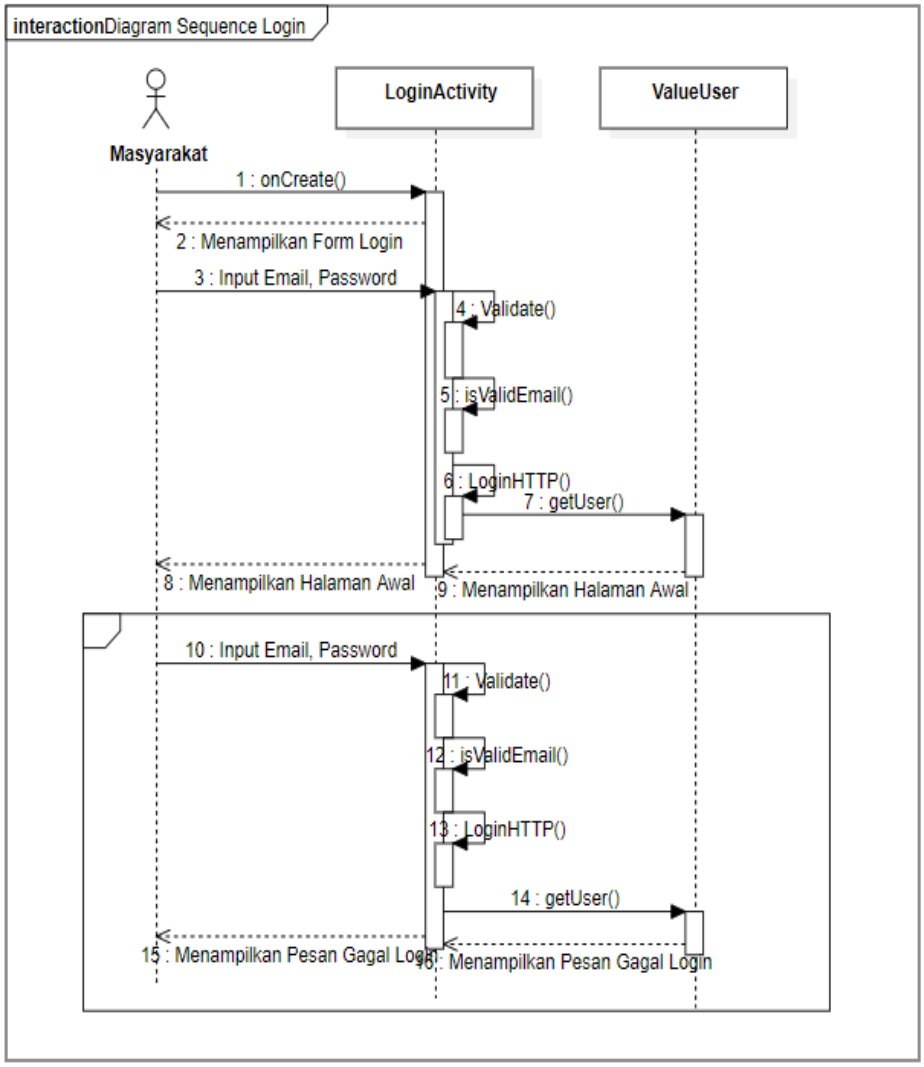
   Berikut ini *sequence diagram* dari use case Daftar



Gambar 3.25 Diagram *Sequence* Daftar

2. *Sequence Diagram Login*

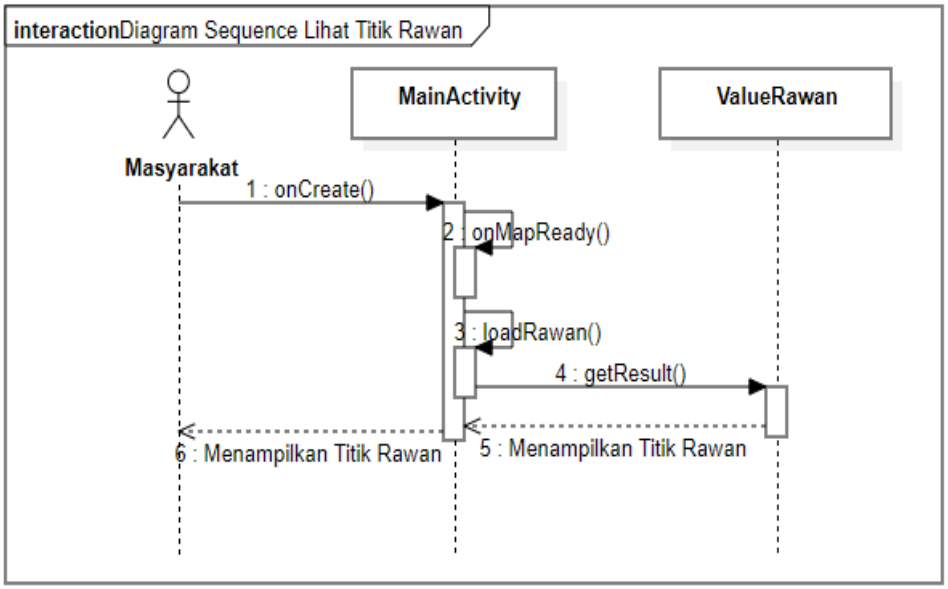
Berikut ini *sequence diagram* dari use case *Login* Masyarakat



Gambar 3.26 Diagram *Sequence Login* Masyarakat

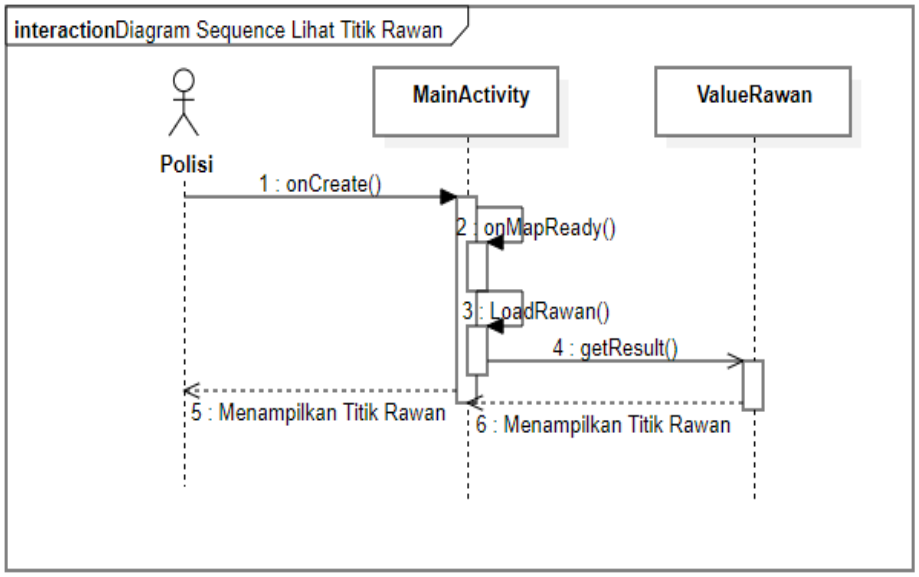
3. *Sequence Diagram* Lihat Titik Rawan

Berikut ini *sequence diagram* dari use case Lihat Titik Rawan Masyarakat



Gambar 3.27 Diagram *Sequence* Lihat Titik Rawan Masyarakat

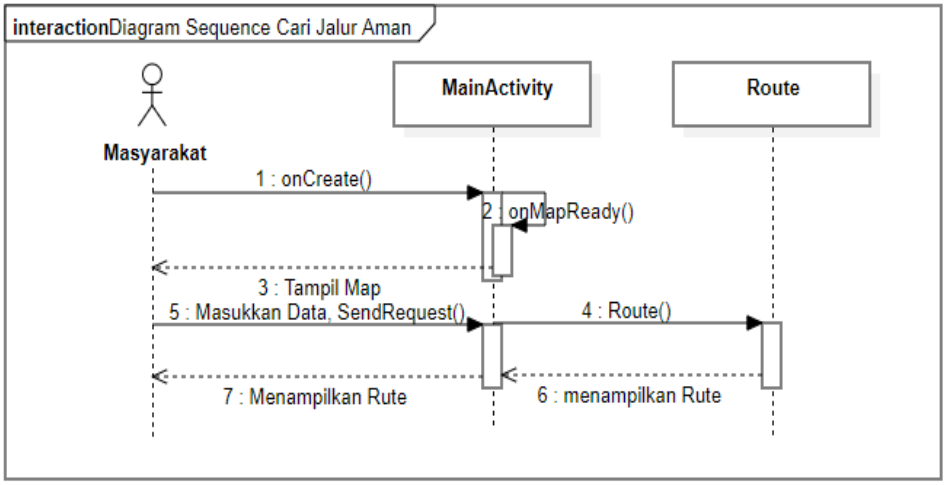
Berikut ini *sequence diagram* dari use case Lihat Titik Rawan Polisi



Gambar 3.28 Diagram *Sequence* Lihat Titik Rawan Polisi

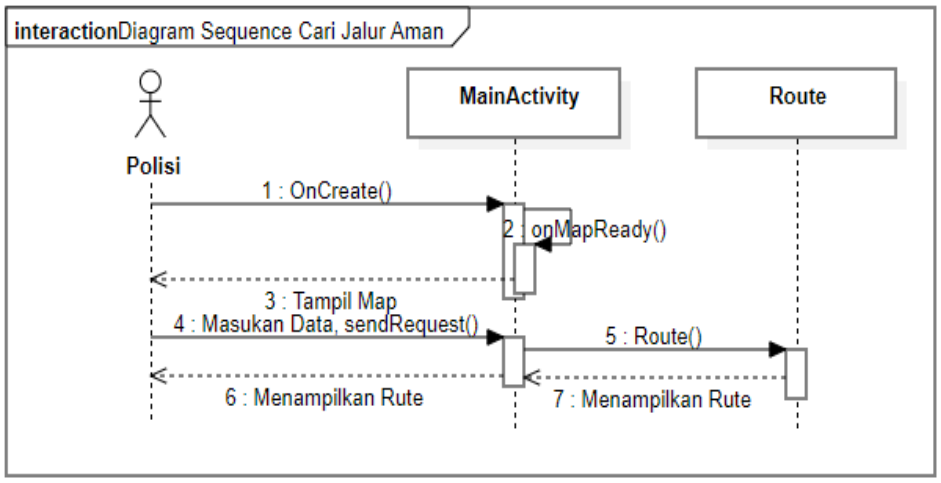
4. *Sequence Diagram* Cari Jalur Aman

Berikut ini *sequence diagram* dari use case Cari Jalur Aman Masyarakat



Gambar 3.29 Diagram *Sequence* Cari Jalur Aman Masyarakat

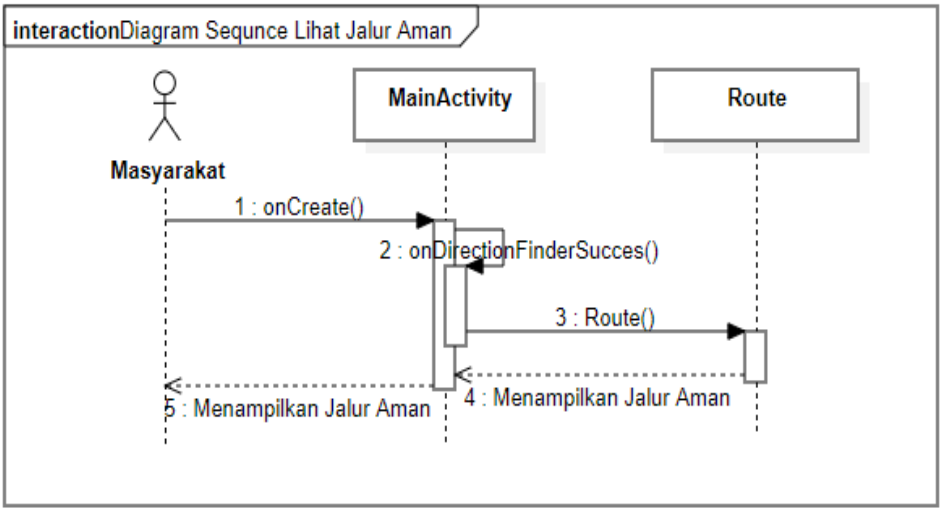
Berikut ini *sequence diagram* dari use case Cari Jalur Aman Polisi



Gambar 3.30 Diagram *Sequence* Cari Jalur Aman Polisi

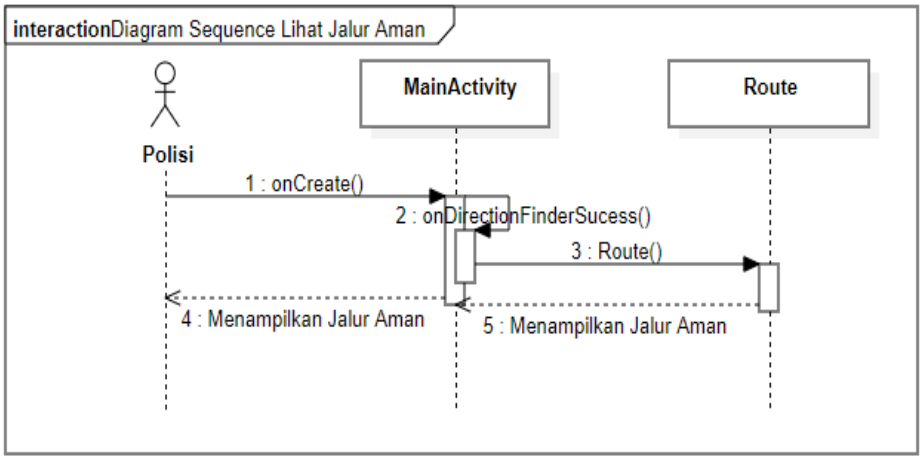
5. *Sequence Diagram* Lihat Jalur Aman

Berikut ini *sequence diagram* dari use case Lihat Jalur Aman Masyarakat



Gambar 3.31 Diagram *Sequence* Lihat Jalur Aman Masyarakat

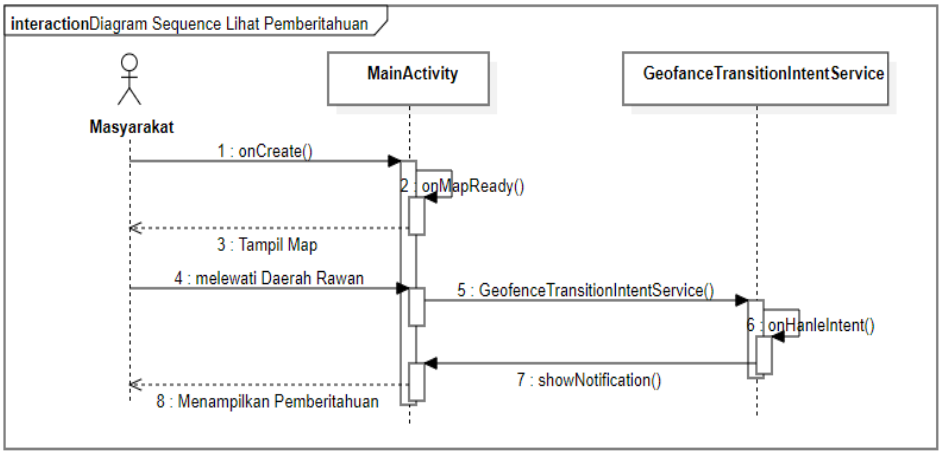
Berikut ini *sequence diagram* dari use case Lihat Jalur Aman Polisi



