

BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Implementasi sistem merupakan penerapan dari perancangan yang telah dilakukan berdasarkan hasil analisis ke dalam bahasa pemrograman tertentu serta penerapan perangkat lunak yang di bangun. Implementasi dan pengujian dilakukan dengan tujuan untuk mengetahui hasil dari sistem yang telah di bangun. Pada penelitian ini metode pengujian terhadap sistem yang dibangun yaitu menggunakan metode pengujian *black box* dan pengujian beta.

4.1.1 Implementasi Perangkat Keras

Implementasi perangkat keras merupakan spesifikasi perangkat keras yang digunakan dalam mengimplementasikan sistem. Berikut merupakan implementasi perangkat keras yang digunakan dalam menjalankan sistem yang di gunakan yang telah dibangun ditunjukkan pada tabel 4.1

Tabel 4.1 Spesifikasi Implementasi Perangkat Keras

No	Komponen	Spesifikasi
1	Processor	Octa-core (4x1.5 GHz Cortex-A53 & 4x1.2 GHz Cortex-A53)
2	RAM	2 GB
3	Memory Internal	16 GB
4	Fitur	Kamera dan GPS
5	Jaringan	GSM/HSPA/LTE

4.1.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak merupakan spesifikasi perangkat lunak yang digunakan dalam mengimplementasikan sistem. Berikut merupakan implementasi perangkat lunak yang digunakan dalam menjalankan sistem yang telah dibangun ditunjukkan pada tabel 4.2

Tabel 4.2 Spesifikasi Implementasi Perangkat Lunak

No	Komponen	Spesifikasi
1	Sistem Operasi	Android 5.0 Lollipop (API Level 22)
2	IDE	Android Studio 3.3
3	Text Editor	Sublime Text

4.1.3 Implementasi Basis Data

Implementasi basis data merupakan deskripsi gambaran perancangan basis data pada aplikasi yang di bangun. Basis data yang dibangun menggunakan Bahasa SQL (*Structured Query Language*). Adapun pada implementasinya, DBMS yang digunakan yaitu adalah MySQL dengan versi 5.1.7.3. Berikut implementasi basis data yang telah dilakukan dapat dilihat pada tabel 4.3

Tabel 4.3 Implementasi Basis Data

No	Nama Tabel	Perintah
1	Pengguna	<pre>CREATE TABLE `pengguna` (`id_pengguna` int(11) NOT NULL AUTO_INCREMENT, `nama` varchar(30) NOT NULL, `email` varchar(50) NOT NULL, `password` varchar(100) NOT NULL, `pass` varchar(100) NOT NULL, `alamat` varchar(100) NOT NULL, `jenis_kelamin` varchar(30) NOT NULL, `status` varchar(30) NOT NULL, `tanggal` date NOT NULL, `kode` varchar(10) NOT NULL, PRIMARY KEY (`id_pengguna`), UNIQUE KEY `email` (`email`)) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1</pre>
2	Laporan	<pre>CREATE TABLE `laporan` (`id_laporan` int(11) NOT NULL AUTO_INCREMENT, `id_pengguna` int(11) NOT NULL, `kota` varchar(30) NOT NULL, `alamat` varchar(100) NOT NULL, `lat` varchar(20) NOT NULL, `lng` varchar(20) NOT NULL, `foto` varchar(60) NOT NULL, `tanggal` datetime NOT NULL, PRIMARY KEY (`id_laporan`),</pre>

		<pre> KEY `id_pengguna` (`id_pengguna`), CONSTRAINT `id_pengguna` FOREIGN KEY (`id_pengguna`) REFERENCES `pengguna` (`id_pengguna`) ON DELETE NO ACTION ON UPDATE NO ACTION) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1 </pre>
3	Sensor	<pre> CREATE TABLE `sensor` (`id_sensor` varchar(30) NOT NULL, `lokasi` varchar(100) NOT NULL, `latitude` varchar(30) NOT NULL, `longitude` varchar(30) NOT NULL, PRIMARY KEY (`id_sensor`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 </pre>
4	Cuaca	<pre> CREATE TABLE `cuaca` (`id_cuaca` int(11) NOT NULL AUTO_INCREMENT, `id_sensor` varchar(30) NOT NULL, `rain` varchar(10) NOT NULL, `humidity` varchar(10) NOT NULL, `temperature` varchar(10) NOT NULL, `wind` varchar(10) NOT NULL, `time` varchar(60) NOT NULL, `tanggal` datetime NOT NULL, PRIMARY KEY (`id_cuaca`), KEY `id_sensor` (`id_sensor`), CONSTRAINT `cuaca_ibfk_1` FOREIGN KEY (`id_sensor`) REFERENCES `sensor` (`id_sensor`) ON DELETE NO ACTION ON UPDATE NO ACTION) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1 </pre>

4.1.4 Implementasi Teknologi Yang Digunakan

Pada perancangan perangkat lunak dalam penelitian ini digunakan beberapa teknologi. Berikut merupakan implementasi dari setiap teknologi yang digunakan dalam pembangunan sistem yang telah dibangun.

4.1.4.1 Implementasi GPS

Teknologi GPS ini digunakan untuk mengetahui keberadaan atau posisi dari pengendara. Adapun implementasi dari teknologi GPS dalam sistem yang dibangun ditunjukkan pada tabel 4.4

Tabel 4.4 Implementasi GPS

```
public void getDeviceLocation() {

    mFusedLocationClient =
    LocationServices.getFusedLocationProviderClient(activitiyreport.
    getApplicationContext());
    LocationRequest locationRequest = new LocationRequest();
    mFusedLocationClient.getLastLocation()
    .addOnSuccessListener(activitiyreport, new
    OnSuccessListener<Location>() {
    @Override
    public void onSuccess(Location location) {
    if (location != null) {
    latitude    = location.getLatitude();
    longititude = location.getLongitude();

    }
    moveCamera(new LatLng(latitude, longititude), rMap);
    });

    locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURA
    CY).setInterval(5);
    LocationCallback locationCallback = new LocationCallback() {
    @Override
    public void onLocationResult(LocationResult locationResult) {
    super.onLocationResult(locationResult);
    if(locationResult == null) {
    return;
    }
    for(Location location : locationResult.getLocations()) {
    latitude    = location.getLatitude();
    longititude = location.getLongitude();

    activitiyreport.getPlace(latitude, longititude);
    moveCamera(new LatLng(latitude, longititude), rMap);

    });
    mFusedLocationClient.requestLocationUpdates(locationRequest,
    locationCallback, null);
    }
}
```

