BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini akan mendeskripsikan tahap dari implementasi dan pengujian terhadap sistem yang akan dibangun. Sistem yang dibangun ini diimplementasikan sesuai dengan analisis dan perancangan ditahap sebelumnya. Selanjutnya, sistem akan dilakukan pengujian. Pada penelitian ini metode pengujian terhadap sistem yang akan dibangun menggunakan metode *black box.*

4.1 Implementasi Sistem

Implementasi sistem merupakan tahap untuk menerapkan hasil dari perancangan yang telah dilakukan terhadap sistem yang akan dibangun. Sehingga sistem tersebut dapat digunakan. Implementasi dalam penelitian ini mencakup implementasi perangkat keras, implementasi perangkat lunak, implementasi basis data serta implementasi antarmuka.

4.1.1 Implementasi Perangkat Keras

Untuk dapat menjalankan sistem yang dibangun, membutuhkan perangkat keras yang dapat mendukung berjalannya sistem. Adapaun perangkat keras yang digunakan untuk menggunakan sistem terbagi menjadi dua, yaitu pada *platform* web dan *platform mobile*.

1. Perangkat keras pada *platform* web

   Berikut ini perangkat keras yang digunakan untuk menjalankan sistem pada *platform* web.

Tabel 4.1 Implementasi Perangkat Keras Pada Web

Nama Perangkat Keras	Keterangan
Prosesor	Intel Dual Core
RAM	2 GB
Hardisk	500 GB

2. Perangkat keras pada *platform mobile*

Berikut ini perangkat keras yang digunakan untuk menjalankan sistem pada *platform mobile*

Tabel 4.2 Implementasi Perangkat Keras Pada *Mobile*

Nama Perangkat Keras	Keterangan
Prosesor	Quad-core 1.8 GHz
Memory	2 GB RAM, 16GB ROM

4.1.2 Implementasi Perangkat Lunak

Selain membutuhkan perangkat keras, untuk dapat menjalankan sistem yang dibangun juga membutuhkan perangkat lunak. Dalam penelitian ini, implementasi perangkat lunak untuk menjalankan sistem terbagi menjadi dua, yaitu perangkat lunak untuk menjalankan sistem pada web serta perangkat lunak untuk menjalankan sistem pada *mobile*. Adapun spesifikasinya adalah sebagai berikut.

1. Perangkat lunak pada *platform* web

   Berikut ini perangkat lunak yang digunakan untuk menggunakan sistem pada *platform* web.

Tabel 4.3 Implementasi Perangkat Lunak Pada Web

Nama Perangkat Lunak	Keterangan
Sistem Operasi	Windows 7
Web Browser	Mozila Firefox, Google Chrome

2. Perangkat lunak pada *platform mobile*

   Berikut ini perangkat lunak yang digunakan untuk menggunakan sistem pada *platform mobile*.

Tabel 4.4 Implementasi Perangkat Lunak pada *Mobile*

Nama Perangkat Lunak	Keterangan
Sistem Operasi	Android versi 5.0 (Lollipop)

4.1.3 Implementasi Teknologi Yang Digunakan

Sistem yang dibangun dalam penelitian ini menggunakan beberapa teknologi. Adapaun implementasi dari teknologi yang digunakan adalah sebagai berikut.

4.1.3.1 Implementasi GPS

Teknologi GPS ini digunakan untuk mengetahui keberadaan atau posisi dari pengguna. Adapun implementasi dari teknologi GPS dalam sistem yang dibangun adalah sebagai berikut.

Tabel 4.5 Implementasi GPS

```
private void getDeviceLocation() {
fusedLocationProviderClient =
LocationServices.getFusedLocationProviderClient(this);
try {
if (mLocationPermissionsGranted) {
com.google.android.gms.tasks.Task location =
fusedLocationProviderClient.getLastLocation();
location.addOnCompleteListener(new OnCompleteListener() {
@Override
public void onComplete(@NonNull
com.google.android.gms.tasks.Task task) {
if (task.isSuccessful()) {
Location currentLocation = (Location) task.getResult();
origin=getCompleteAddressString(currentLocation.getLatitude(),cu
rrentLocation.getLongitude());
}
}
});
}
} catch (SecurityException s) {
}
}
```

Hasil dari *source code* tersebut adalah sebagai berikut.



Gambar 4.1 Hasil Implementasi GPS

4.1.3.2 Implementasi *Google Maps* API

Google Maps API ini digunakan untuk menampilkan peta digital pada aplikasi yang dibangun. Adapun implementasi dari *google maps* API ini adalah sebagai berikut.

Tabel 4.6 Implementasi *Google Maps* API

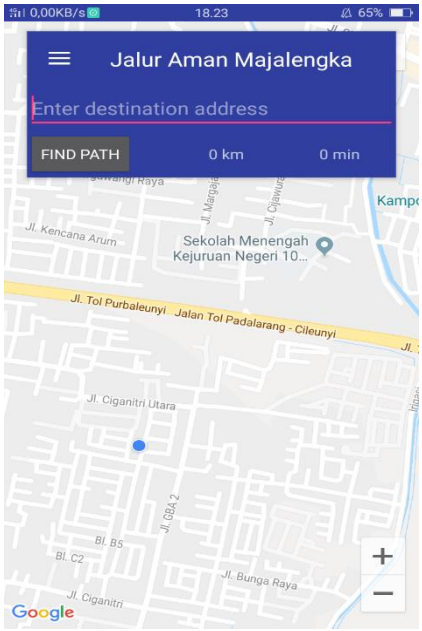
```
<resources>
<string name="google_maps_key" templateMergeStrategy="preserve"
translatable="false">AIzaSyDx07BAdTwfEPqubojWjbN1YGnS-
u10y_4</string>
</resources>
```

Layout Penggunaan *Google Maps* API

Tabel 4.7 Impelementasi Layout Penggunaan *Google Maps* API

```
<fragment
android:id="@+id/map"
android:name="com.google.android.gms.maps.SupportMapFragment"
android:layout_width="match_parent"
android:layout_height="match_parent" />
```

Dari *source code* tersebut maka menghasilkan tampilan seperti dibawah ini.



Gambar 4.2 Hasil Implementasi *Google Maps* API

4.1.3.3 Impelemntasi *Geocoding*

Teknologi *geocoding* ini digunakan untuk mengkonversi *latitude* dan *longtitude* menjadi sebuah alamat yang dapat dengan mudah dibaca oleh pengguna. Impelemntasi dari teknologi *geocoding* dalam penelitian ini adalah sebagai berikut.

Tabel 4.8 Implementasi *Geocoding*

```
private String getCompleteAddressString(double LATITUDE, double
LONGITUDE) {
String strAdd = "";
Geocoder geocoder = new Geocoder(this, Locale.getDefault());
try {
List<android.location.Address> addresses =
geocoder.getFromLocation(LATITUDE, LONGITUDE, 1);
if (addresses != null) {
android.location.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;
}
```

4.1.3.4 Implementasi *Geofencing*

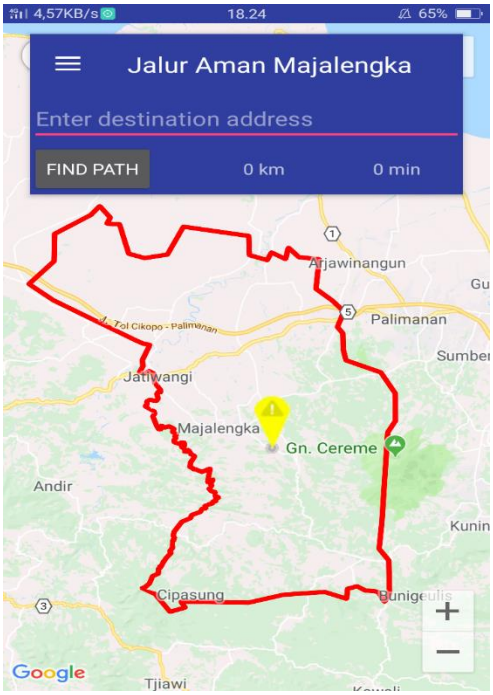
Selain *geocoding*, sistem yang dibangun dalam penelitian ini juga menggunakan teknologi *geofencing*. Teknologi ini digunakan untuk membatasi wilayah daerah rawan dan juga membatasi daerah Majalengka. Berikut ini impelementasi dari teknologi *geofencing*.

1. Implementasi *geofencing* untuk membatasi wilayah Majalengka
   Adapun implementasi *geofencing* untuk membatasi wilayah Majalengka adalah sebagai berikut.

