

## **BAB 3**

### **ANALISIS DAN PERANCANGAN**

#### **3.1 Analisis Sistem**

Analisis sistem dilakukan untuk mengetahui permasalahan – permasalahan yang terdapat pada sistem serta menentukan kebutuhan apa saja yang diperlukan oleh sistem yang akan dibangun. Analisis sistem yang akan dilakukan yaitu meliputi analisis masalah, analisis sistem yang akan dibangun, analisis arsitektur sistem, analisis teknologi yang digunakan, analisis spesifikasi kebutuhan perangkat lunak, analisis kebutuhan Non-fungsional dan kebutuhan fungsional.

##### **3.1.1 Analisis Masalah**

Analisis masalah merupakan uraian dari permasalahan – permasalahan yang ditemukan pada sebuah penelitian. Analisis masalah diperlukan dalam perancangan sistem dengan tujuan untuk mengidentifikasi permasalahan yang ada sehingga pembangunan sistem yang dilakukan menjadi lebih terarah dan tepat sasaran. Analisis masalah dilakukan pada tahap awal perancangan sistem yang mana akan menjadi landasan bagaimana sistem akan dibangun.

Pada penelitian yang akan dilakukan berkaitan dengan kendala yang dialami oleh pengendara bermotor, khususnya sepeda motor saat berkendara pada saat kondisi hujan dan banjir. Pengguna kendaraan sepeda motor tidak mengetahui kondisi cuaca daerah tujuannya maupun daerah sepanjang perjalanannya khususnya cuaca hujan. Permasalahan pun dialami oleh beberapa pengendara seperti kehujanan maupun sulitnya mengenakan jas hujan ketika hujan turun. Selain itu masalah yang juga dialami bagi pengendara motor adalah masalah jalanan banjir. Jalan yang banjir dapat mengganggu kelancaran lalu lintas pengendara, selain itu genangan air banjir yang cukup tinggi dapat menyebabkan air masuk ke mesin kendaraan sehingga kendaraan menjadi mogok.

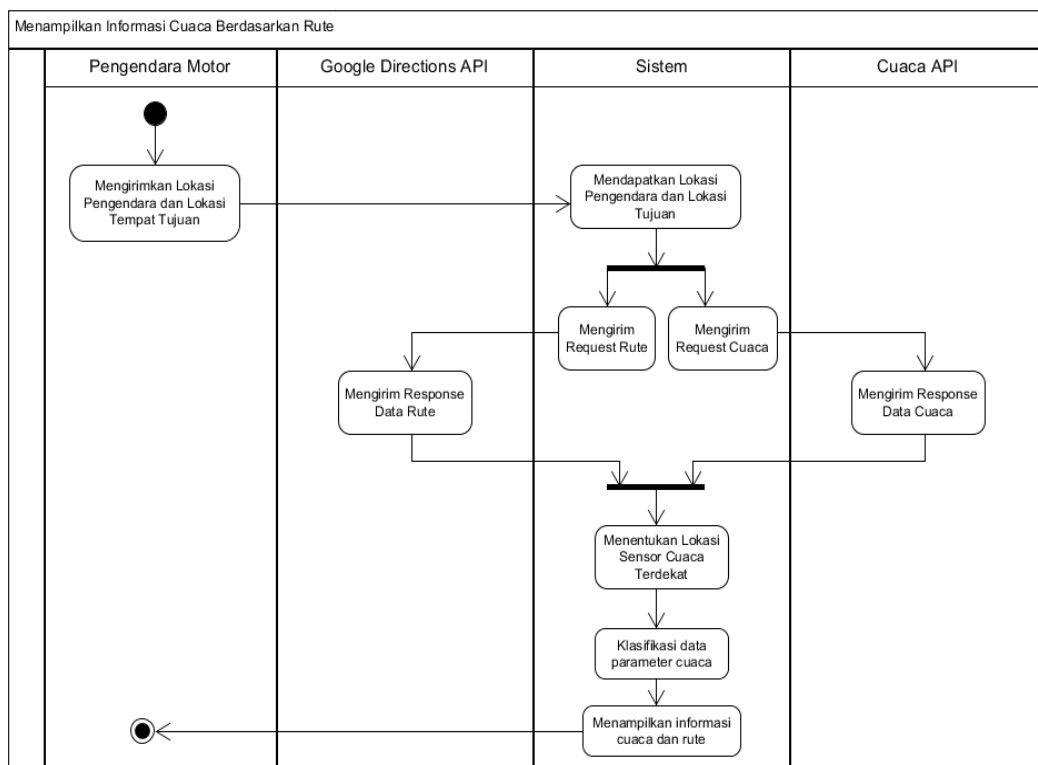
Permasalahan – permasalahan tersebut terjadi disebabkan karena kurangnya informasi yang didapat oleh pengendara sepeda motor tentang kondisi suatu daerah terkait masalah hujan dan banjir. Oleh karena itu diperlukan suatu aplikasi yang

dapat memberikan informasi terkait hujan dan banjir bagi pengendara motor sehingga diharapkan pengguna sepeda motor dapat mempersiapkan diri dalam menghadapi kondisi hujan dan banjir saat berkendara. Adapun data informasi cuaca yang didapat pada penelitian ini yaitu berdasarkan informasi pemantau (sensor) cuaca yang telah dipasang oleh Pemerintah Kota Bandung.

### 3.1.2 Analisis Sistem Yang Akan Dibangun

Analisis sistem yang akan dibangun merupakan analisis yang dilakukan untuk memberikan gambaran umum terkait dengan pembangunan sistem yang akan dibangun. Analisis dilakukan dengan mengacu kepada analisis masalah yang telah dilakukan sebelumnya. Berikut analisis sistem yang akan dibangun yaitu :

#### a) Prosedur Menampilkan Informasi Cuaca Berdasarkan Rute Perjalanan



**Gambar 3.1 Prosedur Menampilkan Informasi Cuaca Berdasarkan Rute**

Berikut penjelasan prosedur yang terdapat pada gambar 3.1 :

1. Pengendara motor mengirimkan lokasi tempat tujuan pengendara serta lokasi pengendara berada ke sistem

2. Selanjutnya sistem akan mengirimkan *request* data parameter cuaca ke Cuaca API untuk mendapatkan informasi parameter sensor cuaca serta mengirimkan *request* rute berdasarkan lokasi asal dan tujuan pengendara ke *Google Direction* API untuk mendapatkan jalur rute lokasi
3. Setelah sistem mendapatkan informasi jalur rute dan informasi cuaca, maka sistem akan menentukan sensor cuaca terdekat berdasarkan jalur rute lokasi pengendara motor
4. Proses selanjutnya, sistem akan melakukan klasifikasi cuaca dari informasi parameter cuaca yang telah didapatkan untuk mendapatkan informasi cuaca
5. Setelah itu informasi akan dikirim kepada pengendara motor berupa informasi cuaca dan jalur rute lokasi

#### 3.1.2.1 Analisis Klasifikasi Cuaca Menggunakan Algoritma C4.5

Pada proses pembangunan sistem digunakan metode algoritma C4.5 untuk melakukan klasifikasi cuaca. Pada penelitian yang telah dilakukan sebelumnya, didapat bahwa parameter cuaca yang paling menentukan kondisi cuaca yaitu *rainin* (curah hujan), *humidity* (kelembaban), *temperature* (suhu) dan *cloud* (awan). Pada penelitian ini hanya akan digunakan tiga parameter yaitu *rainin*, *humidity*, dan *temperature* untuk menyesuaikan dengan sensor cuaca yang ada.

Adapun dalam penelitian ini juga hanya akan dilakukan klasifikasi cuaca dengan status *sunny* (cerah), *light rain* (hujan ringan), dan *heavy rain* (hujan lebat). Beberapa tahapan yang dilakukan dalam melakukan klasifikasi cuaca menggunakan algoritma C4.5 meliputi tahapan *Pre Process* berisi tentang persiapan dataset cuaca, tahapan *Process* berisi tentang tahapan penentuan aturan pohon keputusan (*decission tree*), serta tahapan *Post Process* berisi tentang hasil keluaran yang didapat dari proses sebelumnya dan dilakukan pengujian terhadap tingkat akurasi pohon keputusan yang didapatkan.

##### 1. Pre-Process

Pada tahapan ini dilakukan pengumpulan data berupa dataset cuaca yang akan digunakan sebagai data *training* algoritma C4.5. Adapun dataset yang digunakan dalam penelitian ini yaitu menggunakan data histori cuaca wilayah Kota Bandung

yang diperoleh dari situs *World Weather Online* pada pada bulan November 2018. Berikut dataset cuaca yang telah diperoleh ditunjukkan pada tabel 3.1

**Tabel 3.1 Dataset Cuaca Bulan November 2018**

Tanggal	Waktu	Rainin	Humidity	Temperature	Status
01/11/2018	06:00	0.0	70	82	Sunny
03/11/2018	06:00	0.0	69	82	Sunny
04/11/2018	03:00	1.5	81	79	Light rain shower
06/11/2018	15:00	1.6	63	87	Light rain shower
07/11/2018	03:00	2.7	81	78	Moderate or heavy rain shower
08/11/2018	21:00	0.3	78	79	Light rain shower
11/11/2018	15:00	5.3	72	84	Moderate or heavy rain shower
14/11/2018	09:00	1.9	72	83	Light rain shower
21/11/2018	06:00	0.0	70	82	Sunny
22/11/2018	15:00	2.6	63	86	Moderate or heavy rain shower

Adapun status cuaca yang ditentukan dari data yang didapat sebagai data latih proses klasifikasi pada tabel 3.1 terbagi menjadi tiga yaitu *Sunny*, *Light Rain Shower*, dan *Moderate or Heavy Rain*. Berikut penjelasan dari status cuaca yang terdapat pada dataset cuaca yaitu sebagai berikut

1. Sunny : Menunjukkan kondisi cuaca cerah
2. Light Rain Shower : Menunjukkan kondisi cuaca berupa hujan ringan maupun hujan sedang
3. Moderate or Heavy Rain : Menunjukkan kondisi cuaca berupa hujan lebat

Pada dataset ini digunakan beberapa nilai fitur dari informasi yang terdapat pada tabel 3.1 yaitu curah hujan (*rainin*), informasi kelembaban (*humidity*) dan temperatur (*temperature*). Nilai fitur yang didapatkan menggunakan bertipe numerik.

## 2. Process

Pada proses klasifikasi algoritma C4.5 dilakukan untuk membangun model klasifikasi yang berupa pohon keputusan secara *top-down* (atas ke bawah). Dengan melakukan evaluasi terhadap semua atribut menggunakan ukuran statistik, berupa nilai *gain ratio*, untuk mengukur efektifitas dari suatu atribut dalam melakukan

klasifikasi sekumpulan himpunan data. Adapun tahapan dalam proses algoritma C4.5 meliputi pencarian nilai *entropy* pada atribut selanjutnya menghitung nilai *information gain*. Sebagaimana pada pembahasan bab sebelumnya dalam menghitung nilai *Entropy* dapat dihitung dengan persamaan 1 yang ada pada pembahasan sebelumnya. Kemudian dalam dilakukan perhitungan nilai *gain* dapat dihitung dengan permasamaan 2 yang ada pada pembahasan sebelumnya.

Nilai *gain* terbesar yang didapat akan digunakan sebagai node akar pohon keputusan. Pada dataset cuaca pada tabel 3.1 terdapat 3 kasus, yaitu: Status ( $S_1$ ) = 'Light Rain Shower', Status ( $S_2$ ) = 'Moderate or Heavy Rain', atau Status ( $S_3$ ) = 'Sunny'. Berikut perhitungan yang telah dilakukan dibawah ini

### Proses untuk node ke-1

Perhitungan nilai *entropy* total

$$\begin{aligned} \text{Total} &= 10 \text{ kasus, dengan nilai } P_1 = 4/10, P_2 = 3/10, \text{ dan } P_3 = 3/10 \\ \text{Entropy}(S) &= - (4/10) \log_2 (4/10) - (3/10) \log_2 (3/10) - (3/10) \log_2 (3/10) \\ &= 1.57095 \end{aligned}$$

Perhitungan pada fitur *rainin* dengan nilai = 0.0

$$\begin{aligned} S_{\leq 0.0} &= 3 \text{ kasus, dengan nilai } P_1 = 0/3, P_2 = 0/3, \text{ dan } P_3 = 3/3 \\ \text{Entropy}(S_{\leq 0.0}) &= - (0/3) \log_2 (0/3) - (0/3) \log_2 (0/3) - (3/3) \log_2 (3/3) \\ &= 0.0 \\ S_{> 0.0} &= 7 \text{ kasus, dengan nilai } P_1 = 4/7, P_2 = 3/7, \text{ dan } P_3 = 0/7 \\ \text{Entropy}(S_{> 0.0}) &= - (4/7) \log_2 (4/7) - (3/7) \log_2 (3/7) - (0/7) \log_2 (0/7) \\ &= 0.98522 \\ \text{Gain}(S, \text{Rainin}) &= 1.57095 - (3/10) \text{Entropy}(S_{\leq 0.0}) - (7/10) \text{Entropy}(S_{> 0.0}) \\ &= 0.88129 \end{aligned}$$

Perhitungan pada fitur *rainin* dengan nilai = 0.3

$$\begin{aligned} S_{\leq 0.3} &= 4 \text{ kasus, dengan nilai } P_1 = 1/4, P_2 = 0/4, \text{ dan } P_3 = 3/4 \\ \text{Entropy}(S_{\leq 0.3}) &= - (1/4) \log_2 (1/4) - (0/4) \log_2 (0/4) - (3/4) \log_2 (3/4) \\ &= 0.81127 \\ S_{> 0.3} &= 6 \text{ kasus, dengan nilai } P_1 = 3/6, P_2 = 3/6, \text{ dan } P_3 = 0/6 \\ \text{Entropy}(S_{> 0.3}) &= - (3/6) \log_2 (3/6) - (3/6) \log_2 (3/6) - (0/6) \log_2 (0/6) \end{aligned}$$

$$\begin{aligned}
 &= 1.0 \\
 \text{Gain}(S, \text{Rainin}) &= 1.57095 - (4/10) \text{Entropy}(S_{\leq 0.3}) - (6/10) \text{Entropy}(S_{> 0.3}) \\
 &= 0.64643
 \end{aligned}$$

Perhitungan pada fitur *rainin* dengan nilai = 1.5

$$\begin{aligned}
 S_{\leq 1.5} &= 5 \text{ kasus, dengan nilai } P_1 = 2/5, P_2 = 0/5, \text{ dan } P_3 = 3/5 \\
 \text{Entropy}(S_{\leq 1.5}) &= -(2/5) \log_2(2/5) - (0/5) \log_2(0/5) - (3/5) \log_2(3/5) \\
 &= 0.97095 \\
 S_{> 1.5} &= 5 \text{ kasus, dengan nilai } P_1 = 2/5, P_2 = 3/5, \text{ dan } P_3 = 0/5 \\
 \text{Entropy}(S_{> 1.5}) &= -(2/5) \log_2(2/5) - (3/5) \log_2(3/5) - (0/5) \log_2(0/5) \\
 &= 0.97095 \\
 \text{Gain}(S, \text{Rainin}) &= 1.57095 - (5/10) \text{Entropy}(S_{\leq 1.5}) - (5/10) \text{Entropy}(S_{> 1.5}) \\
 &= 0.59999
 \end{aligned}$$

Perhitungan pada fitur *rainin* dengan nilai = 1.6

$$\begin{aligned}
 S_{\leq 1.6} &= 6 \text{ kasus, dengan nilai } P_1 = 3/6, P_2 = 0/6, \text{ dan } P_3 = 3/6 \\
 \text{Entropy}(S_{\leq 1.6}) &= -(3/6) \log_2(3/6) - (0/6) \log_2(0/6) - (3/6) \log_2(3/6) \\
 &= 1.0 \\
 S_{> 1.6} &= 4 \text{ kasus, dengan nilai } P_1 = 1/4, P_2 = 3/4, \text{ dan } P_3 = 0/4 \\
 \text{Entropy}(S_{> 1.6}) &= -(1/4) \log_2(1/4) - (3/4) \log_2(3/4) - (0/4) \log_2(0/4) \\
 &= 0.81127 \\
 \text{Gain}(S, \text{Rainin}) &= 1.57095 - (6/10) \text{Entropy}(S_{\leq 1.6}) - (4/10) \text{Entropy}(S_{> 1.6}) \\
 &= 0.64643
 \end{aligned}$$

Perhitungan pada fitur *rainin* dengan nilai = 1.9

$$\begin{aligned}
 S_{\leq 1.9} &= 7 \text{ kasus, dengan nilai } P_1 = 4/7, P_2 = 3/7, \text{ dan } P_3 = 0/7 \\
 \text{Entropy}(S_{\leq 1.9}) &= -(4/7) \log_2(4/7) - (3/7) \log_2(3/7) - (0/7) \log_2(0/7) \\
 &= 0.985228 \\
 S_{> 1.9} &= 3 \text{ kasus, dengan nilai } P_1 = 0/3, P_2 = 3/3, \text{ dan } P_3 = 0/3 \\
 \text{Entropy}(S_{> 1.9}) &= -(0/3) \log_2(0/3) - (3/3) \log_2(3/3) - (0/3) \log_2(0/3) \\
 &= 0.0 \\
 \text{Gain}(S, \text{Rainin}) &= 1.57095 - (7/10) \text{Entropy}(S_{\leq 1.9}) - (3/10) \text{Entropy}(S_{> 1.9}) \\
 &= 0.88129
 \end{aligned}$$

Perhitungan pada fitur *rainin* dengan nilai = 2.6

$$\begin{aligned}
S_{\leq 2.6} &= 8 \text{ kasus, dengan nilai } P_1 = 4/8, P_2 = 1/8, \text{ dan } P_3 = 3/8 \\
Entropy(S_{\leq 2.6}) &= -(4/8) \log_2(4/8) - (1/8) \log_2(1/8) - (3/8) \log_2(3/8) \\
&= 1.40563 \\
S_{> 2.6} &= 2 \text{ kasus, dengan nilai } P_1 = 0/2, P_2 = 2/2, \text{ dan } P_3 = 0/2 \\
Entropy(S_{> 2.6}) &= -(0/2) \log_2(0/2) - (2/2) \log_2(2/2) - (0/2) \log_2(0/2) \\
&= 0.0 \\
Gain(S, Rainin) &= 1.57095 - (8/10) Entropy(S_{\leq 2.6}) - (2/10) Entropy(S_{> 2.6}) \\
&= 0.446439
\end{aligned}$$

Nilai *gain* tertinggi didapatkan pada posisi *rainin* dengan nilai 0.0 yaitu sebesar 0.88129. Oleh karena itu dilakukan pemecahan pada atribut *rainin* dengan nilai 0.0. Selanjutnya dilakukan perhitungan terhadap semua nilai yang terdapat pada fitur *humidity* sebagai berikut

Perhitungan pada fitur *humidity* dengan nilai = 63

$$\begin{aligned}
S_{\leq 63} &= 2 \text{ kasus, dengan nilai } P_1 = 1/2, P_2 = 1/2, \text{ dan } P_3 = 0/2 \\
Entropy(S_{\leq 63}) &= -(1/2) \log_2(1/2) - (1/2) \log_2(1/2) - (0/2) \log_2(0/2) \\
&= 1.0 \\
S_{> 63} &= 8 \text{ kasus, dengan nilai } P_1 = 3/8, P_2 = 2/8, \text{ dan } P_3 = 3/8 \\
Entropy(S_{> 63}) &= -(3/8) \log_2(3/8) - (2/8) \log_2(2/8) - (3/8) \log_2(3/8) \\
&= 1.56127 \\
Gain(S, Humidity) &= 1.57095 - (2/10) Entropy(S_{\leq 63}) - (8/10) Entropy(S_{> 63}) \\
&= 0.12192
\end{aligned}$$

Perhitungan pada fitur *humidity* dengan nilai = 69

$$\begin{aligned}
S_{\leq 69} &= 3 \text{ kasus, dengan nilai } P_1 = 1/3, P_2 = 1/3, \text{ dan } P_3 = 1/3 \\
Entropy(S_{\leq 69}) &= -(1/3) \log_2(1/3) - (1/3) \log_2(1/3) - (1/3) \log_2(1/3) \\
&= 1.58496 \\
S_{> 69} &= 7 \text{ kasus, dengan nilai } P_1 = 3/7, P_2 = 2/7, \text{ dan } P_3 = 2/7 \\
Entropy(S_{> 69}) &= -(3/7) \log_2(3/7) - (2/7) \log_2(2/7) - (2/7) \log_2(2/7) \\
&= 1.55665 \\
Gain(S, Humidity) &= 1.57095 - (3/10) Entropy(S_{\leq 69}) - (7/10) Entropy(S_{> 69}) \\
&= 0.00580
\end{aligned}$$

Perhitungan pada fitur *humidity* dengan nilai = 70

$$\begin{aligned}
S_{\leq 70} &= 5 \text{ kasus, dengan nilai } P_1 = 1/5, P_2 = 1/5, \text{ dan } P_3 = 3/5 \\
Entropy(S_{\leq 70}) &= - (1/5) \log_2 (1/5) - (1/5) \log_2 (1/5) - (3/5) \log_2 (3/5) \\
&= 1.37095 \\
S_{> 70} &= 5 \text{ kasus, dengan nilai } P_1 = 3/5, P_2 = 2/5, \text{ dan } P_3 = 0/5 \\
Entropy(S_{> 70}) &= - (3/5) \log_2 (3/5) - (2/5) \log_2 (2/5) - (0/5) \log_2 (0/5) \\
&= 0.97095 \\
Gain(S, Humidity) &= 1.57095 - (5/10) Entropy(S_{\leq 70}) - (5/10) Entropy(S_{> 70}) \\
&= 0.39999
\end{aligned}$$

Perhitungan pada fitur *humidity* dengan nilai = 72

$$\begin{aligned}
S_{\leq 72} &= 7 \text{ kasus, dengan nilai } P_1 = 2/7, P_2 = 2/7, \text{ dan } P_3 = 3/7 \\
Entropy(S_{\leq 72}) &= - (2/7) \log_2 (2/7) - (2/7) \log_2 (2/7) - (3/7) \log_2 (3/7) \\
&= 1.55665 \\
S_{> 72} &= 3 \text{ kasus, dengan nilai } P_1 = 2/3, P_2 = 1/3, \text{ dan } P_3 = 0/3 \\
Entropy(S_{> 72}) &= - (2/3) \log_2 (2/3) - (1/3) \log_2 (1/3) - (0/3) \log_2 (0/3) \\
&= 0.918295 \\
Gain(S, Humidity) &= 1.57095 - (7/10) Entropy(S_{\leq 72}) - (3/10) Entropy(S_{> 72}) \\
&= 0.205802
\end{aligned}$$

Nilai *gain* tertinggi didapatkan pada posisi *humidity* dengan nilai 70 yaitu sebesar 0.39999. Oleh karena itu dilakukan pemecahan pada atribut *humidity* dengan nilai 70. Selanjutnya dilakukan perhitungan terhadap semua nilai yang terdapat pada fitur *temperature* sebagai berikut

Perhitungan pada fitur *tempearture* dengan nilai = 82

$$\begin{aligned}
S_{\leq 82} &= 6 \text{ kasus, dengan nilai } P_1 = 2/6, P_2 = 1/6, \text{ dan } P_3 = 3/6 \\
Entropy(S_{\leq 82}) &= - (2/6) \log_2 (2/6) - (1/6) \log_2 (1/6) - (3/6) \log_2 (3/6) \\
&= 1.45914 \\
S_{> 82} &= 4 \text{ kasus, dengan nilai } P_1 = 2/4, P_2 = 2/4, \text{ dan } P_3 = 0/4 \\
Entropy(S_{> 82}) &= - (2/4) \log_2 (2/4) - (2/4) \log_2 (2/4) - (0/4) \log_2 (0/4) \\
&= 1.0 \\
Gain(S, Temperature) &= 1.57095 - (6/10) Entropy(S_{\leq 82}) - (4/10) Entropy(S_{> 82}) \\
&= 0.295461
\end{aligned}$$

Perhitungan pada fitur *tempearture* dengan nilai = 83



$$\begin{aligned}
S_{\leq 83} &= 7 \text{ kasus, dengan nilai } P_1 = 3/7, P_2 = 1/7, \text{ dan } P_3 = 3/7 \\
Entropy(S_{\leq 83}) &= - (3/7) \log_2 (3/7) - (1/7) \log_2 (1/7) - (3/7) \log_2 (3/7) \\
&= 1.44881 \\
S_{> 83} &= 3 \text{ kasus, dengan nilai } P_1 = 1/3, P_2 = 2/3, \text{ dan } P_3 = 0/3 \\
Entropy(S_{> 83}) &= - (1/3) \log_2 (1/3) - (2/3) \log_2 (2/3) - (0/3) \log_2 (0/3) \\
&= 0.918295 \\
Gain(S, Temperature) &= 1.57095 - (7/10) Entropy(S_{\leq 83}) - (3/10) Entropy(S_{> 83}) \\
&= 0.28129
\end{aligned}$$

Perhitungan pada fitur *tempearture* dengan nilai = 84

$$\begin{aligned}
S_{\leq 84} &= 8 \text{ kasus, dengan nilai } P_1 = 3/8, P_2 = 2/8, \text{ dan } P_3 = 3/8 \\
Entropy(S_{\leq 84}) &= - (3/8) \log_2 (3/8) - (2/8) \log_2 (2/8) - (3/8) \log_2 (3/8) \\
&= 1.56127 \\
S_{> 84} &= 2 \text{ kasus, dengan nilai } P_1 = 1/2, P_2 = 1/2, \text{ dan } P_3 = 0/2 \\
Entropy(S_{> 84}) &= - (1/2) \log_2 (1/2) - (1/2) \log_2 (1/2) - (0/2) \log_2 (0/2) \\
&= 1.0 \\
Gain(S, Temperature) &= 1.57095 - (8/10) Entropy(S_{\leq 84}) - (2/10) Entropy(S_{> 84}) \\
&= 0.12192
\end{aligned}$$

Nilai *gain* tertinggi didapatkan pada posisi *temperature* dengan nilai 82 yaitu sebesar 0.295461. Oleh karena itu dilakukan pemecahan pada atribut *temperature* dengan nilai 73.