4.1.4.2 Implementasi Google Maps API

Google Maps API digunakan untuk memberikan informasi berupa informasi *map* yang digunakan aplikasi dalam menampilkan informasi seperti jalan, kota dan

informasi peta lainnya. Adapun implementasi dari google maps API ini ditunjukkan pada tabel 4.5

Tabel 4.5 Implementasi Google Maps API

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value=" YOUR_API_KEY "/>
```

Implementasi Google Map pada layout aplikasi ditunjukkan pada tabel 4.6

Tabel 4.6 Implementasi Google Maps API pada Layout

```
<fragment
xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/map"
class="com.google.android.gms.maps.SupportMapFragment"
android:layout_width="0dp"
android:layout_height="0dp"
app:layout_constraintBottom_toTopOf="@+id/menu_bar"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.0" />
```

4.1.4.3 Implementasi Google Directions API

Google Directions API merupakan layanan yang dapat memberikan informasi berupa rute dan navigasi antara dua titik lokasi asal dan tujuan. Adapun implementasi dari google maps API ini ditunjukkan pada tabel 4.7

Tabel 4.7 Implementasi Google Directions API

```
public List<List<HashMap<String, String>>> getRoute(String
origin, String destination) {
    JSONArray jRoutes, jLegs, jSteps;
    JSONObject jInfo;
    String data, distance, duration, vdistance;
    List<List<HashMap<String, String>>> routes = new ArrayList<>();
    routeinfo = new ArrayList<>();

    HashMap<String, String> params = new HashMap<>();
    params.put("key", KEY);
    params.put("origin", origin);
    params.put("destination", destination);
```

```

params.put("avoid", "tolls|highways");
params.put("departure_time", "now");
params.put("alternatives", "true");

data = new Request().sendGetRequestParam(URL_ROUTE, params);
try {
jRoutes = new JSONObject(data).getJSONArray("routes");
for (int i = 0; i < jRoutes.length(); i++) {
jLegs = jRoutes.getJSONObject(i).getJSONArray("legs");
List path = new ArrayList();

for (int j = 0; j < jLegs.length(); j++) {
jSteps = jLegs.getJSONObject(j).getJSONArray("steps");

for (int k = 0; k < jSteps.length(); k++) {
String polyline;
polyline =
jSteps.getJSONObject(k).getJSONObject("polyline").get("points").
toString();
List<LatLng> list = decodePoly(polyline);

for (int l = 0; l < list.size(); l++) {
HashMap<String, String> hm = new HashMap<>();
hm.put("lat", Double.toString((list.get(l)).latitude));
hm.put("lng", Double.toString((list.get(l)).longitude));
path.add(hm);
}}}
routes.add(path);
}
} catch (JSONException e) {
e.printStackTrace();
}}

```

4.1.4.4 Implementasi Cuaca API

Cuaca API merupakan layanan yang disediakan oleh Pemerintah Kota Bandung untuk memberikan informasi terkait parameter cuaca di sekitar wilayah

Kota Bandung. Adapun implementasi dari google maps API ini ditunjukkan pada tabel 4.8

Tabel 4.8 Implementasi Cuaca API

```

public List<HashMap<String, String>>
getRecentWeather(List<HashMap<String, String>> sensors) {
    weathers = new ArrayList<>();
    String response, temperature;
    JSONObject recent;

    List<HashMap<String, String>> weatherlist = new ArrayList<>();
    response = new Request().sendGetRequest(URL_LATEST_WEATHER);

    for(int i=0;i< Weather.sensors.size();i++) {
        try {
            recent = new
            JSONArray(response).getJSONArray(0).getJSONObject(i);

            for(int j=0;j< sensors.size(); j++) {
                if(Weather.sensors.get(i).equals(sensors.get(j))) {
                    String rain      = recent.getString("rainin");
                    String humidity = recent.getString("humidity");
                    temperature     = recent.getString("tempf");
                    String wind     = recent.getString("windspeedmph");
                    String time     = recent.getString("time");

                    rain      = rain.equals("null") ? "0" : rain;
                    humidity  = humidity.equals("null") ? "-" : humidity;
                    temperature = temperature.equals("null") ? "-" : temperature;
                    wind      = wind.equals("null") ? "-" : wind + " mil";
                    String status = classifiedWeather(Double.valueOf(rain));

                    HashMap<String, String> formatTime = timeFormat(time);
                    HashMap<String, String> attr      = new HashMap<>();
                    attr.put("id", sensors.get(j).get("id"));
                    attr.put("location", sensors.get(j).get("location"));
                    attr.put("lat", sensors.get(j).get("lat"));
                }
            }
        }
    }
}

```

```

attr.put("lng", sensors.get(j).get("lng"));
attr.put("rain", rain);
    attr.put("humidity", humidity);
attr.put("temp", temperature);
attr.put("wind", wind);
attr.put("timestamp", formatTime.get("timestamp"));
attr.put("status", status);

weatherlist.add(attr);
break;
}}} catch (final JSONException e) {
e.printStackTrace();
}
weathers = weatherlist;
return weatherlist;
}

```

4.1.4.5 Implementasi Accuweather API

Accuweather API merupakan layanan yang memberikan informasi berupa prediksi cuaca yang digunakan dalam perangkat lunak. Berikut implementasi Accuweather API ditunjukkan pada tabel 4.9

Tabel 4.9 Implementasi Accuweather API

```

public HashMap<String, String> getPredictWeather(String lat,
String lng) {
final HashMap<String, String> predict = new HashMap<>();
final HashMap<String, String> params = new HashMap<>();
params.put("apikey", ACCU_KEY);
params.put("q", lat+","+lng);
params.put("language", "id");
params.put("details", "true");

String data = new
Request().sendGetRequestParam(URL_ACCU_LOCATION, params);
try {
JSONObject jsonObject = new JSONObject(data);

```



```

String key          = jsonObject.getString("Key");
String name        = jsonObject.getString("LocalizedName");

String result = new
Request().sendGetRequestParam(URL_ACCU_PREDICT + key, params);

JSONArray jsonArray = new JSONArray(result);
jsonObject = jsonArray.getJSONObject(0);

String status      = jsonObject.getString("IconPhrase");
String rain        =
jsonObject.getJSONObject("Rain").getString("Value");
int temperature    =
jsonObject.getJSONObject("RealFeelTemperature").getInt("Value");
String temp        = String.valueOf((temperature - 32)*5/9);
String humidity    = jsonObject.getString("RelativeHumidity");
String wind        =
jsonObject.getJSONObject("Wind").getJSONObject("Speed").getStrin
g("Value");

predict.put("location", name);
predict.put("status", status);
predict.put("rain", rain);
predict.put("humidity", humidity);
predict.put("temp", temp);
predict.put("wind", wind);