Tabel 4.9 Implementasi *Geofencing* Batas Majalengka

```
private void loadMajalengkaGeofence() {
Retrofit retrofit = new
Retrofit.Builder().baseUrl(Config.URL_SERVICE).addConverterFacto
ry(GsonConverterFactory.create()).build();
ApiCommunication api = retrofit.create(ApiCommunication.class);
Call<ValueMajalengka> call = api.getMajalengkaBorderLine();
call.enqueue(new Callback<ValueMajalengka>() {
@Override
public void onResponse(Call<ValueMajalengka> call,
Response<ValueMajalengka> response) {
String value = response.body().getStatus();
if (value.equals("success")) {
resMajalengka = response.body().getData();
for (int i = 0; i < resMajalengka.size(); i++) {
Double lat = resMajalengka.get(i).getLat();
Double lng = resMajalengka.get(i).getLng();
mMarkers.add(mMap.addMarker(new MarkerOptions().position(new
LatLng(lat,lng))));
                    }
update();
}
}
@Override
public void onFailure(Call<ValueMajalengka> call, Throwable t) {
System.out.println("VALUENYA GAGAL" + t.getMessage());
}
});
}
```

Dari *source code* di atas, maka menghasilkan tampilan seperti dibawah ini.



Gambar 4.3 Hasil Implementasi *Geofencing* Wilayah Majalengka

2. Implementasi *geofencing* untuk membatasi titik rawan

   Adapun implementasi *geofencing* untuk membatasi daerah rawan adalah sebagai berikut.

Tabel 4.10 Implementasi *Geofencing* Batas Titik Rawan

```
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;
```

Dari *source code* di atas, maka menghasilkan tampilan seperti di bawah ini.



Gambar 4.4 Hasil Impelemntasi *Geofencing* Titik Rawan

4.1.3.5 Impelemntasi *Geotagging*

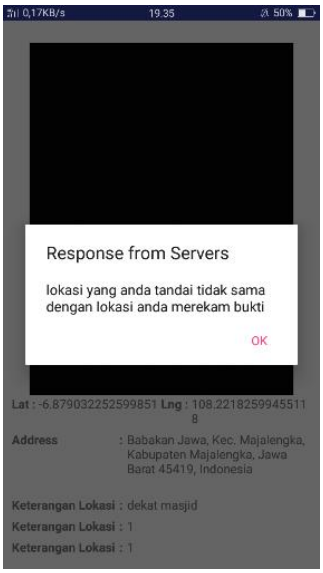
Teknologi *geotagging* ini digunakan pada saat masyarakat melakukan pelaporan melalui aplikasi. Aplikasi ini akan meminta pengguna untuk mengirimkan foto kejadian. Teknologi *geotagging* ini digunakan untuk memvalidasi foto yang dikirimkan oleh masyarakat. Dengan begitu masyarakat yang mengirimkan foto kejadian tidak sesuai dengan posisi kejadian tersebut tidak bisa mengirimkan laporan tersebut. adapun implementasi *geotagging* ini adalah sebagai berikut.

Tabel 4.11 Impelemntasi *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;
}
```

Dari *source code* di atas, maka setiap foto yang akan di kirimkan oleh masyarakat akan di validasi terlebih dahulu. Jika posisi masyarakat yang melaporkan tidak sesuai dengan foto yang diambil maka masyarakat tidak dapat melakukan pelaporan. Seperti pada gambar dibawah ini.



Gambar 4.5 Hasil Implementasi *Geotagging*

4.1.4 Implementasi Perhitungan Level Rawan

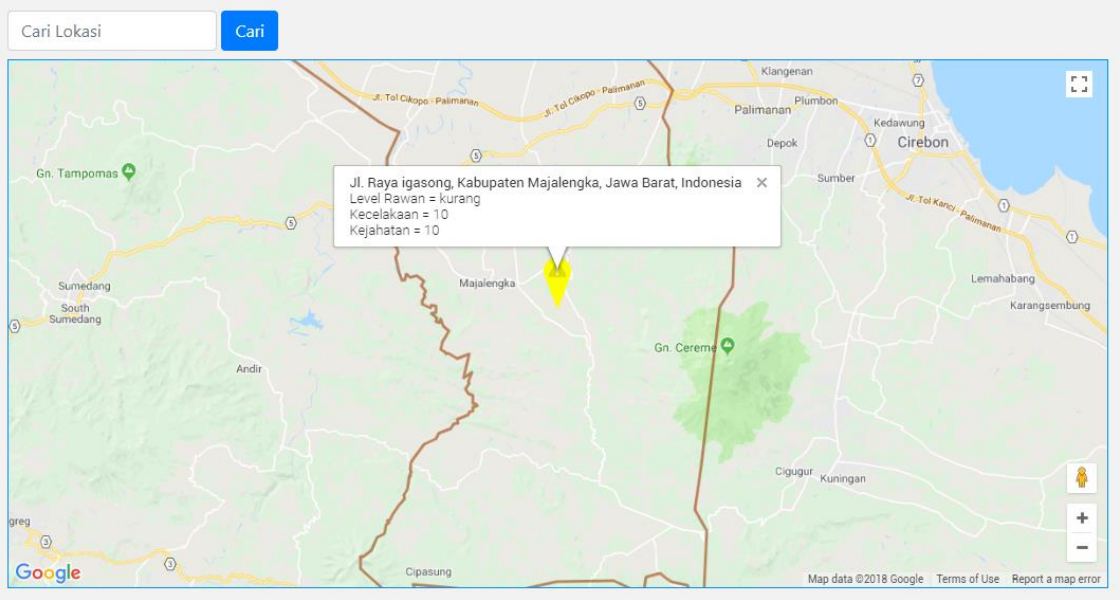
Dalam penelitian ini titik rawan terbagi menjadi tiga level, yaitu level rendah, sedang dan tinggi. Adapun implementasi dari perhitungan titik rawan tersebut adalah sebagai berikut.

Tabel 4.12 Implementasi Level Rawan

```
if (($dataLatbanding<=$dataMasterLat4DigitTerakhir &&
$dataLngbanding<=$dataMasterLng4DigitTerakhir)
                                ||
($dataLatbanding<=$dataMasterLat4DigitTerakhir &&
$dataLngbanding>=$dataMasterLng4DigitTerakhir)
```