Berikut hasil perhitungan nilai *entropy* dan *gain* pada perhitungan node ke-1 disajikan pada tabel 3.2

**Tabel 3.2 Hasil Perhitungan *entropy* dan *gain* pada node ke-1**

Node		Nilai	Jml	Light Rain	Heavy Rain	Sunny	Entropy	Gain
1	Total		10				1.57095	
	<i>rainin</i> '							0.88129
		≤ 0.0	3	0	0	3	0.0	
		> 0.0	7	4	3	0	0.98522	
	<i>humidity</i> '							0.39999
		≤ 70	5	1	1	3	1.37095	
		> 70	5	3	2	0	0.97095	
	<i>termperature</i> '							0.29546
		≤ 82	6	2	1	3	1.45914	
> 82		4	2	2	0	1.0		

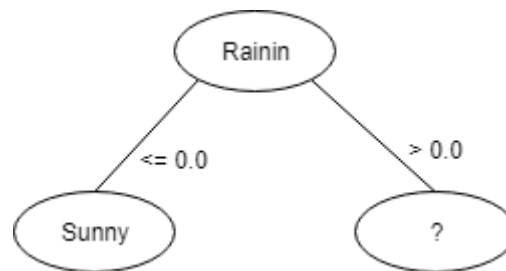
Hasil yang ditunjukkan pada Tabel 3.2 menunjukkan bahwa *gain* tertinggi ada pada fitur *rainin* sehingga *rainin* dijadikan sebagai node akar. Hasil pemisahan data menurut node akar ke-1 ditunjukkan pada tabel 3.3

**Tabel 3.3 Pemisahan data menurut node akar ke-1**

Rainin	Humidity	Temperature	Status
0.0	70	82	Sunny
0.0	69	82	Sunny
0.0	70	82	Sunny
0.3	78	79	Light rain shower
1.5	81	79	Light rain shower
1.6	63	87	Light rain shower
1.9	72	83	Light rain shower
2.6	63	86	Moderate or heavy rain shower
2.7	81	78	Moderate or heavy rain shower
5.3	72	84	Moderate or heavy rain shower

### Proses untuk node ke-2

Untuk node 4 dengan nilai *rainin*  $\leq 0.0$ , nilai entropy yang didapat dipastikan adalah 0 karena semua baris memiliki nilai status yang sama maka node ini dijadikan sebagai node ke-2. Berikut ilustrasi pohon keputusan yang didapat ditunjukkan pada gambar 3.2



**Gambar 3.2 Pohon Keputusan Node ke-2**

### Proses untuk node ke-3

Pada dataset cuaca pada tabel 3.3 terdapat 7 kasus untuk *rainin* dengan nilai  $> 0.0$  yaitu: Status ( $S_1$ ) = '*Light Rain Shower*' atau Status ( $S_2$ ) = '*Moderate or Heavy Rain*'. Berikut perhitungan yang telah dilakukan dibawah ini

Perhitungan nilai *entropy* total

$$\begin{aligned} \text{Jumlah Kelas} &= 7 \text{ kasus, dengan nilai } P_1 = 4/7 \text{ dan } P_2 = 3/7 \\ \text{Entropy}(S) &= - (4/7) \log_2 (4/7) - (3/7) \log_2 (3/7) \\ &= 0.98522 \end{aligned}$$

Perhitungan pada fitur *rainin* dengan nilai = 1.5

$$\begin{aligned} S_{\leq 1.5} &= 2 \text{ kasus, dengan nilai } P_1 = 2/2 \text{ dan } P_2 = 0/2 \\ \text{Entropy}(S_{\leq 1.5}) &= - (2/2) \log_2 (2/2) - (0/2) \log_2 (0/2) \\ &= 0.0 \\ S_{> 1.5} &= 5 \text{ kasus, dengan nilai } P_1 = 2/5 \text{ dan } P_2 = 3/5 \\ \text{Entropy}(S_{> 1.5}) &= - (2/5) \log_2 (2/5) - (3/5) \log_2 (3/5) \\ &= 0.97095 \\ \text{Gain}(S, \text{Rainin}) &= 0.98522 - (2/7) \text{Entropy}(S_{\leq 1.5}) - (5/7) \text{Entropy}(S_{> 1.5}) \\ &= 0.29169 \end{aligned}$$

Perhitungan pada fitur *rainin* dengan nilai = 1.6

$$\begin{aligned} S_{\leq 1.6} &= 3 \text{ kasus, dengan nilai } P_1 = 3/3 \text{ dan } P_2 = 0/3 \\ \text{Entropy}(S_{\leq 1.6}) &= - (3/3) \log_2 (3/3) - (0/3) \log_2 (0/3) \\ &= 0.0 \\ S_{> 1.6} &= 4 \text{ kasus, dengan nilai } P_1 = 1/4 \text{ dan } P_2 = 3/4 \\ \text{Entropy}(S_{> 1.6}) &= - (1/4) \log_2 (1/4) - (3/4) \log_2 (3/4) \\ &= 0.81127 \\ \text{Gain}(S, \text{Rainin}) &= 0.98522 - (3/7) \text{Entropy}(S_{\leq 1.6}) - (4/7) \text{Entropy}(S_{> 1.6}) \\ &= 0.52164 \end{aligned}$$

Perhitungan pada fitur *rainin* dengan nilai = 1.9

$$\begin{aligned} S_{\leq 1.9} &= 4 \text{ kasus, dengan nilai } P_1 = 4/4 \text{ dan } P_2 = 0/4 \\ \text{Entropy}(S_{\leq 1.9}) &= - (4/4) \log_2 (4/4) - (0/4) \log_2 (0/4) \\ &= 0.0 \\ S_{> 1.9} &= 3 \text{ kasus, dengan nilai } P_1 = 0/3 \text{ dan } P_2 = 3/3 \\ \text{Entropy}(S_{> 1.9}) &= - (0/3) \log_2 (0/3) - (3/3) \log_2 (3/3) \\ &= 0.0 \\ \text{Gain}(S, \text{Rainin}) &= 0.98522 - (4/7) \text{Entropy}(S_{\leq 1.9}) - (3/7) \text{Entropy}(S_{> 1.9}) \\ &= 0.98522 \end{aligned}$$

Perhitungan pada fitur *rainin* dengan nilai = 2.6

$$\begin{aligned}
S_{\leq 2.6} &= 5 \text{ kasus, dengan nilai } P_1 = 4/5 \text{ dan } P_2 = 1/5 \\
Entropy(S_{\leq 2.6}) &= - (4/5) \log_2 (4/5) - (1/5) \log_2 (1/5) \\
&= 0.72192 \\
S_{> 2.6} &= 2 \text{ kasus, dengan nilai } P_1 = 0/2 \text{ dan } P_2 = 2/2 \\
Entropy(S_{> 2.6}) &= - (0/2) \log_2 (0/2) - (2/2) \log_2 (2/2) \\
&= 0.0 \\
Gain(S, Rainin) &= 0.98522 - (5/7) Entropy(S_{\leq 2.6}) - (2/7) Entropy(S_{> 2.6}) \\
&= 0.46956
\end{aligned}$$

Nilai *gain* tertinggi didapatkan pada posisi *rainin* dengan nilai 1.9 yaitu sebesar 0.98522. Oleh karena itu dilakukan pemecahan pada atribut *rainin* dengan nilai 1.9. Selanjutnya dilakukan perhitungan terhadap semua nilai yang terdapat pada fitur *humidity* sebagai berikut

Perhitungan pada fitur *humidity* dengan nilai = 63

$$\begin{aligned}
S_{\leq 63} &= 2 \text{ kasus, dengan nilai } P_1 = 1/2 \text{ dan } P_2 = 1/2 \\
Entropy(S_{\leq 63}) &= - (1/2) \log_2 (1/2) - (1/2) \log_2 (1/2) \\
&= 1.0 \\
S_{> 63} &= 5 \text{ kasus, dengan nilai } P_1 = 3/5 \text{ dan } P_2 = 2/5 \\
Entropy(S_{> 63}) &= - (3/5) \log_2 (3/5) - (2/5) \log_2 (2/5) \\
&= 0.97095 \\
Gain(S, Humidity) &= 0.98522 - (2/7) Entropy(S_{\leq 63}) - (5/7) Entropy(S_{> 63}) \\
&= 0.00597
\end{aligned}$$

Perhitungan pada fitur *humidity* dengan nilai = 72

$$\begin{aligned}
S_{\leq 72} &= 4 \text{ kasus, dengan nilai } P_1 = 2/4 \text{ dan } P_2 = 2/4 \\
Entropy(S_{\leq 72}) &= - (2/4) \log_2 (2/4) - (2/4) \log_2 (2/4) \\
&= 1.0 \\
S_{> 72} &= 3 \text{ kasus, dengan nilai } P_1 = 2/3 \text{ dan } P_2 = 1/3 \\
Entropy(S_{> 72}) &= - (2/3) \log_2 (2/3) - (1/3) \log_2 (1/3) \\
&= 0.91829 \\
Gain(S, Humidity) &= 0.98522 - (4/7) Entropy(S_{\leq 72}) - (3/7) Entropy(S_{> 72}) \\
&= 0.02024
\end{aligned}$$

Nilai *gain* tertinggi didapatkan pada posisi *humidity* dengan nilai 72 yaitu sebesar 0.02024. Oleh karena itu dilakukan pemecahan pada atribut *humidity* dengan nilai 72. Selanjutnya dilakukan perhitungan terhadap semua nilai yang terdapat pada fitur *temperature* sebagai berikut

Perhitungan pada fitur *temperature* dengan nilai = 79

$$\begin{aligned}
 S_{\leq 79} &= 3 \text{ kasus, dengan nilai } P_1 = 2/3 \text{ dan } P_2 = 1/3 \\
 Entropy(S_{\leq 79}) &= - (2/3) \log_2 (2/3) - (1/3) \log_2 (1/3) \\
 &= 0.91829 \\
 S_{> 79} &= 4 \text{ kasus, dengan nilai } P_1 = 2/4 \text{ dan } P_2 = 2/4 \\
 Entropy(S_{> 79}) &= - (2/4) \log_2 (2/4) - (2/4) \log_2 (2/4) \\
 &= 1.0 \\
 Gain(S, Temperature) &= 0.98522 - (3/7) Entropy(S_{\leq 79}) - (4/7) Entropy(S_{> 79}) \\
 &= 0.02024
 \end{aligned}$$

Perhitungan pada fitur *temperature* dengan nilai = 83

$$\begin{aligned}
 S_{\leq 83} &= 4 \text{ kasus, dengan nilai } P_1 = 3/4 \text{ dan } P_2 = 1/4 \\
 Entropy(S_{\leq 83}) &= - (3/4) \log_2 (3/4) - (1/4) \log_2 (1/4) \\
 &= 0.81127 \\
 S_{> 83} &= 3 \text{ kasus, dengan nilai } P_1 = 1/3 \text{ dan } P_2 = 2/3 \\
 Entropy(S_{> 83}) &= - (1/3) \log_2 (1/3) - (2/3) \log_2 (2/3) \\
 &= 0.91829 \\
 Gain(S, Temperature) &= 0.98522 - (4/7) Entropy(S_{\leq 83}) - (3/7) Entropy(S_{> 83}) \\
 &= 0.12808
 \end{aligned}$$

Perhitungan pada fitur *temperature* dengan nilai = 84

$$\begin{aligned}
 S_{\leq 84} &= 5 \text{ kasus, dengan nilai } P_1 = 3/5 \text{ dan } P_2 = 2/5 \\
 Entropy(S_{\leq 84}) &= - (3/5) \log_2 (3/5) - (2/5) \log_2 (2/5) \\
 &= 0.97095 \\
 S_{> 84} &= 2 \text{ kasus, dengan nilai } P_1 = 1/2 \text{ dan } P_2 = 1/2 \\
 Entropy(S_{> 84}) &= - (1/2) \log_2 (1/2) - (1/2) \log_2 (1/2) \\
 &= 1.0 \\
 Gain(S, Temperature) &= 0.98522 - (5/7) Entropy(S_{\leq 84}) - (2/7) Entropy(S_{> 84})
 \end{aligned}$$

$$= 0.00597$$

Nilai *gain* tertinggi didapatkan pada posisi *temperature* dengan nilai 83 yaitu sebesar 0.12808. Oleh karena itu dilakukan pemecahan pada atribut *temperature* dengan nilai 83. Berikut hasil perhitungan nilai *entropy* dan *gain* pada perhitungan node ke-3 disajikan pada tabel 3.4

**Tabel 3.4 Hasil perhitungan *entropy* dan *gain* pada node ke-3**

Node		Nilai	Jml	Light Rain	Heavy Rain	Entropy	Gain
3	Total		7			0.98522	
	<i>rainin</i> '						0.98522
		≤ 1.9	4	4	0	0.0	
		> 1.9	3	0	3	0.0	
	<i>humidity</i> '						0.02024
		≤ 72	4	2	2	1.0	
		> 72	3	2	1	0.91829	
	<i>temperature</i> '						0.12808
		≤ 83	4	3	1	0.81127	
		> 83	3	1	2	0.91829	

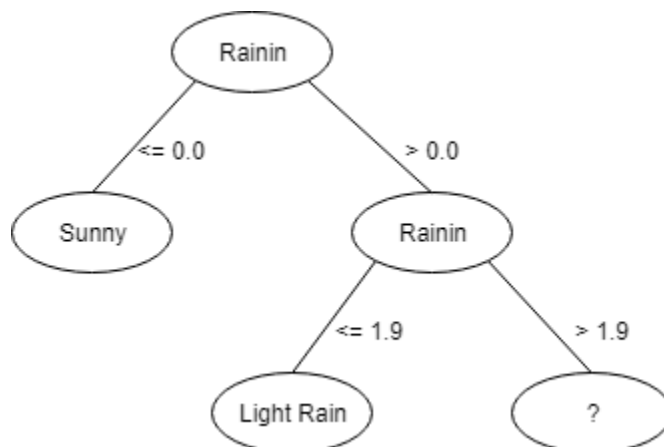
Hasil yang ditunjukkan pada Tabel 3.4 menunjukkan bahwa *gain* tertinggi ada pada fitur *rainin*' dengan nilai 1.9 sehingga *rainin* dijadikan sebagai node ke-3. Hasil pemisahan data menurut node akar ke-3 ditunjukkan pada tabel 3.5

**Tabel 3.5 Pemisahan data menurut node akar ke-3**

Rainin	Humidity	Temperature	Status
0.3	78	79	Light rain shower
1.5	81	79	Light rain shower
1.6	63	87	Light rain shower
1.9	72	83	Light rain shower
2.6	63	86	Moderate or heavy rain shower
2.7	81	78	Moderate or heavy rain shower
5.3	72	84	Moderate or heavy rain shower

#### Proses untuk node ke-4

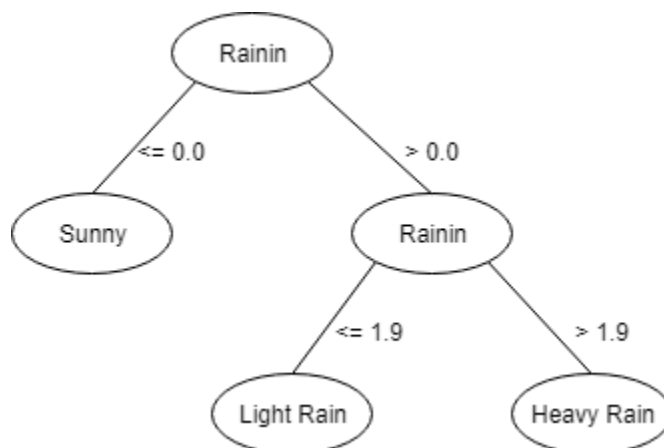
Untuk node 4 dengan  $rainin \leq 1.9$ , nilai *entropy* yang didapat dipastikan adalah 0 karena semua baris memiliki nilai status yang sama yaitu '*Light Rain Shower*' maka node ini dijadikan sebagai node ke-4. Berikut ilustrasi pohon keputusan yang didapat ditunjukkan pada gambar 3.3



**Gambar 3.3 Pohon Keputusan Iterasi Ke-2**

#### Proses untuk node ke-5

Untuk node 5 dengan  $rainin > 1.9$ , nilai entropy yang didapat dipastikan adalah 0 karena semua baris memiliki nilai status yang sama yaitu ‘*Moderate or Heavy Rain*’ maka node ini dijadikan sebagai node ke-5. Berikut ilustrasi pohon keputusan yang didapat ditunjukkan pada gambar 3.4



**Gambar 3.4 Pohon Keputusan Iterasi Ke-2**

Setelah didapatkan model pohon keputusan dari perhitungan yang telah dilakukan maka diperoleh aturan – aturan pengklasifian cuaca sebagai berikut

```

IF rainin <=0.0 THEN status = Cerah
IF rainin >0.0 AND rainin <= 1.9 THEN status = Hujan
IF rainin >0.0 AND rainin >1.9 THEN status = Hujan Lebat
  
```

### 3. Post Process

Setelah dilakukan pemodelan aturan – aturan dalam mengklasifikasikan cuaca, maka langkah terakhir yang harus dilakukan adalah melakukan pengujian pohon keputusan terhadap data uji untuk mengukur keakuratan pohon keputusan. Adapun data uji yang dilakukan menggunakan dataset cuaca milik Pemerintah Kota Bandung bulan Mei 2019. Berikut dataset cuaca yang didapatkan ditunjukkan pada tabel 3.6

**Tabel 3.6 Dataset Uji Cuaca Mei 2019**

No	Tanggal	Lokasi	Rainin	Status
1	09/05/2019	Pascal Hypersquare	1.13	Hujan
2	09/05/2019	Fiberstar	0.43	Hujan
3	10/05/2019	Kecamatan Lengkong	0.0	Cerah
4	10/05/2019	Kecamatan Cidadap	0.0	Cerah
5	09/05/2019	Kecamatan Lengkong	0.57	Hujan

Berikut hasil pengujian yang telah dilakukan ditunjukkan pada tabel 3.7

**Tabel 3.7 Hasil Pengujian Algoritma C4.5**

No	Algoritma C4.5	Dataset	Status
1	Hujan	Hujan	Benar
2	Hujan	Hujan	Salah
3	Cerah	Cerah	Benar
4	Cerah	Cerah	Benar
5	Hujan	Hujan	Benar

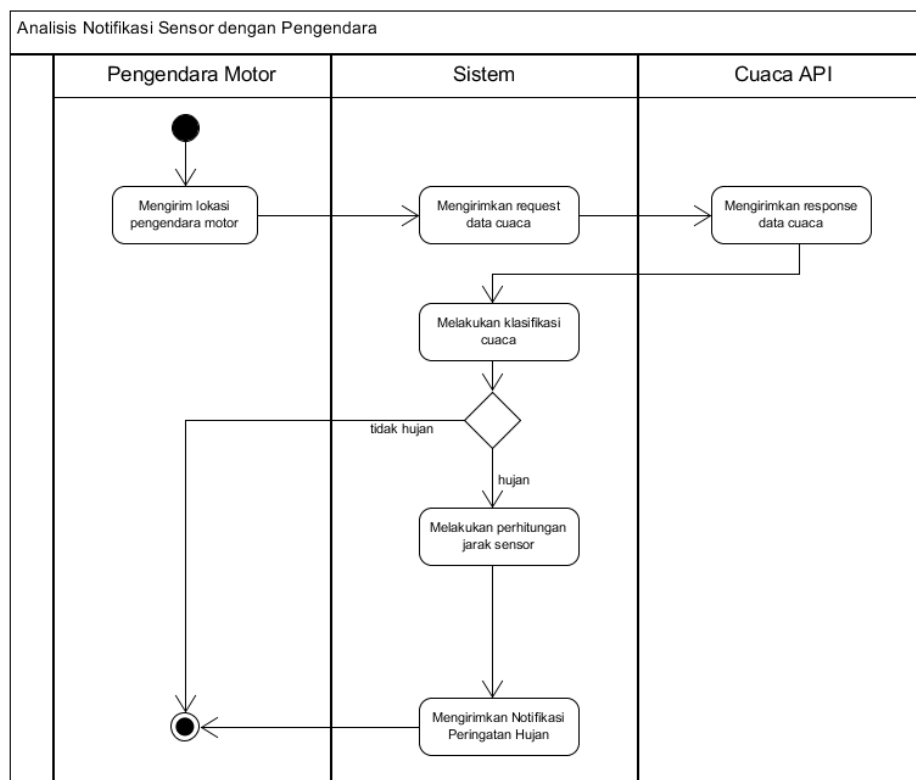
Dari hasil pengujian yang telah dilakukan dapat disimpulkan bahwa pohon keputusan yang didapatkan memiliki keakurasian yang sangat baik dalam mengklasifikasikan kondisi cuaca.

#### 3.1.2.2 Analisis Penerapan Metode Haversine Formula

Metode *Haversine Formula* merupakan metode yang digunakan untuk menghitung jarak antara dua titik posisi berdasarkan nilai *latitude* dan *longitude* berupa garis lurus. Dalam penelitian yang dilakukan, metode ini digunakan untuk menentukan sensor cuaca terdekat berdasarkan jalur rute perjalanan pengendara dan



digunakan untuk melakukan perhitungan jarak sensor terdekat dengan pengendara. Berikut alur proses penggunaan metode *Haversine Formula* dalam menampilkan jarak sensor cuaca dengan pengguna ditunjukkan pada gambar 3.5



**Gambar 3.5 Analisis Prosedur Menampilkan Notifikasi**

Dalam penelitian ini setiap sensor yang ada ditetapkan bahwa sensor dapat memberikan informasi cuaca dengan radius 1 kilometer dengan daerah sekitarnya sehingga ketika sensor cuaca dalam keadaan status hujan sistem akan memberikan notifikasi pada pengendara dengan jarak terdekat sensor dengan pengguna.