} catch (JSONException e) {
e.printStackTrace();
}
return predict;
}

```

4.1.4.6 Implementasi *Firebase Cloud Messaging*

Pada penelitian ini FCM (*Firebase Cloud Messaging*) digunakan untuk memberikan notifikasi berupa informasi hujan dan laporan banjir kepada pengendara motor. Berikut implementasi dari FCM ditunjukkan pada tabel 4.10

Tabel 4.10 Implementasi Firebase Cloud Messaging

```

<service
    android:name=".Handler.Messaging"
    android:stopWithTask="false">
    <intent-filter>
        <action
            android:name="com.google.firebase.MESSAGING_EVENT" />
        </intent-filter>
    </service>

```

4.1.5 Implementasi Metode Klasifikasi Cuaca

Implementasi metode klasifikasi cuaca merupakan penerapan dari hasil analisis perhitungan algoritma yang diterapkan pada aplikasi atau sistem yang dibangun. Implementasi metode atau algoritma klasifikasi cuaca digunakan untuk menentukan status suatu cuaca. Berikut adalah implementasi metode klasifikasi cuaca ditunjukkan pada tabel 4.11

Tabel 4.11 Implementasi Metode Klasifikasi Cuaca

```

private String classifiedWeather(Double rainin) {
    if(rainin > 0.0)
        if(rainin > 1.9)
            return "Hujan Lebat";
        else return "Hujan";
    else return "Cerah";
}

```

4.1.6 Implementasi Metode Haversine Formula

Implementasi metode haversine formula merupakan penerapan dari algoritma haversine formula. Dalam penelitian ini metode ini digunakan untuk melakukan perhitungan jarak. Berikut adalah implementasi metode haversine formula ditunjukkan pada tabel 4.12

Tabel 4.12 Implementasi Metode Haversine Formula

```

public static Double calculatedDistance(Double lat1, Double
lng1, Double lat2, Double lng2) {
    Double x, y;

```