```
                                    ||
($dataLatbanding>=$dataMasterLat4DigitTerakhir &&
$dataLngbanding<=$dataMasterLng4DigitTerakhir)
                                    ||
($dataLatbanding>=$dataMasterLat4DigitTerakhir &&
$dataLngbanding>=$dataMasterLng4DigitTerakhir)
                                    ) {
if ($data2[$i]['id_lokasi']==$data[0]['id_lokasi']) {
array_push($dataLokasiSama, $data2[$i]['id_rawan']);
$tglTerkhirKejadian=$data2[$jmlData2K1]['tanggal_kejadian'];
// echo $tglTerkhirKejadian."<br>";
date_default_timezone_set('Asia/Jakarta');
$waktuSekarang=$data[0]['tanggal_kejadian'];
// echo '<br>Indonesian Timezone: ' . $waktuSekarang;
$tanggal1 = new DateTime($tglTerkhirKejadian);
$tanggal2 = new DateTime($waktuSekarang);
$perbedaan = $tanggal2->diff($tanggal1)->format("%a");
// echo "  perbedaan waktu ke $i adalah =".$perbedaan;
if ($perbedaan<=14) {
$statusRawan='sangat rawan';
$this->apiModel-
>updateRawanApprove($data2[$i]['id_rawan'],$statusRawan);
$this->apiModel->updateRawanApprove($id_rawan,$statusRawan);
}elseif ($perbedaan>14 && $perbedaan<=31) {
$statusRawan='rawan';
$this->apiModel-
>updateRawanApprove($data2[$i]['id_rawan'],$statusRawan);
$this->apiModel->updateRawanApprove($id_rawan,$statusRawan);
}elseif ($perbedaan>31 && $perbedaan<=60) {
$statusRawan='kurang';
$this->apiModel-
>updateRawanApprove($data2[$i]['id_rawan'],$statusRawan);
$this->apiModel->updateRawanApprove($id_rawan,$statusRawan);
}
}
else{
$statusRawan='kurang';
$this->apiModel->updateRawanApprove($id_rawan,$statusRawan);
}
}
```

Hasil dari *source code* diatas adalah sebagai berikut.



Gambar 4.6 Hasil Implementasi Level Rawan

4.1.5 Implementasi Basis Data

Implementasi basis data ini merupakan implementasi dari basis data yang telah di rancang sebelumnya. Dalam penelitian ini basis data atau *database* diterapkan dengan menggunakan Mysql. Adapun sintaks dari pembuatan *database* pencarian jalur aman ini adalah sebagai berikut.

Tabel 4.13 Implementasi Basis Data

No	Nama Tabel	Perintah SQL
1.	lokasi	`CREATE TABLE IF NOT EXISTS `lokasi` ( `id_lokasi` int(3) NOT NULL AUTO_INCREMENT, `nama_jalan` varchar(150) CHARACTER SET latin1 DEFAULT NULL, `lat` float DEFAULT NULL, `lng` float DEFAULT NULL, PRIMARY KEY (`id_lokasi`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=111 ;`
2.	rawan	`CREATE TABLE IF NOT EXISTS `rawan` ( `id_rawan` int(11) NOT NULL AUTO_INCREMENT,`

No	Nama Tabel	Perintah SQL

		`id_lokasi` int(3) NOT NULL,
		`id_user` varchar(16) NOT NULL,
		`desc_lokasi` varchar(50) NOT NULL,
		`kecelakaan` int(11) NOT NULL,
		`kejahatan` int(4) NOT NULL,
		`level_rawan`
		enum('aman','rawan','kurang','sangat rawan')
		CHARACTER SET latin1 NOT NULL,
		`tanggal_kejadian` date NOT NULL,
		`foto` varchar(50) DEFAULT NULL,
		`lat` float NOT NULL,
		`lng` float NOT NULL,
		`status`
		enum('approve','decline','process') NOT NULL,
		PRIMARY KEY (`id_rawan`),
		KEY `id_lokasi` (`id_lokasi`),
		KEY `id_user` (`id_user`)
		ADD CONSTRAINT `rawan_ibfk_1` FOREIGN KEY (`id_lokasi`) REFERENCES `lokasi` (`id_lokasi`) ON DELETE CASCADE ON UPDATE CASCADE;
		) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=105 ;

3.	user	CREATE TABLE IF NOT EXISTS `user` (`id_user` varchar(16) CHARACTER SET latin1 NOT NULL, `password` varchar(100) NOT NULL, `nama_user` varchar(35) CHARACTER SET latin1 NOT NULL, `email` varchar(100) NOT NULL, `no_hp` varchar(16) NOT NULL, `level_user` enum('admin','polisi','masyarakat') NOT NULL,

No	Nama Tabel	Perintah SQL
		`tempat_lahir` varchar(35) CHARACTER SET latin1 NOT NULL, `tanggal_lahir` date NOT NULL, `id_provinsi` char(2) CHARACTER SET latin1 DEFAULT NULL, `id_kota` char(4) CHARACTER SET latin1 DEFAULT NULL, `id_kecamatan` char(7) CHARACTER SET latin1 DEFAULT NULL, `desa` varchar(35) CHARACTER SET latin1 DEFAULT NULL, `rw` int(4) DEFAULT NULL, `rt` int(4) DEFAULT NULL, `alamat` varchar(70) NOT NULL, PRIMARY KEY (`id_user`,`email`)) ENGINE=InnoDB DEFAULT CHARSET=utf8;
4.	provinsi	CREATE TABLE IF NOT EXISTS `provinsi` (`id_provinsi` char(2) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL, `nama_provinsi` varchar(255) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL, PRIMARY KEY (`id_provinsi`)) ENGINE=InnoDB DEFAULT CHARSET=utf8;
5.	kota	CREATE TABLE IF NOT EXISTS `kota` (`id_kota` char(4) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL, `id_provinsi` char(2) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL, `nama_kota` varchar(255) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL, PRIMARY KEY (`id_kota`), KEY `regencies_province_id_index` (`id_provinsi`) ADD CONSTRAINT `kot_pro` FOREIGN KEY (`id_provinsi`) REFERENCES `provinsi`

No	Nama Tabel	Perintah SQL
		(`id_provinsi`) ON DELETE CASCADE ON UPDATE CASCADE;) ENGINE=InnoDB DEFAULT CHARSET=utf8;
6.	kecamatan	CREATE TABLE IF NOT EXISTS `kecamatan` (`id_kecamatan` char(7) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL, `id_kota` char(4) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL, `nama_kecamatan` varchar(255) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL, PRIMARY KEY (`id_kecamatan`), KEY `districts_id_index` (`id_kota`) ADD CONSTRAINT `kec_kot` FOREIGN KEY (`id_kota`) REFERENCES `kota` (`id_kota`) ON DELETE CASCADE ON UPDATE CASCADE;) ENGINE=InnoDB DEFAULT CHARSET=utf8;