Gambar 3.32 Diagram *Sequence* Lihat Jalur Aman Polisi

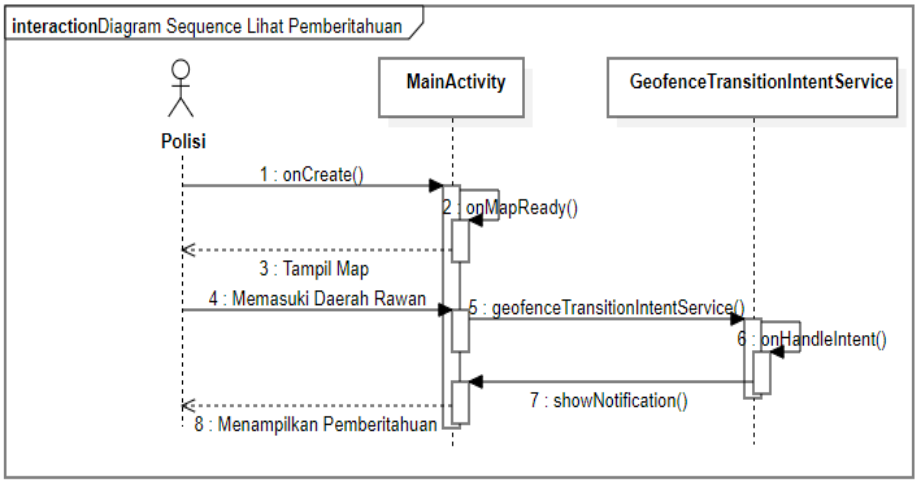
6. *Sequence Diagram* Lihat Pemberitahuan

   Berikut ini *sequence diagram* dari use case Lihat Pemberitahuan Masyarakat



Gambar 3.33 Diagram *Sequence* Lihat Pemberitahuan Masyarakat

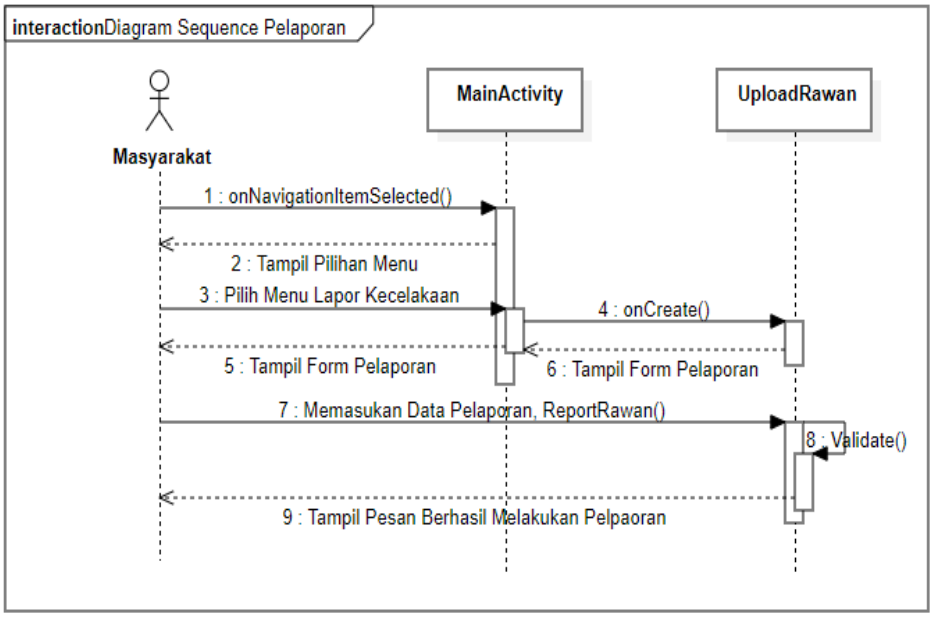
Berikut ini *sequence diagram* dari use case Lihat Pemberitahuan Polisi



Gambar 3.34 Diagram *Sequence* Lihat Pemberitahuan Polisi

7. *Sequence Diagram* Lakukan Pelaporan

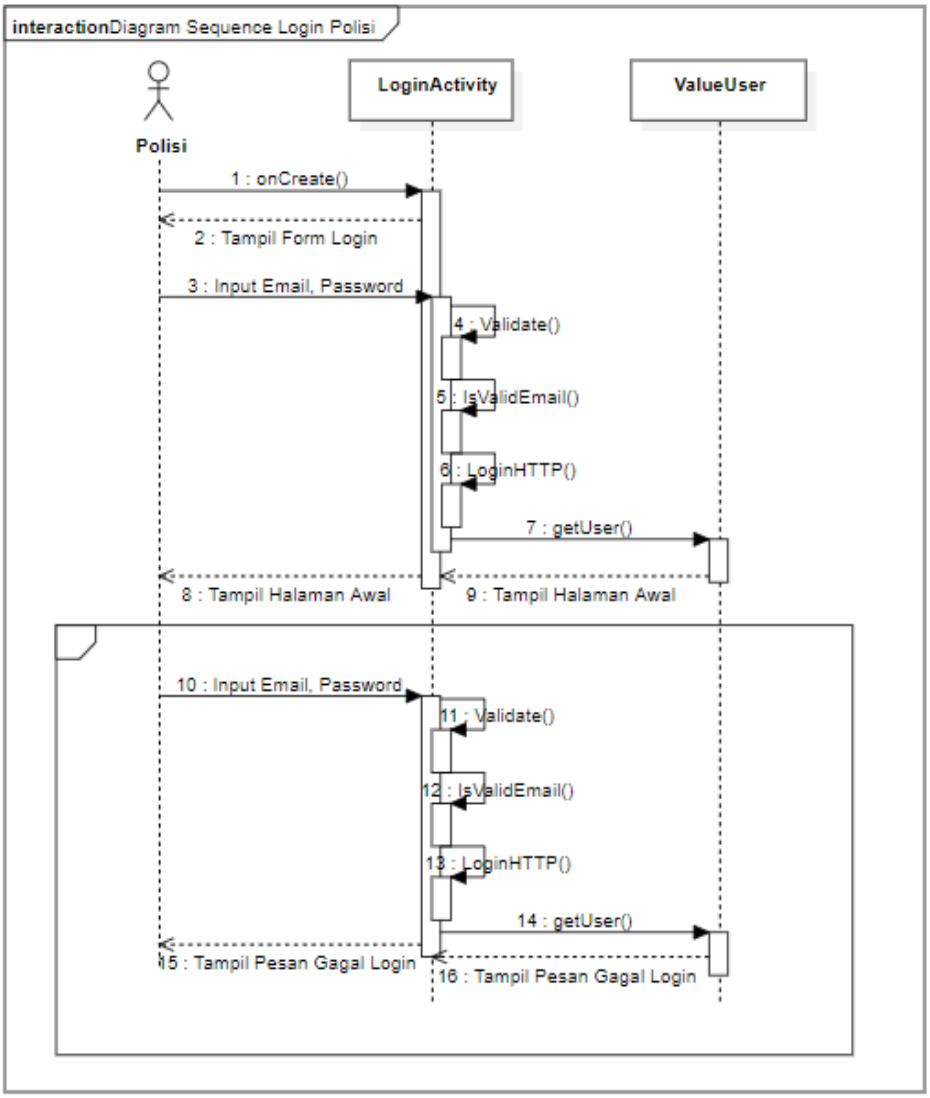
   Berikut ini *sequence diagram* dari use case Lakukan Pelaporan



Gambar 3.35 Diagram *Sequence* Lakukan Pelpaoran

8. *Sequence Diagram Login*

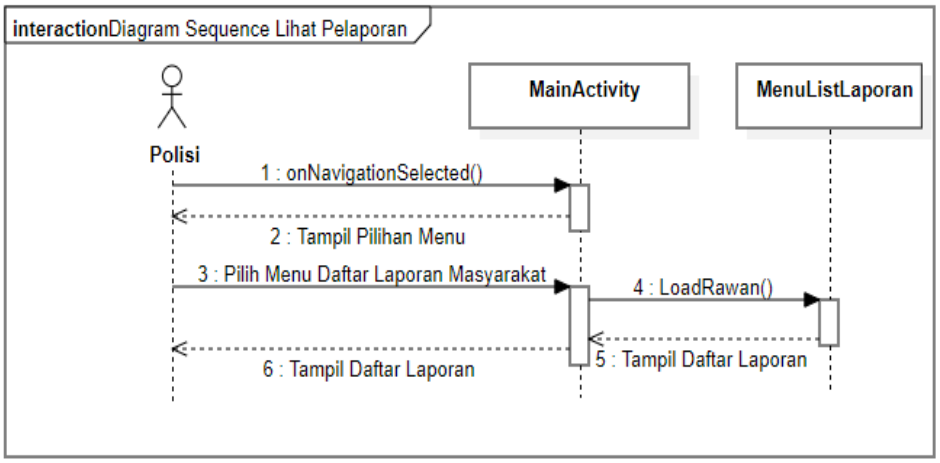
Berikut ini *sequence diagram* dari use case *Login* Polisi



Gambar 3.36 Diagram *Sequence Login* Polisi

9. *Sequence Diagram* Lihat Pelaporan

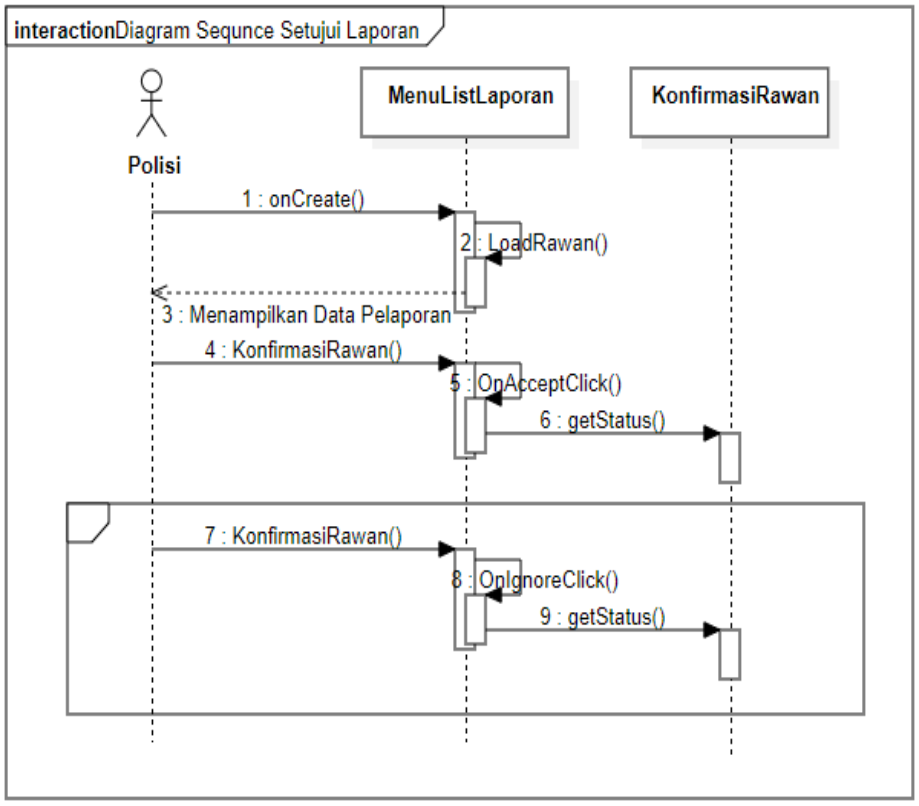
   Berikut ini *sequence diagram* dari use case Lihat Pelaporan



Gambar 3.37 Diargam *Sequence* Lihat Pelaporan

10. *Sequence Diagram* Setujui Pelaporan

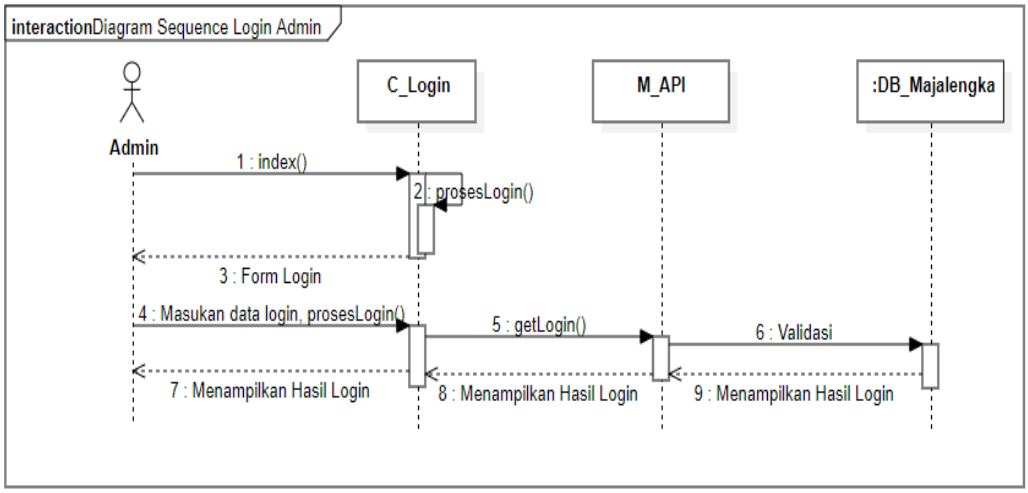
    Berikut ini *sequence diagram* dari use case Setujui Pelaporan



Gambar 3.38 Diagram *Sequence* Setujui Pelaporan

11. *Sequence Diagram Login*

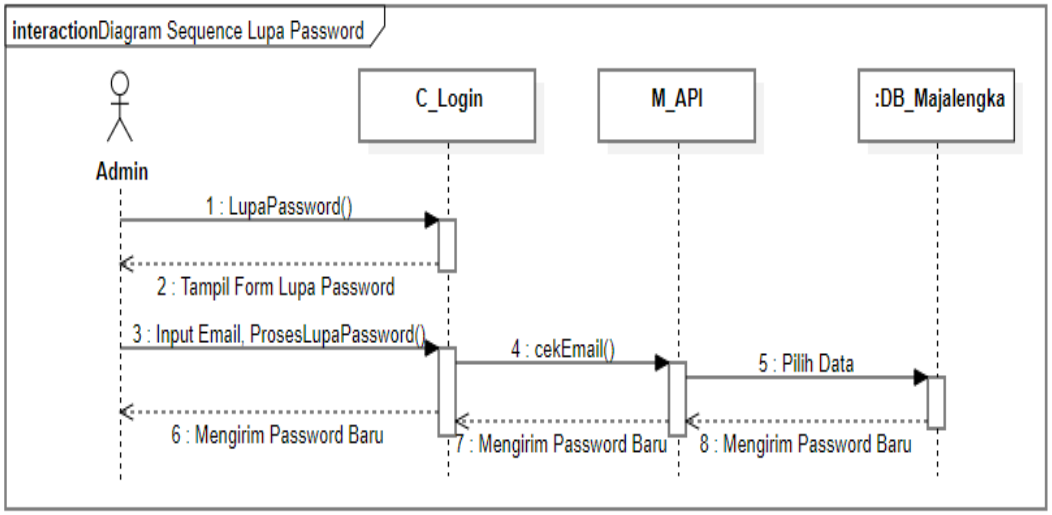
Berikut ini *sequence diagram* dari use case *Login* Admin