```

lat1 = Math.toRadians(lat1);
lng1 = Math.toRadians(lng1);
lat2 = Math.toRadians(lat2);
lng2 = Math.toRadians(lng2);
x = (lng2 - lng1) * Math.cos((lat1+lat2)/2);
y = lat2 - lat1;
return Math.sqrt(x*x+ y*y) * 6371;
}

```

4.1.7 Implementasi Kelas

Implementasi kelas mendeskripsikan setiap *class* yang digunakan pada aplikasi yang dibangun. Adapun implementasi kelas dalam pembangunan perangkat lunak menggunakan bahasa pemrograman Java. Berikut merupakan implementasi *class* ditunjukkan pada tabel 4.13

Tabel 4.13 Implementasi Kelas

No	Nama Kelas	Nama File
1	LoginActivity	LoginActivity.java
2	MapActivity	MapActivity.java
3	RegisterActivity	RegisterActivity.java
4	ForgetPassword	ForgetPassword.java
5	Verification	Verification.java
6	ReportActivity	ReportActivity.java
7	Map	Map.java
8	Notify	Notify.java
9	Report	Report.java
10	Route	Route.java
11	User	User.java
12	Weather	Weather.java
13	FindPlace	FindPlace.java
14	ItemClickListener	ItemClickListener.java
15	Messaging	Messaging.java
16	ReportAdapter	ReportAdapter.java
17	Request	Request.java
18	RouteAdapter	RouteAdapter.java
19	WeatherAdapter	WeatherAdapter.java

4.1.8 Implementasi Antarmuka

Implementasi antarmuka mendeskripsikan setiap halaman antarmuka dari sistem yang telah dibangun. Implementasi antarmuka dilakukan pada seluruh halaman aplikasi berdasarkan dengan perancangan antarmuka yang telah dilakukan

sebelumnya. Adapun hasil dari implementasi halaman antarmuka akan dilampirkan pada lembar lampiran. Berikut implementasi antarmuka yang telah dilakukan ditunjukkan pada tabel 4.14

Tabel 4.14 Implementasi Antarmuka

No	Halaman	Deskripsi	Nama File
1	Halaman Login	Halaman ini digunakan pengendara untuk melakukan login agar dapat menggunakan aplikasi	activity_login.xml
3	Halaman Lupa Password	Halaman ini digunakan pengendara untuk meminta informasi akun	activity_forget_password.xml
3	Halaman Daftar	Halaman ini digunakan pengendara untuk melakukan pendaftaran akun aplikasi	activity_register.xml
4	Halaman Utama	Halaman ini merupakan halaman utama dari aplikasi ketika aplikasi pertama kali dijalankan, setelah login	activity_map.xml
5	Halaman Menu Utama	Halaman ini digunakan oleh pengendara untuk melakukan navigasi menu yang ada	nav_header.xml
6	Halaman Rute Hujan dan Banjir	Halaman ini digunakan untuk menampilkan rute pengendara dan juga info terkait lokasi tujuan	info_route.xml
7	Halaman Prediksi Cuaca	Halaman ini digunakan untuk menampilkan informasi prediksi cuaca lokasi tujuan pengendara	detail_weather_predict.xml
8	Halaman Peringatan Hujan	Halaman ini digunakan untuk menampilkan informasi peringatan hujan kepada pengendara	alert_box_weather.xml
9	Halaman Rute Alternatif	Halaman ini digunakan untuk menampilkan informasi peringatan titik banjir dan juga rekomendasi jalur alternatif	alert_box_report.xml
10	Halaman Menu Informasi Cuaca	Halaman ini digunakan untuk memberikan informasi cuaca dalam bentuk list	info_weather.xml

No	Halaman	Deskripsi	Nama File
11	Halaman Detail Informasi Cuaca	Halaman ini digunakan untuk menampilkan detail informasi dan titik lokasi cuaca	deatail_weather.xml
12	Halaman Menu Informasi Banjir	Halaman ini digunakan untuk memberikan informasi banjir dalam bentuk list	info_report.xml
13	Halaman Detail Informasi Banjir	Halaman ini digunakan untuk menampilkan detail informasi dan titik lokasi banjir	deatail_report.xml
14	Halaman Pelaporan Banjir	Halaman ini digunakan oleh pengendara untuk melakukan pelaporan lokasi banjir	activitiy_report.xml

4.2 Pengujian Perangkat Lunak

Pengujian perangkat lunak dimaksudkan untuk menemukan sebuah kesalahan atau kekurangan dari sistem perangkat lunak yang telah dibangun. Sehingga dapat diketahui, apakah perangkat lunak yang dibangun telah memenuhi kriteria atau sudah sesuai dengan tujuan dari penelitian atau tidak. Pengujian yang akan dilakukan yaitu pengujian blackbox dan pengujian beta.

4.2.1 Pengujian *Blackbox*

Pengujian *Blackbox* dilakukan dengan cara menguji setiap fungsional yang terdapat didalam aplikasi yang telah dibangun. Pengujian *Blackbox* terdiri dari skenario pengujian perangkat lunak dan hasil pengujian aplikasi.

4.2.1.1 Skenario Pengujian Perangkat Lunak

Pengujian ini dilakukan pada seluruh fungsional perangkat lunak *mobile* yang telah dibangun. Pengujian akan dilakukan dengan pengujian *blackbox*. Berikut skenario pengujian sistem perangkat lunak *mobile* yang akan diuji dapat dilihat pada tabel 4.15

Tabel 4.15 Pengujian Sistem Perangkat *Mobile*

No	Kelas Uji	Poin Pengujian	Jenis Pengujian
1.	Daftar	a. Memasukan data pengguna baru b. Memasukan email yang telah terdaftar c. Memasukan data kosong d. Verifikasi akun	<i>Black Box</i>
2.	Login Pengendara	a. Memasukan data login pengendara b. Memasukan email yang belum terdaftar c. Memasukan password salah d. Memasukan data kosong	<i>Black Box</i>
3.	Lupa Password	a. Memasukan data Email yang telah terdaftar b. Memasukan data Email yang belum terdaftar c. Memasukan Email kosong	<i>Black Box</i>
4.	Menampilkan Informasi Cuaca	a. Menampilkan informasi cuaca pada halaman utama	<i>Black Box</i>