7.	geomajalengka	CREATE TABLE `geomajalengka` (`lat` float DEFAULT NULL, `lng` float DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1;
8.	loginlog	CREATE TABLE IF NOT EXISTS `loginlog` (`id_log` int(11) NOT NULL AUTO_INCREMENT, `email` varchar(100) NOT NULL, `login_time` datetime NOT NULL, `logout_time` datetime DEFAULT NULL, PRIMARY KEY (`id_log`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=54 ;

4.1.6 Implementasi Antar Muka

Implementasi antar muka dalam penelitian ini mendeskripsikan setiap tampilan dari sistem yang akan dibangun. Adapun implementasi antarmuka terdiri dari nama antar muka dan file yang mewakilinya sebegi berikut.

1. Implementasi Antarmuka Web

   Adapun implementasi antarmuka pada web adalah sebagai berikut.

Tabel 4.14 Implementasi Antarmuka Web

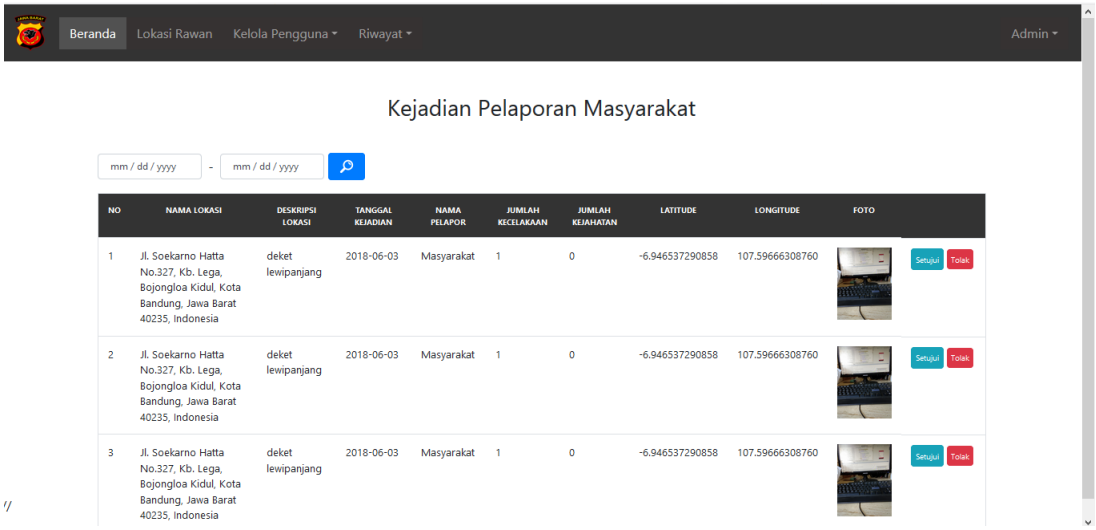
Menu	Deskripsi	Nama File
Masuk	Merupakan halaman awal untuk dapat masuk pada aplikasi	vLogin.php
Lupa kata sandi	Merupakan halaman jika admin lupa kata sandi	vLupaPassword.php
Kejadian Pelaporan Masyarakat	Merupakan halaman untuk mengelola data pelaporan dari masyarakat	vKlarifikasi.php
Lokasi rawan dan tambah data rawan	Merupakan halaman untuk mengelola data lokasi atau titik-titik rawan kecelakaan dan kejahatan di Majalengka	mapsLokasiRawan.php
Detail Lokasi Rawan	Merupakan halaman untuk melihat detail lokasi rawan di setiap ruas jalan	mapsLokasiRawanDetail.php
Data Pengguna	Merupakan halaman untuk admin melihat data-data pengguna yang terdaftar dalam aplikasi	vUserManagement.php
Tambah pengguna	Merupakan halaman untuk admin menambahkan data pengguna baru	vUserManagementTambah.php
Riwayat pelaporan	Merupakan halaman untuk admin melihat data riwayat pelaporan dari masyarakat	vLogKlarifikasi.php

Menu	Deskripsi	Nama File
Log Masuk	Merupakan halaman untuk admin dapat melihat keluar masuknya pengguna dalam aplikasi	vLog.php
Profile Saya	Merupakan halaman untuk melihat data pribadi	vProfileSaya.php

Dari tabel implementasi diatas, adapun beberapa tampilan antarmuka dari web adalah sebagai berikut.

1. Halaman kejadian pelaporan dari masyarakat

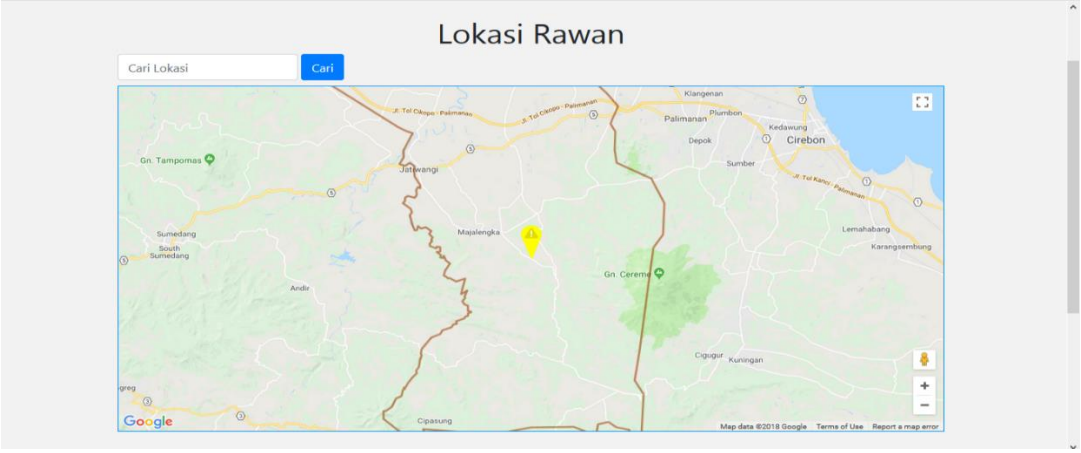
   Berikut ini tampilan antarmuka halaman laporan kejadian kecelakaan atau kejahatan dari masyarakat.



Gambar 4.7 Hasil Implementasi Antarmuka Pelaporan Masyarakat

2. Halaman lokasi rawan wilayah Majalengka

   Berikut ini tampilan antarmuka halaman lokasi rawan kecelaan dan kejahatan di Majalengka.



Gambar 4.8 Hasil Implementasi Antarmuka Lokasi Rawan

3. Implementasi Antarmuka *Mobile*

   Implementasi antarmuka mobile pada penelitian ini terbagi menjadi dua, yaitu antarmuka untuk bagian polisi serta antarmuka untuk masyarakat. Adapun implementasi antarmuka pada mobile adalah sebagai berikut.

Tabel 4.15 Implementasi Antarmuka *Mobile*

Menu	Deskripsi	Nama File
Login	Merupakan halaman awal untuk dapat masuk pada aplikasi	activity_login.xml
Registrasi	Merupakan halaman untuk mendaftarkan akun agar dapat melakukan login	activity_register.xml
Titik-Titik Rawan	Merupakan halaman awal aplikasi, berupa peta digital yang menampilkan titi-titik rawan di majalengka	activity_main.xml

Menu	Deskripsi	Nama File
Lapor Kecelakaan	Merupakan halaman untuk masyarakat melakukan pelaporan kecelakaan atau kejahatan	activity_upload_rawan.xml
Upload Kecelakaan	Merupakan halaman untuk mengupload foto sebagai tanda bukti kecelakaan.	activity_preview_upload_rawan.xml
List Laporan	Merupakan halaman untuk melihat daftar laporan kecelakaan atau kejahatan dari masyarakat	activity_menu_list_laporan.xml

Dari tabel implementasi antarmuka *mobile* atau tampilan *mobile* aplikasi jalur aman ini, adapun beberapa hasil implementasinya adalah sebagai berikut.

1. Halaman Titik Rawan

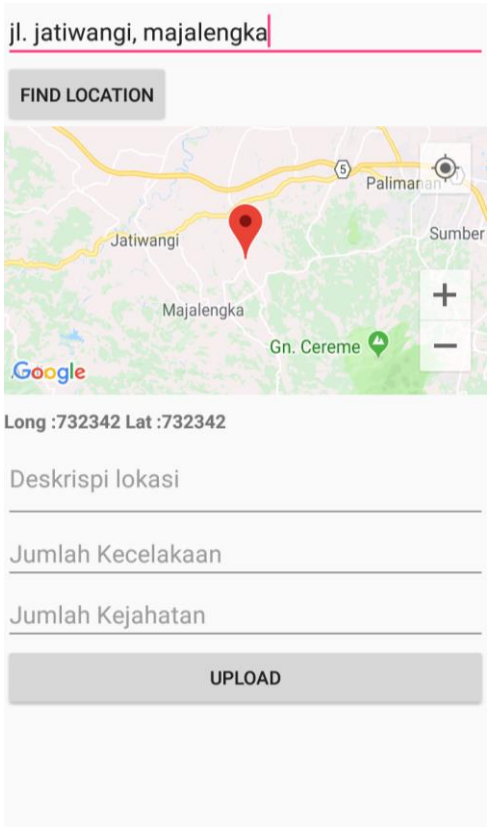
   Berikut ini tampilan halaman titik rawan yang merupakan halaman awal aplikasi ketika pengguna berhasil *login* menggunakan *mobile* android.



Gambar 4.9 Hasil Implementasi Halaman Awal Aplikasi *Mobile*

2. Halaman lapor kejadian kecelakaan atau kejahatan

   Berikut ini hasil implementasi tampilan lapor kejadian kecelakaan atau kejahatan. Halaman ini digunakan untuk masyarakat yang ingin melaporkan kejadian kecelakaan atau kejahatan kepada polisi.



Gambar 4.10 Hasil Implementasi Halaman Lapor Kejadian Kecelakaan Atau Kejahatan

4.2 Pengujian Sistem

Tahap pengujian ini dilakukan untuk menemukan kesalahan atau kekurangan dari sistem yang telah dibangun. Pengujian ini dimaksudkan untuk mengetahui apakah sistem yang dibangun sudah memenuhi kriteria yang dibutuhkan atau tidak. Adapun hasil pengujian pada sistem yang dibangun adalah sebagai berikut.

4.2.1 Pengujian Fungsional

Pada penelitian ini, pengujian sistem yang dilakukan menggunakan metode *black box*. Pengujian *black box* ini dilakukan untuk menguji fungsi-fungsi yang terdapat pada sistem yang dibangun. Adapun skenario pengujian pada sistem yang dibangun adalah sebagai berikut.

1. Skenario Pengujian Pada Web

Adapun skenario pengujian fungsional pada web dalam penelitian ini adalah sebagai berikut.