Gambar 3.39 Diagram *Sequence Login* Admin

12. *Sequence Diagram* Lupa *Password*

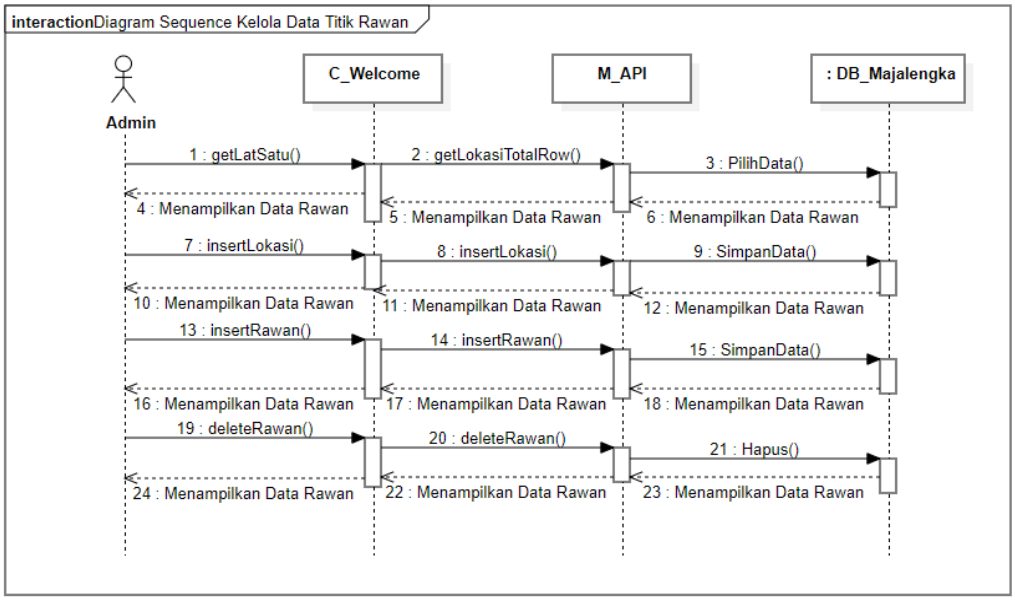
Berikut ini *sequence diagram* dari use case Lupa *Password*



Gambar 3.40 Diagram *Sequence* Lupa *Password*

13. *Sequence Diagram* Kelola Data Titik Rawan

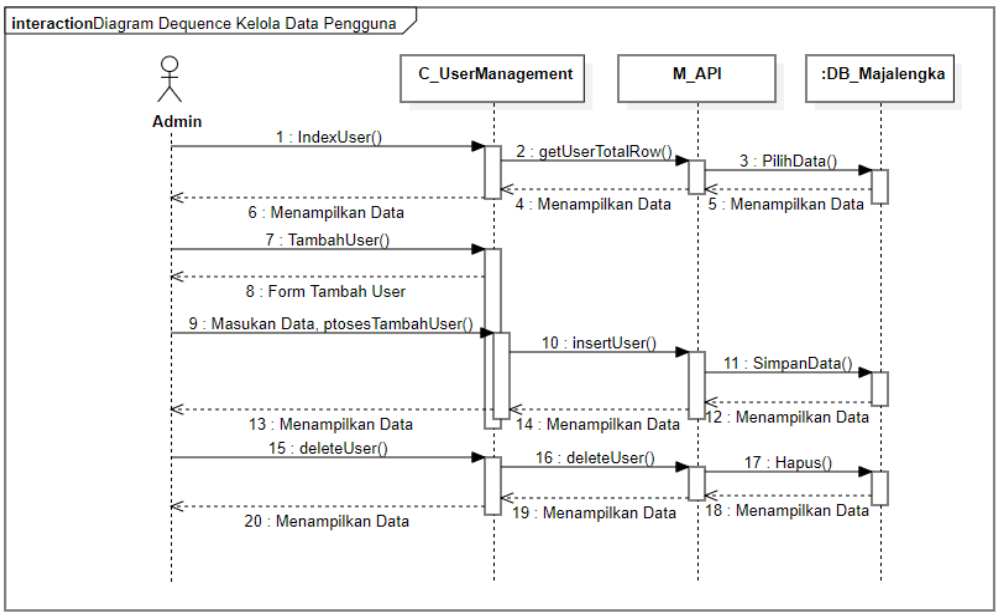
Berikut ini *sequence diagram* dari use case Kelola Data Titik Rawan



Gambar 3.41 Diagram *Sequence* Kelola Data Titik Rawan

14. *Sequence Diagram* Kelola Data Pengguna

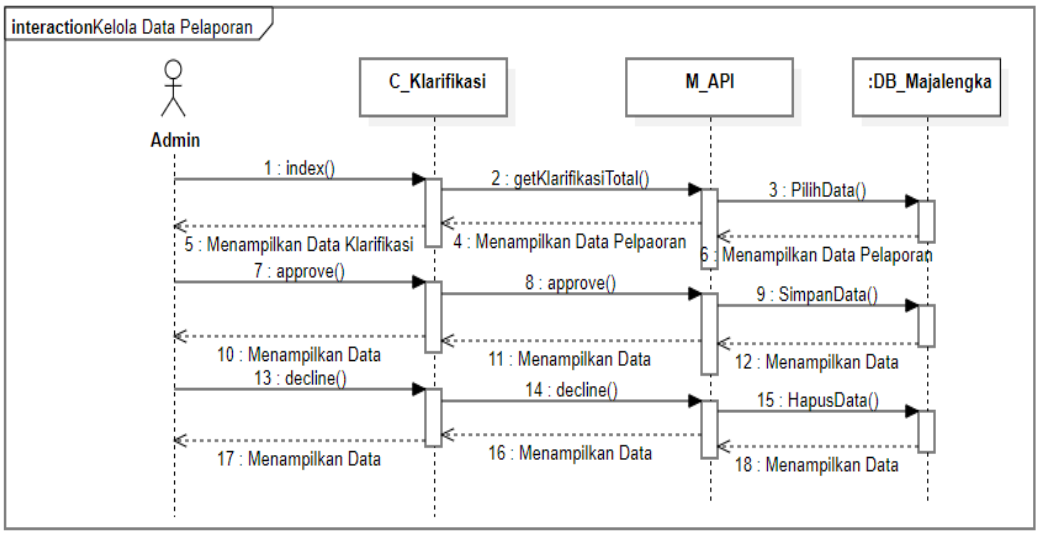
Berikut ini *sequence diagram* dari use case Kelola Data Pengguna



Gambar 3.42 Diagram *Sequence* Kelola Data Pengguna

15. *Sequence Diagram* Kelola Data Pelaporan

Berikut ini *sequence diagram* dari use case Kelola Data Pelaporan



Gambar 3.43 Diagram *Sequence* Kelola Data Pelaporan

3.1.9.6 Analisis Data

Dalam sistem yang akan dibangun, terdapat tiga kategori untuk menentukan level dari setiap titik rawan, diantaranya level tinggi, sedang dan rendah. Level tersebut di berikan berdasarkan waktu dan jumlah kejadian kecelakaan maupun kejahatan. Data yang diperoleh dalam penelitian ini adalah data jumlah kecelakaan pada tahun 2000 sampai dengan 2017. Namun, data yang akan di amati adalah data jumlah kecelakaan di tahun 2017. Adapun sebagaian data yang telah diperoleh adalah sebagai berikut.

Tabel 3.30 Jumlah Kejadian Kecelakaan

JUMLAH KEJADIAN		JUMLAH KORBAN			
Tahun	**Jumlah**	**Meninggal**	**Luka Berat**	**Luka Ringan**	**Total**
2015	290	82	223	305	610
2016	219	90	93	349	532
2017	**358**	105	187	482	774

Dari tabel 3.27 diatas, terlihat bahwa pada tahun 2017 terjadi sebanyak 358 kejadian kecelakaan di titik yang berbeda. Dengan begitu, hasil pengamatan dari data tersebut adalah sebagai berikut.

Tabel 3.31 Perkiraan Jumlah Kecelakaan

Bulan	Kejaian Per Bulan	Kejadian Per Minggu	Kejadian Per Hari
1 – 11	29 kejadian	7 kejadian	1 kejadian
12	39 kejadian	9 kejadian	2 kejadian

Pada tabel 3.28 diatas, dapat di perkirakan kejadian kecelakaan pada tahun 2017, terdapat sekitar 11 bulan dengan jumlah 29 kejadian dan ada satu bulan yang kejadiannya sekitar 39 kejadian. Dari hasil jumlah kejadian perbulan tersebut, diturunkan lagi menjadi per minggu dengan hasil dimana sekitar 11 bulan terdapat 7 kejadian dalam waktu seminggu dan terdapat satu bulan yang terjadi sekitar 9 kejadian dalam 1 minggunya. Sehingga dapat di perkirakan jumlah kecelakaan dalam 1 hari sekitar 1 sampai dengan 2 kejadian.

Dari perkiraan diatas, karena hasilnya dalam 1 minggu terdapat minimal 7 kejadian, maka hasil yang seperti itu akan masuk pada level tingkat tinggi. Jika lokasi atau titik kecelakaan tersebut berada pada satu titik. Karena pada level

tersebut telah ditentukan bahwa jumlah kejadian kecelakaan yang terjadi minimal 1 kali dalam 1 minggu pada setiap titiknya.

3.2 Perancangan Sistem

Pada penelitian ini, tahap perancangan sistem dimulai setelah tahap analisis. Tahap perancangan sistem ini. Adapun perancangan sistem pada penelitian ini meliputi pembuatan skema relasi, struktur tabel, struktur menu, tampilan antarmuka sistem serta pembuatan jaringan semantik.

3.2.1 Perancangan Data

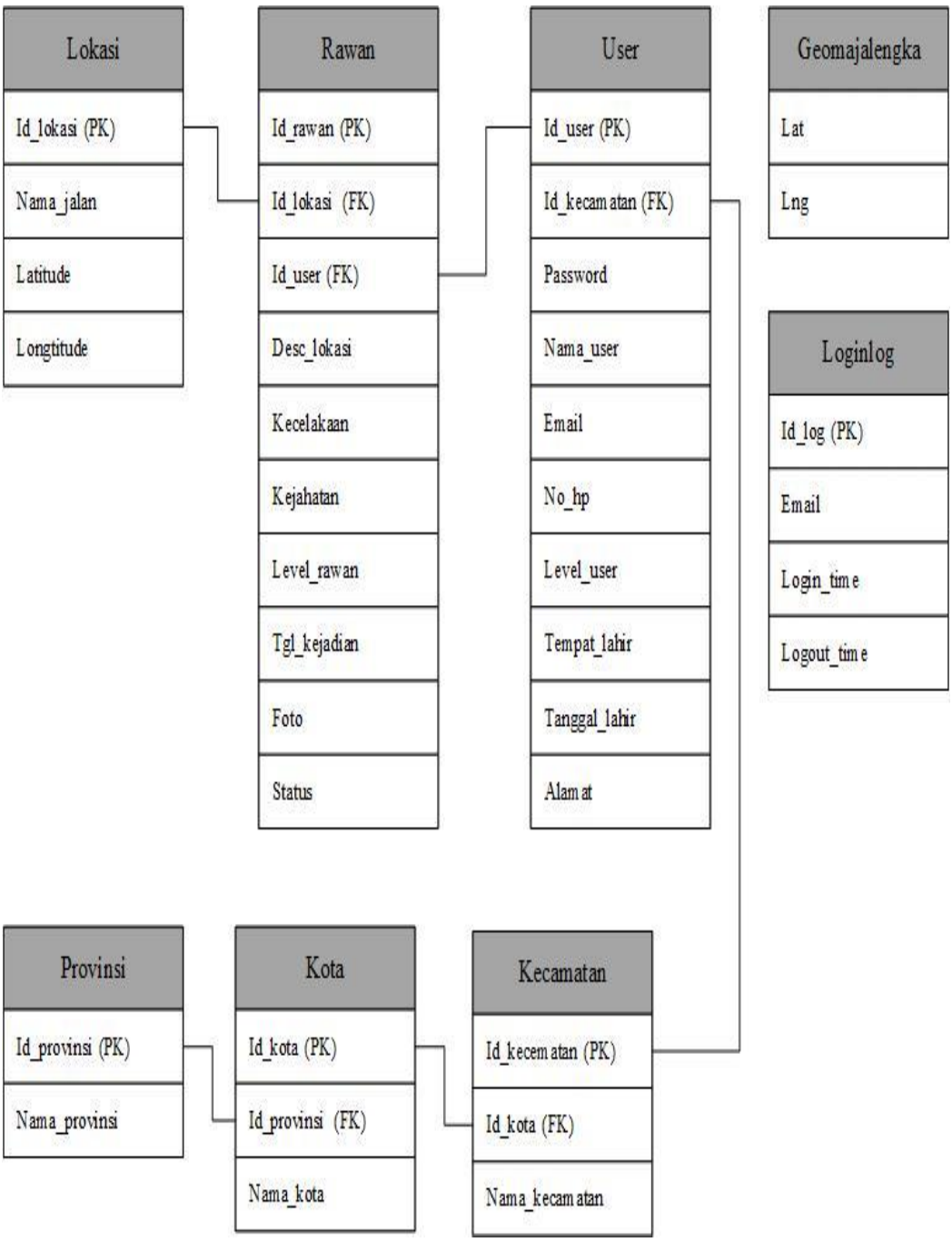
Perancangan data atau lebih dikenal dengan perancangan basis data yaitu menciptakan, atau merancang data yang terhubung dan disimpan secara bersama-sama. Dalam menggambarkan hubungan antar data, digunakan skema relasi dan struktur tabel.

3.2.1.1 Skema Relasi

Skema relasi merupakan rangkaian hubungan antara dua tabel atau lebih pada sistem basis data. Adapun data-data yang digunakan dalam sistem yang akan dibangun adalah sebagai berikut.

1. Lokasi : { *id_lokasi, nama_jalan, latitude, longtitude}
2. Rawan : { *id_rawan, **id_lokasi, **id_user, decs_lokasi, kecelakaan, kejahatan, level_rawan, tanggal_kejadian, foto, status}
3. *User* : { *id_user, **id_kecamatan, password, nama_user, email, no_hp, level_user, tempat_lahir, tanggal_lahir, Rt, Rw, alamat}
4. Provinsi : { *id_provinsi, nama_provinsi}
5. Kota : { *id_kota, **id_provinsi, nama_kota}
6. Kecamatan : { *id_kecamatan, **id_kota, nama_kecamatan}
7. Geomajalengka : { lat, lng}
8. *Loginlog* : { *id_log, email, login_time. Logout_time}

Adapaun skema relasi dari data-data diatas dapat dilihat pada gambar 3.44 dibawah ini.



Gambar 3.44 Skema Relasi

3.2.1.2 Struktur Tabel

Adapun struktur tabel dari skema relasi yang telah dibuat adalah sebagai berikut.