Berikut penjelasan dari alur proses yang ditunjukkan pada gambar 3.6

1. Pengendara motor mengirimkan lokasi pengendara berada ke sistem
2. Sistem melakukan *request* ke Cuaca API untuk mendapatkan informasi parameter cuaca
3. Setelah itu sistem melakukan pengecekan status cuaca pada tiap sensor yang didapatkan. Jika status cuaca menunjukkan kondisi cuaca hujan maka dilakukan perhitungan jarak antara titik sensor dengan titik pengendara, jika tidak maka kondisi berakhir.

4. Ketika terdapat titik lokasi – lokasi hujan, maka sistem akan menghitung jarak sensor terdekat untuk ditampilkan kepada pengguna.

Berikut contoh perhitungan jarak antara pengendara dengan titik sensor menggunakan metode *haversine formula*. Diasumsikan lokasi pengguna berada di Jl. Kebon Jati dengan titik koordinat yaitu *latitude* adalah -6.9163745 dan *longitude* adalah 107.5958897. Adapun titik lokasi sensor yang ada ditunjukkan pada tabel 3.8

**Tabel 3.8 Data Lokasi Sensor Cuaca**

ID	Lokasi	Latitude	Longitude
IBANDUNG10	Pascal Hypersquare	-6.914	107.597
IBANDUNG8	Balai Kota Bandung	-6.911	107.609

Perhitungan jarak arak sensor “IBANDUNG10” dengan pengguna

Diketahui : Lat1 = -6.9163745, Lon1 = 107.5958897

Lat2 = -6.914, Lon2 = 107.597

$$\begin{aligned}
 x &= ((107.597 * 0.0174532925) - (107.597 * 0.0174532925)) * \\
 &\quad \text{Cos}((( -6.9163745 * 0.0174532925) + (-6.914 * \\
 &\quad 0.0174532925)) / 2) \\
 &= 1.9237421
 \end{aligned}$$

$$\begin{aligned}
 y &= (-6.914 * 0.0174532925) - (-6.9163745 * 0.0174532925) \\
 &= 4.1442843
 \end{aligned}$$

$$\begin{aligned}
 \text{Jarak} &= \text{sqrt}(1.9237421 * 1.9237421 + 4.1442843 * 4.1442843) * \\
 &\quad 6371 \\
 &= 0.291 \text{ Km}
 \end{aligned}$$

Perhitungan jarak arak sensor “IBANDUNG8” dengan pengguna

Diketahui : Lat1 = -6.9163745, Lon1 = 107.5958897

Lat2 = -6.911, Lon2 = 107.609

$$\begin{aligned}
 x &= ((107.609 * 0.0174532925) - (107.597 * 0.0174532925)) * \\
 &\quad \text{Cos}((( -6.9163745 * 0.0174532925) + -6.911 * \\
 &\quad 0.0174532925)) / 2) \\
 &= 0.000227
 \end{aligned}$$

$$\begin{aligned}
 y &= (-6.911 * 0.0174532925) - (-6.9163745 * 0.0174532925) \\
 &= 9.380272
 \end{aligned}$$

$$\begin{aligned} \text{Jarak} &= \sqrt{(1.9237421 * 1.9237421 + 4.1442843 * 4.1442843)} * \\ &6371 \\ &= 1.5657 \text{ Km} \end{aligned}$$

Dari perhitungan yang telah dilakukan didapat jarak antara pengendara motor dengan sensor cuaca dengan id bernama “IBANDUNG10” adalah 291 meter sedangkan untuk sensor cuaca dengan id bernama “IBANDUNG8” memiliki jarak dengan pengendara sebesar 1,56 km. Sehingga dapat disimpulkan bahwa notifikasi akan disampaikan kepada pengguna adalah dari sensor “IBANDUNG10”.

### 3.1.2.3 Analisis Laporan Banjir

Dalam penelitian ini, aplikasi yang akan dibangun memiliki fungsi untuk dapat melakukan pelaporan banjir bagi pengendara motor. Adapun laporan tersebut akan menjadi informasi banjir bagi pengendara motor lainnya. Banjir yang terjadi pada setiap daerah memiliki radius luas banjir yang beragam yang mana hal tersebut dipengaruhi oleh intensitas hujan maupun kondisi daerah tersebut. Oleh karena itu pada penelitian ini, setiap laporan banjir yang dikirimkan hanya akan dibatasi dengan luas radius 50 meter. Setiap laporan yang berada dalam satu radius tersebut akan dijadikan sebagai satu buah laporan dengan beberapa pelapor, hal ini dilakukan dalam meningkatkan ke validan laporan.

Berikut contoh dari pengelompokan laporan banjir berdasarkan jarak ditunjukkan pada tabel 3.9

**Tabel 3.9 Data Laporan Banjir**

ID	Jalan	Latitude	Longitude
A	Jl. Raden Dewi Sartika No.86	-6.9285561811904905	107.6056938518365
B	Jl. Raden Dewi Sartika No.86	-6.92861915752031	107.60569051526552
C	Jl. Pungkur No.28	-6.92805452896461	107.6049472684995

Selanjutnya dihitung jarak antar laporan banjir dengan metode *Haversine Formula*. Berikut hasil perhitungan jarak ditunjukkan pada tabel 3.10

**Tabel 3.10 Perhitungan Jarak Laporan Banjir**

Lokasi 1	Lokasi 2	Jarak
Laporan A	Laporan B	7.01 meter
Laporan A	Laporan C	99.51 meter

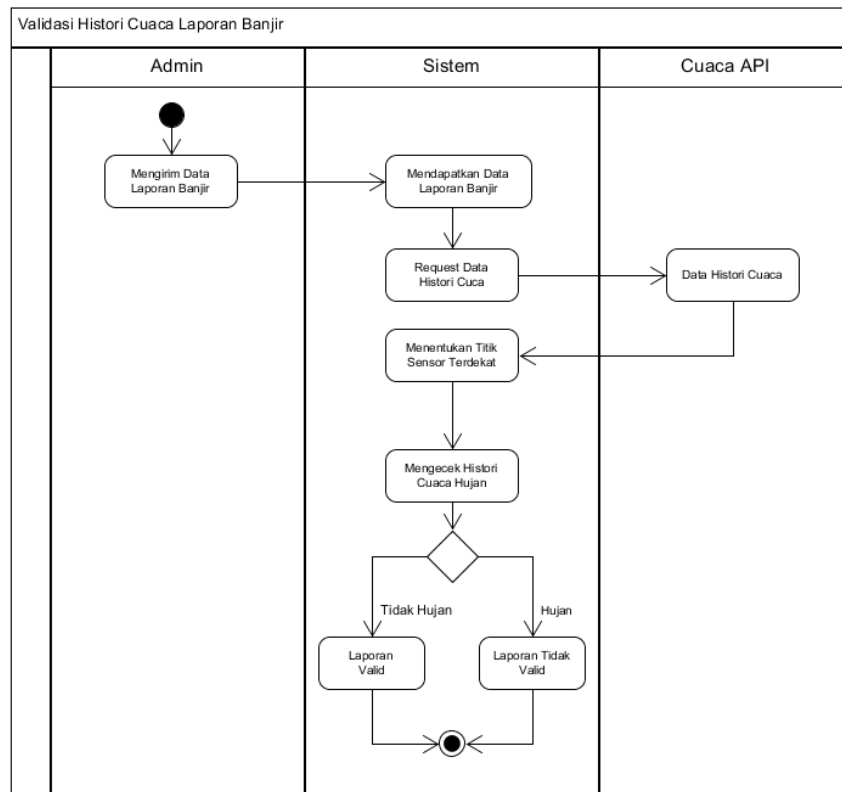
Dapat dilihat bahwa Laporan A dan Laporan B memiliki jarak kurang dari sama dengan 80 meter sehingga akan dijadikan 1 laporan sedangkan Laporan C menjadi 1 laporan tersendiri. Berikut hasil pengelompokan akhir ditunjukkan pada tabel 3.11

**Tabel 3.11 Pengelompokan Laporan Banjir**

Laporan	Lokasi	Pelapor
Laporan 1	Laporan A	2 pelapor
	Laporan B	
Laporan 2	Laporan C	1 pelapor

#### 3.1.2.3.1 Validasi Informasi Banjir

Pada penelitian ini laporan banjir yang dikirim juga akan divalidasi dengan melihat histori cuaca sekitar lokasi pelaporan berdasarkan sensor yang ada. Diasumsikan bahwa titik banjir yang dilaporkan merupakan banjir yang disebabkan oleh curah hujan yang ada di lokasi pelaporan dan bukan merupakan “banjir kiriman” (banjir yang dikirim dari daerah lain yang memiliki daerah yang lebih tinggi). Berikut tahapan validasi laporan banjir yang dilakukan berikut ditampilkan pada gambar 3.6



**Gambar 3.6 Validasi Data Laporan Banjir**

Adapun penjelasan dari gambar 3.6 yaitu sebagai berikut

1. Sistem akan melakukan pengambilan informasi titik lokasi pengiriman laporan banjir dilakukan
2. Selanjutnya sistem melakukan pengecekan terhadap histori informasi cuaca terdekat dari lokasi titik pelaporan banjir.
3. Jika informasi histori cuaca yang didapat menyatakan bahwa kondisi cuaca sebelumnya adalah hujan maka laporan banjir akan dinyatakan valid.
4. Jika tidak maka laporan banjir tidak akan dinyatakan valid.

Namun didapat bahwa sensor cuaca yang ada belum secara berkala dapat mengirimkan kondisi cuaca secara *real-time*, terkadang sensor mengalami keterlambatan dalam mengirimkan kondisi cuaca suatu daerah sehingga proses validasi menjadi kurang optimal. Maka dari itu hasil validasi yang dilakukan hanya akan dijadikan sebagai informasi bagi pengendara untuk memvalidasi laporan.

### 3.1.2.3.2 Validasi Lalu Lintas Banjir

Pada pembahasan sebelumnya diketahui bahwa banjir dapat menyebabkan kepadatan lalu lintas. Dalam penelitian ini kondisi lalu lintas pada daerah banjir ketika pengendara mencari rute akan ditampilkan sebagai informasi tambahan, sehingga pengendara dapat menentukan sendiri apakah pengendara akan menggunakan jalur alternatif ataupun tidak, adapun tahapan pengambilan informasi yang dilakukan yaitu sebagai berikut :

1. Dilakukan pengecekan titik terdekat banjir dengan jalur pengendara dengan menggunakan metode perhitungan jarak (*haversnine formula*)
2. Jika titik banjir didapatkan, maka tahapan selanjutnya adalah mendapatkan kondisi sepanjang jalan yang dilalui titik banjir tersebut.
3. Kondisi jalan yang didapatkan dari Google Directions API berupa informasi *duration* (waktu rata-rata pengendara melalui jalan tersebut pada jam – jam sebelumnya) dan informasi *duraton\_in\_traffic* (waktu rata-rata pengendara melalui jalan tersebut saat ini)
4. Dengan membandingkan kedua waktu tersebut, yaitu jika waktu di jalan (*duraton\_in\_traffic*) lebih besar dari waktu rata-rata (*duration*) maka ditentukan bahwa jalanan tersebut dalam keadaan padat
5. Namun jika tidak, maka sistem akan menentukan bahwa lalu lintas di sekitar titik banjir tidak sedang dalam keadaan padat.
6. Selanjutnya informasi kondisi lalu lintas akan ditampilkan kepada pengendara sebagai informasi tambahan laporan banjir

Berikut data response yang didapatkan dari Google Directions API sebagai berikut

**Tabel 3.12 Response Data Lalu Lintas Google Directions API**

```
{
  "geocoded_waypoints" : [
    {
      "geocoder_status" : "OK",
      "place_id" : "ChIJf0dSgjnmaC4RshXo05MfahQ",
      "types" : [ "locality", "political" ]
    },
    {
      "geocoder_status" : "OK",
      "place_id" : "ChIJs1-XR2bvaC4RYeo7VPhrEKE",

```

```

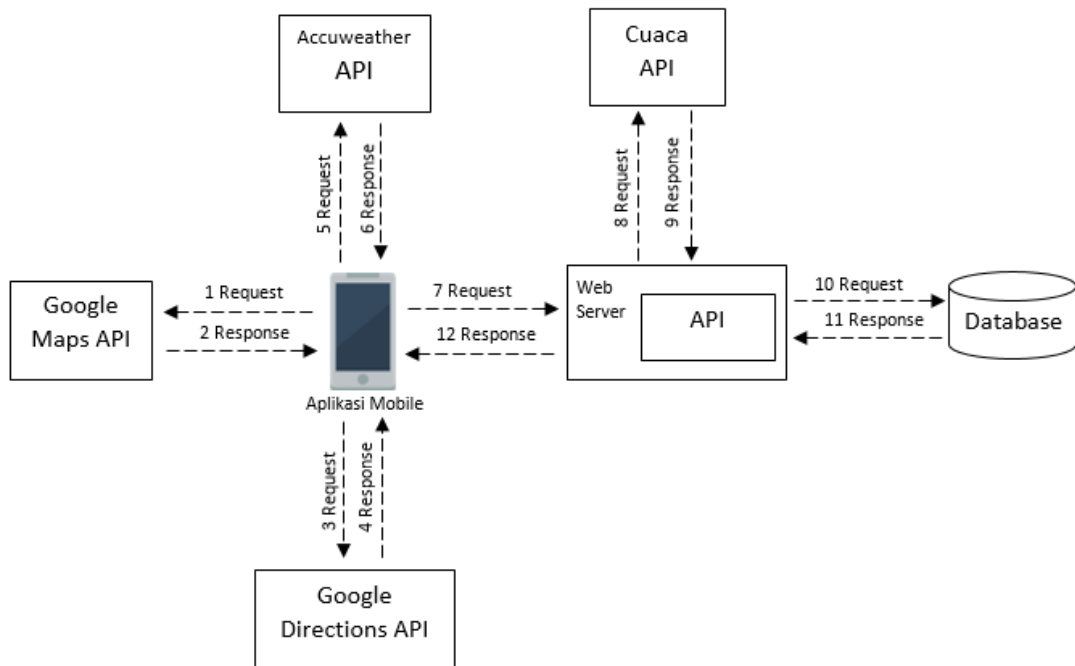
    "types" : [ "administrative_area_level_4", "political" ]
  }
],
"routes" : [
  {
    "bounds" : {
      "northeast" : {
        "lat" : -6.914624099999999,
        "lng" : 107.6273653
      },
      "southwest" : {
        "lat" : -6.9387662,
        "lng" : 107.5468238
      }
    },
    "copyrights" : "Data peta ©2019 Google",
    "legs" : [
      {
        "distance" : {
          "text" : "13,0 km",
          "value" : 13047
        },
        "duration" : {
          "text" : "42 menit",
          "value" : 2509
        },
        "duration_in_traffic" : {
          "text" : "28 menit",
          "value" : 1685
        },
        "end_address" : "Margaasih, Kec. Margaasih, Bandung, Jawa Barat, Indonesia",
        "end_location" : {
          "lat" : -6.938703899999999,
          "lng" : 107.5478936
        },
        "start_address" : "Bandung, Kota Bandung, Jawa Barat, Indonesia",
        "start_location" : {
          "lat" : -6.9168957,
          "lng" : 107.6192613
        }
      },
    ]
  },
]

```

### 3.1.3 Analisis Arsitektur Sistem

Analisis arsitektur sistem bertujuan untuk memberikan gambaran serta identifikasi terhadap sistem yang akan dibangun. Analisis arsitektur dilakukan untuk memberi gambaran bagaimana sistem saling berhubungan dan berinteraksi satu dengan yang lainnya. Pada penelitian ini analisis arsitektur yang dilakukan

yaitu sistem yang terdapat pada perangkat *mobile*. Adapun sistem operasi *mobile* yang digunakan pada perancangan sistem yang akan dibangun yaitu menggunakan *OS Android*. Dalam sistem ini terdapat satu pengguna saja yaitu pengendara sepeda motor yang menggunakan sistem pada *platform* ini. Berikut merupakan gambaran ilustrasi dari arsitektur sistem pada *platform mobile* yang akan dibangun ditampilkan pada gambar 3.7



**Gambar 3.7 Ilustrasi arsitektur sistem pada *platform mobile***

Berikut penjelasan dari ilustrasi pada gambar 3.7 :

1. Aplikasi pada *platform mobile* dapat melakukan *request* berupa permintaan data *map* dengan mengirim *request* ke *Google Maps API*. Untuk melakukan *request* diperlukan sebuah *API Key* sebagai perizinan permintaan data
2. Setelah itu data akan dikirimkan kembali ke *platform android* dalam bentuk *response* berupa informasi yang dibutuhkan dalam menampilkan *map*
3. Untuk menampilkan rute jalur maka *request* akan dikirimkan *Google Directions API* berupa titik koordinat asal dan juga titik koordinat tujuan
4. Setelah itu *Google Direction API* akan mengirimkan sebuah *response* yang berisi informasi titik – titik rute jalur, waktu tempuh, dan jarak rute



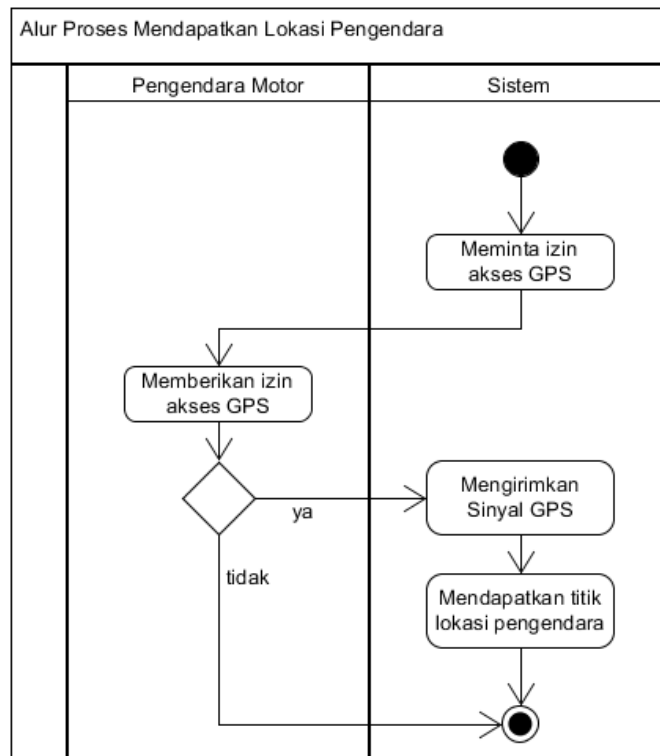
5. Pada aplikasi yang akan dibangun terdapat fitur prediksi cuaca lokasi tujuan pengendara. Dalam mendapatkan informasi tersebut sistem pada *platform mobile* akan mengirimkan *request* ke *Accuweather API*.
6. Selanjutnya *Accuweather API* akan mengirim informasi prediksi cuaca akan ke *platform android* dalam bentuk *response*
7. Selain layanan *API* diatas, *web server* juga menyediakan *API* sendiri untuk mendapatkan data dari database. *Request* disesuaikan dengan kebutuhan sistem pada *platform mobile*
8. Untuk mendapatkan informasi cuaca yang disediakan oleh Cuaca *API*, *web server* akan mengirimkan *request* ke penyedia Cuaca *API*, setiap *request* memerlukan *access token* sebagai perizinan permintaan data
9. Selanjutnya informasi cuaca akan dikirimkan oleh Cuaca *API* dalam bentuk *response* yang berisi data lokasi sensor cuaca dan parameter – parameter cuaca kembali ke *web server* untuk disimpan di dalam *database*.
10. Untuk melakukan permintaan data ke *database* yaitu dengan melakukan *request* yang telah disediakan oleh sistem. Adapun data yang terdapat dalam *database* berupa data informasi cuaca yang didapat dari Cuaca *API*, laporan banjir serta data pengguna.
11. Selanjutnya data yang didapat dari database akan dikirimkan dalam bentuk *response* berisi data informasi *database* kembali ke *web server*.
12. Setiap *response* yang dikirimkan kembali ke aplikasi dengan *platform mobile* akan berformat JSON.

#### **3.1.4 Analisis Teknologi Yang Digunakan**

Analisis teknologi yang digunakan dalam penelitian ini dimaksudkan untuk memberikan deskripsi gambaran terhadap teknologi apa saja yang digunakan pada pembangunan perangkat lunak yang dilakukan. Berikut teknologi yang dibutuhkan oleh sistem pada perangkat lunak yang dibangun yaitu sebagai berikut.

### 3.1.4.1 Global Position System (GPS)

*Global Position System (GPS)* adalah sebuah teknologi yang dapat dimanfaatkan untuk memberikan informasi lokasi berupa koordinat *latitude* dan *longitude* posisi perangkat pengendara motor berada. Dalam penelitian ini teknologi GPS digunakan untuk membantu sistem dalam mengetahui keberadaan lokasi pengendara berada. Informasi lokasi pengendara motor selanjutnya akan digunakan sebagai penentuan rute jalur dan juga penentuan titik terdekat sensor cuaca dengan titik lokasi pengendara. Berikut alur proses untuk mendapatkan lokasi pengendara pada sistem yang akan dibangun ditampilkan pada gambar 3.8



**Gambar 3.8 Alur Proses Mendapatkan Lokasi Pengendara**

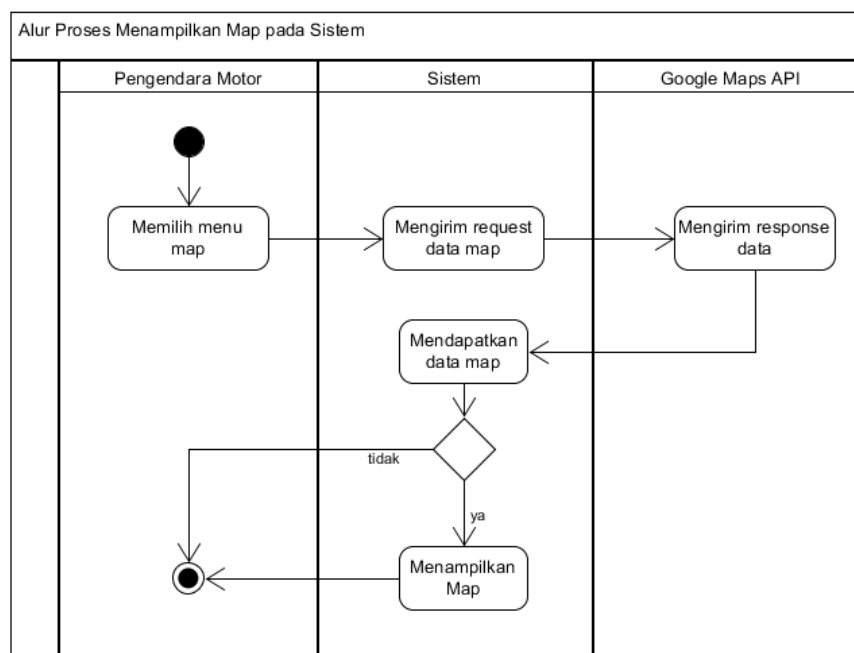
Berikut penjelasan dari alur proses yang terdapat pada gambar 3.8

1. Pada saat sistem dijalankan pertama kali maka sistem akan meminta izin untuk dapat mengakses GPS
2. Selanjutnya pengendara motor akan melakukan konfirmasi permintaan izin oleh sistem untuk mengakses GPS

3. Jika akses diberikan, maka sistem akan mengirimkan sinyal GPS dari perangkat pengendara motor sehingga sistem dapat mendapatkan titik lokasi pengendara berada
4. Jika akses tidak diberikan maka sistem tidak dapat mengakses GPS

### 3.1.4.2 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. Dengan menggunakan layanan ini akan memudahkan pengendara motor dalam mengetahui lokasi keberadaanya dalam bentuk visual. Berikut alur proses untuk menampilkan *map* pada sistem yang akan dibangun ditunjukkan pada gambar 3.9



**Gambar 3.9 Alur Proses Google Map API pada sistem**

Berikut penjelasan dari proses pada gambar 3.9 :