5.	Menampilkan Rute Hujan dan Banjir	a. Menampilkan rute lokasi tujuan pada map, informasi titik cuaca dan terdekat rute pengendara	<i>Black Box</i>
6.	Menampilkan Notifikasi Hujan	a. Menampilkan dialog peringatan hujan dan rekomendasi persiapan jas hujan	<i>Black Box</i>
7.	Menampilkan Rekomendasi Jalur Alternatif Banjir	a. Menampilkan pilihan rute jalur alternatif b. Memilih jalur dari rekomendasi jalur alternatif	<i>Black Box</i>
8.	Mendapatkan prediksi cuaca	a. Menampilkan informasi prediksi cuaca	<i>Black Box</i>
9.	Menampilkan Informasi Banjir	a. Menampilkan informasi titik banjir pada map	<i>Black Box</i>
10.	Melaporkan Titik Banjir	a. Mengirim laporan banjir b. Mengirim laporan banjir tanpa foto	<i>Black Box</i>

4.2.1.2 Hasil Pengujian Perangkat Lunak

1. Hasil Pengujian Daftar

Adapun hasil pengujian dari kelas uji Daftar yang telah dilakukan berikut ditunjukkan pada tabel 4.16

Tabel 4.16 Hasil Pengujian Daftar

Hasil Pengujian Data Benar			
Data Masukan	Hasil Yang Diharapkan	Hasil Pengamatan	Kesimpulan
Nama : Risco Ramdani Email : risco80@gmail.com	Sistem menyimpan data pengguna dan	Data disimpan oleh sistem dan sistem	[✓] Diterima [] Ditolak

Password : risco898 Alamat : Jl. Jati Satu Jenis Kelamin : Laki-laki	mengirimkan kode verifikasi	mengirimkan kode verifikasi	
Kode : S0QWA	Akun diaktifasi dan masuk ke halaman utama	Akun berhasil diaktifasi dan masuk ke halaman utama	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
Hasil Pengujian Data Salah			
Data Masukan	Hasil Yang Diharapkan	Hasil Pengamatan	Kesimpulan
Email : risco80@gmail.com	Tampil pesan "Email Telah Terdaftar"	Tampil pesan "Email Telah Terdaftar"	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
Nama : {Kosong}	Tampil pesan "Nama harus di lengkapi"	Tampil pesan "Nama harus di lengkapi"	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
Email : {Kosong}	Tampil pesan "Nama harus di lengkapi"	Tampil pesan "Nama harus di lengkapi"	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
Password : {Kosong}	Tampil pesan "Nama harus di lengkapi"	Tampil pesan "Nama harus di lengkapi"	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
Password : risco	Tampil pesan "Password minimal perlu 6 huruf"	Tampil pesan "Password minimal perlu 6 huruf"	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
Alamat : {Kosong}	Tampil pesan "Nama harus di lengkapi"	Tampil pesan "Nama harus di lengkapi"	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
Kode: ZZZZ	Tampil pesan "Kode Verfikasi Salah"	Tampil pesan "Kode Verfikasi Salah"	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

2. Hasil Pengujian Login

Adapun hasil pengujian dari kelas uji Login yang telah dilakukan berikut ditunjukkan pada tabel 4.17

Tabel 4.17 Hasil Pengujian Login Pengendara

Hasil Pengujian Data Benar			
Data Masukan	Hasil Yang Diharapkan	Hasil Pengamatan	Kesimpulan
Email : risco80@gmail.com Password : risco898	Berhasil login dan tampil halaman utama aplikasi.	Berhasil login dan tampil halaman utama aplikasi.	[✓] Diterima [] Ditolak
Hasil Pengujian Data Salah			
Data Masukan	Hasil Yang Diharapkan	Hasil Pengamatan	Kesimpulan
Email : sandi78@gmail.com Password : sandi898	Tampil pesan “Email Tidak Terdaftar”	Tampil pesan “Email Tidak Terdaftar”	[✓] Diterima [] Ditolak
Email : risco80@gmail.com Password : risco555	Tampil pesan “Password Anda Salah”	Tampil pesan “Password Anda Salah”	[✓] Diterima [] Ditolak
Email : {Kosong}	Tampil pesan “Email harus di lengkapi”	Tampil pesan “Email harus di lengkapi”	[✓] Diterima [] Ditolak
Password : {Kosong}	Tampil pesan “Password harus di lengkapi”	Tampil pesan “Password harus di lengkapi”	[✓] Diterima [] Ditolak

3. Hasil Pengujian Lupa Password

Adapun hasil pengujian dari kelas uji Lupa Password yang telah dilakukan berikut ditunjukkan pada tabel 4.18

Tabel 4.18 Hasil Pengujian Lupa Password

Hasil Pengujian Data Benar			
Data Masukan	Hasil Yang Diharapkan	Hasil Pengamatan	Kesimpulan
Email : risco80@gmail.com	Tampil pesan “Berhasil mengirim password ke email	Tampil pesan “Berhasil mengirim password ke email	[✓] Diterima [] Ditolak

	anda” dan email berhasil dikirimkan	anda” dan email berhasil dikirimkan	
Hasil Pengujian Data Salah			
Data Masukan	Hasil Yang Diharapkan	Hasil Pengamatan	Kesimpulan
Email : sandi78@gmail.com	Tampil pesan “Email Belum Terdaftar”	Tampil pesan “Email Belum Terdaftar”	[✓] Diterima [] Ditolak
Email : {Kosong}	Tampil pesan “Email harus di lengkapi”	Tampil pesan “Email harus di lengkapi”	[✓] Diterima [] Ditolak

4. Hasil Pengujian Menampilkan Informasi Cuaca

Adapun hasil pengujian dari kelas uji Menampilkan Informasi Cuaca yang telah dilakukan berikut ditunjukkan pada tabel 4.20

Tabel 4.19 Hasil Pengujian Menampilkan Informasi Cuaca

Hasil Pengujian Data Benar			
Data Masukan	Hasil Yang Diharapkan	Hasil Pengamatan	Kesimpulan
Lokasi: Bojongkoneng latitude : -6.885, longitude : 107.647, Lokasi: Babakan Siliwangi latitude : -6.888, longitude : 107.61, Lokasi: Balai Kota Bandung latitude: -6.911 longitude: 107.609,	Sistem menampilkan informasi cuaca pada map (sensor Babakan Siliwangi, Bojongkoneng, Balai Kota Bandung)	Sistem berhasil menampilkan informasi cuaca pada map (sensor Babakan Siliwangi, Bojongkoneng, Balai Kota Bandung)	[✓] Diterima [] Ditolak