1. Tabel Lokasi

Tabel 3.32 Struktur Tabel Lokasi

Atribut	Tipe Data	Ukuran	Keterangan
Id_lokasi	Int	3	*Primary Key*
Nama_Jalan	Varchar	150	
Latitude	Float		
Longtitude	float		

2. Tabel Rawan

Tabel 3.33 Struktur Tabel Rawan

Atribut	Tipe Data	Ukuran	Keterangan
Id_rawan	Int	11	*Primary Key*
Id_lokasi	int	3	*Foreign Key*
Id_user	Varchar	16	*Foreign Key*
Desc_lokasi	Varchar	50	
Kecelakaan	Int	11	
Kejahatan	Int	4	
Level_rawan			
Tgl_kejadian	Date		
Foto	Varchar	50	
Status			

3. Tabel *User*

Tabel 3.34 Struktur Tabel *User*

Atribut	Tipe Data	Ukuran	Keterangan
Id_user	Varchar	16	*Primary Key*
Id_kecamatan	Char	7	*Foreign Key*
Password	Varchar	100	
Nama_user	Varchar	35	
Email	Varchar	100	
No_hp	Number		
Level_user			
Tempat_lahir	Varchar	35	
Tanggal_lahir	Date		
Desa	Varchar	35	
Rt	Int	4	
Rw	Int	4	
Alamat	Varchar	70	

4. Tabel Provinsi

Tabel 3.35 Struktur Tabel Provinsi

Atribut	Tipe Data	Ukuran	Keterangan
Id_provinsi	Char	2	*Primary Key*
Nama_provinsi	Varchar	255	

5. Tabel Kota

Tabel 3.36 Struktur Tabel Kota

Atribut	Tipe Data	Ukuran	Keterangan
Id_kota	Char	4	*Primary Key*
Id_provinsi	Char	2	*Foreign Key*
Nama_kota	Varchar	255	

6. Tabel Kecamatan

Tabel 3.37 Struktur Tabel Kecamatan

Atribut	Tipe Data	Ukuran	Keterangan
Id_kecamatan	Char	7	*Primary Key*
Id_kota	Char	4	*Foreign Key*
Nama_kecamatan	Varchar	255	

7. Tabel Geomajalengka

Tabel 3.38 Struktur Tabel Geomajalengka

Atribut	Tipe Data	Ukuran	Keterangan
Lat	Float		
Lng	Float		

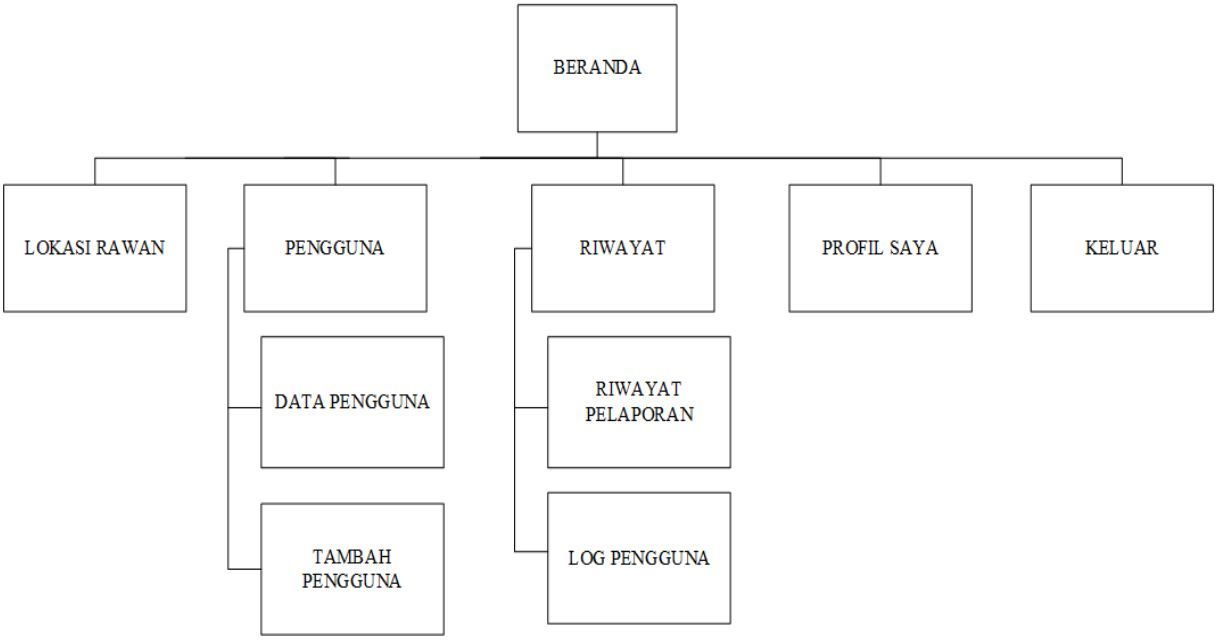
8. Tabel *Loginlog*

Tabel 3.39 Struktur Tabel Loginlog

Atribut	Tipe Data	Ukuran	Keterangan
Id_log	Int	11	*Primary Key*
Email	Varchar	100	
Login_log	Datetime		
Logout_log	datetime		

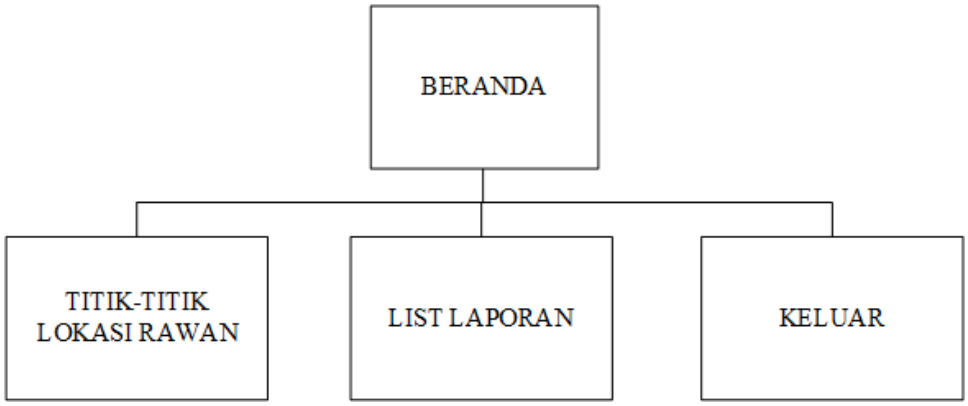
3.2.2 Perancangan Struktur Menu

Pada sistem yang akan dibangun ini terdapat dua flatform, yaitu flatform Web untuk Admin dan Mobile Anroid untuk masyarakat dan polisi. Berikut ini struktur menu pada flatform Web.



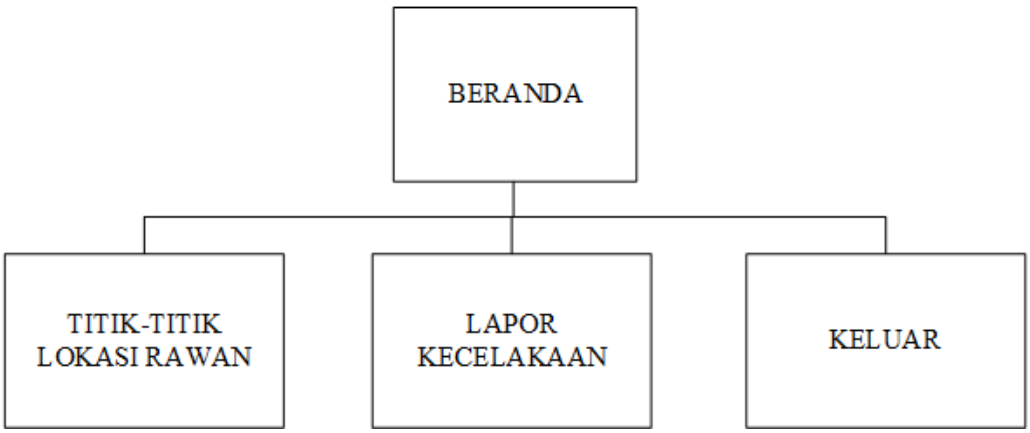
Gambar 3.45 Struktur Menu Web

Struktur menu halaman polisi adalah sebagai berikut.



Gambar 3.46 Struktur Menu Halaman Polisi

Dan struktur menu halaman masyarakat adalah sebagai berikut.



Gambar 3.47 Struktur Menu Halaman Masyarakat

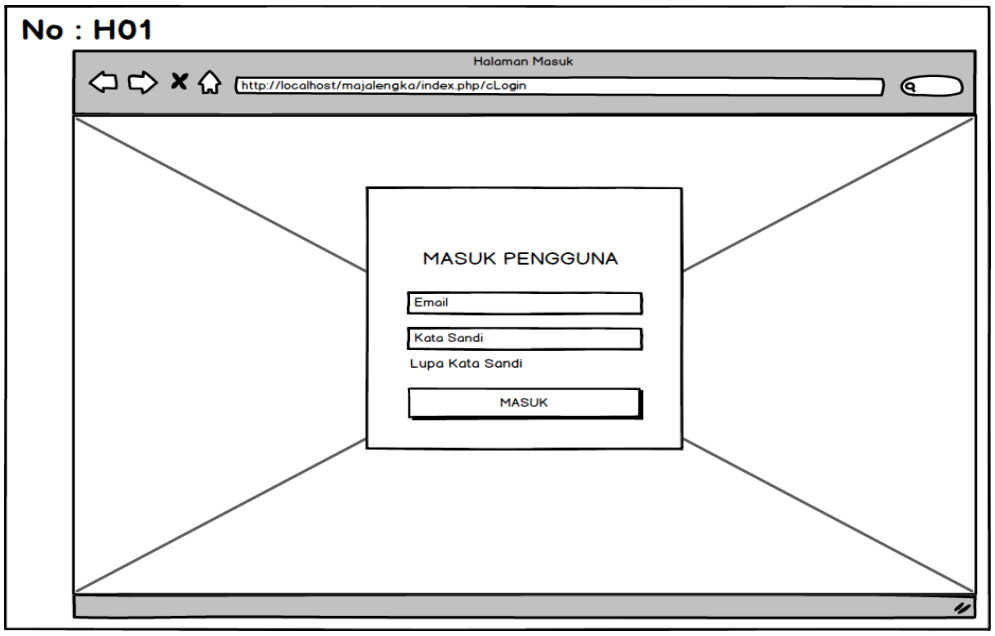
3.2.3 Perancangan Antarmuka

Pada sistem yang akan dibangun ini terdapat dua flatform, yaitu flatform Web untuk Admin dan Mobile Anroid untuk masyarakat dan polisi. Berikut ini perancangan antarmuka dari pada flatform Web.

3.2.3.1 Perancangan Antarmuka Halaman Admin

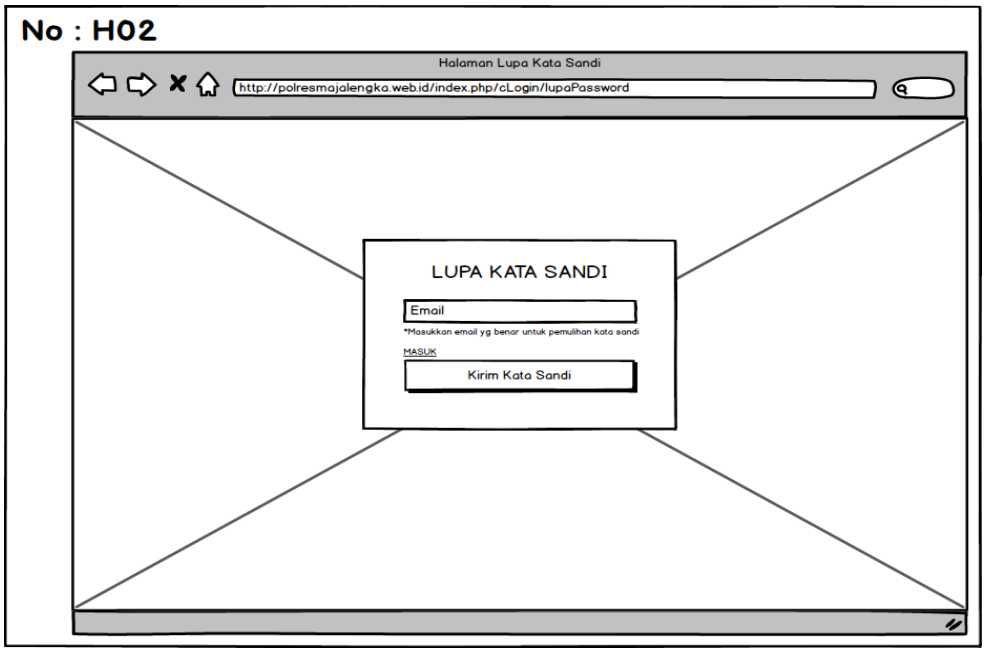
Berikut ini perancangan antarmuka pada flatform web untuk admin dari sistem yang akan dibangun.

1. Perancangan antarmuka halaman masuk Admin (H01)



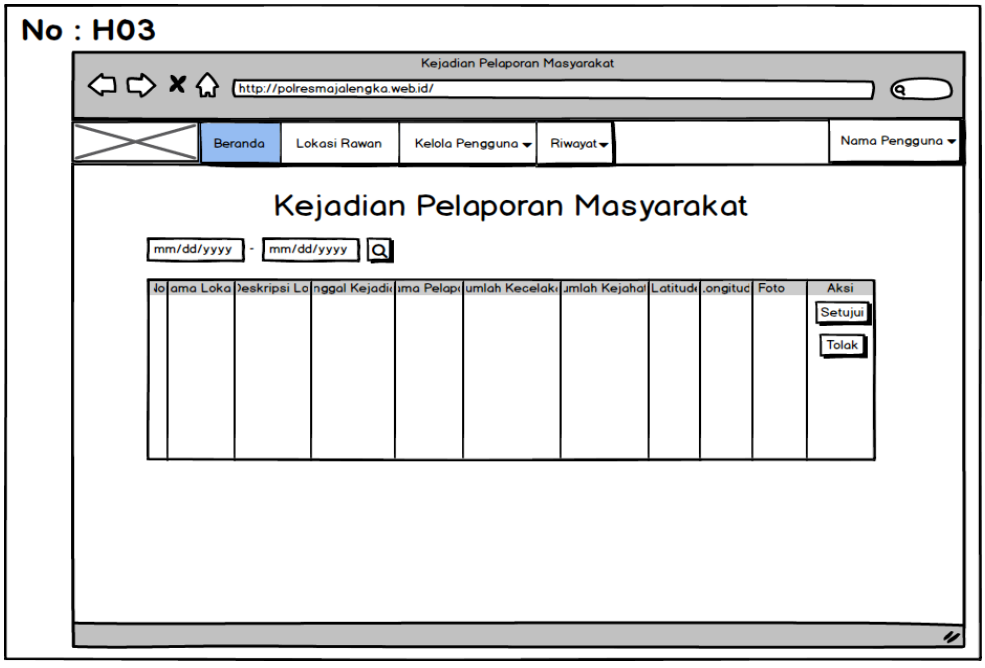
Gambar 3.48 Perancangan Halaman Masuk Admin