1. Untuk dapat menampilkan *map* pada sistem, pengendara motor dapat memilih menu *map*

2. Selanjutnya sistem akan melakukan permintaan data *map* dengan mengirimkan *request* ke *Google Maps API*. Setiap *request* yang dilakukan memerlukan *API Key* sebagai izin permintaan data
3. Setelah itu *Google Maps API* akan mengirimkan *response* data
4. Jika *response* yang dikirim merupakan data *map* maka sistem akan menampilkan visual *map* pada sistem kepada pengendara motor
5. Jika tidak maka visual *map* tidak akan ditampilkan

Berikut merupakan sintaks untuk menyertakan *API Key* pada sistem

**Tabel 3.13 Contoh API Key Google Maps API**

```

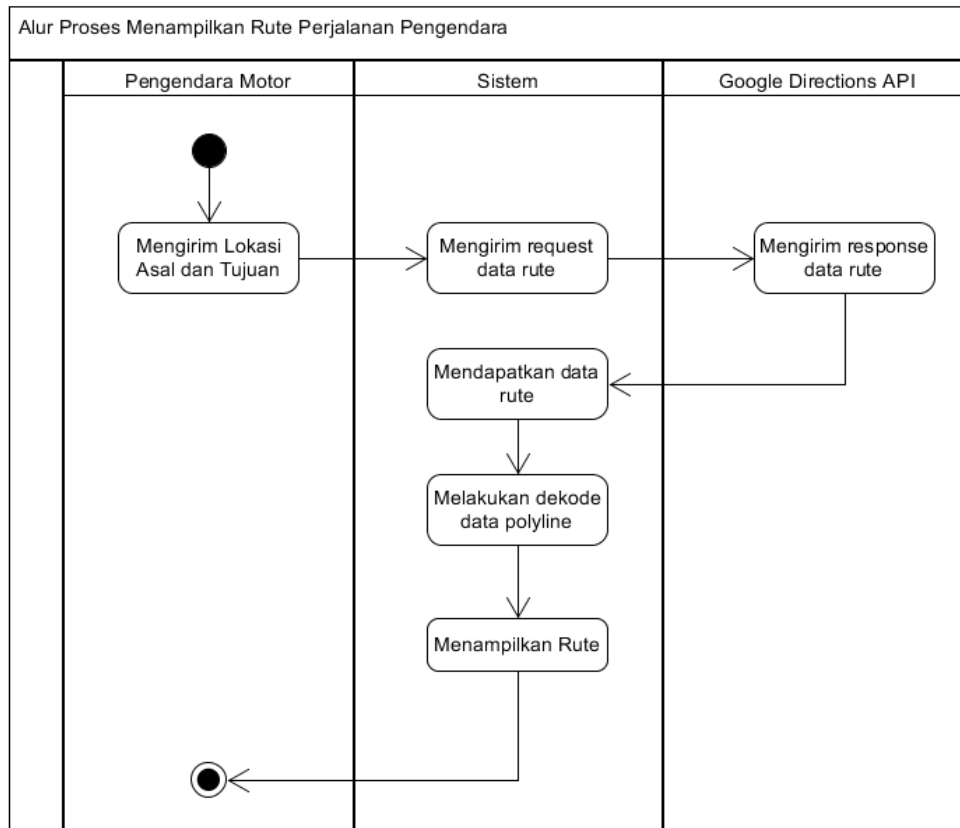
<meta-data
  android:name="com.google.android.geo.API_KEY"
  android:value=" YOUR_API_KEY "/>
```

Adapun penggunaan *Google Maps API* pada sistem ini yaitu diantaranya adalah

1. Menampilkan posisi lokasi pengendara motor pada sebuah *Map*
2. Menampilkan rute pengendara motor pada *Map* sepanjang titik lokasi pengendara hingga titik lokasi tujuan pengendara
3. Menampilkan informasi titik – titik daerah hujan dan banjir dalam bentuk *marker* pada *map*

### 3.1.4.3 Google Directions API

*Google Directions API* merupakan layanan yang dapat memberikan informasi berupa rute dan navigasi antara dua titik lokasi asal dan tujuan. Parameter yang dibutuhkan dalam mendapatkan rute adalah titik *latitude* dan *longitude* tempat asal dan tujuan. Pada perangkat lunak yang akan dibangun *Google Directions API* digunakan untuk memudahkan pengendara motor dalam menentukan jalur perjalanan ke tempat tujuannya. Berikut alur proses dalam menampilkan rute perjalanan pengendara motor ditunjukkan pada gambar 3.10



**Gambar 3.10 Alur Proses Menampilkan Rute Perjalanan Pengendara**

Berikut penjelasan dari alur proses yang terdapat pada gambar 3.10

1. Pengendara motor akan mengirimkan titik lokasi pengendara berada dengan lokasi tempat tujuan pengendara
2. Selanjutnya sistem akan mengirimkan *request* data rute ke Google Directions API. Setiap *request* diperlukan *API Key* untuk perizinan permintaan data
3. Selanjutnya Google Directions API akan mengirimkan *response* berupa data rute perjalanan seperti jarak tempuh, waktu tempuh, *star address*, *end address* dan data *polyline*. Data *polyline* berisi data *points* yang bernilai titik – titik koordinat rute yang telah di encode
4. Setelah sistem mendapatkan data, sistem akan melakukan dekode pada data *polyline* untuk mendapatkan titik – titik koordinat rute perjalanan
5. Setiap titik koordinat yang telah di dekode akan ditampilkan dalam bentuk gambar garis lurus membentuk jalur kepada pengendara

Berikut merupakan contoh *response* yang didapat dari *Google Directions API*

**Tabel 3.14 Response Data Rute Google Directions API**

```
{
  "geocoded_waypoints" : [
    {
      "geocoder_status" : "OK",
      "place_id" : "ChIJX9bcf0zmaC4Ram_Jz2d1QYM",
      "types" : [ "establishment",
"local_government_office", "point_of_interest" ]
    },
    {
      "geocoder_status" : "OK",
      "place_id" : "ChIJteyvcUvmaC4RyVnZiMYgLLo",
      "types" : [ "atm", "establishment", "finance",
"point_of_interest" ]
    }
  ],
  "routes" : [
    {
      "bounds" : {
        "northeast" : {
          "lat" : -6.9027138,
          "lng" : 107.6191971
        },
        "southwest" : {
          "lat" : -6.9029995,
          "lng" : 107.6190037
        }
      },
      "copyrights" : "Data peta ©2019 Google",
      "legs" : [
        {
          "distance" : {
            "text" : "52 m",
            "value" : 52
          },
          "duration" : {
            "text" : "1 menit",
            "value" : 21
          },
          "end_address" : "Unnamed Road, Citarum,
Bandung Wetan, Kota Bandung, Jawa Barat 40115, Indonesia",
          "end_location" : {
            "lat" : -6.9029995,
            "lng" : 107.6191971
          },
          "start_address" : "Jl. Diponegoro No.22,
Citarum, Bandung Wetan, Kota Bandung, Jawa Barat 40115,
Indonesia",
          "start_location" : {
            "lat" : -6.9027138,
            "lng" : 107.6190037
          }
        }
      ]
    }
  ]
}
```

```

    "steps" : [
      {
        "distance" : {
          "text" : "52 m",
          "value" : 52
        },
        "duration" : {
          "text" : "1 menit",
          "value" : 21
        },
        "end_location" : {
          "lat" : -6.9029995,
          "lng" : 107.6191971
        },
        "html_instructions" : "Ke arah  

\u003cb\u003etimur\u003c/b\u003e\u003cdiv style=\  

size:0.9em\`\u003eJalan dengan penggunaan  

terbatas\u003c/div\u003e",
        "polyline" : {
          "points" :
"|dci@wizoS?A?_@?A?A@?`@AT?"
        },
        "start_location" : {
          "lat" : -6.9027138,
          "lng" : 107.6190037
        },
        "travel_mode" : "DRIVING"
      }
    ],
    "traffic_speed_entry" : [],
    "via_waypoint" : []
  }
},
"overview_polyline" : {
  "points" : "|dci@wizoS?e@x@A"
},
"summary" : "",
"warnings" : [],
"waypoint_order" : []
}
],
"status" : "OK"
}

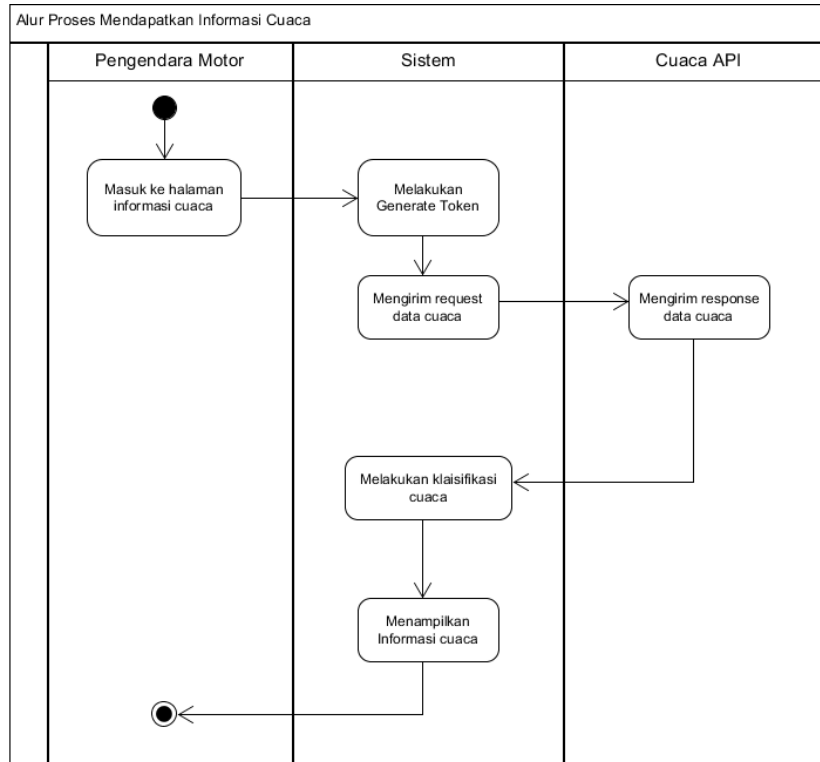
```

#### 3.1.4.4 Cuaca API

Cuaca API pada penelitian ini digunakan untuk menampilkan informasi titik cuaca dan juga sebagai validasi peringatan hujan kepada pengendara. Adapaun pada penelitian ini didapat informasi sebanyak 46 titik sensor cuaca yang meliputi kelurahan dan kecamatan di Kota Bandung. Namun didapat bahwa data yang diberikan oleh sensor cuaca ini terkadang mengalami keterlambatan, berbeda

dengan informasi yang didapat pada pembahasan sebelumnya, sehingga informasi menjadi kurang *real-time*.

Berikut alur proses dari penggunaan Cuaca API pada sistem yang akan dibangun dalam mendapatkan informasi terkait cuaca suatu daerah di wilayah kota bandung ditunjukkan pada gambar 3.11



**Gambar 3.11 Alur Proses Mendapatkan Informasi Cuaca**

Berikut penjelasan dari alur proses yang terdapat pada gambar 3.11

1. Penderita motor masuk ke halaman informasi cuaca yang ada pada sistem
2. Selanjutnya sistem akan melakukan *generate token* untuk mendapatkan *access token*. Dalam melakukan *generate token* diperlukan *username* dan *password* yang telah terdaftar di Cuaca API
3. Setelah itu, sistem akan mengirimkan *request* ke Cuaca API untuk meminta informasi data cuaca, *request* dilakukan dengan menyertakan *access token* sebagai perizinan permintaan data
4. Setelah Cuaca API mengirimkan *response* data cuaca ke sistem, sistem akan melakukan klasifikasi cuaca berdasarkan informasi parameter cuaca yang didapatkan



5. Selanjutnya sistem akan menampilkan informasi cuaca pada sistem ke pengendara motor

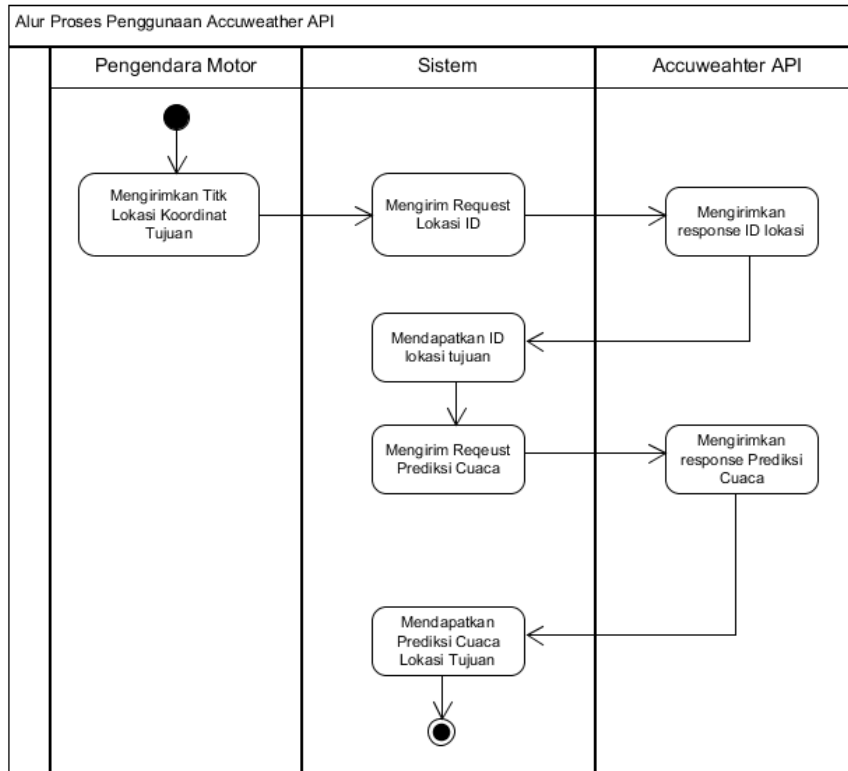
Berikut contoh *response* yang didapatkan dari Cuaca API

**Tabel 3.15 Response Data Cuaca API**

```
[
  [
    {
      "weeklyrainin": "0.00",
      "yearlyrainin": "6.58",
      "baromin": "29.85",
      "windchillf": "79.7",
      "uv": null,
      "indoortempf": "79.9",
      "rainin": "0.00",
      "windgustmph": "2.46",
      "dailyrainin": "0.00",
      "indoorhumidity": "58",
      "time": "2017-09-11T02:56:52.789216144Z",
      "ID": "IBANDUNG4",
      "windspeedmph": "0.22",
      "dewptf": "62.6",
      "humidity": "56",
      "solarradiation": "84.41",
      "monthlyrainin": "0.09",
      "tempf": "79.7",
      "winddir": "97"
    }
  ]
]
```

#### 3.1.4.5 Accuweather API

Accuweather API merupakan layanan servis yang menyediakan informasi dan juga prediksi cuaca. Dalam penelitian ini Accuweather API digunakan untuk memberikan informasi prediksi cuaca lokasi tujuan pengendara ketika pengendara mencari rute. Adapun alur proses dari penggunaan Accuweather API pada sistem yang akan dibangun ditunjukkan pada gambar 3.12



**Gambar 3.12 Alur Proses Mendapatkan Prediksi Cuaca**

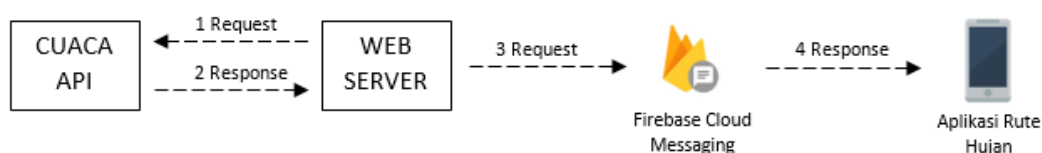
Berikut penjelasan dari alur proses yang terdapat pada gambar 3.12

1. Pengendara akan mengirimkan titik lokasi tujuan ke dalam sistem
2. Selanjutnya sistem akan mengirim titik lokasi tujuan tersebut yang berupa titik koordinat *latitude* dan *longitude* ke Accuweather API untuk mendapatkan ID lokasi tujuan
3. Setelah sistem mendapatkan ID lokasi tersebut, selanjutnya ID lokasi tersebut akan digunakan sebagai parameter permintaan prediksi cuaca tujuan lokasi
4. Setelah sistem mendapatkan informasi prediksi cuaca dari Accuweather API maka akan ditampilkan kepada pengendara

#### 3.1.4.6 Firebase Cloud Messaging

*Firebase Cloud Messaging* (FCM) merupakan layanan yang disediakan oleh *google* untuk mengirimkan pesan notifikasi ke semua perangkat android yang terhubung dengan *server*. Pada penelitian ini FCM digunakan untuk memberikan

notifikasi berupa informasi hujan kepada pengendara motor ketika aplikasi berada di *background* sehingga pengendara motor dapat mengetahui daerah – daerah mana yang sedang terjadi hujan dan banjir secara *real-time*. Pengiriman notifikasi pesan pada aplikasi dilakukan dengan menggunakan *messaging topic* kepada semua perangkat aplikasi yang terdaftar, aplikasi klien akan secara otomatis berlangganan ke topik pada saat pengendara motor melakukan pendaftaran pengguna. Berikut ilustrasi dari penggunaan FCM pada sistem ditunjukkan pada gambar 3.13



**Gambar 3.13 Notifikasi Banjir dengan FCM**

Berikut tahapan dari penggunaan *firebase cloud messaging* pada sistem ketika pengendara melaporkan laporan banjir sebagai berikut :

1. *Web server* akan mengirimkan *request* secara berkala untuk mendapatkan informasi cuaca terbaru ke Cuaca API.
2. Selanjutnya Cuaca API akan mengirimkan *response* berupa informasi data cuaca dari setiap sensor cuaca yang ada kembali ke *web server*.
3. Dari data cuaca yang didapat maka *web server* akan melihat apakah terdapat titik hujan dari data cuaca sensor yang ada. Jika ada maka *web server* akan melakukan *request* pengiriman pesan ke FCM (*firebase cloud messaging*)
4. Setelah FCM menerima *request* dari *web server*, FCM akan meneruskan pesan berupa informasi daerah – daerah hujan dalam bentuk *response* kepada seluruh aplikasi rute hujan dalam bentuk *push notification*.

### 3.1.5 Analisis Spesifikasi Kebutuhan Perangkat Lunak

Analisis spesifikasi kebutuhan perangkat lunak dilakukan untuk memberikan gambaran kebutuhan apa saja yang terdapat pada perangkat lunak yang akan dibangun serta spesifikasi kebutuhan apa saja yang diperlukan dalam menjalankan sistem. Spesifikasi kebutuhan perangkat lunak pada sistem yang akan

diabangun terbagi menjadi dua yaitu spesifikasi kebutuhan fungsional dan spesifikasi kebutuhan non-fungsional.

Spesifikasi kebutuhan perangkat lunak fungsional akan memberikan deskripsi hal apa saja yang terdapat pada sistem yang harus disediakan. Adapun analisis yang dilakukan yaitu pada sistem dengan *platform mobile*. Berikut spesifikasi kebutuhan perangkat lunak fungsional ditunjukkan pada tabel 3.16

**Tabel 3.16 Spesifikasi Kebutuhan Perangkat Lunak Fungsional**

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-F-001	Sistem <i>mobile</i> menyediakan fasilitas bagi pengendara motor untuk dapat melakukan pendaftaran pengguna
SKPL-F-002	Sistem <i>mobile</i> menyediakan fasilitas bagi pengendara motor untuk dapat melakukan <i>login</i> pengguna
SKPL-F-003	Sistem <i>mobile</i> menyediakan fasilitas bagi pengendara motor untuk memberikan informasi akun ketika pengendara lupa password
SKPL-F-004	Sistem <i>mobile</i> menyediakan fasilitas bagi pengendara motor untuk menampilkan informasi titik cuaca pada <i>map</i>
SKPL-F-005	Sistem <i>mobile</i> menyediakan fasilitas bagi pengendara motor untuk menampilkan informasi hujan dan banjir berdasarkan rute perjalanan
SKPL-F-006	Sistem <i>mobile</i> menyediakan fasilitas bagi pengendara motor untuk mendapatkan notifikasi peringatan hujan saat terdapat titik hujan pada rute perjalanan pengendara
SKPL-F-007	Sistem <i>mobile</i> menyediakan fasilitas bagi pengendara motor untuk mendapatkan rekomendasi jalur alternatif dalam menghindari banjir pada rute perjalanannya
SKPL-F-008	Sistem <i>mobile</i> menyediakan fasilitas bagi pengendara motor untuk menampilkan informasi prediksi cuaca lokasi tujuannya
SKPL-F-009	Sistem <i>mobile</i> menyediakan fasilitas bagi pengendara motor untuk menampilkan informasi laporan banjir
SKPL-F-010	Sistem <i>mobile</i> menyediakan fasilitas bagi pengendara motor untuk melakukan pelaporan lokasi banjir terkini
SKPL-F-011	Sistem <i>mobile</i> menyediakan fasilitas bagi pengendara motor untuk dapat melakukan <i>logout</i> pengguna

Selain spesifikasi kebutuhan fungsional, adapun kebutuhan spesifikasi non-fungsional. Kebutuhan non-fungsional memberikan batasan – batasan layanan yang ditawarkan oleh sistem yang akan dibangun. Adapun spesifikasi perangkat lunak kebutuhan non-fungsional dilakukan pada platform *mobile*. Berikut spesifikasi kebutuhan perangkat lunak non-fungsional pada *platform mobile* ditunjukkan pada tabel 3.17

**Tabel 3.17 Spesifikasi Kebutuhan Perangkat Lunak Non-Fungsional**

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-NF-001	Sistem <i>mobile</i> yang dibangun berbasis sistem operasi android
SKPL-NF-002	Sistem <i>mobile</i> dapat dijalankan pada sistem android dengan versi minimal 5.0 (Lollipop)
SKPL-NF-003	Sistem <i>mobile</i> memerlukan akses GPS pada <i>smartphone</i>
SKPL-NF-004	Sistem <i>mobile</i> perlu dijalankan dengan menggunakan akses internet
SKPL-NF-005	Sistem <i>mobile</i> dapat diakses selama 24 jam

### 3.1.6 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non-fungsional dilakukan untuk menentukan spesifikasi kebutuhan yang diperlukan oleh sistem yang akan dibangun agar dapat berjalan dengan baik. Spesifikasi ini meliputi sekumpulan elemen dan komponen apa saja yang diperlukan oleh sistem yang akan dibangun sebelum diimplementasikan. Selain itu ditentukan pula spesifikasi masukan apa saja yang diperlukan oleh sistem hingga menghasilkan suatu keluaran. Dalam kebutuhan non-fungsional dilakukan beberapa analisis yaitu analisis kebutuhan perangkat keras, analisis kebutuhan perangkat lunak, serta analisis pengguna.