5. Hasil Pengujian Menampilkan Rute Hujan dan Banjir

Adapun hasil pengujian dari kelas uji Menampilkan Rute Hujan dan Banjir yang telah dilakukan berikut ditunjukkan pada tabel 4.20

Tabel 4.20 Hasil Pengujian Menampilkan Rute Hujan dan Banjir

Hasil Pengujian Data Benar			
Data Masukan	Hasil Yang Diharapkan	Hasil Pengamatan	Kesimpulan
search: UNIKOM origin: -6.9366348, 107.5423954 destination: - 6.8868635000000005, 107.61530920000001 sensor: Pascal Hypersquare latitude: -6.914 longitude: 107.597 sensor:Kelurahan Sekeloa latiude: -6.883 longitude: 107.62 reportlat: -6.936635 reportlng: 107.542390	Sistem menampilkan rute lokasi tujuan pada map serta menampilkan titik cuaca dan banjir terdekat disekitar map	Sistem menampilkan rute lokasi tujuan pada map serta menampilkan titik cuaca dan banjir terdekat disekitar map	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

6. Hasil Pengujian Menampilkan Notifikasi Hujan

Adapun hasil pengujian dari kelas uji Menampilkan Notifikasi Hujan yang telah dilakukan berikut ditunjukkan pada tabel 4.21

Tabel 4.21 Hasil Pengujian Menampilkan Notifikasi Hujan

Hasil Pengujian Data Benar			
Data Masukan	Hasil Yang Diharapkan	Hasil Pengamatan	Kesimpulan
Lokasi: Pascal Hypersquare latitude: -6.914 longitude: 107.597 status: hujan userloc: -6.88, 107.628	Tampil dialog “Peringatan Hujan” dan lokasi hujan pada rute pengendara.	Sitem menampilkan “Peringatan Hujan” serta lokasi hujan pada rute pengendara.	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

7. Hasil Pengujian Menampilkan Rekomendasi Jalur Alternatif Banjir

Adapun hasil pengujian dari kelas uji Menampilkan Rekomendasi Jalur Alternatif Banjir yang telah dilakukan berikut ditunjukkan pada tabel 4.22

Tabel 4.22 Hasil Pengujian Menampilkan Rekomendasi Jalur Alternatif Banjir

Hasil Pengujian Data Benar			
Data Masukan	Hasil Yang Diharapkan	Hasil Pengamatan	Kesimpulan
Search:UNIKOM origin: -6.88, 107.628 destination: - 6.8868635000000005, 107.61530920000001 reportlat: -6.936635 reportlng: 107.542390 routelat: -6.93666 routelng: 107.54245 Alternatives: true	Tampil dialog peringatan banjir pada aplikasi dan Tampil pilihan jalur alternatif pada map	Sistem berhasil menampilkan dialog peringatan banjir Sistem berhasil menampilkan pilihan jalur alternatif pada map	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
origin: -6.88, 107.628 destination: - 6.8868635000000005, 107.61530920000001 position : 1 routes:1	Tampil jalur alternatif yang dipilih pada map	Sistem berhasil menampilkan jalur alternatif yang dipilih pada map	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

8. Hasil Pengujian Mendapatkan Informasi Prediksi Cuaca

Adapun hasil pengujian dari kelas uji Mendapatkan Informasi Prediksi Cuaca yang telah dilakukan berikut ditunjukkan pada tabel 4.23

Tabel 4.23 Mendapatkan Informasi Prediksi Cuaca

Hasil Pengujian Data Benar			
Data Masukan	Hasil Yang Diharapkan	Hasil Pengamatan	Kesimpulan

Lokasi:UNIKOM lat: -6.8868635000000005 lng: 107.61530920000001 key: 1855673	Sistem menampilkan informasi dialog prediksi cuaca	Sistem berhasil menampilkan informasi dialog prediksi cuaca	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
--	---	--	--

9. Hasil Pengujian Menampilkan Informasi Banjir Terkini

Adapun hasil pengujian dari kelas uji Menampilkan Informasi Banjir Terkini yang telah dilakukan berikut ditunjukkan pada tabel 4.24

Tabel 4.24 Menampilkan Informasi Banjir Terkini

Hasil Pengujian Data Benar			
Data Masukan	Hasil Yang Diharapkan	Hasil Pengamatan	Kesimpulan
latitude: -6.93666 longitude: 107.54245	Sistem menampilkan info banjir pada map	Sistem berhasil menampilkan info banjir pada map	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

10. Hasil Pengujian Melaporkan Titik Banjir

Adapun hasil pengujian dari kelas uji Melaporkan Titik Banjir yang telah dilakukan berikut ditunjukkan pada tabel 4.25

Tabel 4.25 Hasil Pengujian Melaporkan Titik Banjir

Hasil Pengujian Data Benar			
Data Masukan	Hasil Yang Diharapkan	Hasil Pengamatan	Kesimpulan
id_user: 107 kota: Kecamatan Margahayu alamat: Jl. Nata Endah Blok D No.10 lat: -6.9613446 lng: 107.5740554 foto: 1563181290192.jpg	Sistem mengirimkan laporan banjir dan tampil pesan sukses	Sistem berhasil mengirimkan laporan banjir dan tampil pesan sukses	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
Hasil Pengujian Data Salah			
Data Masukan	Hasil Yang Diharapkan	Hasil Pengamatan	Kesimpulan

id_user: 107 kota: Kecamatan Margahayu alamat: Jl. Nata Endah Blok D No.10 lat: -6.9613446 lng: 107.5740554 foto: {kosong}	Tampil pesan “File foto harus di lengkapi”	Tampil pesan “File foto harus di lengkapi”	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
---	--	--	--

4.2.1.3 Kesimpulan Pengujian *Blackbox*

Berdasarkan hasil pengujian *Blackbox* yang telah dilakukan pada kasus uji maka dapat disimpulkan bahwa dalam Aplikasi Peringatan Hujan dan Banjir ini tidak terdapat kesalahan proses dan fungsional serta dapat berjalan sesuai dengan yang diharapkan. Sehingga dapat di simpulkan bahwa aplikasi yang dibangun telah dapat digunakan dengan baik.