Tabel 4.16 Rencana Pengujian Web

No	Komponen Yang Di Uji	Skenario	Jenis Pengujian
1.	*Login*	Masukan data email dan *password* yang benar	*Blackbox*
		Masukan data email dan *password* yang salah	
2.	Lupa *Password*	Masukan data email	*Blackbox*
		Masukan data email kosong	
3.	Daftar pelaporan	Menampilkan daftar pelaporan masyarakat	*Blackbox*
		Klik setujui laporan masyarakat	
4.	Tambah data pengguna	Masukan data pengguna secara lengkap dengan benar	*Blackbox*
		Masukan data pengguna kosong	
5.	Lokasi Rawan	Menampilkan titik-titik rawan	*Blackbox*

No	Komponen Yang Di Uji	Skenario	Jenis Pengujian
6.	Tambah data titik rawan	Memasukan pencarian lokasi rawan	*Blackbox*
		Memasukan data secara lengkap dan menambah data titik rawan	
7.	Lihat detail titik rawan	Menampilkan detail titik rawan pada ruas jalan	*Blackbox*
8.	Tambah pengguna	Masukan data pengguna secara lengkap dengan benar	*Blackbox*
		Masukan data pengguna kosong	
9.	Daftar pengguna	Menampilkan daftar list pengguna	*Blackbox*
		Klik detail pada data pengguna	
10.	Riwayat pelaporan	Menampilkan daftar riwayat pelaporan masyarakat	*Blackbox*
11.	Lihat dan hapus riwayat *login* pengguna	Menampilkan daftar riwayat masuk pengguna	*Blackbox*
		Klik hapus pada data riwayat masuk pengguna	

2. Skenario Pengujian Pada *Mobile*

Adapun skenario pengujian pada *mobile* android dalam penelitian ini adalah sebagai berikut.

Tabel 4.17 Rencana Pengujian *Mobile*

No	Komponen Yang Di Uji	Skenario	Jenis Pengujian
1.	Daftar	Masukan data diri pengguna secara lengkap dan benar	*Blackbox*
		Masukan data diri pengguna kosong	
2.	*Login*	Masukan data email dan *password* yang benar	*Blackbox*
		Masukan data email dan *password* yang salah	
3.	Halaman titik rawan	Masukan data pencarian lokasi dengan benar	*Blackbox*
		Rekomendasi rute aman	
4.	Halaman pelaporan masyarakat	Masukan data laporan secara lengkap dengan benar	*Blackbox*
		Masukan data laporan kosong	
5.	Halaman daftar laporan masyarakat	Menampilkan list daftar laporan masyarakat	*Blackbox*
		Klik setujui laporan masyarakat	

4.2.2 Skenario dan Hasil Pengujian Fungsionalitas

Hasil pengujian ini menampilkan benar atau tidaknya funsgionalitas dari sistem yang dibangun. Adapun hasilnya adalah sebagai berikut.

1. Hasil Pengujian Pada Web

Berikut ini hasil dari pengujian pada sistem yang dibangun dalam *platform* web.

Tabel 4.18 Hasil Pengujian Pada Web

No	Komponen Yang Di Uji	Skenario	Hasil Yang Diharapkan	Hasil Pengujian
1.	*Login*	Masukan data email dan *password* yang benar	Menampilkan halaman awal aplikasi	(√) Berhasil ( ) Gagal
		Masukan data email dan *password* yang salah	Menampilkan pesan *password* yang dimasukan salah	(√) Berhasil ( ) Gagal
2.	Lupa *Password*	Masukan data email	Mengirimkan *password* baru melalui email	(√) Berhasil ( ) Gagal
		Masukan data email kosong	*Password* baru tidak terkirim melalui email	(√) Berhasil ( ) Gagal
3.	Daftar pelaporan	Menampilkan daftar pelaporan masyarakat	Tampil daftar pelaporan dari masyarakat	(√) Berhasil ( ) Gagal
		Klik setujui laporan masyarakat	Titik rawan bertambah setelah di setujui	(√) Berhasil ( ) Gagal

No	Komponen Yang Di Uji	Skenario	Hasil Yang Diharapkan	Hasil Pengujian
4.	Tambah data pengguna	Masukan data pengguna secara lengkap dengan benar	Berhasil menambah data pengguna	(√) Berhasil ( ) Gagal
		Masukan data pengguna kosong	Menampilkan pesan data masukan harus diisi	(√) Berhasil ( ) Gagal
5.	Lokasi Rawan	Menampilkan titik-titik rawan	Tampil data titik rawan berupa peta digital	(√) Berhasil ( ) Gagal
6.	Tambah data titik rawan	Memasukan pencarian lokasi rawan	Menampilkan lokasi rawan yang di cari	(√) Berhasil ( ) Gagal
		Memasukan data secara lengkap dan menambah data titik rawan	Berhasil menambah lokasi rawan	(√) Berhasil ( ) Gagal
7.	Lihat detail titik rawan	Menampilkan detail titik rawan pada ruas jalan	Menampilkan detail titik rawan	(√) Berhasil ( ) Gagal
8.	Tambah pengguna	Masukan data pengguna secara lengkap dengan benar	Berhasil menambha data pengguna	(√) Berhasil ( ) Gagal
		Masukan data pengguna kosong	Menampilkan pesan data harus diisi	(√) Berhasil ( ) Gagal
9.	Daftar pengguna	Menampilkan daftar list pengguna	Tampil daftar data pengguna	(√) Berhasil ( ) Gagal

No	Komponen Yang Di Uji	Skenario	Hasil Yang Diharapkan	Hasil Pengujian
		Klik detail pada data pengguna	Berhasil menampilkan detail data pengguna	(√) Berhasil ( ) Gagal
10.	Riwayat pelaporan	Menampilkan daftar riwayat pelaporan masyarakat	Tampil daftar data riwayat pelaporan masyarakat	(√) Berhasil ( ) Gagal
11.	Lihat dan hapus riwayat *login* pengguna	Menampilkan daftar riwayat masuk pengguna	Tampil daftar riwayat masuk pengunna aplikasi	(√) Berhasil ( ) Gagal
		Klik hapus pada data riwayat masuk pengguna	Berhasil menghapus data riwayat masuk pengguna aplikasi	(√) Berhasil ( ) Gagal

1. Hasil Pengujian Pada *Mobile*

Berikut ini hasil dari pengujian pada sistem yang dibangun dalam *platform mobile*.

Tabel 4.19 Hasil Pengujian *Mobile*

No	Komponen Yang Di Uji	Skenario	Hasil Yang Diharapkan	Hasil Pengujian
1.	Daftar	Masukan data diri pengguna secara lengkap dan benar	Berhasil membuat akun	(√) Berhasil ( ) Gagal