#### 3.1.6.1 Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras dilakukan untuk menentukan kebutuhan non-fungsional yang berkaitan dengan spesifikasi perangkat keras yang diperlukan oleh sistem agar aplikasi dapat berjalan dengan baik. Adapun spesifikasi kebutuhan perangkat keras yang dilakukan terhadap sistem yang berjalan pada *platform mobile*. Berikut spesifikasi kebutuhan perangkat keras minimum yang dibutuhkan oleh aplikasi pada *platform mobile* ditunjukkan pada tabel 3.18

**Tabel 3.18 Spesifikasi Kebutuhan Perangkat Keras Pada Platform Mobile**

No	Spesifikasi Kebutuhan Perangkat Keras
1	Minimum Processor Chipset Quad-core 1.2 GHz Cortex-A53
2	Minimum RAM 1 GB
3	Minimum <i>Memory Internal</i> 8 GB
4	Memiliki sensor GPS
5	Memiliki fitur <i>Camera</i> dengan resolusi 5.0 Megapixel
6	Memiliki Koneksi Internet 3G/4G

### 3.1.6.2 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak merupakan spesifikasi perangkat lunak (*software*) yang diperlukan dalam membangun aplikasi yang akan dibangun serta kebutuhan yang diperlukan agar aplikasi dapat berjalan. Adapun spesifikasi kebutuhan perangkat lunak yang ditunjukkan pada tabel 3.19

**Tabel 3.19 Spesifikasi kebutuhan perangkat lunak**

Jenis	Spesifikasi
Sistem Operasi Mobile	Android OS 5.0 Lollipop
Sistem Operasi Desktop	Microsoft Windows Xp
IDE ( <i>Integrated Development Environment</i> )	Android Studio 3.3
Kode Editor	Sublime 3
Mockup Tool	Balsamiq Mockup
UML Tool	UMLet

### 3.1.6.3 Analisis Pengguna

Analisis pengguna dimaksudkan untuk menentukan kriteria pengguna yang dapat berinteraksi serta menjalankan sistem yang akan dibangun. Analisis pengguna bertujuan untuk mengetahui karakteristik pengguna yang terlibat dengan sistem sehingga sistem dapat dijalankan dengan optimal. Berikut karakteristik pengguna yang terlibat dengan sistem ditunjukkan pada tabel 3.20

**Tabel 3.20 Analisis Pengguna**

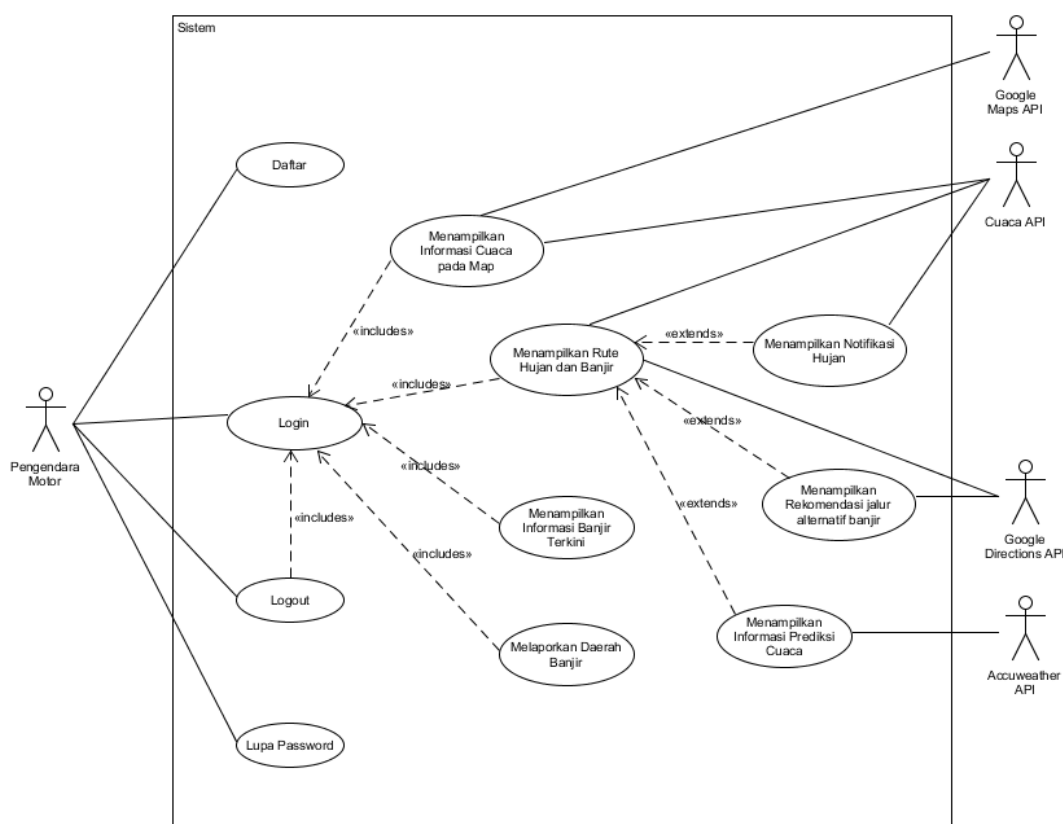
Pengguna	Karakteristik
Pengendara Motor	Memiliki pengetahuan dasar terhadap penggunaan aplikasi di android
	Tidak memiliki masalah dalam penglihatan
	Terbiasa mengoperasikan perangkat smartphone android
	Memiliki pemahaman dalam menggunakan aplikasi navigasi rute

### 3.1.7 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dimaksudkan untuk memberikan gambaran proses kegiatan yang terdapat dalam sistem serta menjelaskan kebutuhan fungsional apa saja yang diperlukan oleh sistem agar sistem dapat berjalan dengan baik. Analisis kebutuhan fungsional yang dilakukan dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Adapun tahapan – tahapan pemodelan spesifikasi kebutuhan fungsional menggunakan UML meliputi *use case diagram*, *activity diagram*, *class diagram*, *sequence diagram*.

#### 3.1.7.1 Use Case Diagram

*Use Case* diagram menggambarkan hubungan antara aktor atau pengguna yang terlibat dengan sistem dengan sistem yang akan dibangun. Aktor merupakan seorang yang berinteraksi langsung dengan sistem, sedangkan *use case* menunjukkan fungsionalitas apa saja yang terdapat dalam sistem. Berikut merupakan diagram *use case* dari sistem yang akan dibangun ditunjukkan pada gambar 3.14



**Gambar 3.14 Use Case Diagram**

Berikut definisi dan peran aktor yang terlibat dalam sistem yang dibangun ditunjukkan pada tabel 3.21

**Tabel 3.21 Deskripsi Aktor**

<b>Aktor</b>	<b>Deskripsi</b>
Pengendara Motor	Aktor utama yang berperan untuk menggunakan fungsionalitas pada sistem yang dibangun pada <i>platform mobile</i> pada level <i>interface</i>
<i>Google Maps API</i>	Aktor pendukung yang berperan memberikan data yang dibutuhkan oleh sistem dalam menampilkan <i>map</i>
<i>Google Directions API</i>	Aktor pendukung yang berperan memberikan data yang dibutuhkan oleh sistem dalam menampilkan rute jalur
Cuaca API	Aktor pendukung yang berperan memberikan data yang dibutuhkan oleh sistem dalam mendapatkan informasi parameter cuaca
<i>Accuweather API</i>	Aktor pendukung yang berperan memberikan data yang dibutuhkan oleh sistem dalam mendapatkan informasi prediksi cuaca

Berikut deskripsi dari *use case* yang terdapat dalam sistem yang dibangun ditunjukkan pada tabel 3.22

**Tabel 3.22 Deskripsi Use Case**

<b>No</b>	<b>Use Case</b>	<b>Deskripsi</b>
1.	Daftar	Sistem menyediakan fasilitas bagi pengendara motor untuk melakukan pendaftaran pengguna baru pada aplikasi <i>mobile</i>
2.	Login	Sistem menyediakan fasilitas bagi pengendara motor untuk melakukan <i>login</i> untuk dapat menggunakan aplikasi <i>mobile</i>
3.	Lupa Password	Sistem <i>mobile</i> menyediakan fasilitas bagi pengendara motor untuk memberikan informasi akun ketika pengendara lupa password
4.	Menampilkan Informasi Cuaca Pada Map	Sistem menyediakan fasilitas bagi pengendara motor untuk menampilkan informasi cuaca berupa <i>marker</i> pada map yang berisi informasi kondisi cuaca dari sensor yang ada
5.	Menampilkan Rute Hujan dan Banjir	Sistem menyediakan fasilitas bagi pengendara motor untuk menampilkan rute dan juga informasi daerah hujan dan banjir berdasarkan rute perjalanannya
6.	Menampilkan Notifikasi Hujan	Sistem menyediakan fasilitas bagi pengendara motor untuk mendapatkan notifikasi berupa peringatan hujan dan rekomendasi pemakaian jas hujan ketika daerah pada rute perjalanannya terindikasi hujan



No	Use Case	Deskripsi
7.	Menampilkan Rekomendasi Jalur Alternatif Banjir	Sistem menyediakan fasilitas bagi pengendara motor untuk memberikan rekomendasi jalur alternatif banjir untuk menghindari titik lokasi banjir berdasarkan laporan yang ada
8.	Mendapatkan informasi prediksi cuaca	Sistem menyediakan fasilitas bagi pengendara motor untuk mendapatkan informasi prediksi kondisi cuaca pada lokasi tujuannya pada jam selanjutnya
9.	Menampilkan Informasi Banjir Terkini	Sistem menyediakan fasilitas bagi pengendara motor dalam memperoleh informasi lokasi banjir terkini dari pelaporan banjir yang ada
10.	Melaporkan Titik Banjir	Sistem menyediakan fasilitas bagi pengendara motor untuk melaporkan titik lokasi banjir daerah tertentu
11.	Logout	Sistem <i>mobile</i> menyediakan fasilitas bagi pengendara motor untuk dapat melakukan <i>logout</i> pengguna

### 3.1.7.2 Skenario Use Case

*Use case scenario* digunakan untuk memberi pemahaman lebih jauh dari setiap proses yang terdapat pada *use case* yang dibangun. Pada *Use Case Scenario* dijelaskan setiap tahapan alur proses berupa interaksi aktor dengan *use case* yang ada pada sistem. Adapun *use case scenario* dari setiap *use case* yaitu sebagai berikut.

#### 1. Use Case Daftar

Berikut adalah skenario *use case* daftar yang ditunjukkan pada tabel 3.23

**Tabel 3.23 Skenario Use Case Daftar**

Nama Use Case:	Daftar	
ID Use Case	SUC-001	
Prioritas	Tinggi	
Pelaku Sistem Utama	Pengendara Motor	
Pelaku Partisipan Lain	-	
Prakondisi:	Pengendara berada di halaman login	
Pemicu:	Pengendara motor memilih menu “daftar”	
	Kegiatan Pelaku	Respons Sistem

Bidang Khas Suatau Event:	<b>Langkah-1:</b> Pengendara masuk ke halaman pendaftaran	<b>Langkah-2:</b> Sistem menampilkan form identitas pendaftaran pengguna
	<b>Langkah-3:</b> Pengendara memasukkan data diri berupa nama, <i>email</i> dan password, alamat ke form pendaftaran pengguna.	
	<b>Langkah-4:</b> Pengendara motor memilih tombol daftar.	<b>Langkah-5:</b> Sistem melakukan validasi data pendaftaran pengguna.
		<b>Langkah-6:</b> Sistem mengirimkan kode verifikasi dan menampilkan halaman “verifikasi kode”.
	<b>Langkah-7:</b> Pengendara memasukkan kode verifikasi dengan benar.	<b>Langkah-8:</b> Sistem melakukan aktivasi akun pengendara.
Bidang Alternatif:	<b>Alt-Step-5:</b> Jika data pengendara motor tidak valid maka sistem akan menampilkan pesan kesalahan	
Postkondisi:	Pengendara motor berhasil melakukan pendaftaran akun pengguna baru	

## 2. Use Case Login Pengendara

Berikut adalah skenario *use case* login bagi pengendara motor yang ditunjukkan pada tabel 3.24

**Tabel 3.24 Skenario Use Case Login Pengendara**

Nama Use Case:	Login Pengendara	
ID Use Case	SUC-002	
Prioritas	Tinggi	
Pelaku Sistem Utama	Pengendara Motor	
Pelaku Partisipan Lain	-	
Prakondisi:	Pengendara telah melakukan pendaftaran pengguna	
Pemicu:	Pengendara motor memilih menu “login”	
	Kegiatan Pelaku	Responns Sistem

Bidang Khas Suatu Event:	<b>Langkah-1:</b> Pengendara masuk ke halaman login	<b>Langkah-2:</b> Sistem menampilkan form <i>login</i> pengguna
	<b>Langkah-3:</b> Pengendara memasukan data berupa <i>email</i> dan <i>password</i> ke form <i>login</i> pengguna.	
	<b>Langkah-4:</b> Pengendara motor memilih tombol login.	<b>Langkah-5:</b> Sistem melakukan validasi data <i>login</i> pengendara motor
		<b>Langkah-6:</b> Sistem menampilkan pesan berhasil melakukan <i>login</i> dan menampilkan halaman utama.
Bidang Alternatif:	<b>Alt-Step-5:</b> Jika data <i>login</i> tidak valid maka sistem akan menampilkan pesan kesalahan pada halaman login	
Postkondisi:	Pengendara motor berhasil melakukan login, dan sistem menampilkan halaman utama	

### 3. Use Case Lupa Password

Berikut adalah skenario *use case* lupa password bagi pengendara motor yang ditunjukkan pada tabel 3.25

**Tabel 3.25 Skenario Use Case Login Pengendara**

Nama Use Case:	Lupa Password	
ID Use Case	SUC-003	
Prioritas	Tinggi	
Pelaku Sistem Utama	Pengendara Motor	
Pelaku Partisipan Lain	-	
Prakondisi:	Pengendara telah melakukan pendaftaran pengguna	
Pemicu:	Pengendara motor memilih menu “lupa password”	
Bidang Khas Suatu Event:	Kegiatan Pelaku	Responns Sistem
	<b>Langkah-1:</b> Pengendara masuk ke halaman lupa password	<b>Langkah-2:</b> Sistem menampilkan form halaman lupa password
	<b>Langkah-3:</b> Pengendara memasukan data berupa <i>email</i> ke dalam form.	

	<b>Langkah-4:</b> Pengendara motor memilih tombol kirim password.	<b>Langkah-5:</b> Sistem mencari data email yang telah terdaftar
		<b>Langkah-6:</b> Sistem mengirimkan informasi akun kepada pengguna melalui email.
Bidang Alternatif:	<b>Alt-Step-5:</b> Jika data tidak ditemukan maka akan ditampilkan pesan bahwa email belum terdaftar	
Postkondisi:	Pengendara motor berhasil mendapatkan informasi akun pengendara berupa email dan password	

#### 4. Use Case Menampilkan Informasi Cuaca Pada Map

Berikut adalah skenario *use case* untuk menampilkan informasi cuaca pada *map* yang ditunjukkan pada tabel 3.26

**Tabel 3.26 Skenario Use Case Menampilkan Informasi Cuaca Pada Map**

Nama Use Case:	Menampilkan Informasi Cuaca pada Map	
ID Use Case	SUC-004	
Prioritas	Tinggi	
Pelaku Sistem Utama	Pengendara Motor	
Pelaku Partisipan Lain	<i>Google Maps</i> API, Cuaca API	
Prakondisi:	Pengendara motor telah melakukan login	
Pemicu:	Pengendara motor masuk ke halaman utama	
Bidang Khas Suatu Event:	Kegiatan Pelaku	Respons Sistem
	<b>Langkah-1:</b> Pengendara masuk ke halaman utama.	<b>Langkah-2:</b> Sistem mengirimkan <i>request</i> data map ke <i>Google Maps</i> API
		<b>Langkah-3:</b> Sistem mendapatkan <i>response</i> data maps dan menampilkan <i>map</i> pada halaman utama
		<b>Langkah-4:</b> Sistem mengirimkan <i>request</i> ke Cuaca API untuk mendapatkan data parameter cuaca
		<b>Langkah-5:</b> Sistem melakukan klasifikasi parameter cuaca untuk

		menentukan kondisi cuaca yang didapat
		<b>Langkah-6:</b> Sistem menampilkan informasi cuaca pada <i>map</i> dalam bentuk <i>icon</i>
Bidang Alternatif:	<b>Alt-Step-4:</b> Jika sistem tidak mendapatkan <i>response</i> data cuaca sistem tidak akan menampilkan informasi cuaca pada <i>map</i>	
Postkondisi:	Pengendara motor berhasil mendapatkan informasi cuaca yang ditampilkan pada sebuah <i>map</i>	

### 5. Use Case Menampilkan Rute Hujan dan Banjir

Berikut adalah skenario *use case* untuk menampilkan rute hujan dan banjir yang ditunjukkan pada tabel 3.27

**Tabel 3.27 Skenario Use Case Menampilkan Rute Hujan dan Banjir**

Nama Use Case:	Menampilkan Rute Hujan dan Banjir	
ID Use Case	SUC-005	
Prioritas	Tinggi	
Pelaku Sistem Utama	Pengendara Motor	
Pelaku Partisipan Lain	<i>Google Directions</i> API, Cuaca API	
Prakondisi:	Pengendara motor telah masuk ke halaman utama	
Pemicu:	Pengendara motor memilih tempat pada kolom pencarian	
Bidang Khas Suatu Event:	Kegiatan Pelaku	Respons Sistem
	<b>Langkah-1:</b> Pengendara memasukan alamat tempat tujuan pengendara	<b>Langkah-2:</b> Sistem mendapatkan lokasi pengendara dan lokasi tujuan lalu mengirimkan <i>request</i> ke <i>Google Directions</i> API
		<b>Langkah-3:</b> Sistem mendapatkan <i>response</i> rute lalu lintas
		<b>Langkah-4:</b> Sistem mengirimkan <i>request</i> ke Cuaca API untuk mendapatkan data parameter cuaca dan posisi sensor cuaca
		<b>Langkah-8:</b> Sistem melakukan klasifikasi parameter cuaca untuk

		menentukan kondisi cuaca yang didapat
		<b>Langkah-9:</b> Sistem mencari data daerah pelaporan banjir
		<b>Langkah-10:</b> sistem menentukan titik banjir dan hujan pada rute pengendara
		<b>Langkah-11:</b> Sistem menampilkan rute dan informasi cuaca dan banjir pada sebuah <i>map</i> dalam sistem
Bidang Alternatif:	<b>Alt-Step-2:</b> Jika sistem tidak mendapatkan <i>response</i> data rute maka rute tidak akan ditampilkan	
	<b>Alt-Step-4:</b> Jika sistem tidak mendapatkan <i>response</i> data cuaca maka informasi cuaca tidak akan ditampilkan	
	<b>Alt-Step-9:</b> Jika data laporan banjir tidak ditemukan maka sistem tidak akan menentukan titik banjir pada rute	
Postkondisi:	Pengendara motor berhasil mendapatkan informasi cuaca dan banjir berdasarkan rute perjalanan yang ditampilkan pada <i>map</i>	

## 6. Use Case Menampilkan Notifikasi Hujan

Berikut adalah skenario *use case* menampilkan notifikasi hujan yang ditunjukkan pada tabel 3.28

**Tabel 3.28 Skenario Use Case Menampilkan Notifikasi Hujan**

Nama Use Case:	Menampilkan Notifikasi Hujan	
ID Use Case	SUC-006	
Prioritas	Tinggi	
Pelaku Sistem Utama	Pengendara Motor	
Pelaku Partisipan Lain	Cuaca API	
Prakondisi:	Pengendara motor telah memasukan alamat tujuan pengendara dan memiliki rute perjalanan	
Pemicu:	Pengendara motor memilih tempat tujuan	
Bidang Khas Suatu	Kegiatan Pelaku	Respons Sistem
Event:	<b>Langkah-1:</b> Pengendara mengirimkan lokasi pengendara berada	<b>Langkah-2:</b> Sistem melakukan <i>request</i> ke Cuaca API untuk

		mendapatkan informasi parameter cuaca sensor terdekat
		<b>Langkah-3:</b> Sistem melakukan klasifikasi parameter cuaca untuk mengetahui kondisi cuaca
		<b>Langkah-4:</b> Jika kondisi cuaca yang didapat adalah hujan, Sistem akan melakukan pengecekan jarak titik lokasi sensor dengan titik lokasi pengendara
		<b>Langkah-5:</b> Sistem menampilkan notifikasi hujan dengan informasi lokasi dan jarak terdekat titik hujan
Bidang Alternatif:	<b>Alt-Step-4:</b> Jika kondisi cuaca menyatakan tidak hujan maka sistem tidak akan menampilkan notifikasi peringatan hujan kepada pengguna	
Postkondisi:	Pengendara motor mendapatkan notifikasi peringatan hujan pada saat mendekati daerah hujan pada rute perjalanannya	

#### 7. Use Case Menampilkan Rekomendasi Jalur Alternatif Banjir

Berikut adalah skenario *use case* menampilkan rekomendasi jalur alternatif banjir yang ditunjukkan pada tabel 3.29

**Tabel 3.29 Skenario Use Case Menampilkan Rekomendasi Jalur Alternatif Banjir**

Nama Use Case:	Menampilkan Rekomendasi Jalur Alternatif Banjir	
ID Use Case	SUC-007	
Prioritas	Tinggi	
Pelaku Sistem Utama	Pengendara Motor	
Pelaku Partisipan Lain	<i>Google Directions API</i>	
Prakondisi:	Pengendara motor telah memasukan alamat tujuan lokasi pengendara	
Pemicu:	Sistem mendeteksi bahwa terdapat titik banjir pada daerah rute perjalanan pengendara	
	Kegiatan Pelaku	Respons Sistem

Bidang Khas Suatau Event:	<b>Langkah-1:</b> Pengendara memasukkan lokasi tujuan pengendara	<b>Langkah-2:</b> Sistem mengirimkan <i>request</i> ke <i>Google Directions API</i> untuk mendapatkan rute
		<b>Langkah-2:</b> Sistem melakukan pengecekan titik banjir pada rute perjalanan
		<b>Langkah-2:</b> Jika terdeteksi banjir sistem akan mengirimkan <i>request</i> ke <i>Google Directions API</i> untuk mendapatkan rute alternatif lain sesuai lokasi tujuan pengendara
		<b>Langkah-3:</b> Sistem mendapatkan <i>response</i> , dan sistem memberikan rekomendasi jalur alternatif kepada pengendara motor
		<b>Langkah-5:</b> Sistem menampilkan rute pada <i>map</i>
Bidang Alternatif:	<b>Alt-Step-1:</b> Jika titik banjir tidak ditemukan maka sistem tidak akan memberikan rekomendasi rute jalur alternatif	
Postkondisi:	Pengendara motor mendapatkan rute jalur alternatif lain dalam menghindari titik lokasi banjir	

## 8. Use Case Mendapatkan Informasi Prediksi Cuaca

Berikut adalah skenario *use case* menampilkan informasi prediksi cuaca yang ditunjukkan pada tabel 3.30

**Tabel 3.30 Skenario Use Case Menampilkan Info Prediksi Cuaca**

Nama Use Case:	Menampilkan Informasi Prediksi Cuaca
ID Use Case	SUC-008
Prioritas	Tinggi
Pelaku Sistem Utama	Pengendara Motor
Pelaku Partisipan Lain	<i>Accuweather API</i>
Prakondisi:	Pengendara motor memasukkan alamat lokasi tujuan pengendara
Pemicu:	Pengendara motor memilih tombol “prediksi cuaca”



Bidang Khas Suatu	Kegiatan Pelaku	Respons Sistem
Event:	<b>Langkah-1:</b> Pengendara motor menekan tombol “prediksi cuaca”	<b>Langkah-2:</b> Sistem melakukan <i>request</i> ke <i>Accuweather</i> API untuk mendapatkan informasi prediksi berdasarkan titik tujuan pengendara
		<b>Langkah-3:</b> Sistem mendapatkan <i>response</i> data berupa prediksi cuaca pada jam – jam selanjutnya
		<b>Langkah-4:</b> Sistem menampilkan informasi prediksi cuaca
Bidang Alternatif:	<b>Alt-Step-3:</b> Jika sistem gagal mendapatkan data prediksi cuaca maka informasi prediksi tidak akan ditampilkan	
Postkondisi:	Pengendara motor mendapatkan informasi terkait prediksi cuaca berdasarkan lokasi pengendara berada	

### 9. Use Case Menampilkan Informasi Banjir Terkini

Berikut adalah skenario *use case* menampilkan informasi banjir terkini kepada pengendara yang ditunjukkan pada tabel 3.31

**Tabel 3.31 Skenarion Use Case Menampilkan Informasi Banjir Terkini**

Nama Use Case:	Menampilkan Informasi Banjir Terkini	
ID Use Case	SUC-009	
Prioritas	Tinggi	
Pelaku Sistem Utama	Pengendara Motor	
Pelaku Partisipan Lain	-	
Prakondisi:	Pengendara motor telah masuk ke halaman utama	
Pemicu:	Pengendara motor memilih menu “info banjir”	
Bidang Khas Suatu Event:	Kegiatan Pelaku	Respons Sistem
	<b>Langkah-1:</b> Pengendara motor memilih menu “informasi banjir”	<b>Langkah-2:</b> Sistem melakukan pencarian data laporan banjir terkini yang dikirim pengendara motor lain
		<b>Langkah-3:</b> Sistem mendapatkan data laporan banjir terkini
		<b>Langkah-4:</b> Sistem menampilkan informasi laporan banjir