2. Perancangan antarmuka halaman lupa kata sandi (H02)



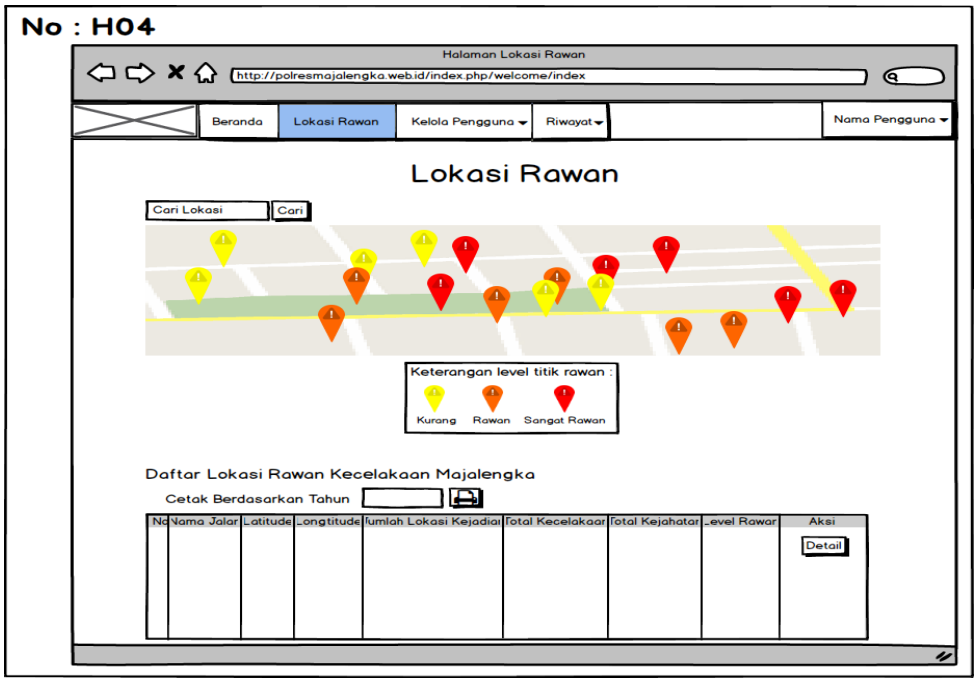
Gambar 3.49 Perancangan Antarmuka Halaman Lupa Kata Sandi

3. Perancangan antarmuka halaman awal Admin (H03)



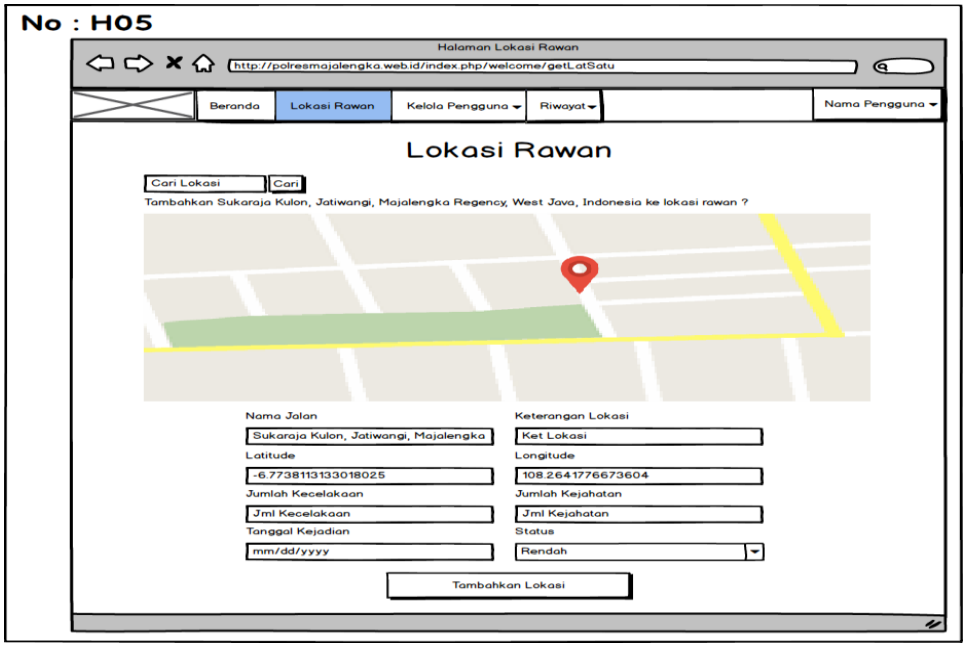
Gambar 3.50 Perancangan Halaman Awal Admin

4. Perancangan antarmuka halaman data lokasi rawan (H04)



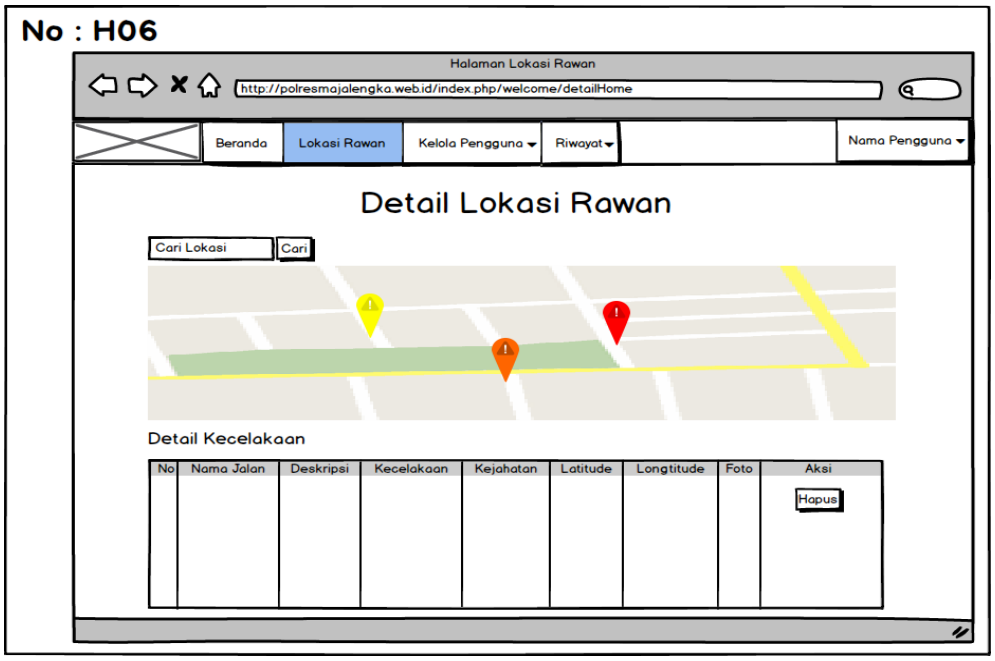
Gambar 3.51 Perancangan Halaman Data Lokasi Rawan

5. Perancangan antarmuka halaman tambah data lokasi rawan (H05)



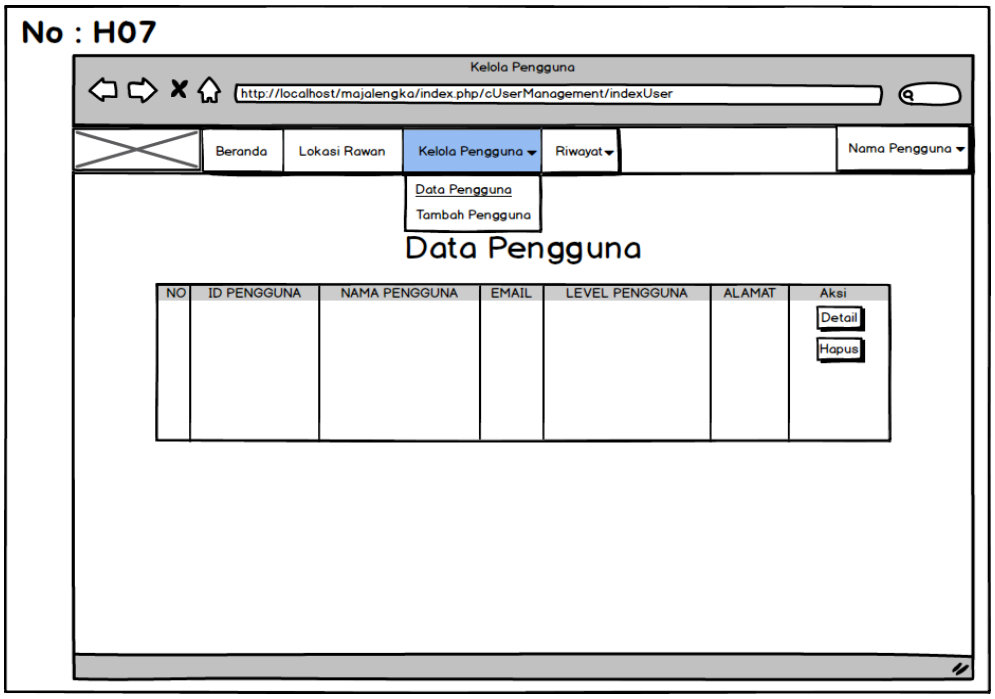
Gambar 3.52 Perancangan Halaman Tambah Data Lokasi Rawan

6. Perancangan antarmuka halaman detail lokasi rawan (H06)



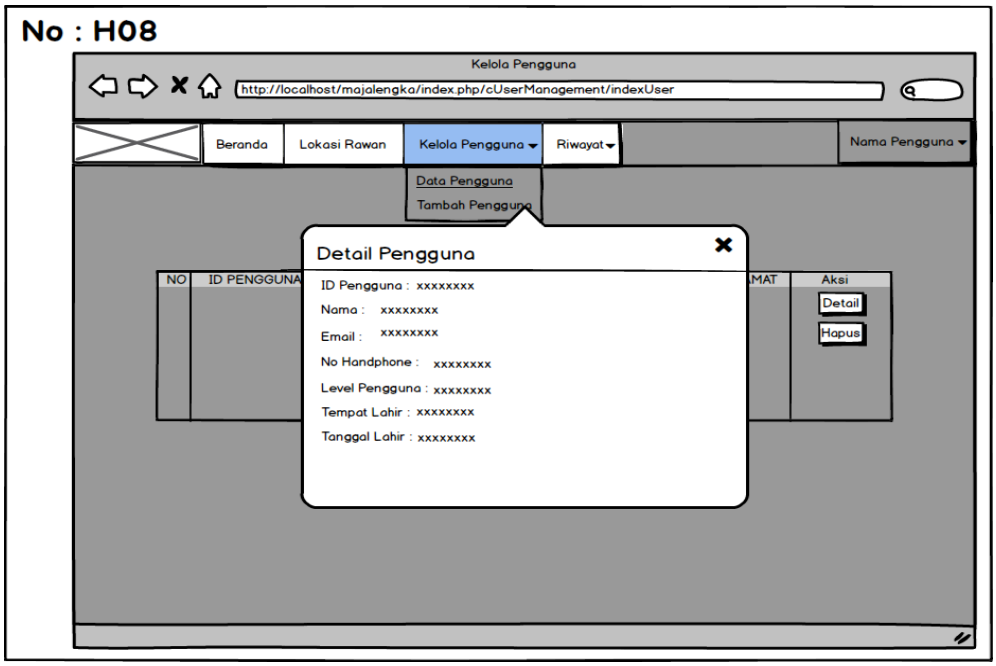
Gambar 3.53 Perancangan Halaman Detail Lokasi Rawan

7. Perancangan antarmuka halaman data pengguna (H07)



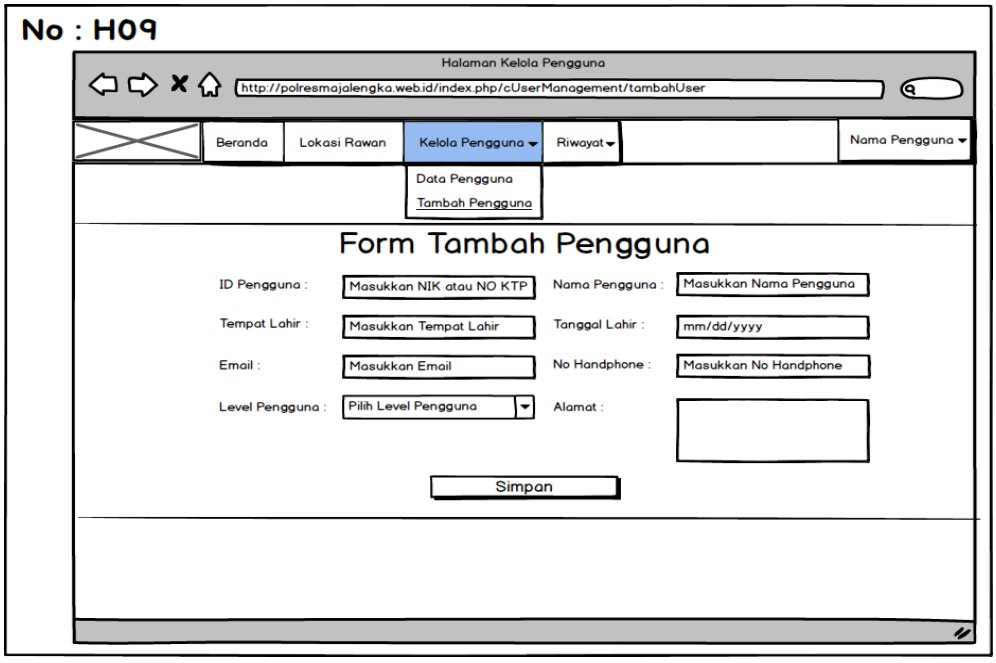
Gambar 3.54 Perancangan Halaman Data Pengguna

8. Perancangan antarmuka halaman popup detail pengguna (H08)



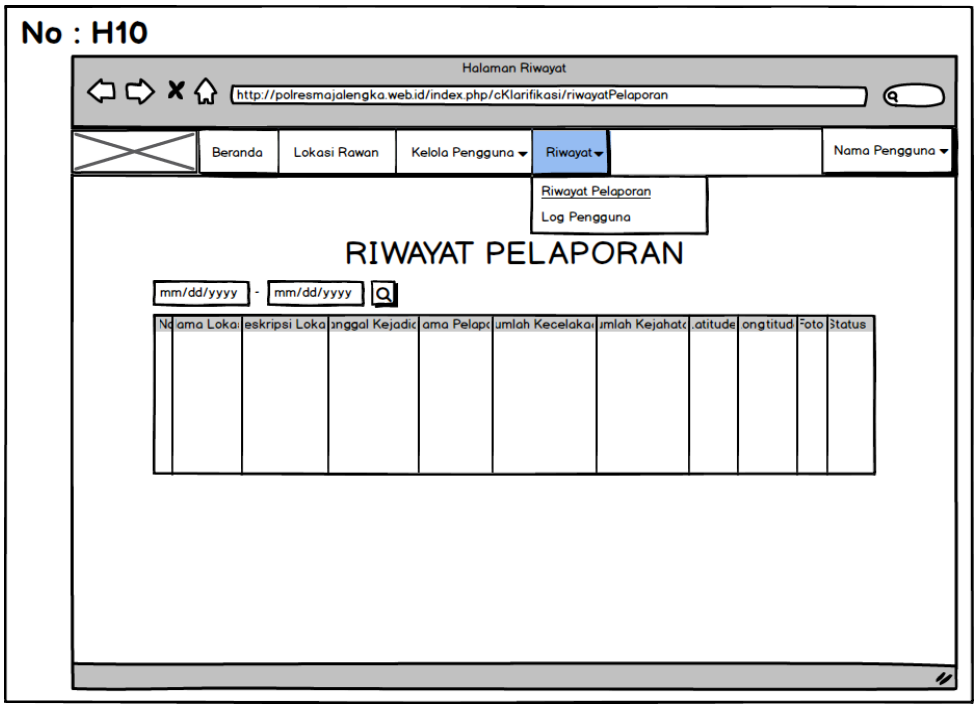
Gambar 3.55 Perancangan Halaman Detail Pengguna