No	Komponen Yang Di Uji	Skenario	Hasil Yang Diharapkan	Hasil Pengujian
			dan dapat melakukan login	
		Masukan data diri pengguna kosong	Menampilkan pesan kesalahan data diri harus diisi	(√) Berhasil ( ) Gagal
2.	*Login*	Masukan data email dan *password* yang benar	Berhasil masuk pada halaman awal aplikasi	(√) Berhasil ( ) Gagal
		Masukan data email dan *password* yang salah	Menapilkan pesan bahwa *password* salah	(√) Berhasil ( ) Gagal
3.	Halaman titik rawan	Masukan data pencarian lokasi dengan benar	Berhasil menampilkan lokasi pencarian	(√) Berhasil ( ) Gagal
		Rekomendasi rute aman	Menampilkan rute yang terhindar dari titik rawan	(√) Berhasil ( ) Gagal
4.	Halaman pelaporan masyarakat	Masukan data laporan secara lengkap dengan benar	Berhasil mengirim laporan kepada polisi serta admin	(√) Berhasil ( ) Gagal
		Masukan data laporan kosong	Menampilkan pesan data pelaporan harus diisi	(√) Berhasil ( ) Gagal

No	Komponen Yang Di Uji	Skenario	Hasil Yang Diharapkan	Hasil Pengujian
5.	Halaman daftar laporan masyarakat	Menampilkan list daftar laporan masyarakat	Tampil daftra laporan masyarakat	(√) Berhasil ( ) Gagal
		Klik setujui laporan masyarakat	Berhasil menyimpan titik rawan baru dari laporan masyarakat	(√) Berhasil ( ) Gagal

4.2.3 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif, dilakukan secara langsung oleh pengguna sistem. Pengujian ini dilakukan untuk dapat mengetahui sejauh mana sistem yang telah dibangun dapat membantu dan memudahkan pengguna. Dalam penelitian ini pengujian beta dilakukan dengan menggunakan teknik kuesioner dan wawancara.

4.2.3.1 Skenario Pengujian Beta

Pengujian yang dilakukan dalam penelitian ini juga menggunakan teknik kuesioner untuk dapat menarik kesimpulan. Kuesioner ini diajukan kepada masyarakat Majalengka serta Polisi yang menggunakan aplikasi melalui Mobile android. Dari setiap pertanyaan akan diberikan 5 buah pilihan jawaban . Dari jawaban setiap pengguna tersebut akan dihitung persentasenya menggunakan skala *likert.* Adapun detail dari skala *likert* pada penelitian ini adalah sebagai berikut.

Tabel 4.20 Penilaian *Skala Likert*

KATEGORI	NILAI
Sangat setuju	5
Setuju	4
Ragu-ragu	3

KATEGORI	NILAI
Tidak setuju	2
Sangat tidak setuju	1

Untuk menghitung jumlah skor kriterium jumlah penilaian = 5, dan jumlah pengisi kuesioner = 30 dapat menggunakan rumus :

Kriterium = Jumlah Penilaian x Jumlah Responden

Jadi, nilai dari kuesioner yang dibuat untuk masyarakat : 5 x 30 = 150. Sedangkan untuk polisi 5 x 10 = 50.

Untuk menghitung jawaban dari responden dalam bentuk persentse menggunakan rumus :

$$P = \frac{Total\ nilai}{Skor\ ideal} \times 100\%$$

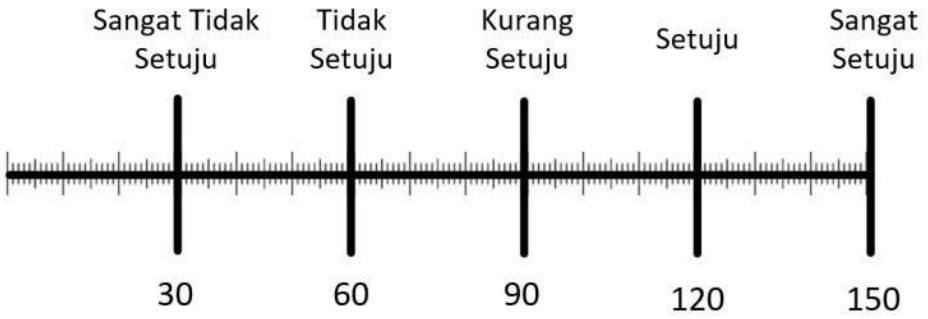
Keterangan :

P = Nilai persentase yang dicari

Total nilai = Jumlah frekuensi dikalikan nilai yang di tetapkan jawaban

Skor ideal = Nilai tertinggi dilakukan dengan jumlah sampel

Skor yang telah di dapat, selanjutnya dimasukan ke dalam bentuk interpretasi skor berikut ini.



Gambar 4.11 *Interval Rating Scale*

Adapun daftar pertanyaan kuesioner yang diajukan kepada masyarakat dan polisi sebagai pengguna aplikasi adalah sebagai berikut.

Tabel 4.21 Daftar Pertanyaan Kuesioner Masyarakat

NO	PERTANYAAN
1.	Apakah aplikasi jalur aman pada android ini dapat digunakan dengan mudah?
2.	Apakah aplikasi jalur aman pada android ini sudah dapat menampilkan titik-titik rawan di Majalengka?
3.	Apakah aplikasi jalur aman pada android ini mudah digunakan untuk mencari jalur alternatif?
4.	Apakah aplikasi jalur aman pada android ini sudah dapat menghindari daerah yang rawan di Majalengka?
5.	Apakah aplikasi jalur aman pada android ini sudah dapat digunakan untuk melaporkan kejadian kecelakaan atau kejahatan di Majalengka?

Tabel 4.22 Daftar Pertanyaan Kuesioner Polisi

NO	PERTANYAAN
1.	Apakah aplikasi jalur aman pada android ini dapat digunakan dengan mudah?
2.	Apakah aplikasi jalur aman pada android ini sudah dapat menampilkan titik-titik rawan di Majalengka?

NO	PERTANYAAN
3.	Apakah aplikasi jalur aman pada android ini mudah digunakan untuk mencari jalur alternatif?
4.	Apakah aplikasi jalur aman pada android ini sudah dapat menghindari daerah rawan di Majalengka?
5.	Apakah aplikasi jalur aman pada android ini sudah dapat menerima dan menyetujui laporan dari masyarakat?

4.2.3.2 Perhitungan Hasil Kuesioner

Dengan menggunakan rumus yang telah dijelaskan diatas, adapun hasil perhitungn kuesioner yang telah diajukan kepada 30 orang masyarakat Majalengka dan10 orang polisi yang bekerja di Polres Majalengka adalah sebagai berikut.

Berikut ini hasil perhitungan dari setiap pertanyaan yang diajukan untuk masyarakat. Hasilnya adalah :

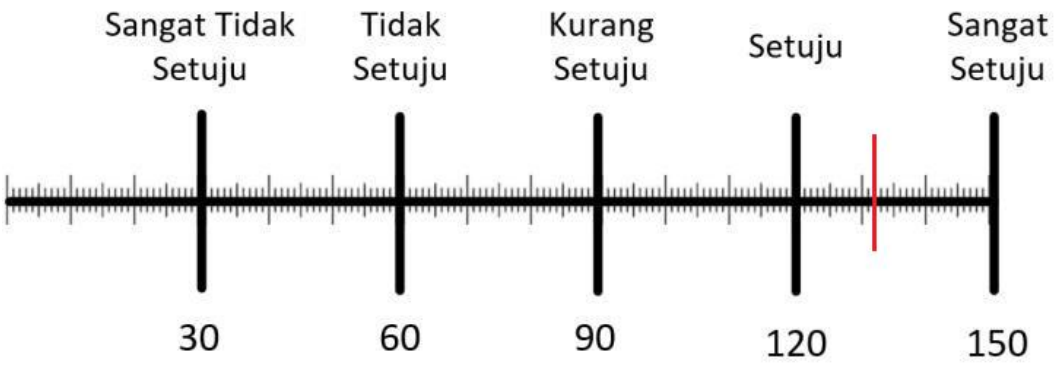
1. Apakah aplikasi jalur aman pada android ini dapat digunakan dengan mudah?

Tabel 4.23 Hasil Perhitungan Jawaban Masyarakat Nomor 1

Kategori Jawaban	Skor	Frekuensi jawaban	Total skor	Nilai persentase	Keputusan
Sangat setuju	5	14	70	(133 / (30x5)) x 100% = **88.67%**	Setuju
Setuju	4	15	60		
Ragu-ragu	3	1	3		
Tidak setuju	2	0	0		
Sangat tidak setuju	1	0	0		
Jumlah		30	133		