Bidang Alternatif:	<b>Alt-Step-3:</b> Jika sistem tidak dapat menemukan data laporan banjir terkini, maka sistem mengirim pesan kepada pengguna bahwa informasi banjir tidak ditemukan
Postkondisi:	Pengendara motor dapat memperoleh informasi banjir terkini pada halaman menu informasi banjir

## 10. Use Case Melaporkan Titik Banjir

Berikut adalah skenario *use case* melaporkan titik banjir yang ditunjukkan pada tabel 3.32

**Tabel 3.32 Skenario Use Case Melaporkan Titik Banjir**

Nama Use Case:	Melaporkan Titik Banjir	
ID Use Case	SUC-010	
Prioritas	Tinggi	
Pelaku Sistem Utama	Pengendara Motor	
Pelaku Partisipan Lain	-	
Prakondisi:	Pengendara motor telah masuk ke halaman utama	
Pemicu:	Pengendara motor memilih menu “lapor banjir”	
Bidang Khas Suatu Event:	Kegiatan Pelaku	Respons Sistem
	<b>Langkah-1:</b> Pengendara masuk ke halaman laporan banjir	<b>Langkah-2:</b> Sistem menampilkan halaman laporan banjir yang berisi form pelaporan pengendara
	<b>Langkah-3:</b> Pengendara motor memasukan data laporan banjir berupa foto dan lokasi pengirim	
	<b>Langkah-4:</b> Pengendara motor memilih tombol “kirim laporan”	<b>Langkah-5:</b> Sistem melakukan validasi data banjir yang dikirim
		<b>Langkah-6:</b> Sistem mengirimkan <i>request</i> ke Cuaca API untuk mendapatkan histori cuaca
		<b>Langkah-7:</b> Sistem melakukan validasi histori cuaca

		<b>Langkah-8:</b> Sistem menyimpan data laporan banjir ke dalam sistem
Bidang Alternatif:	<b>Alt-Step-3:</b> Jika data laporan banjir tidak lengkap maka akan tampil pesan kesalahan	
Postkondisi:	Pengendara motor dapat melakukan pelaporan banjir pada titik lokasi pengendara berada	

## 11. Use Case Logout

Berikut adalah skenario *use case* melaporkan titik banjir yang ditunjukkan pada tabel 3.33

**Tabel 3.33 Skenario Use Case Logout**

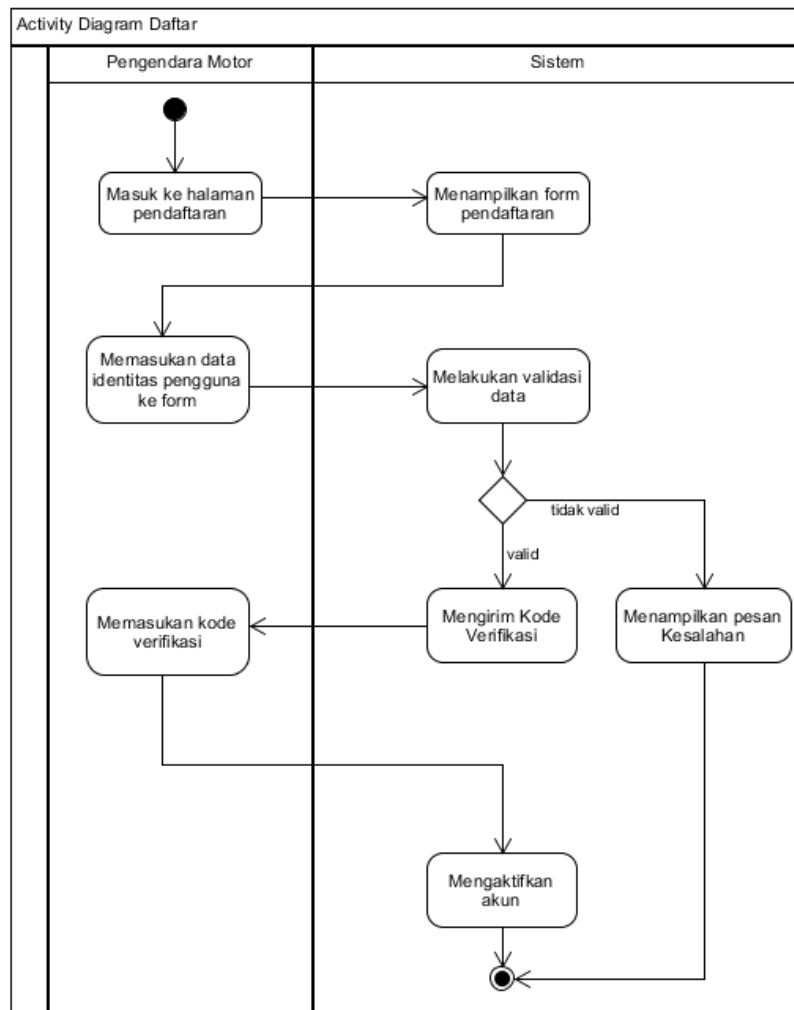
Nama Use Case:	Logout	
ID Use Case	SUC-011	
Prioritas	Tinggi	
Pelaku Sistem Utama	Pengendara Motor	
Pelaku Partisipan Lain	-	
Prakondisi:	Pengendara telah melakukan login	
Pemicu:	Pengendara motor memilih menu “keluar”	
Bidang Khas Suatu Event:	Kegiatan Pelaku	Respons Sistem
	<b>Langkah-1:</b> Pengendara memilih menu “Keluar”	<b>Langkah-2:</b> Sistem akan menghapus data pengguna dalam aplikasi
		<b>Langkah-3:</b> Sistem keluar dari halaman utama dan menampilkan halaman login
		<b>Langkah-4:</b> Sistem menampilkan pesan berhasil melakukan <i>logout</i> .
Postkondisi:	Pengendara motor berhasil melakukan logout, dan sistem menampilkan halaman login	

### 3.1.7.3 Activity Diagram

*Activity Diagram* merupakan suatu model yang memberi gambaran urutan proses atau aktivitas dalam sebuah *use case*. *Activity diagram* bertujuan memberikan pemahaman terhadap proses bisnis dari sebuah sistem secara keseluruhan. Adapun pembuatan *activity diagram* didasarkan pada *use case diagram* dan *scenario use case* yang telah dirancang sebelumnya. Berikut rincian dari *activity diagram* yang telah disusun yaitu sebagai berikut

#### 1. *Activity Diagram* Daftar

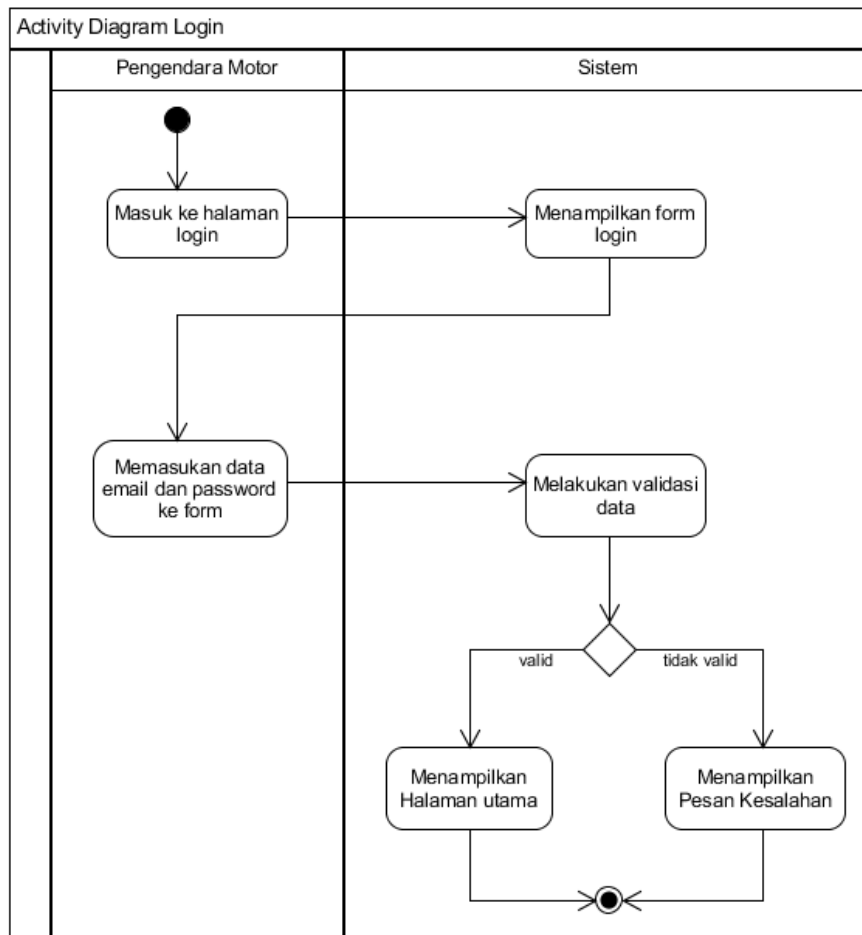
Berikut adalah skenario *activity diagram* daftar yang ditunjukkan pada gambar 3.15



**Gambar 3.15 Acitivity Diagram Daftar**

## 2. Activity Diagram Login

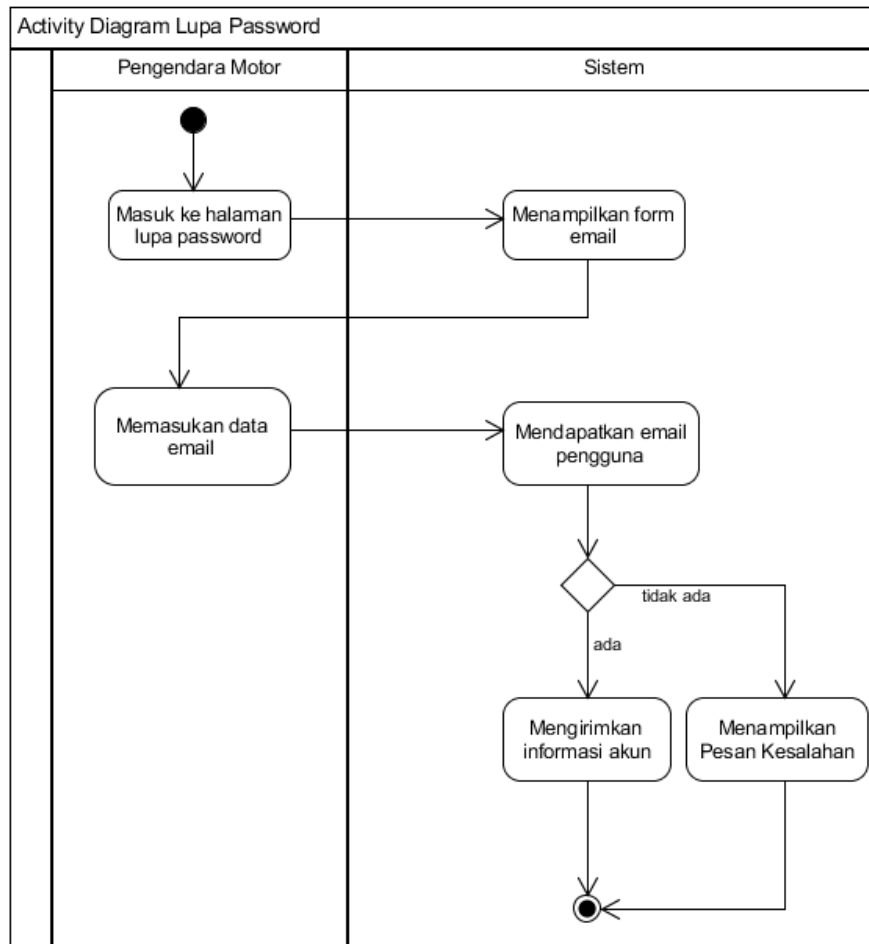
Berikut adalah skenario *activity diagram* login oleh pengendara motor yang ditunjukkan pada gambar 3.16



**Gambar 3.16 Activity Diagram Login Pengendara**

### 3. Activity Diagram Lupa Password

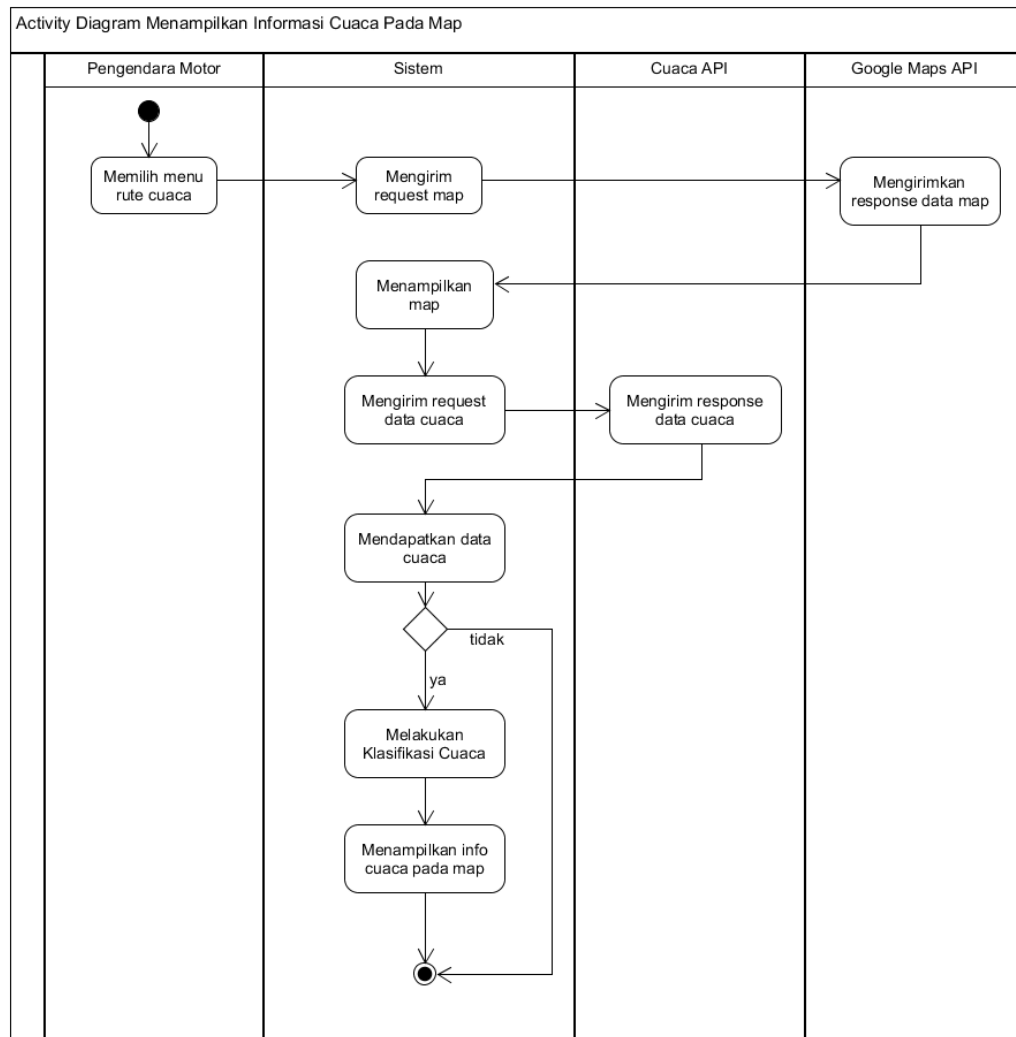
Berikut adalah skenario *activity diagram* lupa password oleh pengendara motor yang ditunjukkan pada gambar 3.17



**Gambar 3.17 Activity Diagram Lupa Password**

#### 4. *Activity Diagram* Menampilkan Informasi Cuaca Pada Map

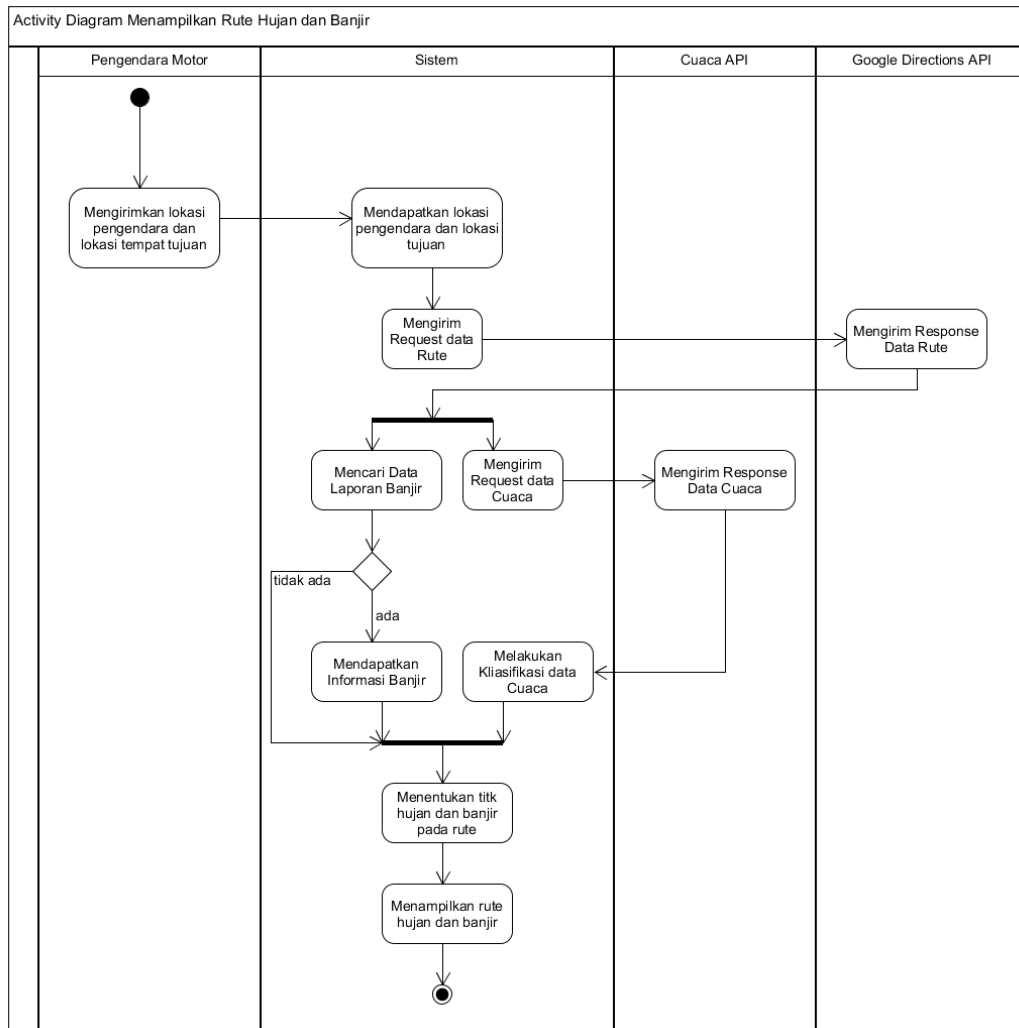
Berikut adalah skenario *activity diagram* menampilkan informasi cuaca pada *map* yang ditunjukkan pada gambar 3.18



**Gambar 3.18 Activity Diagram Menampilkan Informasi Cuaca Pada Map**

### 5. Activity Diagram Menampilkan Rute Hujan dan Banjir

Berikut adalah skenario *activity diagram* menampilkan rute hujan dan banjir yang ditunjukkan pada gambar 3.19

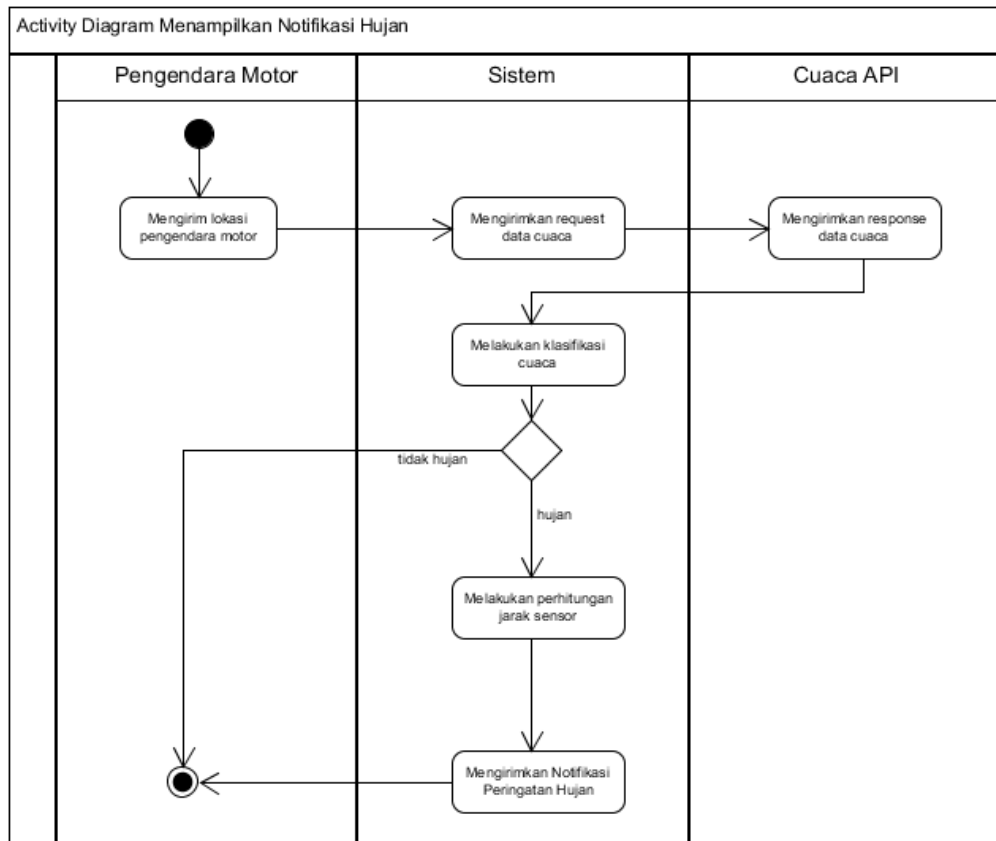


**Gambar 3.19 Activity Diagram Menampilkan Rute Hujan dan Banjir**



## 6. Activity Diagram Menampilkan Notifikasi Hujan

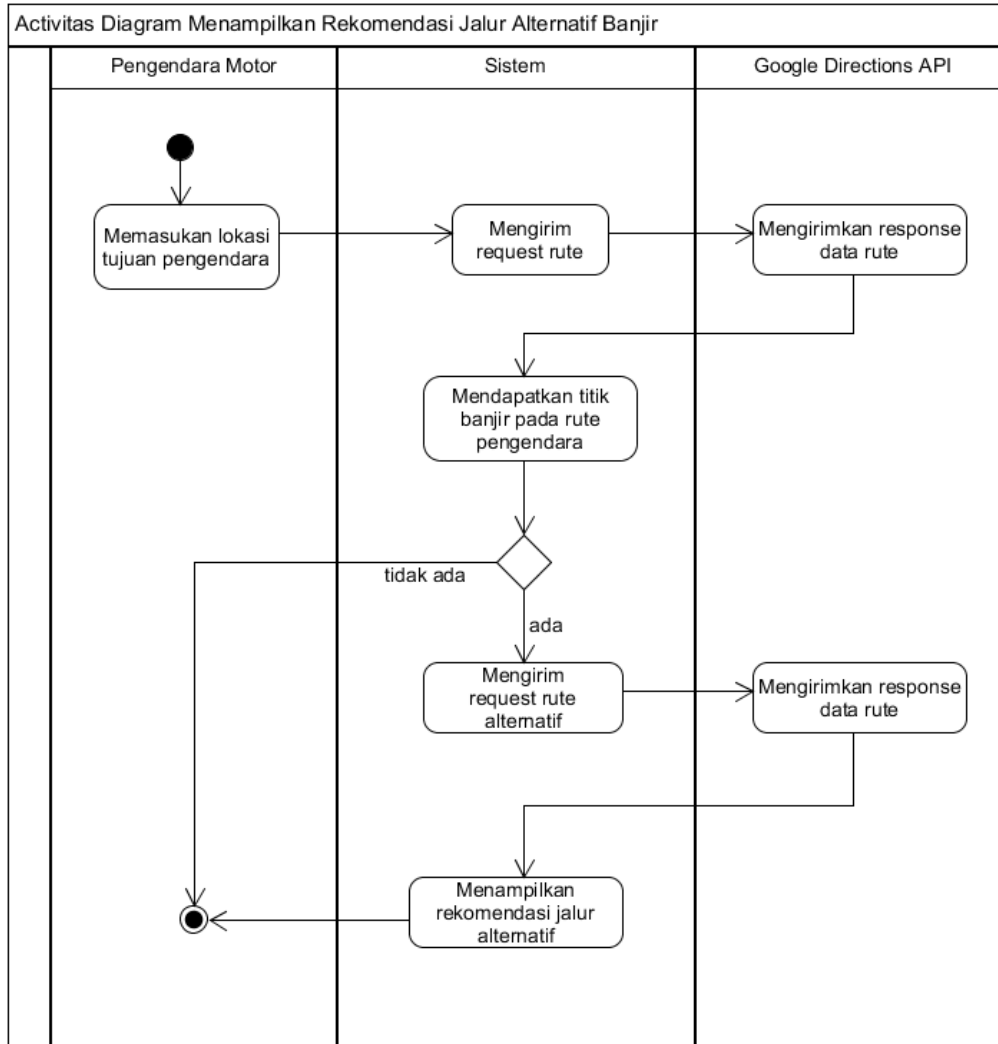
Berikut adalah skenario *activity diagram* menampilkan notifikasi hujan yang ditunjukkan pada gambar 3.20