4.2.2 Pengujian Beta

Pengujian beta merupakan pengujian yang bersifat objektif yang mana pengujian ini dilakukan kepada pengguna secara langsung, hal ini bertujuan untuk mengetahui sejauh mana sistem dapat berjalan sesuai perancangan yang telah dilakukan. Adapun teknik yang digunakan dalam pengujian beta dilakukan dengan memberikan kuesioner kepada calon pengguna.

4.2.2.1 Skenario Pengujian Beta

Pengujian beta yang dilakukan dalam penelitian ini menggunakan teknik kuesioner untuk dapat menarik kesimpulan. 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 dapat dilihat pada tabel 4.26

Tabel 4.26 Skala Jawaban Kuisisioner

Inisial	Skala	Skor
SS	Sangat Setuju	5
S	Setuju	4
R	Ragu - ragu	3
TS	Tidak Setuju	2
STS	Sangat Tidak Setuju	1

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

$$Kriterium = Skor Penilaian \times Jumlah Responden$$

Sehingga didapatkan skor ideal dari setiap skala yang ditunjukkan pada tabel 4.27 sebagai berikut

Tabel 4.27 Skor Kriterium

Skala	Skor Kriterium
Sangat Setuju	5 x 30 = 150
Setuju	4 x 30 = 120
Ragu - ragu	3 x 30 = 90
Tidak Setuju	2 x 30 = 60
Sangat Tidak Setuju	1 x 30 = 30

Dalam mencari nilai persentase dari masing-masing jawaban kuesioner digunakan rumus skala likert sebagai berikut.

$$P = \frac{f}{n} \times 100 \%$$

Keterangan :

P = Nilai persentase yang dicari

f = Jumlah frekuensi dikalikan dengan nilai yang ditetapkan jawaban

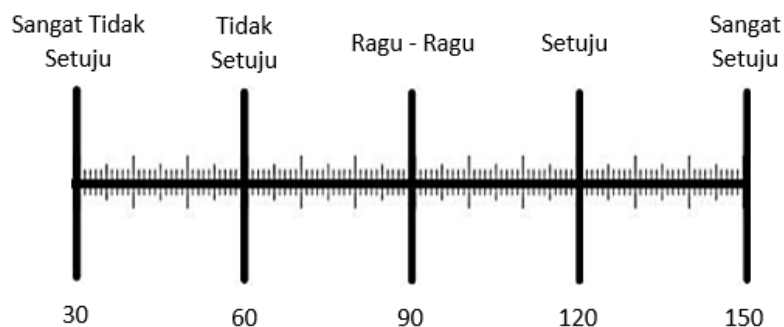
n = Nilai tertinggi dikalikan dengan jumlah sampel

Penerapan hasil perhitungan yang didapat selanjutnya dimasukkan ke dalam bentuk interpretasi skor berikut ditunjukkan pada tabel 4.28

Tabel 4.28 Rating Scale

Skala	Skor Kriteria	Rating Scale
Sangat Setuju	150	121 – 150
Setuju	120	91 – 120
Ragu - ragu	90	61 – 90
Tidak Setuju	60	31 – 60
Sangat Tidak Setuju	30	30

Berikut ilustrasi interval dari rating scale yang telah ditentukan pada gambar 4.1

**Gambar 4.1 Interval Rating Scale**

Adapun daftar pertanyaan kuesioner yang diajukan kepada pengendara sebagai pengguna aplikasi ditunjukkan pada Tabel 4.28

Tabel 4.29 Daftar Pertanyaan Kuisisioner

No	Pertanyaan
1.	Aplikasi Rute Hujan dapat membantu pengguna dalam mendapatkan informasi cuaca berdasarkan rute perjalanan ?
2.	Aplikasi Rute Hujan dapat membantu pengguna dalam mengetahui lokasi – lokasi yang sedang hujan saat berkendara ?
3.	Aplikasi Rute Hujan dapat membantu pengguna dalam mempersiapkan diri akan cuaca hujan saat berkendara ?
4.	Aplikasi Rute Hujan dapat memberikan informasi banjir yang ada, baik lokasi maupun foto ?
5.	Aplikasi Rute Hujan dapat memudahkan pengguna dalam membagikan lokasi banjir ?
6.	Aplikasi Rute Hujan dapat membantu pengguna dalam menghadapi banjir dengan adanya jalur alternatif ?
7.	Aplikasi Rute Hujan mudah untuk digunakan oleh pengguna ?

4.2.2.2 Perhitungan Hasil Kuisisioner

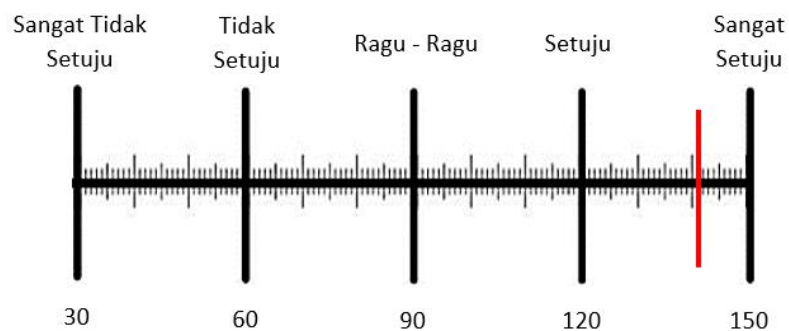
Dengan menggunakan rumus yang telah dijelaskan diatas, adapun hasil perhitungn kuesioner yang telah diajukan kepada 30 orang responden dengan terdapat 7 pertanyaan yang diberikan. Berikut hasil dari setiap pertanyaan yang diajukan kepada para pengendara motor :

1. Aplikasi Rute Hujan dapat membantu pengguna dalam mendapatkan informasi cuaca berdasarkan rute perjalanan ?

Tabel 4.30 Hasil Perhitungan Kuisisioner Pertanyaan 1

Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor	Nilai Persentase	Keputusan
Sangat Setuju	5	21	105	$(141 / (30 \times 5)) \times 100\% = 94\%$	Sangat Setuju
Setuju	4	9	36		
Ragu - Ragu	3	0	3		
Tidak Setuju	2	0	0		
Sangat Tidak Setuju	1	0	0		
Jumlah		30	141		

Berdasarkan perhitungan diatas, didapat total skor yang diperoleh adalah 141 dengan nilai persentasenya menghasilkan nilai 94% sehingga skala kategori masuk pada kategori sangat setuju. Maka dapat disimpulkan bahwa aplikasi yang dibangun telah dapat memberikan informasi cuaca berdasarkan rute kepada pengendara motor dengan baik. Berikut *interval rating scale* untuk hasil kuisisioner pertanyaan diatas ditunjukkan pada gambar 4.2



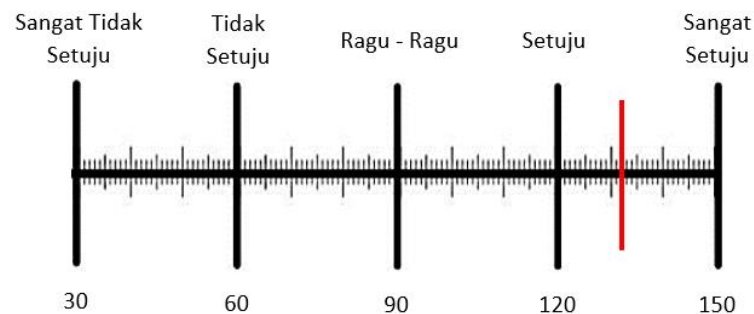
Gambar 4.2 Interval Rating Scale Petanyaan 1

2. Aplikasi Rute Hujan dapat membantu pengguna dalam mengetahui lokasi – lokasi yang sedang hujan saat berkendara ?

Tabel 4.31 Hasil Perhitungan Kuisisioner Pertanyaan 2

Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor	Nilai Persentase	Keputusan
Sangat Setuju	5	12	60	$(132 / (30 \times 5)) \times 100\% = 88\%$	Sangat Setuju
Setuju	4	18	72		
Ragu -Ragu	3	0	0		
Tidak Setuju	2	0	0		
Sangat Tidak Setuju	1	0	0		
Jumlah		30	132		

Berdasarkan perhitungan diatas, didapat total skor yang diperoleh adalah 132 dengan nilai persentasenya menghasilkan nilai 88% sehingga skala kategori masuk pada kategori sangat setuju. Maka dapat disimpulkan bahwa aplikasi yang dibangun telah dapat membantu pengguna dalam memberikan informasi lokasi – lokasi yang sedang hujan. Berikut *interval rating scale* untuk hasil kuisisioner pertanyaan diatas ditunjukkan pada gambar 4.3