Berdasarkan perhitungan diatas, total skor yang diperoleh adalah 133 dengan skala kategori masuk pada kategori setuju sedangkan untuk nilai persentasenya menghasilkan nilai 88.67% maka dapat dinyatakan bahwa aplikasi

yang dibangun sudah dapat memenuhi tujuannya. Berikut ini *interval rating scale* untuk pertanyaan pertama.



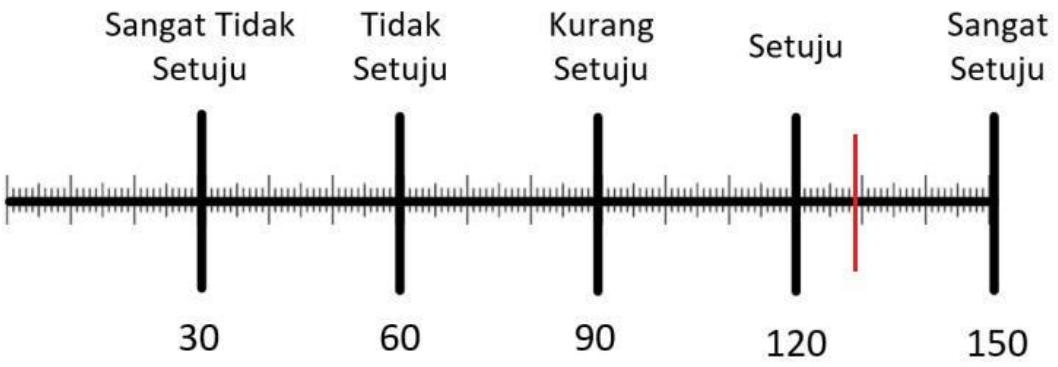
Gambar 4.12 *Interval Rating Scale* Pertanyaan 1

2. Apakah aplikasi jalur aman pada android ini sudah dapat menampilkan titik-titik rawan di Majalengka?

Tabel 4.24 Hasil Perhitungan Jawaban Masyarakat Nomor 2

Kategori Jawaban	Skor	Frekuensi jawaban	Total skor	Nilai persentase	Keputusan
Sangat setuju	5	11	55	(129 / (30x5)) x 100% = **86%**	Setuju
Setuju	4	17	68		
Ragu-ragu	3	2	6		
Tidak setuju	2	0	0		
Sangat tidak setuju	1	0	0		
Jumlah		30	129		

Berdasarkan perhitungan diatas, total skor yang diperoleh adalah 129 dengan skala kategori masuk pada kategori setuju sedangkan untuk nilai persentasenya menghasilkan nilai 86% maka dapat dinyatakan bahwa aplikasi yang dibangun sudah dapat memenuhi tujuannya. Berikut ini *interval rating scale* untuk pertanyaan kedua.



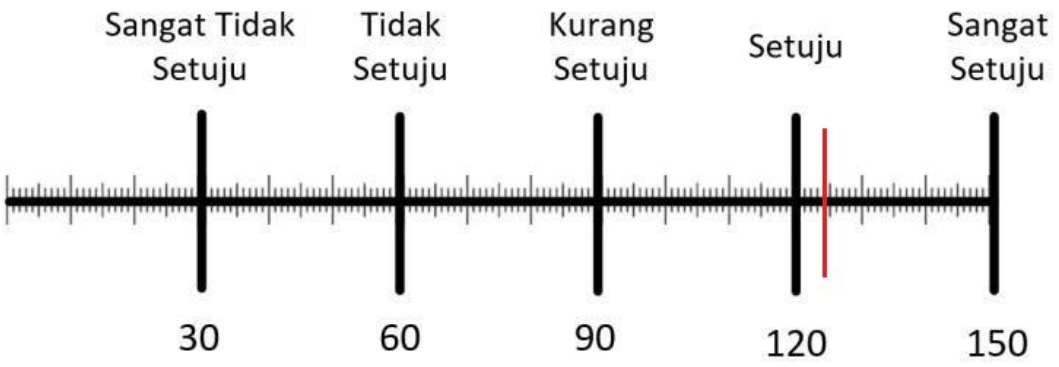
Gambar 4.13 *Interval Rating Scale* Pertanyaan 2

3. Apakah aplikasi jalur aman pada android ini mudah digunakan untuk mencari jalur alternatif?

Tabel 4.25 Hasil Perhitungan Jawaban Masyarakat Nomor 3

Kategori Jawaban	Skor	Frekuensi jawaban	Total skor	Nilai persentase	Keputusan
Sangat setuju	5	7	35	(123 / (30x5)) x 100% = **82%**	Setuju
Setuju	4	19	76		
Ragu-ragu	3	4	12		
Tidak setuju	2	0	0		
Sangat tidak setuju	1	0	0		
Jumlah		30	123		

Berdasarkan perhitungan diatas, total skor yang diperoleh adalah 124 dengan skala kategori masuk pada kategori setuju sedangkan untuk nilai persentasenya menghasilkan nilai 82% maka dapat dinyatakan bahwa aplikasi yang dibangun sudah dapat memenuhi tujuannya. Berikut ini *interval rating scale* untuk pertanyaan ketiga.



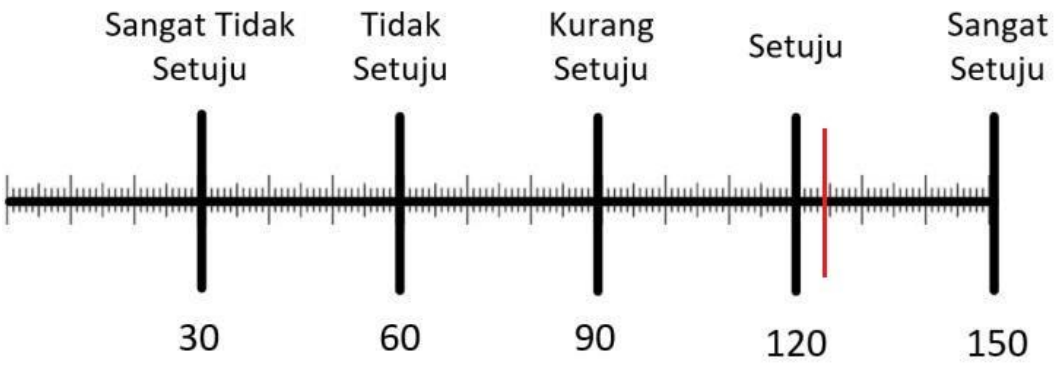
Gambar 4.14 Interval Rating Scale Pertanyaan 3

4. Apakah aplikasi jalur aman pada android ini sudah dapat menghindari daerah yang rawan di Majalengka?

Tabel 4.26 Hasil Perhitungan Jawaban Masyarakat Nomor 4

Kategori Jawaban	Skor	Frekuensi jawaban	Total skor	Nilai persentase	Keputusan
Sangat setuju	5	9	45	(124 / (30x5)) x 100% = **82.67%**	Setuju
Setuju	4	16	64		
Ragu-ragu	3	5	15		
Tidak setuju	2	0	0		
Sangat tidak setuju	1	0	0		
Jumlah		30	124		

Berdasarkan perhitungan diatas, total skor yang diperoleh adalah 124 dengan skala kategori masuk pada kategori setuju sedangkan untuk nilai persentasenya menghasilkan nilai 82.67% maka dapat dinyatakan bahwa aplikasi yang dibangun sudah dapat memenuhi tujuannya. Berikut ini *interval rating scale* untuk pertanyaan keempat.



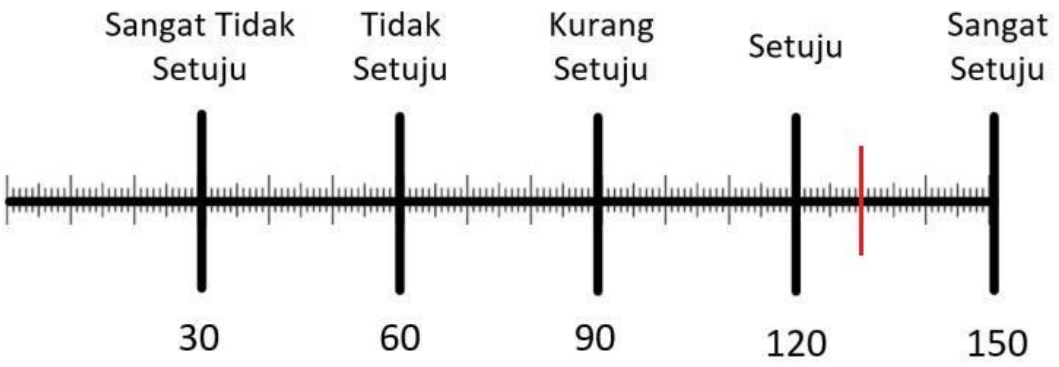
Gambar 4.15 *Interval Rating* Scale Pertanyaan 4

5. Apakah aplikasi jalur aman pada android ini sudah dapat digunakan untuk melaporkan kejadian kecelakaan atau kejahatan di Majalengka?

Tabel 4.27 Hasil Perhitungan Jawaban Masyarakat Nomor 5

Kategori Jawaban	Skor	Frekuensi jawaban	Total skor	Nilai persentase	Keputusan
Sangat setuju	5	13	65	(130 / (30x5)) x 100% = **86.67%**	Setuju
Setuju	4	14	56		
Ragu-ragu	3	3	9		
Tidak setuju	2	0	0		
Sangat tidak setuju	1	0	0		
Jumlah		30	130		

Berdasarkan perhitungan diatas, total skor yang diperoleh adalah 130 dengan skala kategori masuk pada kategori setuju sedangkan untuk nilai persentasenya menghasilkan nilai 86.67% maka dapat dinyatakan bahwa aplikasi yang dibangun sudah dapat memenuhi tujuannya. Berikut ini *interval rating scale* untuk pertanyaan kelima.



Gambar 4.16 *Interval Rating* Scale Pertanyaan 5

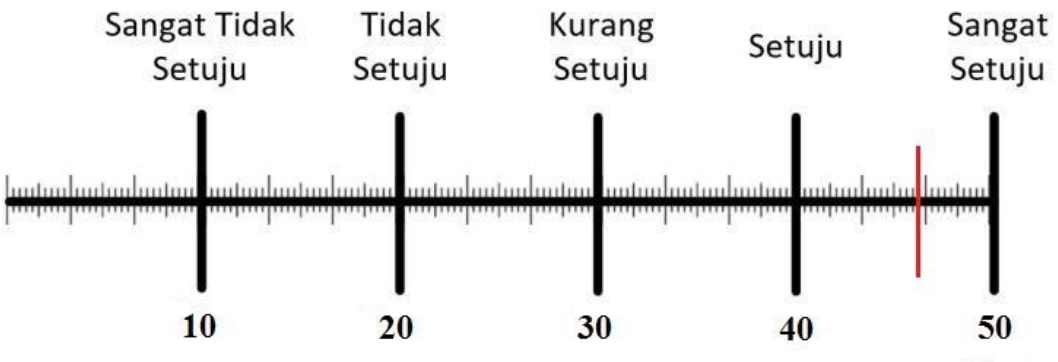
Sedangkan hasil perhitungan kuesioner yang diajukan untuk bagian kepolisian dalam penelitian ini adalah sebagai berikut.

1. Apakah aplikasi jalur aman pada android ini dapat digunakan dengan mudah?

Tabel 4.28 Hasil Perhitungan Jawaban Polisi Nomor 1

Kategori Jawaban	Skor	Frekuensi jawaban	Total skor	Nilai persentase	Keputusan
Sangat setuju	5	7	35	(47 / (10x5)) x 100% = **94%**	Sangat Setuju
Setuju	4	3	12		
Ragu-ragu	3	0	0		
Tidak setuju	2	0	0		
Sangat tidak setuju	1	0	0		
Jumlah		10	47		

Berdasarkan perhitungan diatas, total skor yang diperoleh adalah 47 dengan skala kategori masuk pada kategori sangat setuju sedangkan untuk nilai persentasenya menghasilkan nilai 94% maka dapat dinyatakan bahwa aplikasi yang dibangun sudah dapat memenuhi tujuannya. Berikut ini *interval rating scale* untuk pertanyaan pertama untuk polisi.



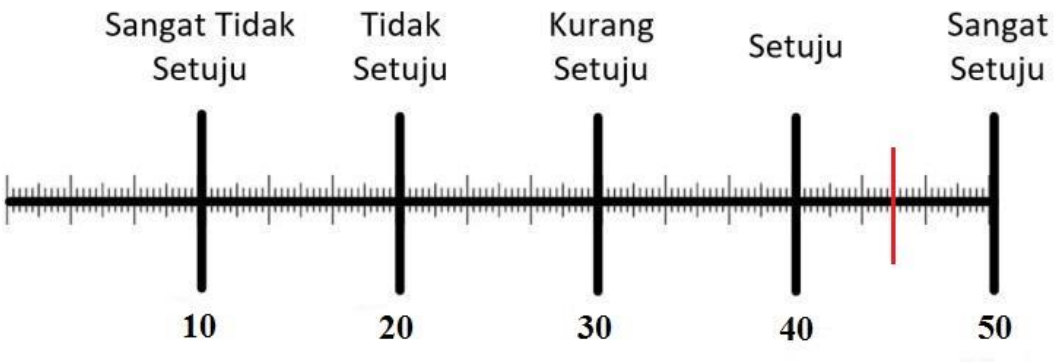
Gambar 4.17 *Interval Rating Scale* Pertanyaan 1 Polisi

2. Apakah aplikasi jalur aman pada android ini sudah dapat menampilkan titik-titik rawan di Majalengka?

Tabel 4.29 Hasil Perhitungan Jawaban Polisi Nomor 2

Kategori Jawaban	Skor	Frekuensi jawaban	Total skor	Nilai persentase	Keputusan
Sangat setuju	5	5	25	(45 / (10x5)) x 100% = **90%**	Setuju
Setuju	4	5	20		
Ragu-ragu	3	0	0		
Tidak setuju	2	0	0		
Sangat tidak setuju	1	0	0		
Jumlah		10	45		

Berdasarkan perhitungan diatas, total skor yang diperoleh adalah 45 dengan skala kategori masuk pada kategori setuju sedangkan untuk nilai persentasenya menghasilkan nilai 90% maka dapat dinyatakan bahwa aplikasi yang dibangun sudah dapat memenuhi tujuannya. Berikut ini *interval rating scale* untuk pertanyaan kedua untuk polisi.