9. Perancangan antarmuka halaman tambah data pengguna (H09)



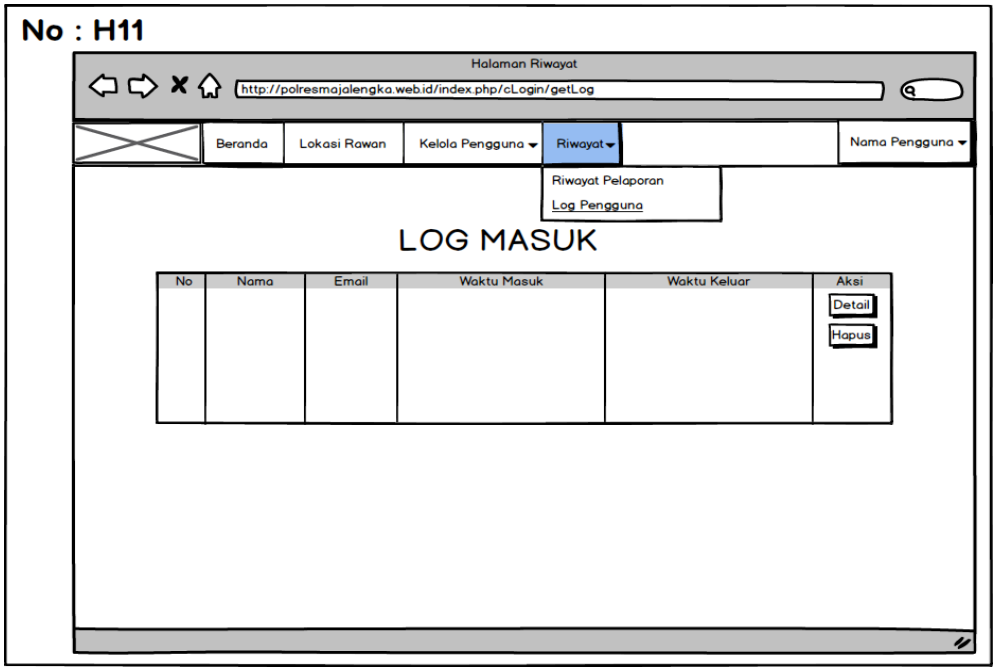
Gambar 3.56 Perancangan Halaman Tambah Data Pengguna

10. Perancangan antarmuka halaman riwayat pelaporan (H10)



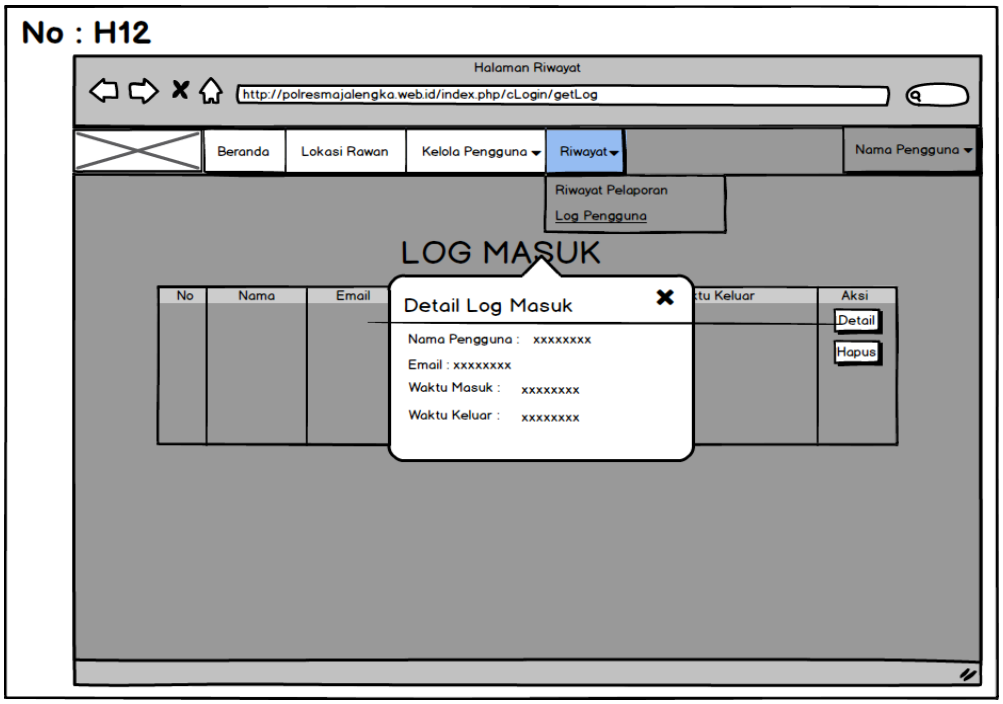
Gambar 3.57 Perancangan Halaman Riwayat Pelaporan

11. Perancangan antarmuka halaman log masuk (H11)



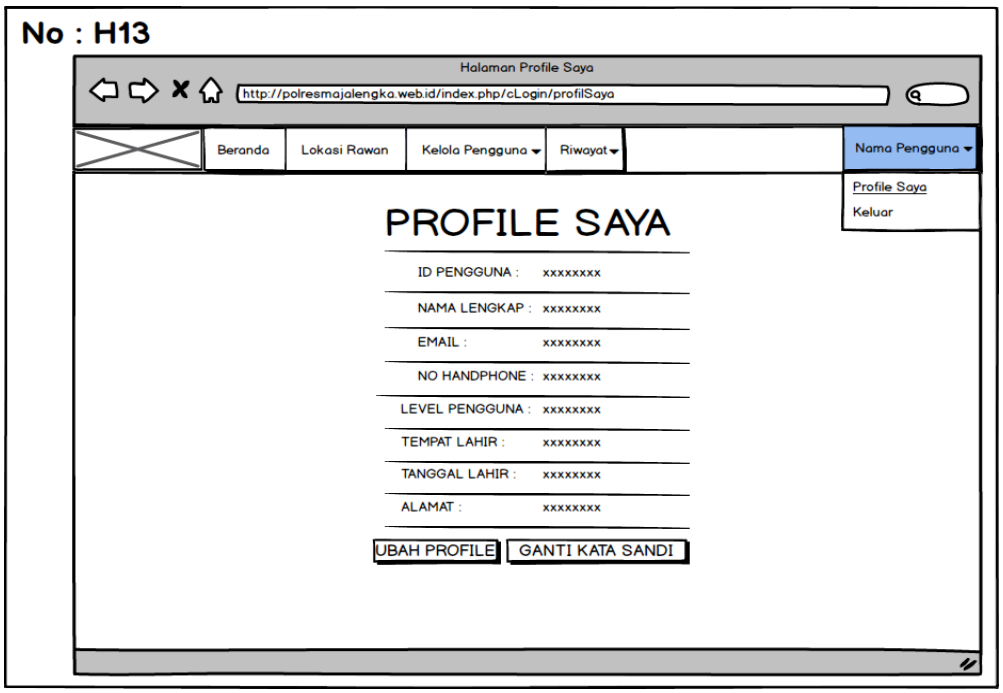
Gambar 3.58 Perancangan Halaman Log Masuk

12. Perancangan antarmuka halaman detail log masuk (H12)



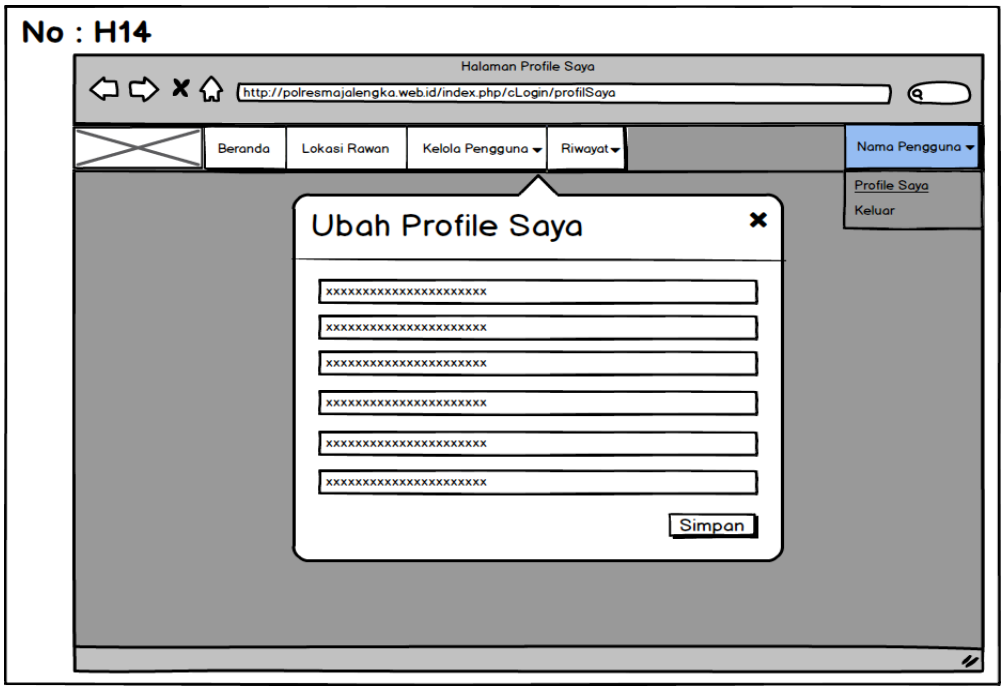
Gambar 3.59 Perancangan Halaman Detail Log

13. Perancangan antarmuka halaman profile saya (H13)



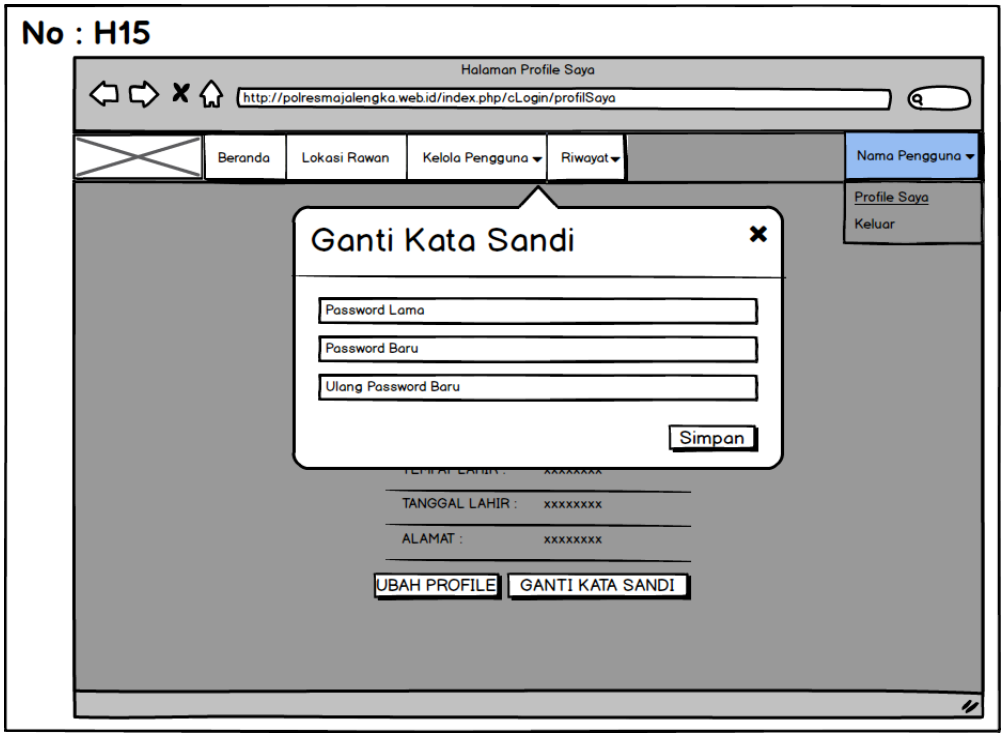
Gambar 3.60 Perancangan Halaman Profile Saya

14. Perancangan antarmuka halaman ubah profile (H14)



Gambar 3.61 Perancangan Halaman Ubah Profile

15. Perancangan antarmuka halaman ganti kata sandi (H15)

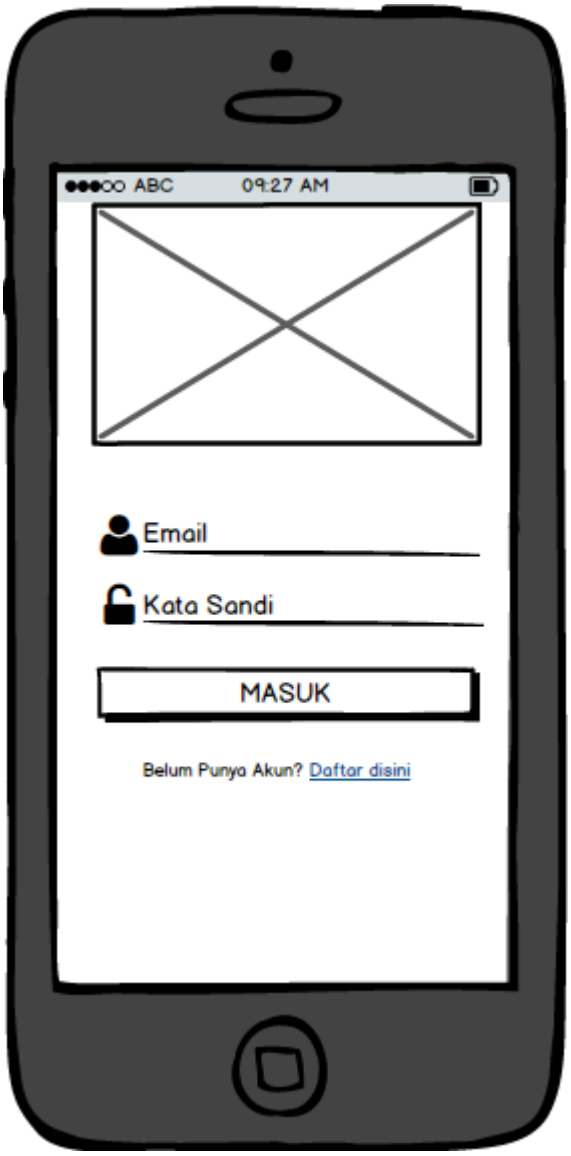


Gambar 3.62 Perancangan Halaman Ganti Kata Sandi

3.2.3.2 Perancangan Antarmuka Halaman Polisi

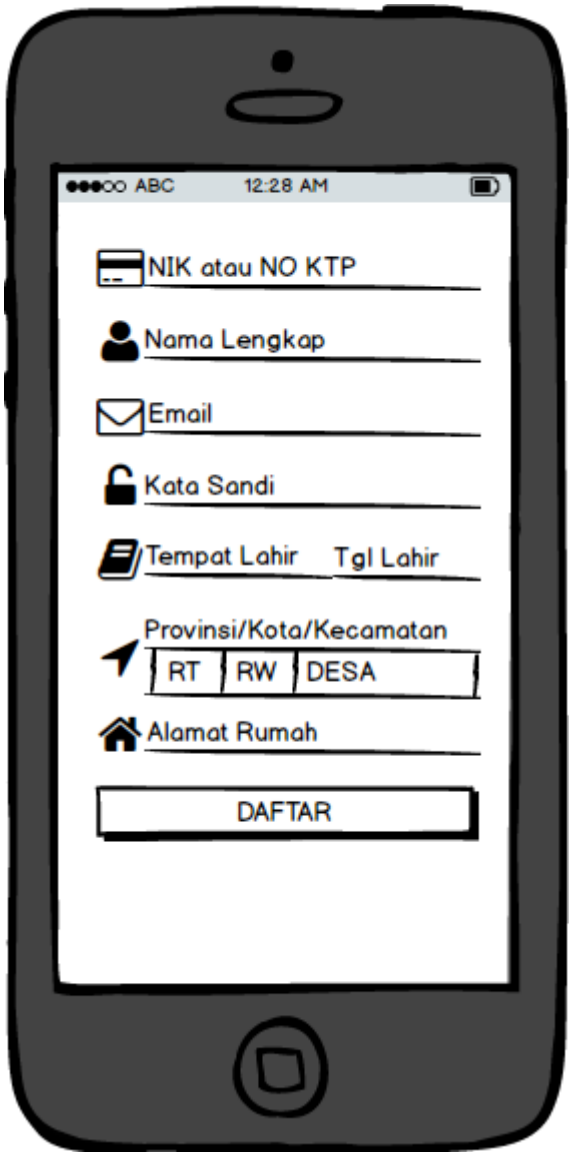
Berikut ini perancangan antarmuka pada *platform mobile* untuk polisi dari sistem yang akan dibangun.