**Gambar 3.20 Activity Diagram Menampilkan Notifikasi Hujan**

### 7. Activity Diagram Menampilkan Rekomendasi Jalur Alternatif Rute Banjir

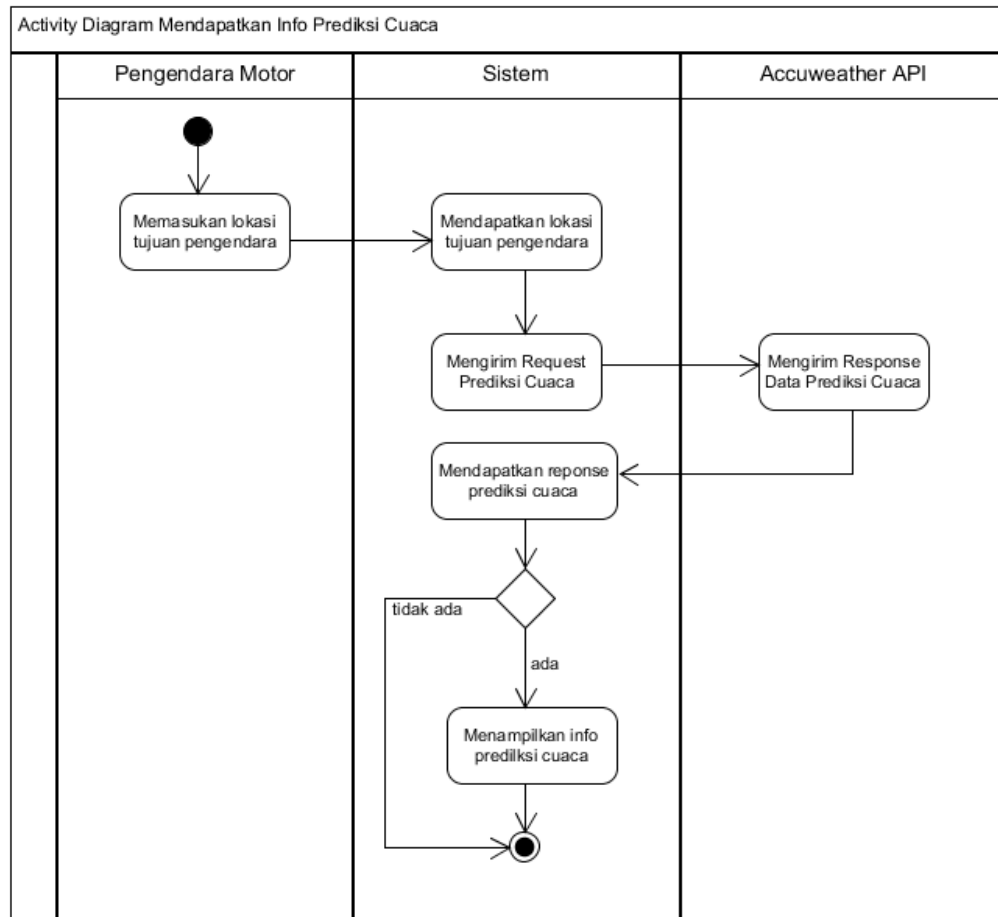
Berikut adalah skenario *activity diagram* rekomendasi jalur alternatif banjir yang ditunjukkan pada gambar 3.21



**Gambar 3.21 Activity Diagram Menampilkan Rekomendasi Jalur Alternatif Banjir**

### 8. *Activity Diagram* Mendapatkan Info Prediksi Cuaca

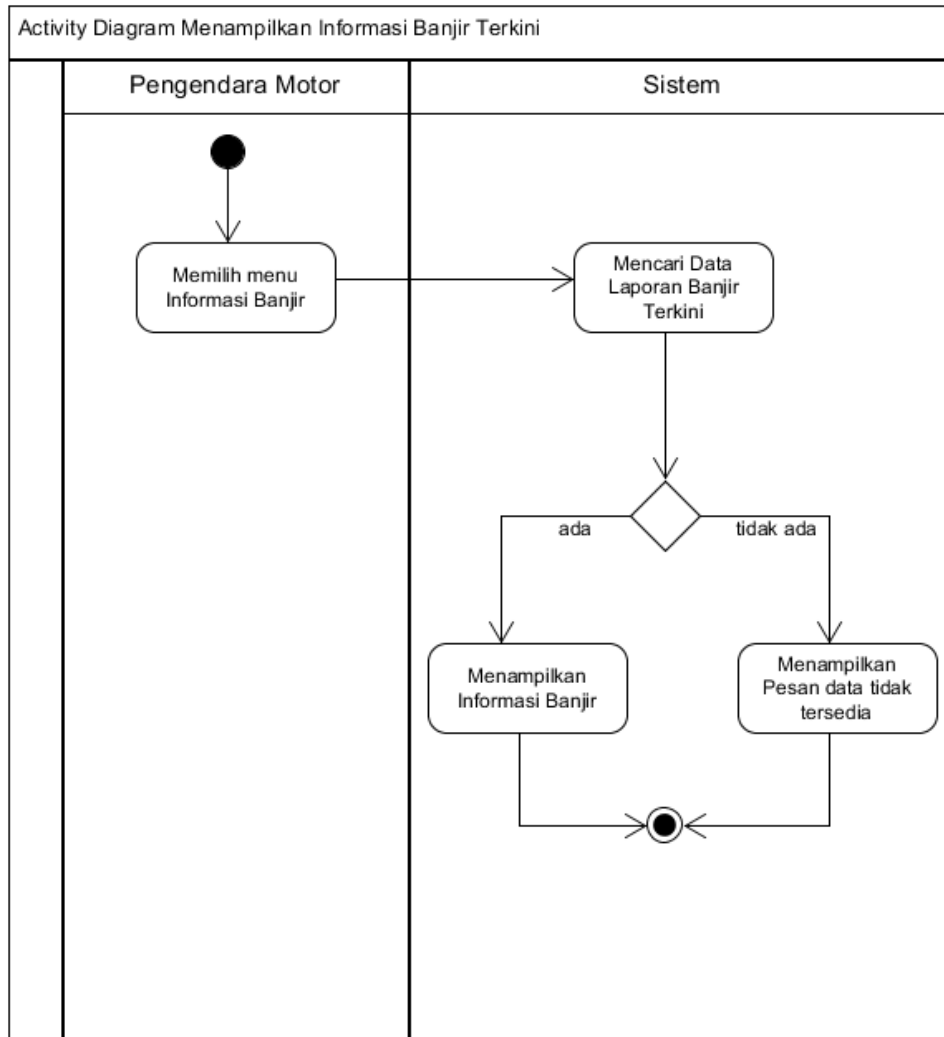
Berikut adalah skenario *activity diagram* mendapatkan info prediksi cuaca bagi pengendara yang ditunjukkan pada gambar 3.22



**Gambar 3.22** Activity Diagram Mendapatkan Info Prediksi Cuaca

### 9. *Activity Diagram* Menampilkan Informasi Banjir Terkini

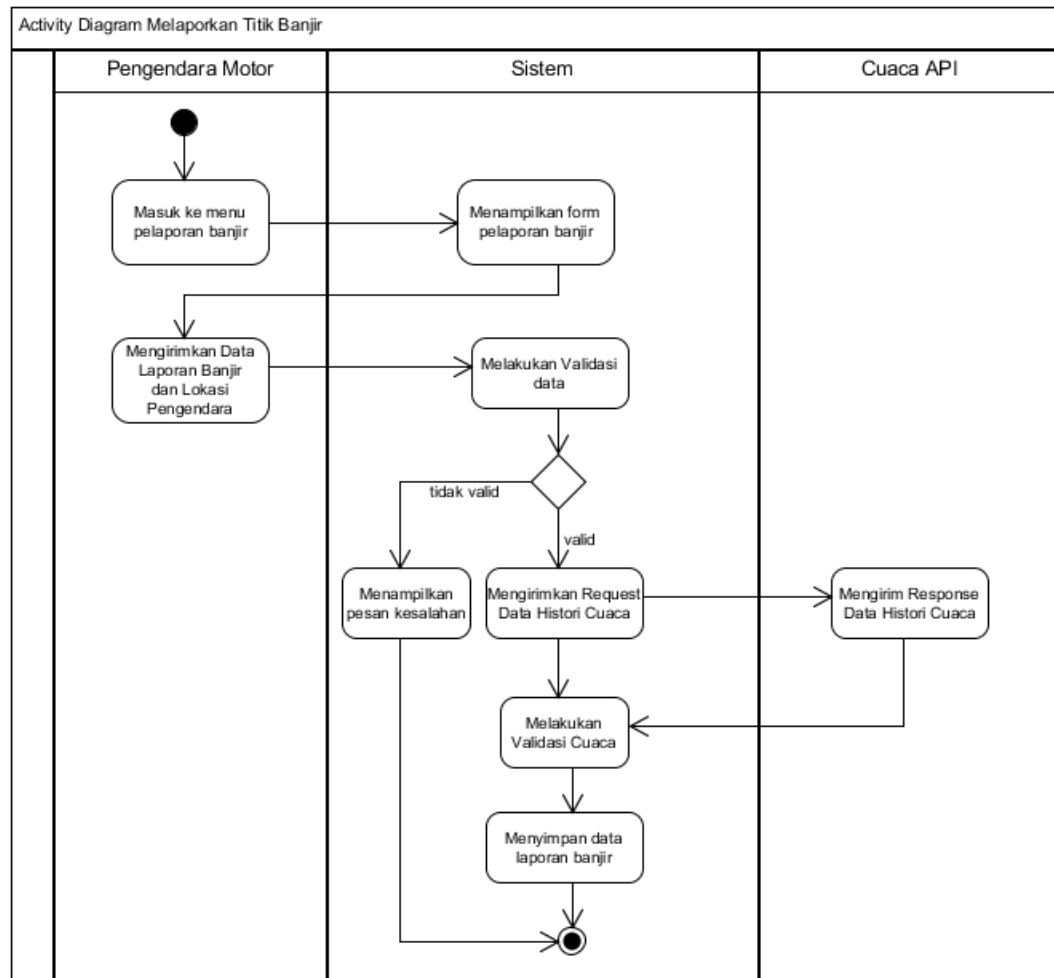
Berikut adalah skenario *activity diagram* menampilkan informasi banjir terkini yang ditunjukkan pada gambar 3.23



**Gambar 3.23** Activity Diagram Menampilkan Informasi Banjir Terkini

### 10. Activity Diagram Melaporkan Titik Banjir

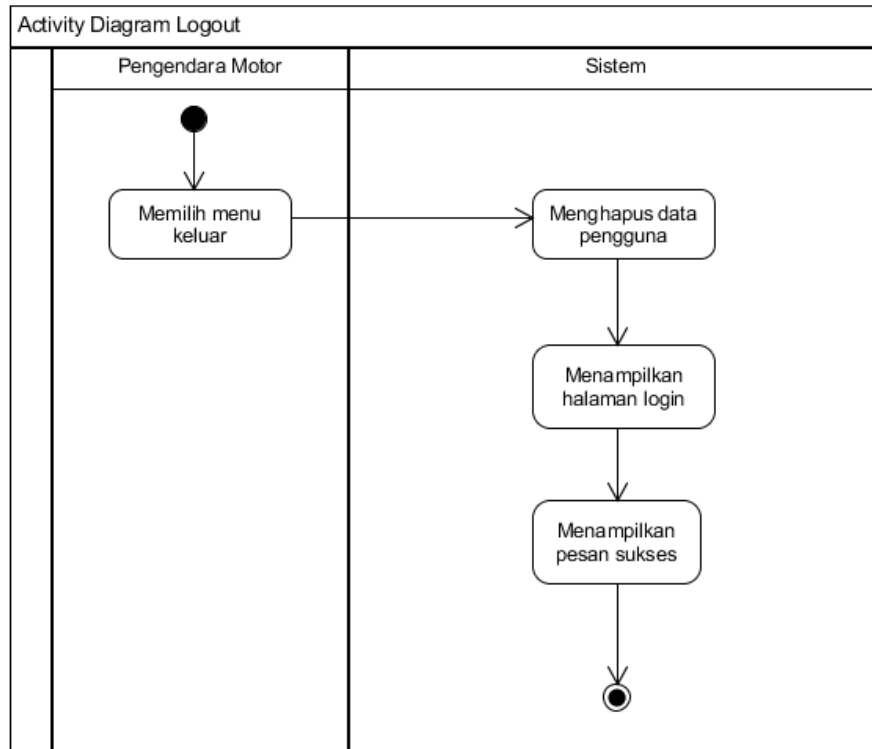
Berikut adalah skenario *activity diagram* melaporkan titik banjir yang ditunjukkan pada gambar 3.24



**Gambar 3.24 Activity Diagram Melaporkan Titik Banjir**

### 11. Activity Diagram Logout

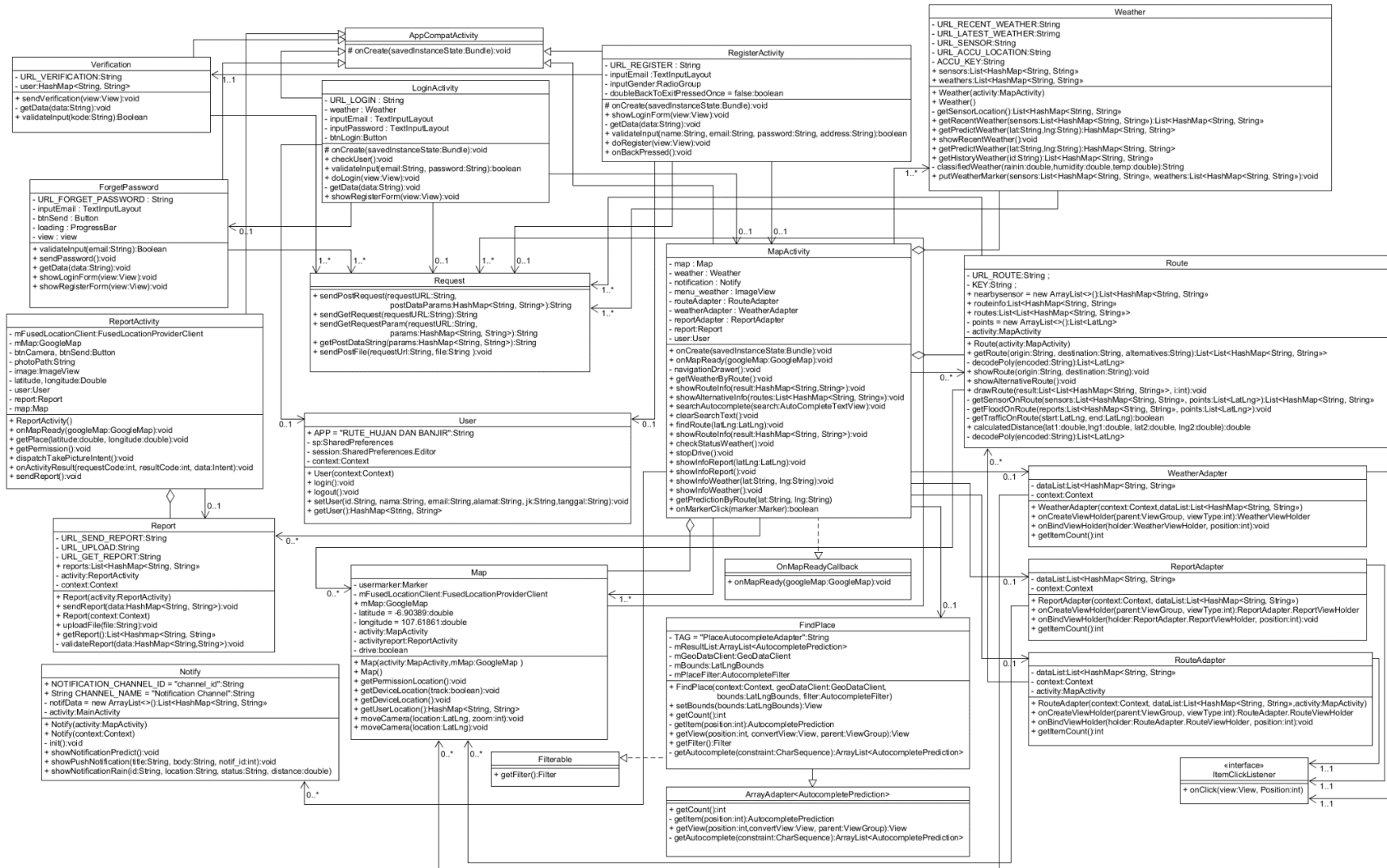
Berikut adalah skenario *activity diagram* logout pengguna yang ditunjukkan pada gambar 3.25



**Gambar 3.25 Activity Diagram Logout**

#### 3.1.7.4 Class Diagram

*Class Diagram* merupakan suatu model yang memberi gambaran struktur dari segi pendefinisian kelas yang terlibat didalam sistem. *Class diagram* bertujuan memberikan pemahaman terhadap hubungan antar kelas serta memberikan rincian atribut – atribut serta metode – metode yang terdapat pada kelas. Berikut model *class diagram* yang telah dibuat ditunjukkan pada gambar 3.26



Gambar 3.26 Class Diagram

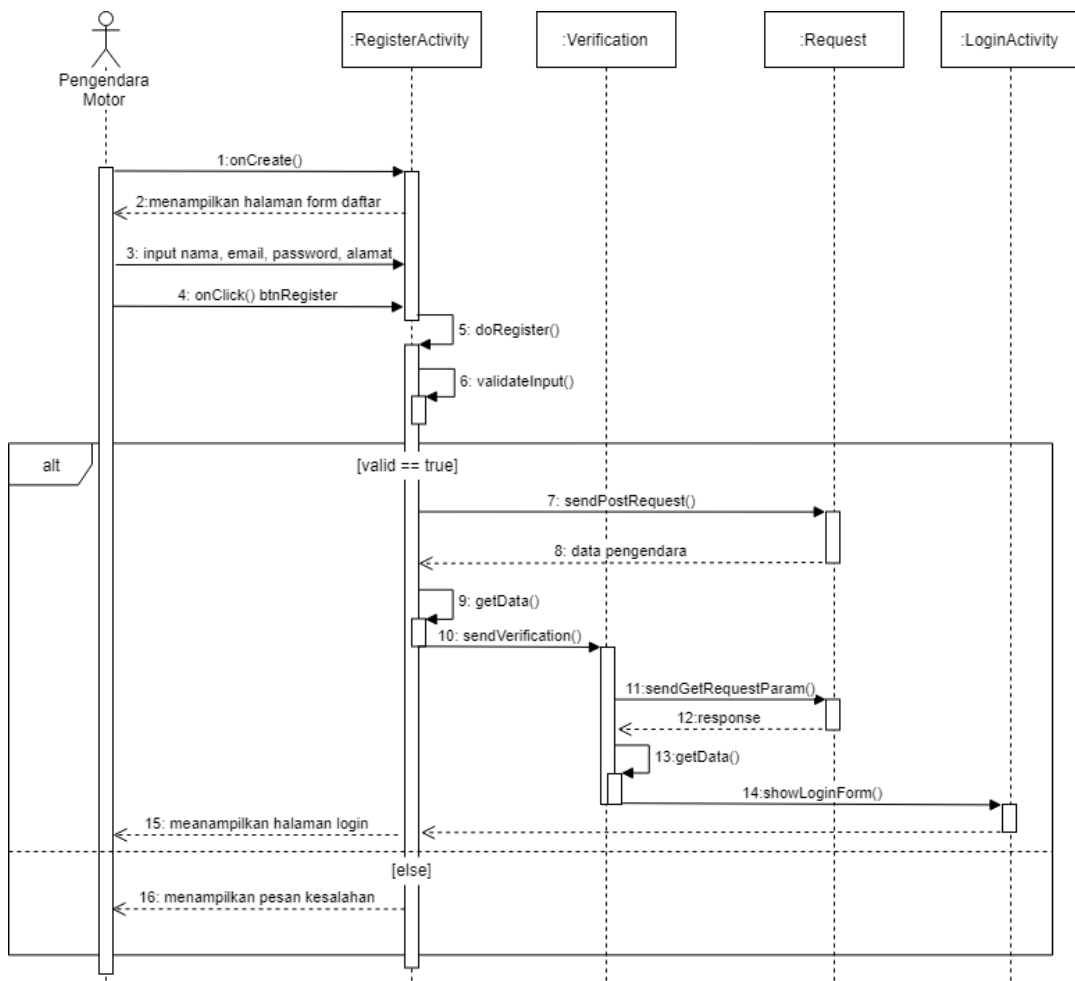
### 3.1.7.5 Sequence Diagram

*Sequence diagram* merupakan diagram yang menggambarkan suatu interaksi antar satu objek dengan objek lainnya yang terdapat dalam sebuah sistem yang akan dibangun. *Sequence diagram* dapat memberikan gambaran *event* atau kelakuan yang terjadi pada suatu objek dalam mendeskripsikan waktu hidup dan juga pesan – pesan yang dikirimkan antar objek. Adapaun pembuatan *sequence diagram* didasarkan pada *use case diagram* dan *scenario use case* yang telah dirancang sebelumnya. Berikut rincian dari *sequence diagram* yang telah disusun yaitu sebagai berikut.