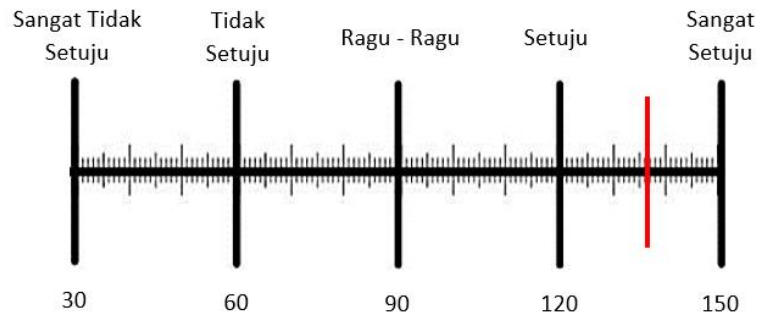
Gambar 4.3 Interval Rating Scale Petanyaan 2

3. Aplikasi Rute Hujan dapat membantu pengguna dalam mempersiapkan diri akan cuaca hujan saat berkendara ?

Tabel 4.32 Hasil Perhitungan Kuisisioner Pertanyaan 3

Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor	Nilai Persentase	Keputusan
Sangat Setuju	5	16	80	$(136 / (30 \times 5)) \times 100\% = 90,66\%$	Sangat Setuju
Setuju	4	14	56		
Ragu - Ragu	3	0	0		
Tidak Setuju	2	0	0		
Sangat Tidak Setuju	1	0	0		
Jumlah		30	136		

Berdasarkan perhitungan diatas, didapat total skor yang diperoleh adalah 136 dengan nilai persentasenya menghasilkan nilai 90,66% sehingga skala kategori masuk pada kategori sangat setuju. Maka dapat disimpulkan bahwa aplikasi yang dibangun telah dapat membantu pengguna dalam mempersiapkan diri akan cuaca hujan saat berkendara Berikut *interval rating scale* untuk hasil kuisisioner pertanyaan diatas ditunjukkan pada gambar 4.4

**Gambar 4.4 Interval Rating Scale Petanyaan 3**

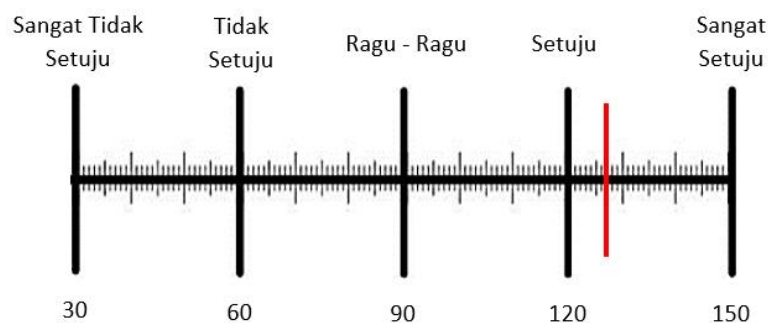
4. Aplikasi Rute Hujan dapat memberikan informasi banjir yang ada, baik lokasi maupun foto ?

Tabel 4.33 Hasil Perhitungan Kuisisioner Pertanyaan 4

Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor	Nilai Persentase	Keputusan
Sangat Setuju	5	12	60	$(127 / (30 \times 5)) \times 100\% = 84,66\%$	Sangat Setuju
Setuju	4	13	52		
Ragu - Ragu	3	5	15		
Tidak Setuju	2	0	0		

Sangat Tidak Setuju	1	0	0		
Jumlah		30	127		

Berdasarkan perhitungan diatas, didapat total skor yang diperoleh adalah 127 dengan nilai persentasenya menghasilkan nilai 84,66% sehingga skala kategori masuk pada kategori sangat setuju. Maka dapat disimpulkan bahwa aplikasi yang dibangun telah dapat membantu pengguna dalam mendapatkan informasi banjir yang ada. Berikut *interval rating scale* untuk hasil kuisisioner pertanyaan diatas ditunjukkan pada gambar 4.5



Gambar 4.5 Interval Rating Scale Petanyaan 4

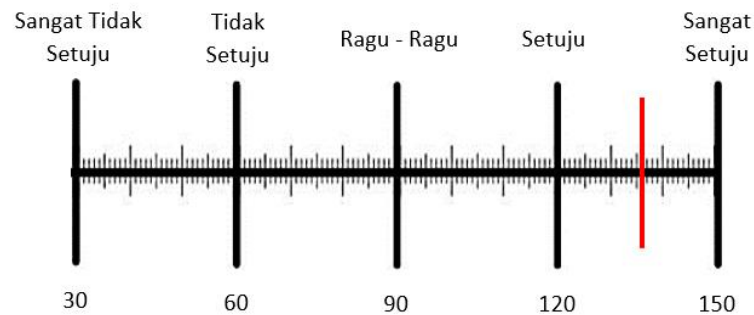
5. Aplikasi Rute Hujan dapat memudahkan pengguna dalam membagikan lokasi banjir ?

Tabel 4.34 Hasil Perhitungan Kuisisioner Pertanyaan 5

Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor	Nilai Persentase	Keputusan
Sangat Setuju	5	18	90	$(136 / (30 \times 5)) \times 100\% = 90,66\%$	Sangat Setuju
Setuju	4	10	40		
Ragu - Ragu	3	2	6		
Tidak Setuju	2	0	0		
Sangat Tidak Setuju	1	0	0		
Jumlah		30	136		

Berdasarkan perhitungan diatas, didapat total skor yang diperoleh adalah 136 dengan nilai persentasenya menghasilkan nilai 90,66% sehingga skala kategori masuk pada kategori sangat setuju. Maka dapat disimpulkan bahwa aplikasi yang dibangun telah dapat memudahkan pengguna dalam membagikan

informasi banjir. Berikut *interval rating scale* untuk hasil kuisisioner pertanyaan diatas ditunjukkan pada gambar 4.6



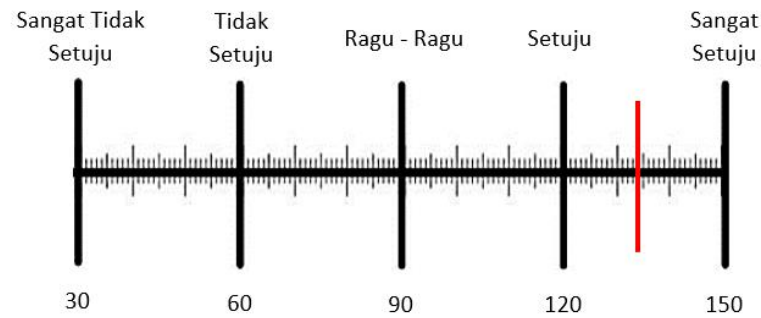
Gambar 4.6 Interval Rating Scale Petanyaan 5

6. Aplikasi Rute Hujan dapat membantu pengguna dalam menghadapi banjir dengan adanya jalur alternatif ?

Tabel 4.35 Hasil Perhitungan Kuisisioner Pertanyaan 6

Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor	Nilai Persentase	Keputusan
Sangat Setuju	5	16	80	$(134 / (30 \times 5)) \times 100\% = 89,33\%$	Sangat Setuju
Setuju	4	12	48		
Ragu - Ragu	3	2	6		
Tidak Setuju	2	0	0		
Sangat Tidak Setuju	1	0	0		
Jumlah		30	134		

Berdasarkan perhitungan diatas, didapat total skor yang diperoleh adalah 134 dengan nilai persentasenya menghasilkan nilai 89,33% sehingga skala kategori masuk pada kategori sangat setuju. Maka dapat disimpulkan bahwa dengan adanya jalur alternatif pada aplikasi yang dibangun telah membantu pengguna dalam menghadapi banjir. Berikut *interval rating scale* untuk hasil kuisisioner pertanyaan diatas ditunjukkan pada gambar 4.7



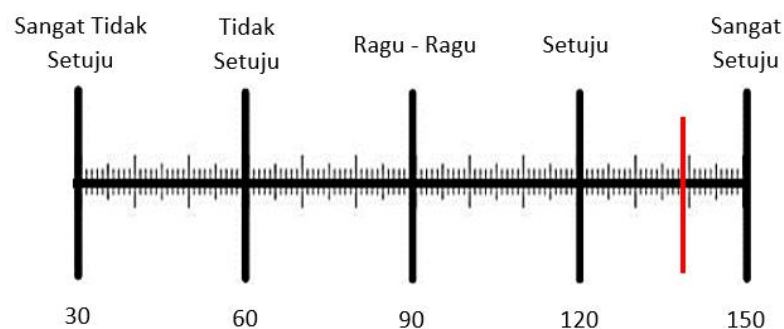
Gambar 4.7 Interval Rating Scale Petanyaan 6

7. Aplikasi Rute Hujan mudah untuk digunakan oleh pengguna ?

Tabel 4.36 Hasil Perhitungan Kuisisioner Pertanyaan 7

Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor	Nilai Persentase	Keputusan
Sangat Setuju	5	19	95	$(139 / (30 \times 5)) \times 100\% = 92,66\%$	Sangat Setuju
Setuju	4	11	44		
Ragu - Ragu	3	0	0		
Tidak Setuju	2	0	0		
Sangat Tidak Setuju	1	0	0		
Jumlah		30	139		

Berdasarkan perhitungan diatas, didapat total skor yang diperoleh adalah 139 dengan nilai persentasenya menghasilkan nilai 92,66% sehingga skala kategori masuk pada kategori sangat setuju. Maka dapat disimpulkan bahwa aplikasi yang dibangun mudah digunakan oleh pengguna. Berikut *interval rating scale* untuk hasil kuisisioner pertanyaan diatas ditunjukkan pada gambar 4.8



Gambar 4.8 Interval Rating Scale Petanyaan 7

4.2.2.3 Kesimpulan Pengujian Beta

Berdasarkan hasil pengujian Beta yang telah dilakukan kepada 30 responden dapat dilihat bahwa poin yang diperoleh cukup tinggi, mayoritas responden menyatakan sangat setuju dari setiap pertanyaan yang diajukan. Sehingga dapat disimpulkan bahwa Aplikasi Peringatan Hujan dan Banjir ini telah dapat digunakan oleh pengendara motor dengan baik dalam mengatasi masalah – masalah yang ada berupa cuaca hujan dan banjir.