1. Perancangan tampilan halaman login (H01)



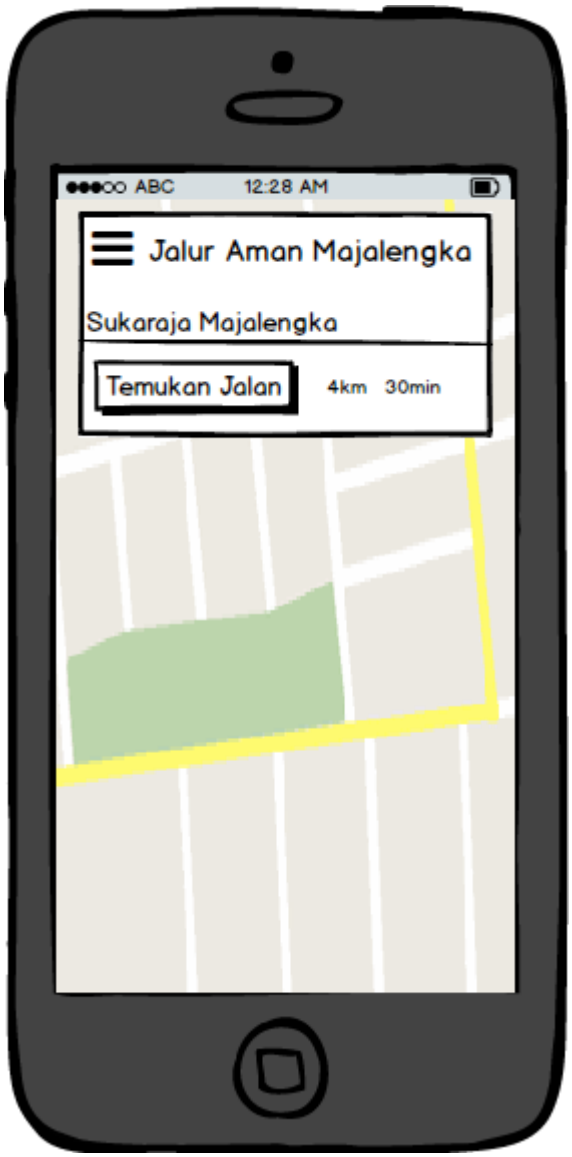
Gambar 3.63Perancangan Tampilan Halaman Login

2. Perancangan tampilan halaman daftar aplikasi (H02)



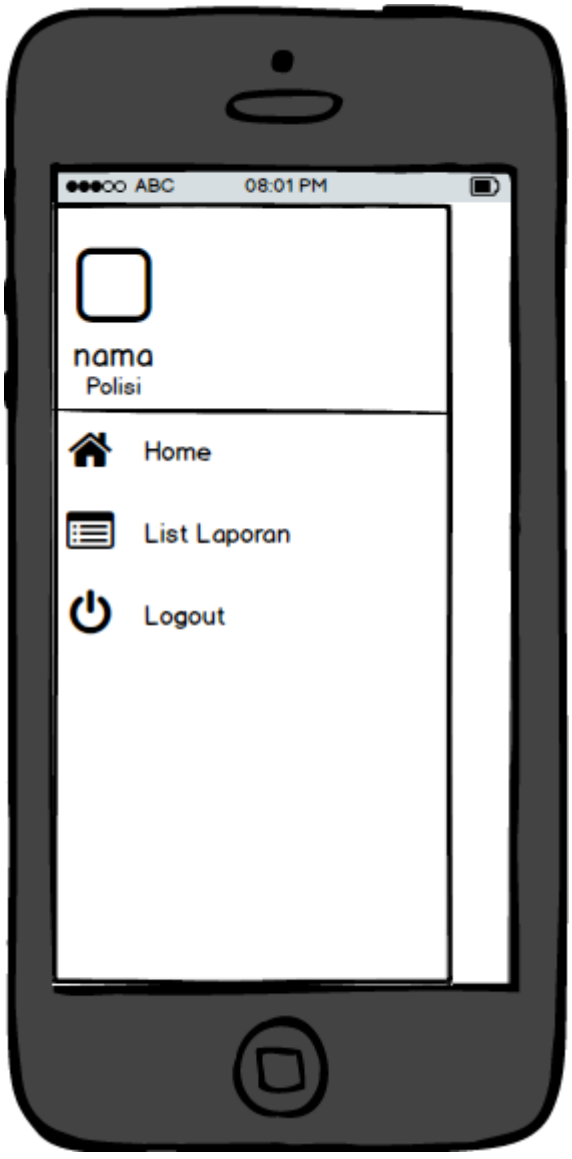
Gambar 3.64 Perancangan Tampilan Halaman Daftar

3. Perancangan tampilan halaman awal berhasil login (H03)



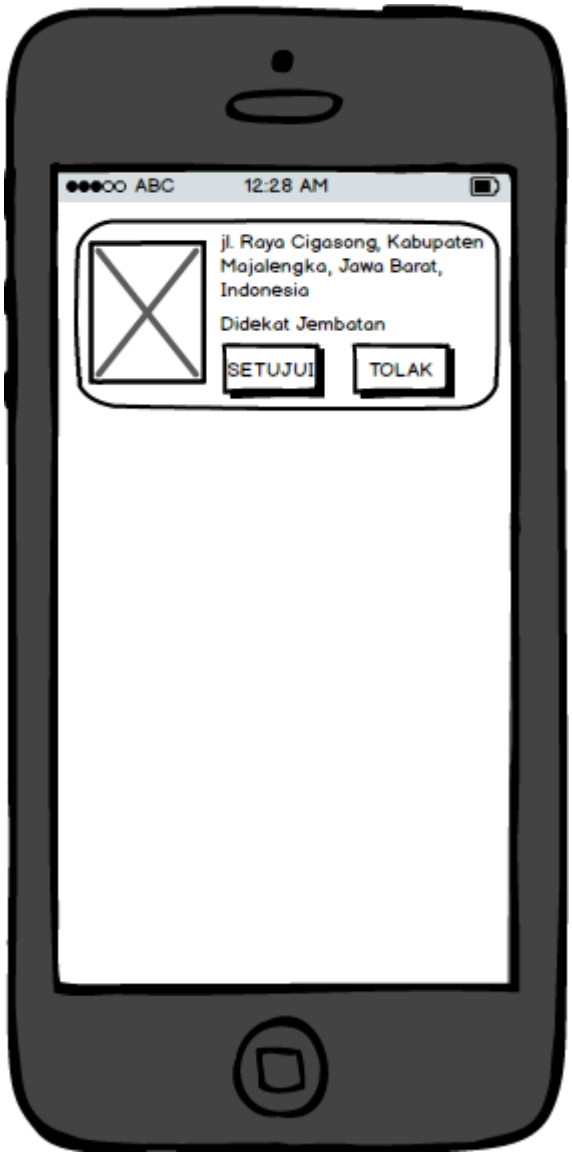
Gambar 3.65 Perancangan Tampilan Halaman Awal Berhasil Login

4. Perancangan tampilan daftar menu polisi (H04)



Gambar 3.66 Perancangan Tampilan Daftar Menu Polisi

5. Perancangan tampilan halaman daftar laporan masyarakat (H05)

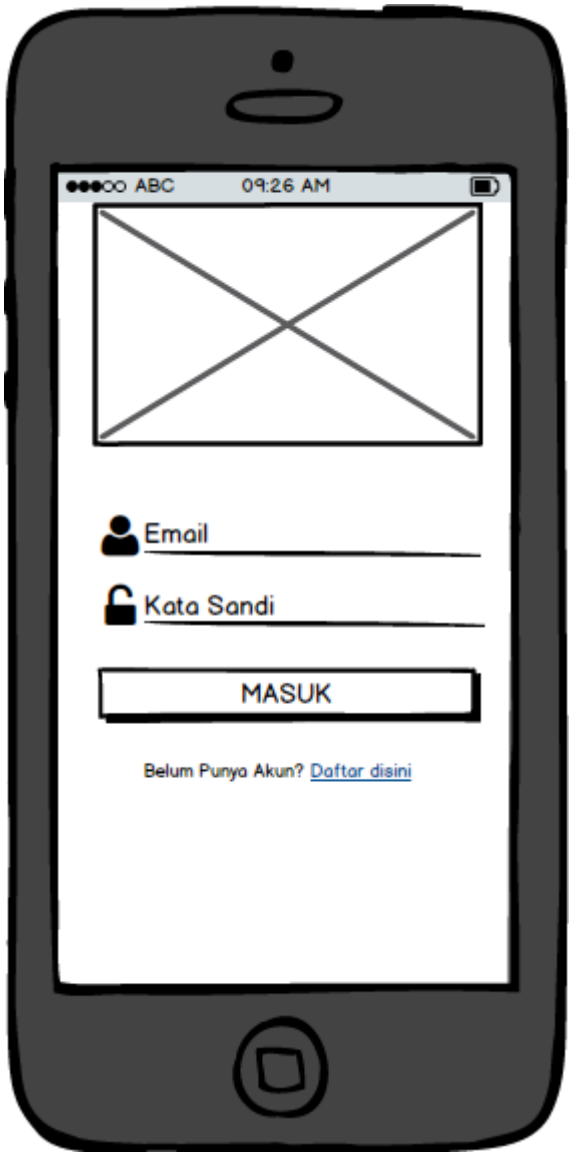


Gambar 3.67 Perancangan Tampilan Halaman Daftar Laporan Masyarakat

3.2.3.3 Perancangan Antarmuka Halaman Masyarakat

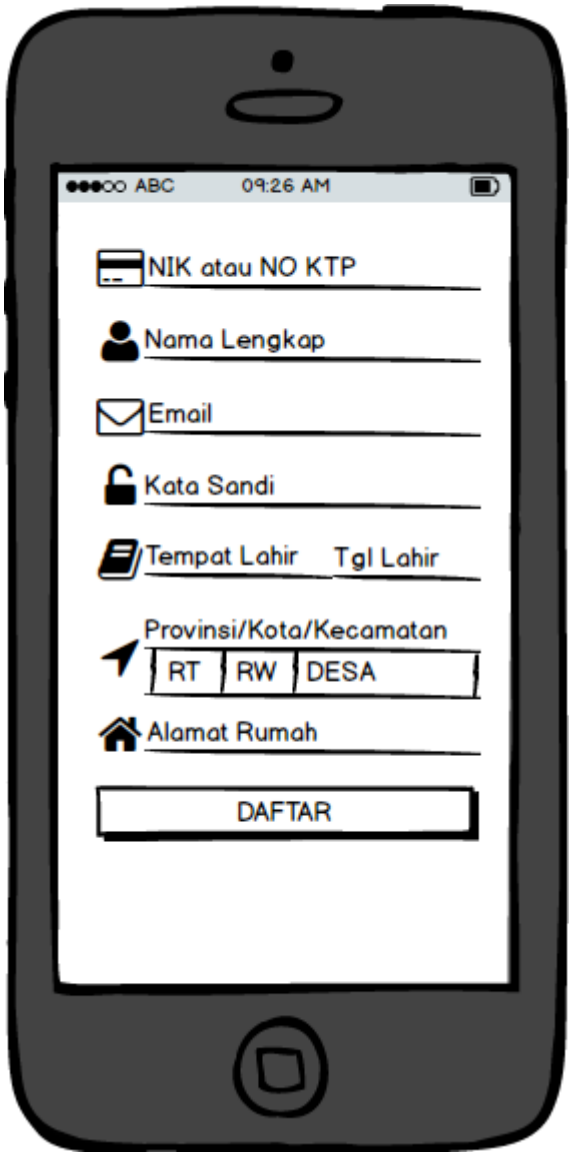
Berikut ini perancangan antarmuka pada *flatform mobile* untuk masyarakat dari sistem yang akan dibangun.

1. Perancangan tampilan halaman awal login aplikasi (H01)



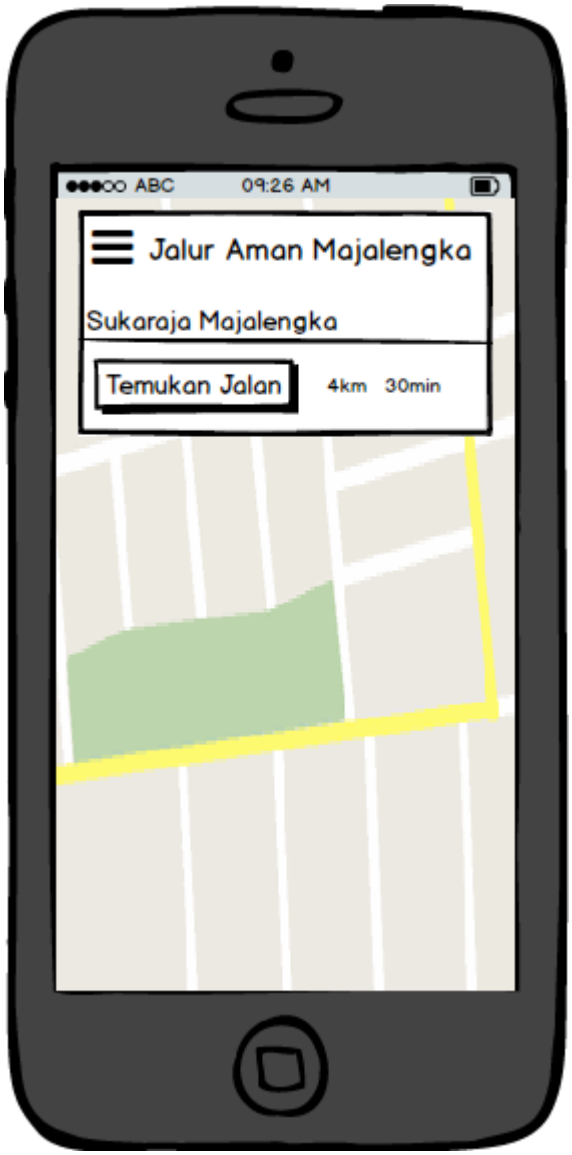
Gambar 3.68 Pearancangan Tampilan Halaman Login