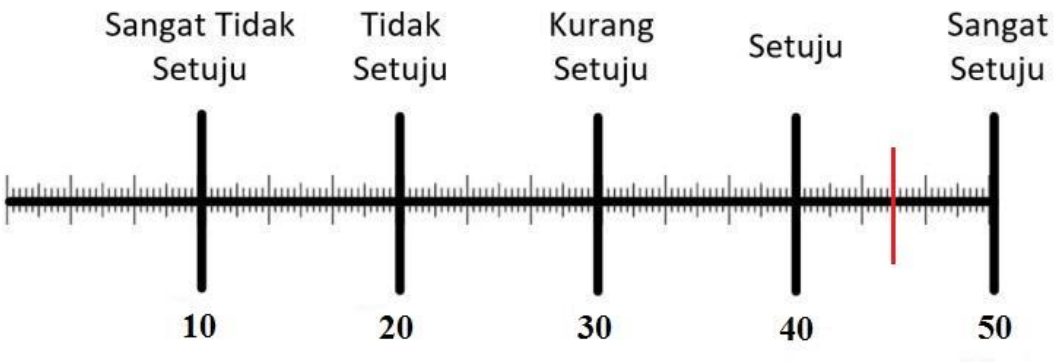
Gambar 4.18 *Interval Rating Scale* Pertanyaan 2 Polisi

3. Apakah aplikasi jalur aman pada android ini mudah digunakan untuk mencari jalur alternatif?

Tabel 4.30 Hasil Perhitungan Jawaban Polisi Nomor 3

Kategori Jawaban	Skor	Frekuensi jawaban	Total skor	Nilai persentase	Keputusan
Sangat setuju	5	5	25	(45 / (10x5)) x 100% = **90%**	Setuju
Setuju	4	5	20		
Ragu-ragu	3	0	0		
Tidak setuju	2	0	0		
Sangat tidak setuju	1	0	0		
Jumlah		10	45		

Berdasarkan perhitungan diatas, total skor yang diperoleh adalah 45 dengan skala kategori masuk pada kategori setuju sedangkan untuk nilai persentasenya menghasilkan nilai 90% maka dapat dinyatakan bahwa aplikasi yang dibangun sudah dapat memenuhi tujuannya. Berikut ini *interval rating scale* untuk pertanyaan ketiga untuk polisi.



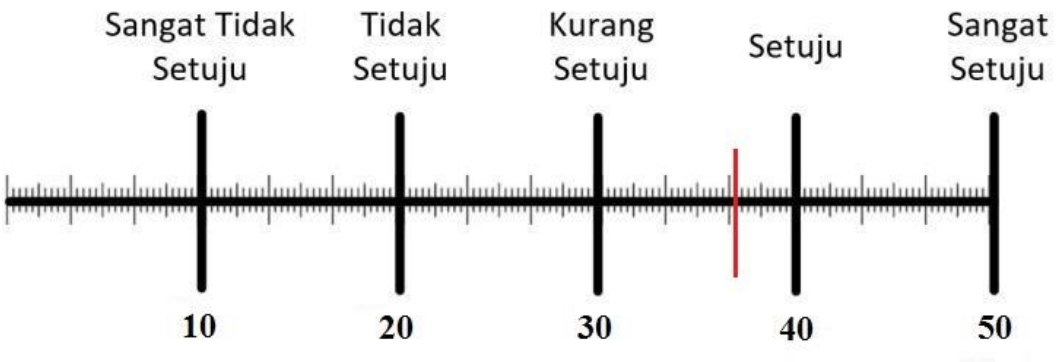
Gambar 4.19 *Interval Rating Scale* Pertanyaan 3 Polisi

4. Apakah aplikasi jalur aman pada android ini sudah dapat menghindari daerah rawan di Majalengka?

Tabel 4.31 Hasil Perhitungan Jawaban Polisi Nomor 4

Kategori Jawaban	Skor	Frekuensi jawaban	Total skor	Nilai persentase	Keputusan
Sangat setuju	5	0	0	(37 / (10x5)) x 100% = **74%**	Setuju
Setuju	4	7	28		
Ragu-ragu	3	3	9		
Tidak setuju	2	0	0		
Sangat tidak setuju	1	0	0		
Jumlah		10	37		

Berdasarkan perhitungan diatas, total skor yang diperoleh adalah 37 dengan skala kategori masuk pada kategori setuju sedangkan untuk nilai persentasenya menghasilkan nilai 74% maka dapat dinyatakan bahwa aplikasi yang dibangun sudah dapat memenuhi tujuannya. Berikut ini *interval rating scale* untuk pertanyaan ketiga untuk polisi.



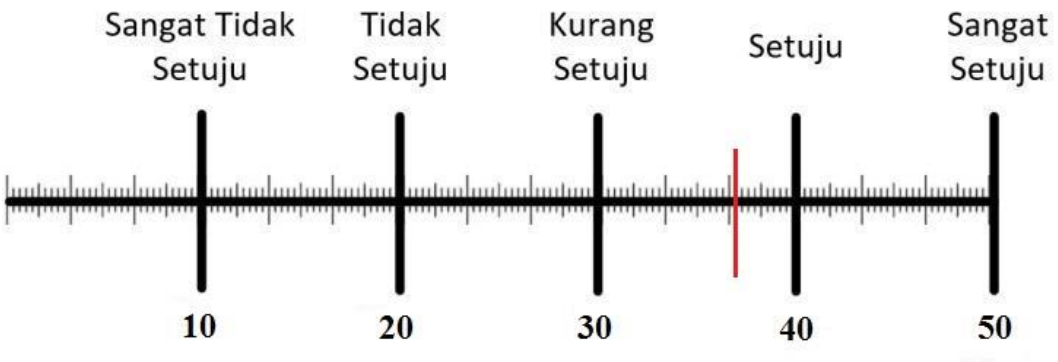
Gambar 4.20 *Interval Rating Scale* Pertanyaan 4 Polisi

5. Apakah aplikasi jalur aman pada android ini sudah dapat menerima dan menyetujui laporan dari masyarakat?

Tabel 4.32 Hasil Perhitungan Jawaban Polisi Nomor 5

Kategori Jawaban	Skor	Frekuensi jawaban	Total skor	Nilai persentase	Keputusan
Sangat setuju	5	1	5	(37 / (10x5)) x 100% = **74%**	Setuju
Setuju	4	5	20		
Ragu-ragu	3	4	12		
Tidak setuju	2	0	0		
Sangat tidak setuju	1	0	0		
Jumlah		10	37		

Berdasarkan perhitungan diatas, total skor yang diperoleh adalah 37 dengan skala kategori masuk pada kategori setuju sedangkan untuk nilai persentasenya menghasilkan nilai 74% maka dapat dinyatakan bahwa aplikasi yang dibangun sudah dapat memenuhi tujuannya. Berikut ini *interval rating scale* untuk pertanyaan ketiga untuk polisi.



Gambar 4.21 *Interval Rating Scale* Pertanyaan 5 Polisi

4.2.4 Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian dengan menggunakan metode *black box* yang telah dilakukan, maka fungsionalitas yang terdapat pada aplikasi pencarian jalur aman pada *paltform mobile* android serta web sudah dapat diterima. Dengan begitu, aplikasi tersebut sudah dapat digunakan oleh masyarakat, polisi dan admin sebagai pengguna.

Untuk hasil pengujian beta, hasilnya dapat dilihat pada tabel dibawah ini. Berikut ini hasil dari pengujian beta untuk pengguna masyarakat.

Tabel 4.33 Kesimpulan Hasil Pengujian Masyarakat

No.	Pertanyaan	Skor	Persentase	Keputusan
1.	Apakah aplikasi jalur aman pada android ini dapat digunakan dengan mudah?	133	88.67%	Setuju
2.	Apakah aplikasi jalur aman pada android ini sudah dapat menampilkan titik-titik rawan di Majalengka?	129	86%	Setuju
3.	Apakah aplikasi jalur aman pada android ini mudah	123	82%	Setuju

No.	Pertanyaan	Skor	Persentase	Keputusan
	digunakan untuk mencari jalur alternatif?			
4.	Apakah aplikasi jalur aman pada android ini sudah dapat menghindari daerah yang rawan di Majalengka?	124	82.67%	Setuju
5.	Apakah aplikasi jalur aman pada android ini sudah dapat digunakan untuk melaporkan kejadian kecelakaan atau kejahatan di Majalengka?	130	86.67%	Setuju

Sedangkan untuk hasil pengujian beta terhadap jawaban kuesioner polisi adalah sebagai berikut.

Tabel 4.34 Kesimpulan Hasil Pengujian Polisi

No.	Pertanyaan	Skor	Persentase	Keputusan
1.	Apakah aplikasi jalur aman pada android ini dapat digunakan dengan mudah?	47	94%	Sangat Setuju
2.	Apakah aplikasi jalur aman pada android ini sudah dapat menampilkan titik-titik rawan di Majalengka?	45	90%	Setuju
3.	Apakah aplikasi jalur aman pada android ini mudah digunakan untuk mencari jalur alternatif?	45	90%	Setuju

No.	Pertanyaan	Skor	Persentase	Keputusan
4.	Apakah aplikasi jalur aman pada android ini sudah dapat menghindari daerah yang rawan di Majalengka?	37	74%	Setuju
5.	Apakah aplikasi jalur aman pada android ini sudah dapat menerima dan menyetujui laporan dari masyarakat?	37	74%	Setuju

Dari hasil rekapitulasi tersebut, terlihat bahwa poin yang dihasilkan bernilai cukup baik dan tinggi. Masyarakat Majalengka banyak yang memberikan jawaban yang baik terhadap aplikasi yang dibangun. Selain itu, bagian kepolisian pun banyak memberikan jawaban setuju terhadap aplikasi jalur aman ini. Dengan begitu, aplikasi jalur aman yang dibangun ini sudah dapat membantu masyarakat dan polisi dalam mendapatkan informasi mengenai daerah rawan di Majalengka.