2. Perancangan tampilan halaman daftar aplikasi (H02)



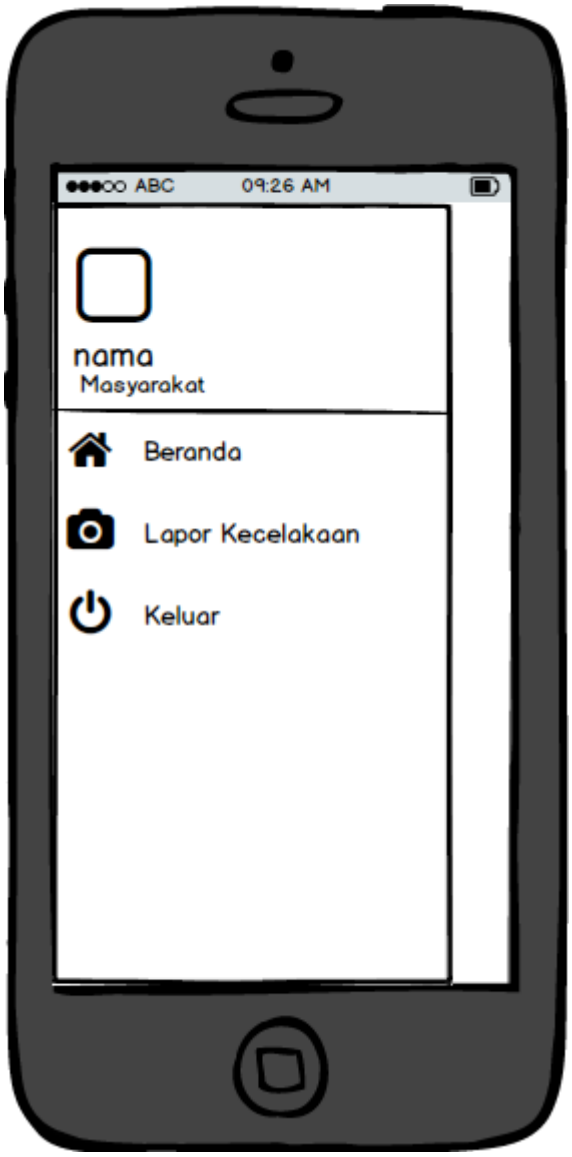
Gambar 3.69 Perancangan Tampilan Halaman Daftar

3. Perancangan tampilan halaman awal berhasil login pada aplikasi (H03)



Gambar 3.70 Perancangan Tampilan Awal Berhasil Login

4. Perancangan tampilan daftar menu masyarakat (H04)



Gambar 3.71 Perancangan Tampilan Daftar Menu Masyarakat

5. Perancangan tampilan halaman pelaporan kejadian kecelakaan dan kejahatan (H05)



Gambar 3.72 Perancangan Tampilan Halaman Pelaporan Kejadian

3.2.4 Perancangan Pesan Kesalahan

Perancangan pesan kesalahan pada sistem yang akan dibangun bertujuan untuk memberikan rancangan pesan penanganan kesalahan atas kesalahan yang mungkin dilakukan oleh pengguna. Adapun perancangan pesan keslahan dalam sistem yang akan dibangun adalah sebagai berikut.

1. Perancangan Pesan Kesalahan Web

   Berikut ini perancangan pesan kesalahan pada flatform web

Tabel 3.40 Perancangan Pesan Kesahalan Web

Kode Pesan	Pesan
M01	Masukkan Email dan Password yang Benar
M02	Masukkan Email Yang Valid
M03	Anda Setuju Menyetujui Lokasi ini? Klik Ok Untuk Iya, Jika Tidak Klik Cancel
M04	Anda Yakin Tidak Akan Menambahkan Lokasi ini? Klik Ok Untuk Iya, Jika Tidak Klik Cancel
M05	Berhasil Menambahkan Data Lokasi Rawan
M06	Gagal Menambah Data Lokasi Rawan
M07	Yakin Akan Menghapus "Nama Jalan". Klik Ok Untuk Iya, Jika Tidak Klik Cancel
M08	Yakin Akan Menghapus Pengguna "nama pengguna". Klik Ok Untuk Iya, Jika Tidak Klik Cancel
M09	Berhasil Menambah Pengguna
M10	Berhasil Menghapus Pengguna
M11	Anda yakin akan menghapus dari log no “nomor urut”. Klik Ok Untuk Iya, Jika Tidak Klik Cancel
M12	Berhasil Menghapus Log
M13	Sukses Ubah Profile
M14	Gagal Ubah Profile

Kode Pesan	Pesan
M15	Berhasil Mengganti Kata Sandi
M16	Gagal Mengganti Kata Sandi

2. Perancangan Pesan Kesalahan *Mobile*

   Berikut ini perancangan pesan kesalahan pada *platform mobile*

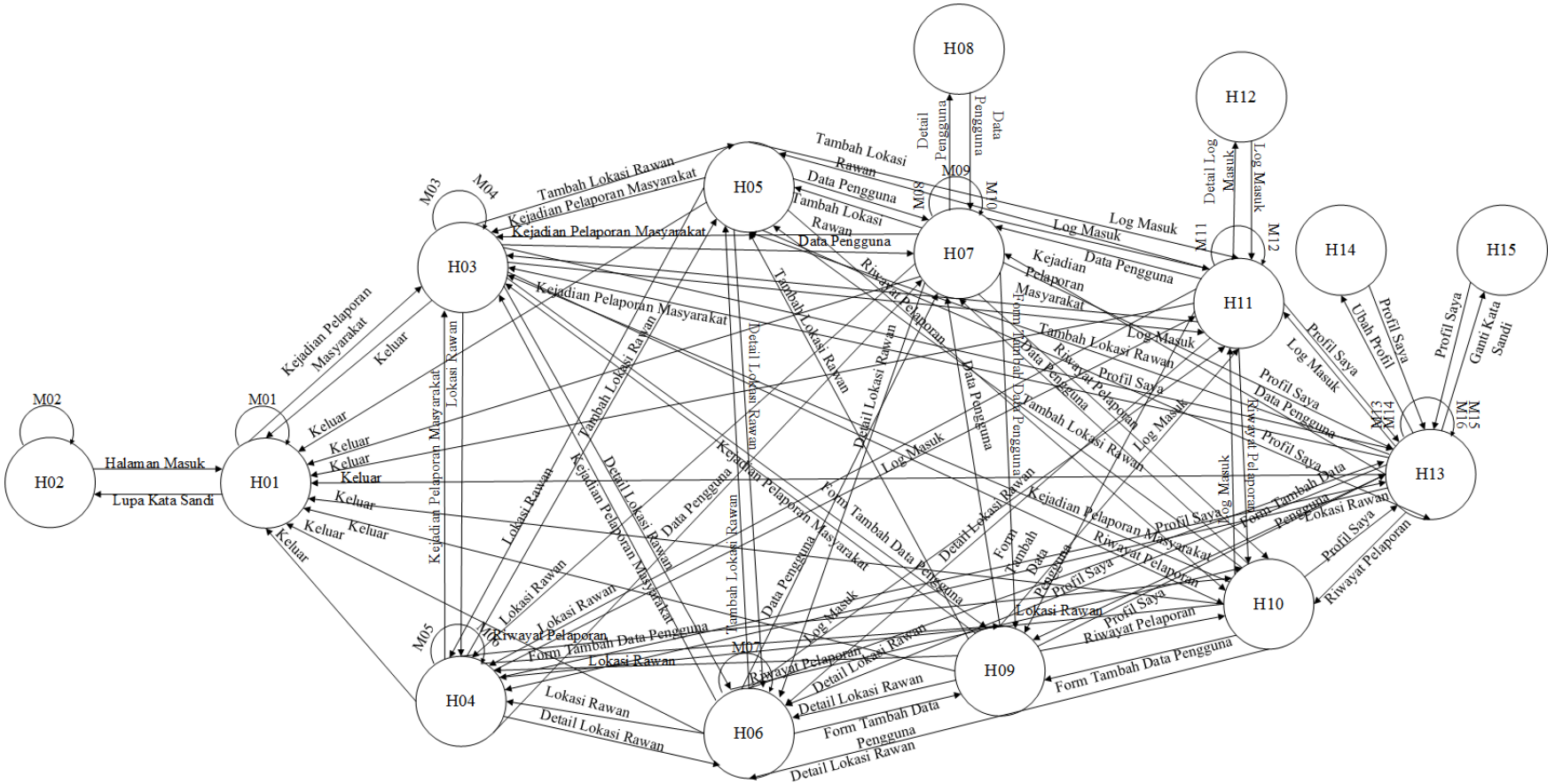
Tabel 3.41 Perancangan Pesan Kesalahan *Mobile*

Kode Pesan	Pesan
M17	Gagal Login!
M18	Laporan di setujui!
M19	Laporan di tolak!
M20	Apakah anda ingin keluar dari aplikasi ini? Pilih Iya atau Pilih Tidak.

3.2.5 Jaringan Semantik

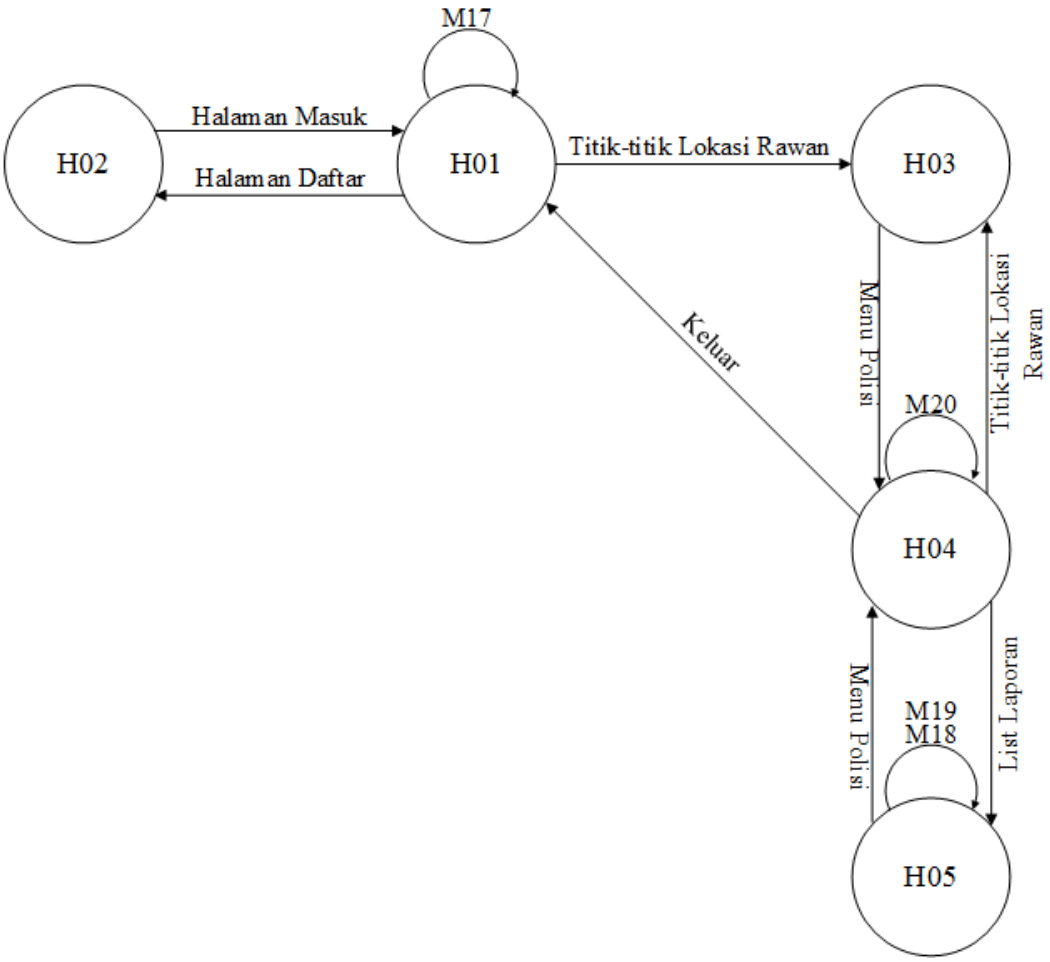
Pada penelitian ini, jaringan semantik dibuat untuk menggambarkan hubungan antar objek atau hubungan antar menu. Adapun jaringan semantik pada aplikasi yang akan dibangun adalah sebagai berikut.

1. Jaringan Semantik Web Admin



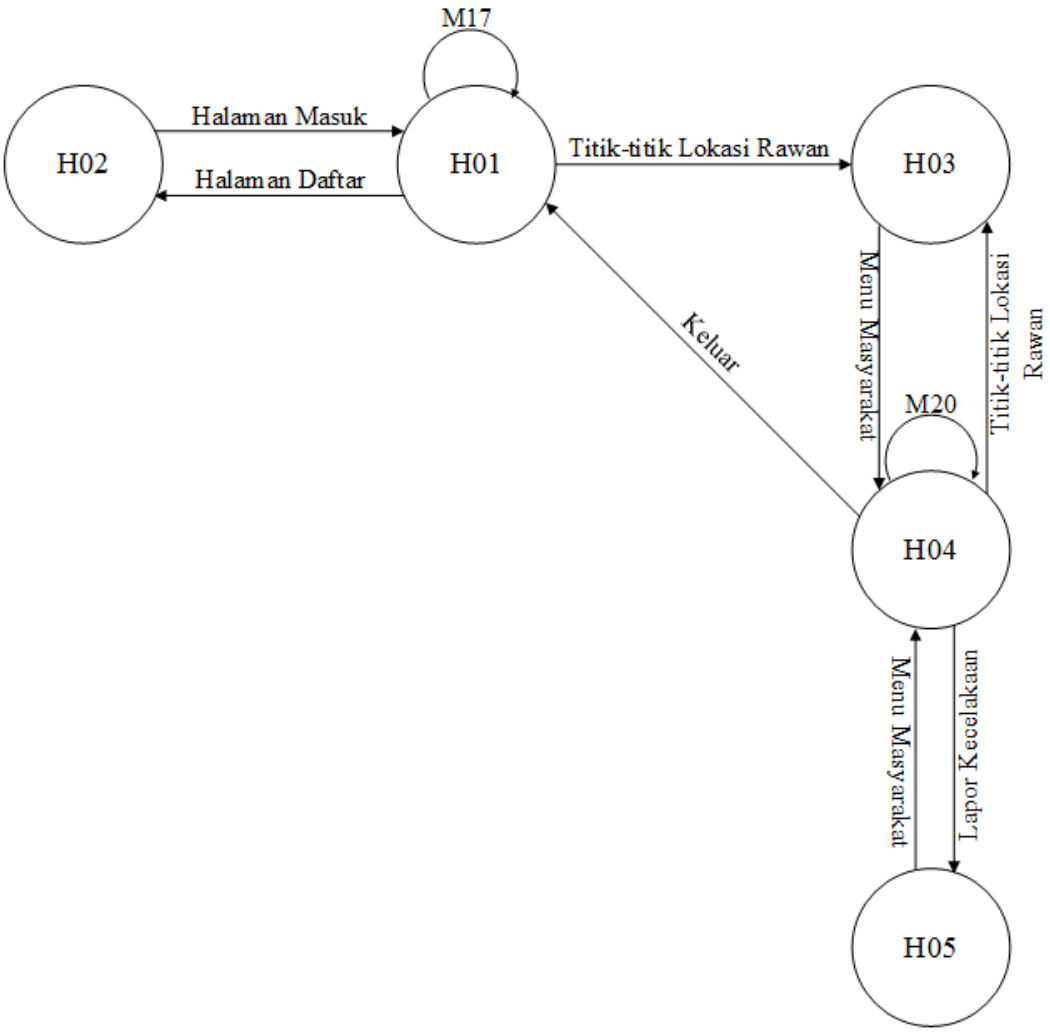
Gambar 3.73 Jaringan Semantik Web

2. Jaringan Semantik *Mobile* Polisi



Gambar 3.74 Jaringan Semantik *Mobile* Polisi

3. Jaringan Semantik *Mobile* Masyarakat



Gambar 3.75 Jaringan Semantik *Mobile* Masyarakat