#### 1. Sequence Diagram Daftar

Berikut adalah skenario *sequence diagram* daftar yang ditunjukkan pada gambar

3.27

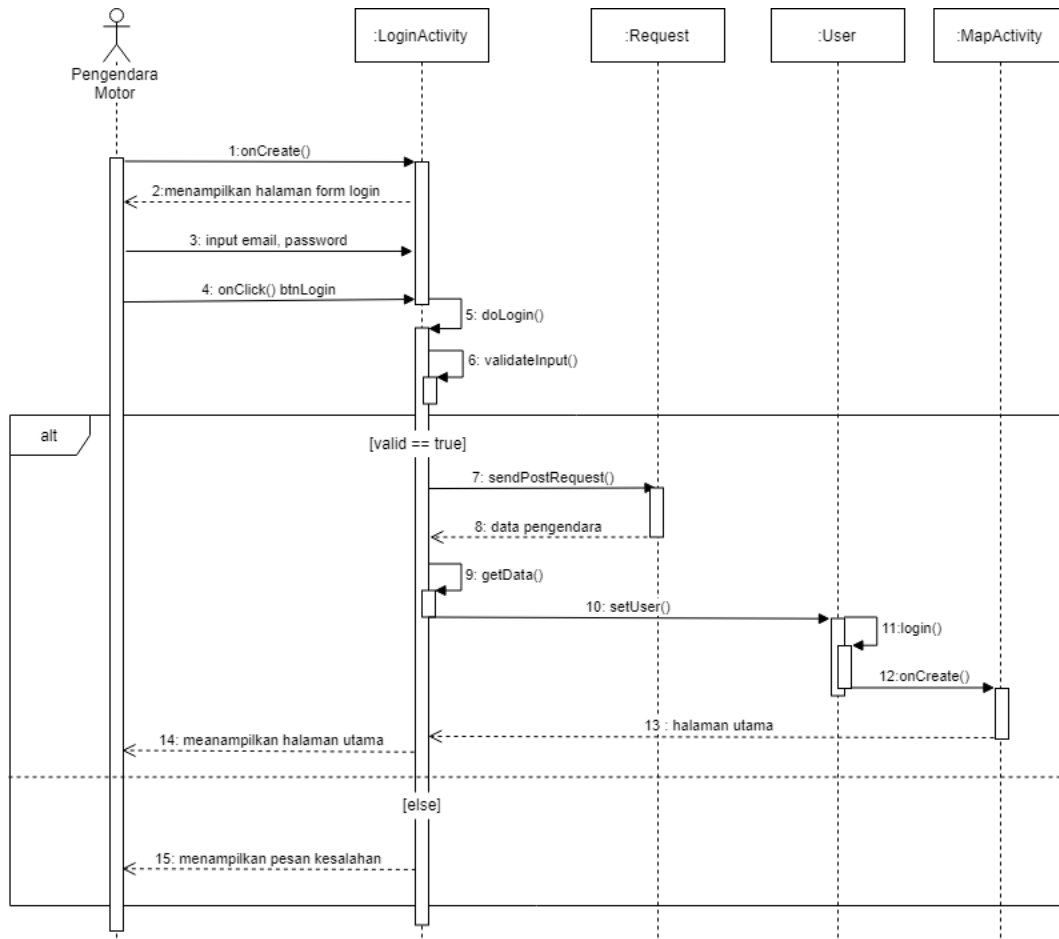


Gambar 3.27 Sequence Diagram Daftar



## 2. Sequence Diagram Login

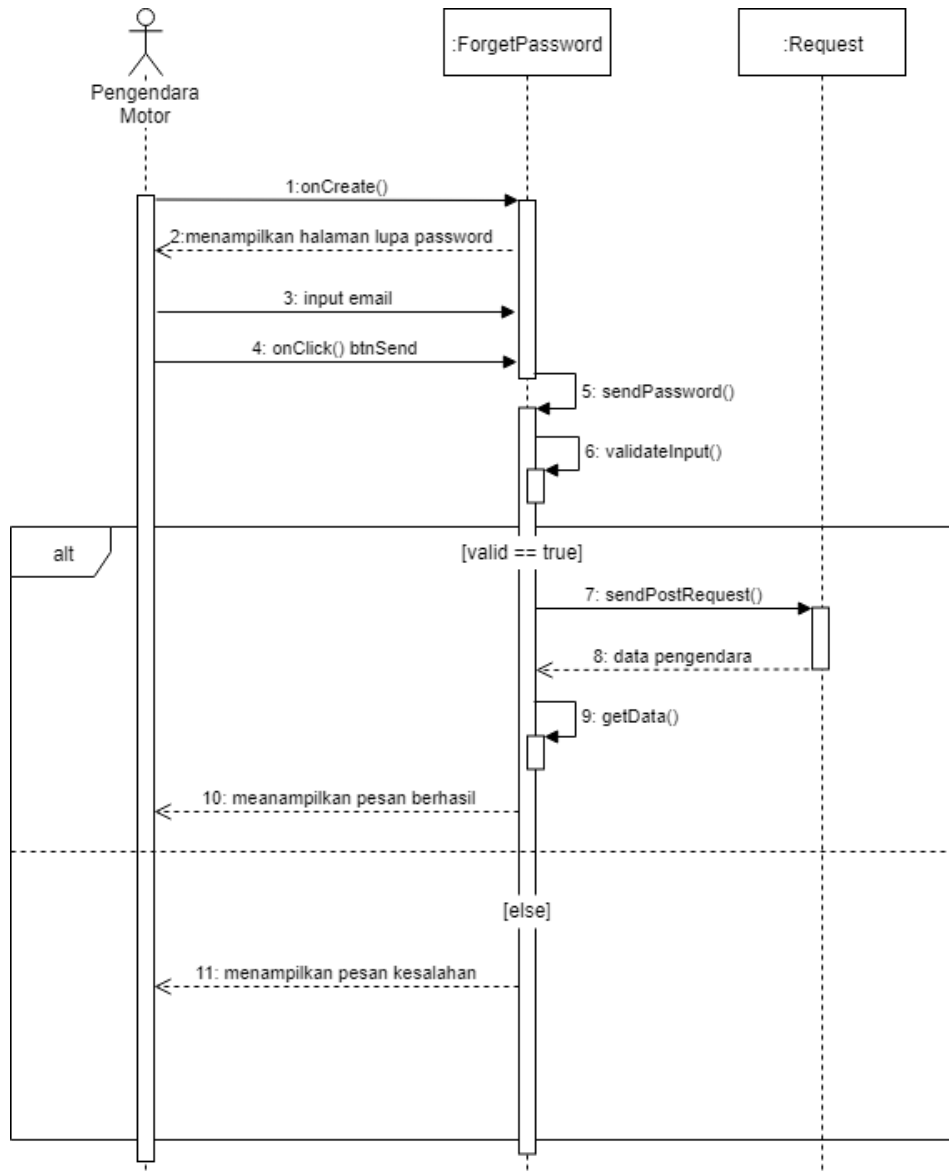
Berikut adalah skenario *sequence diagram* login bagi pengendara yang ditunjukkan pada gambar 3.28



**Gambar 3.28 Sequence Diagram Login Pengendara**

### 3. Sequence Diagram Lupa Password

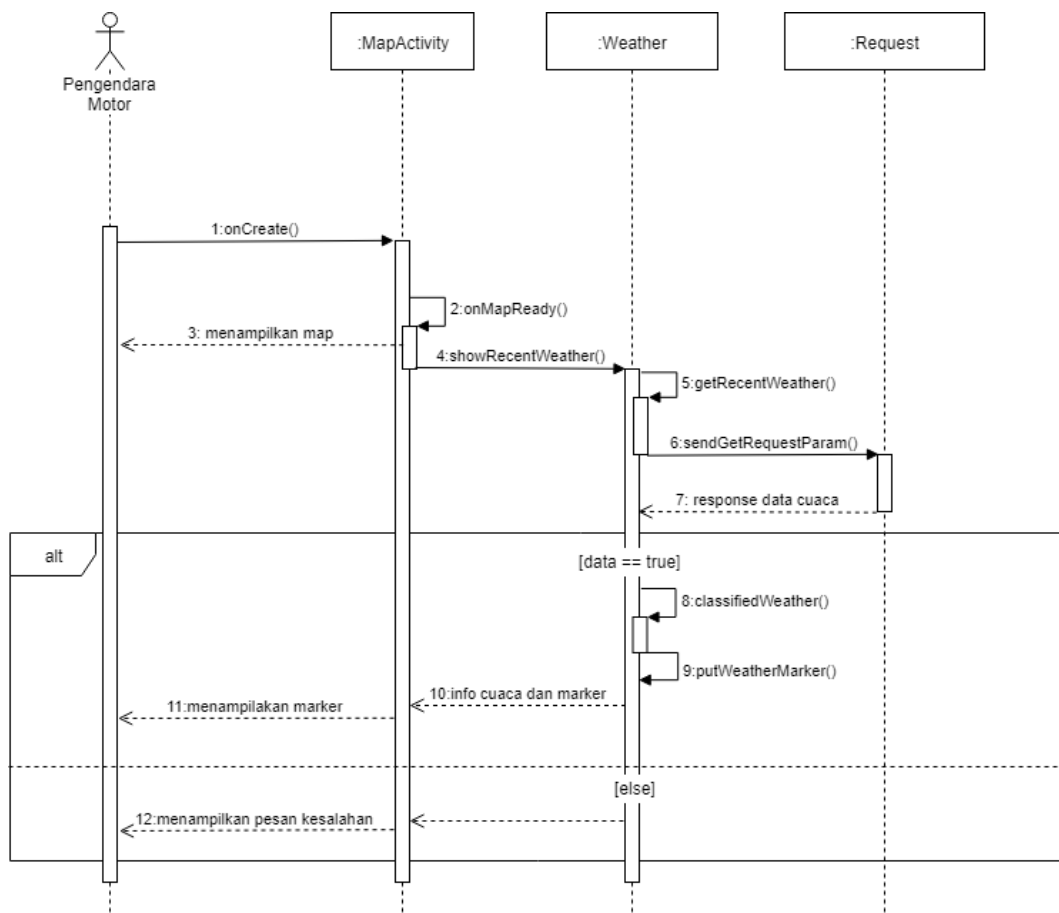
Berikut adalah skenario *sequence diagram* lupa password bagi pengendara yang ditunjukkan pada gambar 3.29



**Gambar 3.29 Sequence Diagram Lupa Password**

#### 4. *Sequence Diagram* Menampilkan Informasi Cuaca Pada Map

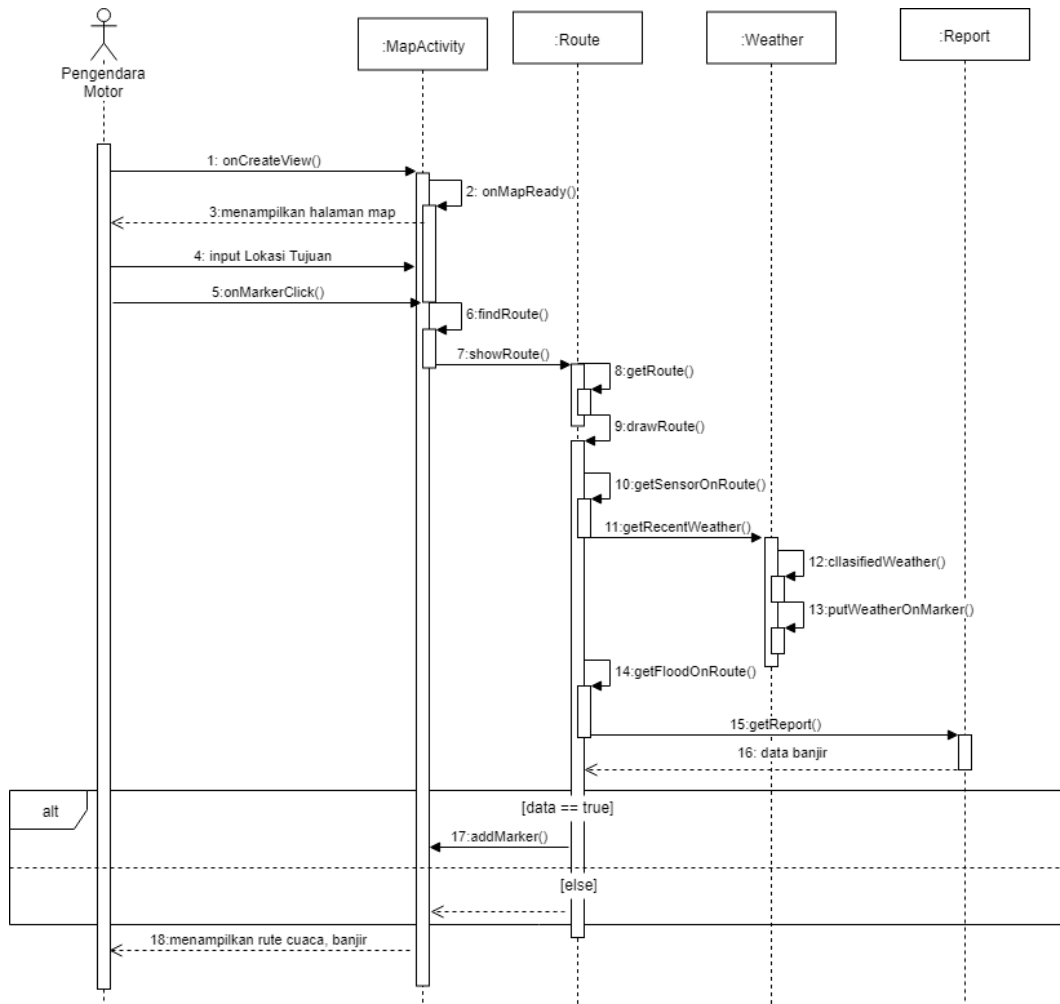
Berikut adalah skenario *sequence diagram* menampilkan informasi cuaca pada *map* yang ditunjukkan pada gambar 3.30



**Gambar 3.30** Sequence Diagram Menampilkan Informasi Cuaca Pada Map

### 5. Sequence Diagram Menampilkan Rute Hujan dan Banjir

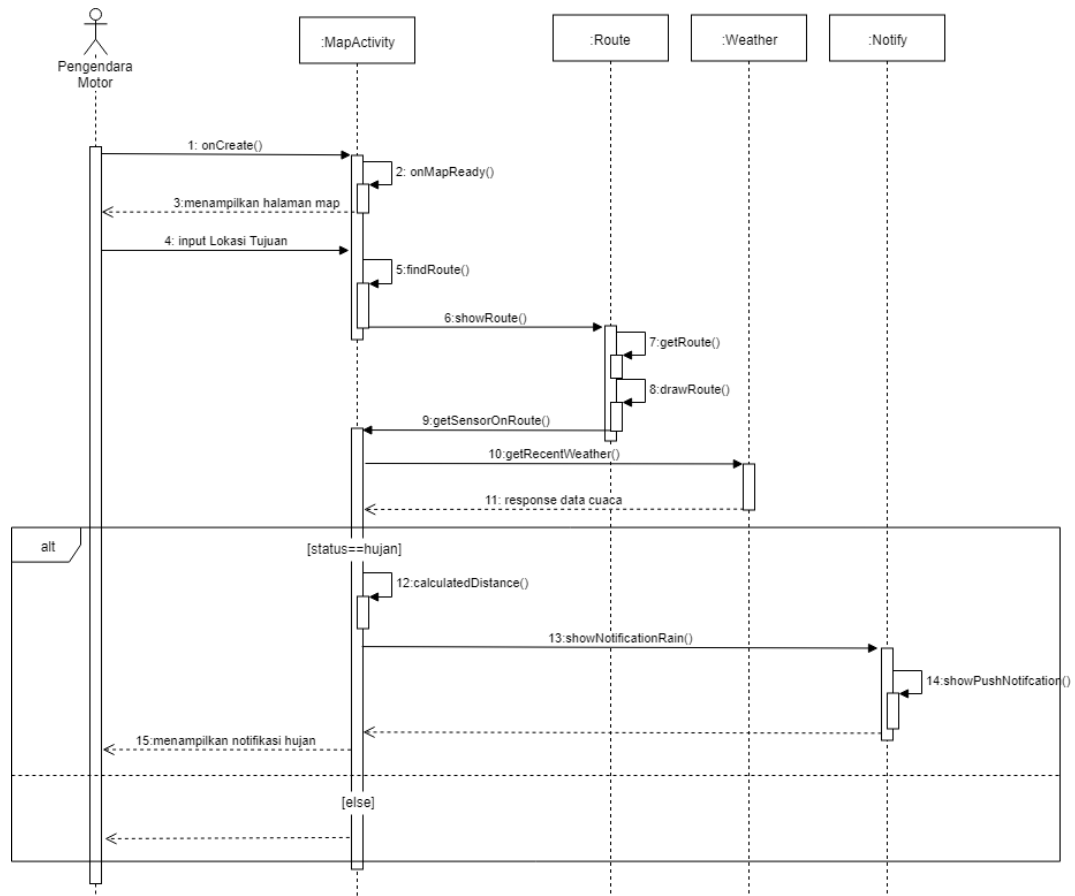
Berikut adalah skenario *sequence diagram* menampilkan rute hujan dan banjir yang ditunjukkan pada gambar 3.31



**Gambar 3.31 Sequence Diagram Informasi Rute Banjir**

## 6. Sequence Diagram Menampilkan Notifikasi Hujan

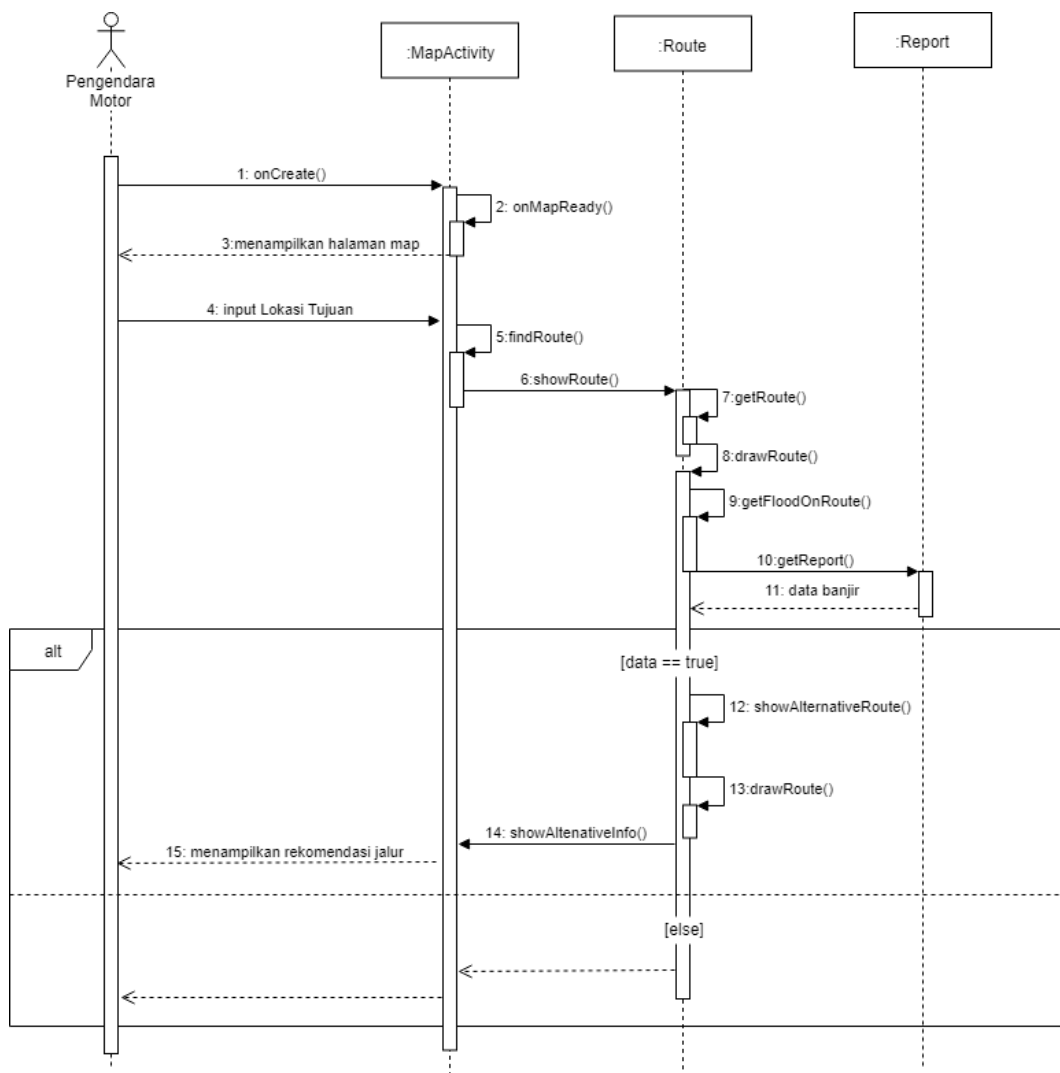
Berikut adalah skenario *sequence diagram* menampilkan notifikasi hujan yang ditunjukkan pada gambar 3.32



**Gambar 3.32 Sequence Diagram Menampilkan Notifikasi Hujan**

### 7. Sequence Diagram Menampilkan Rekomendasi Jalur Alternatif Banjir

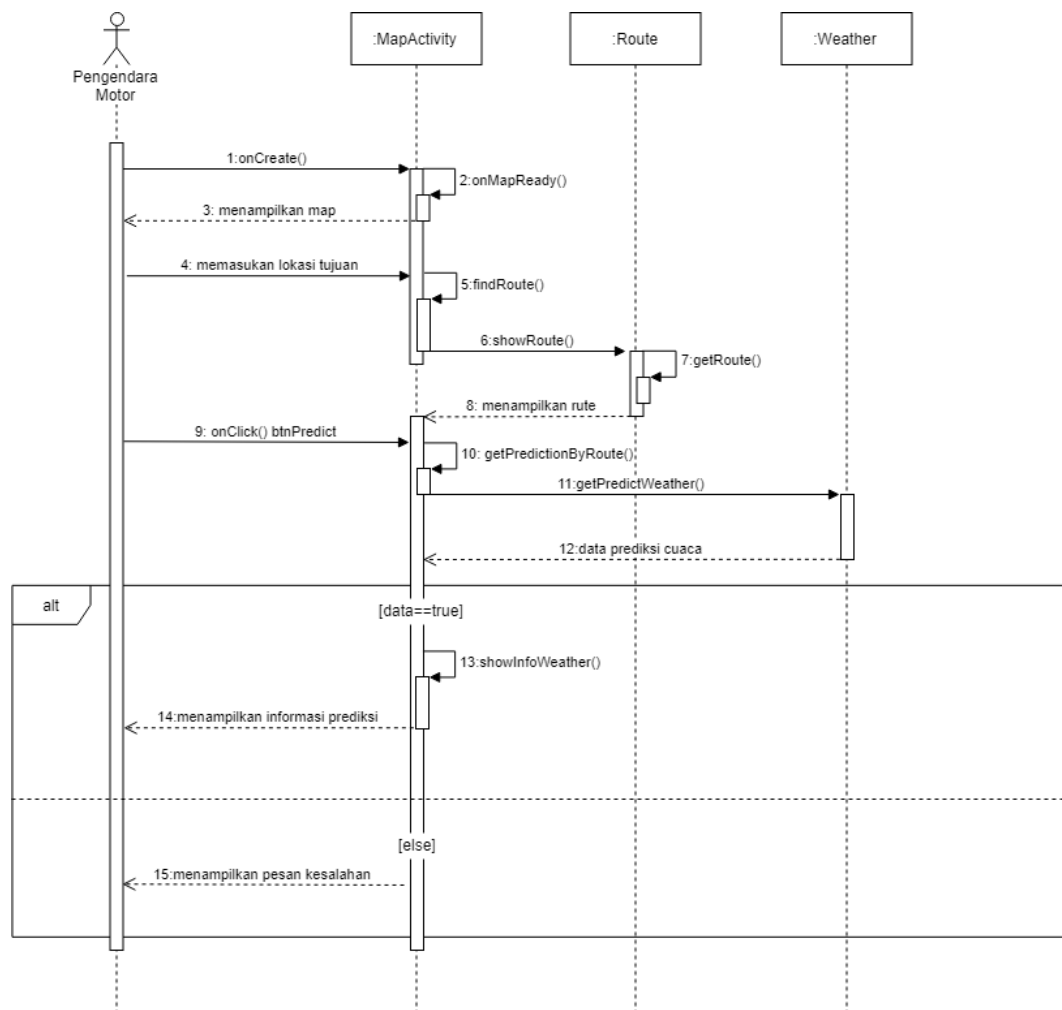
Berikut adalah skenario *sequence diagram* menampilkan rekomendasi jalur alternatif yang ditunjukkan pada gambar 3.33



**Gambar 3.33 Sequence Diagram Menampilkan Rekomendasi Jalur Alternatif Banjir**

## 8. Sequence Diagram Mendapatkan Info Prediksi Cuaca

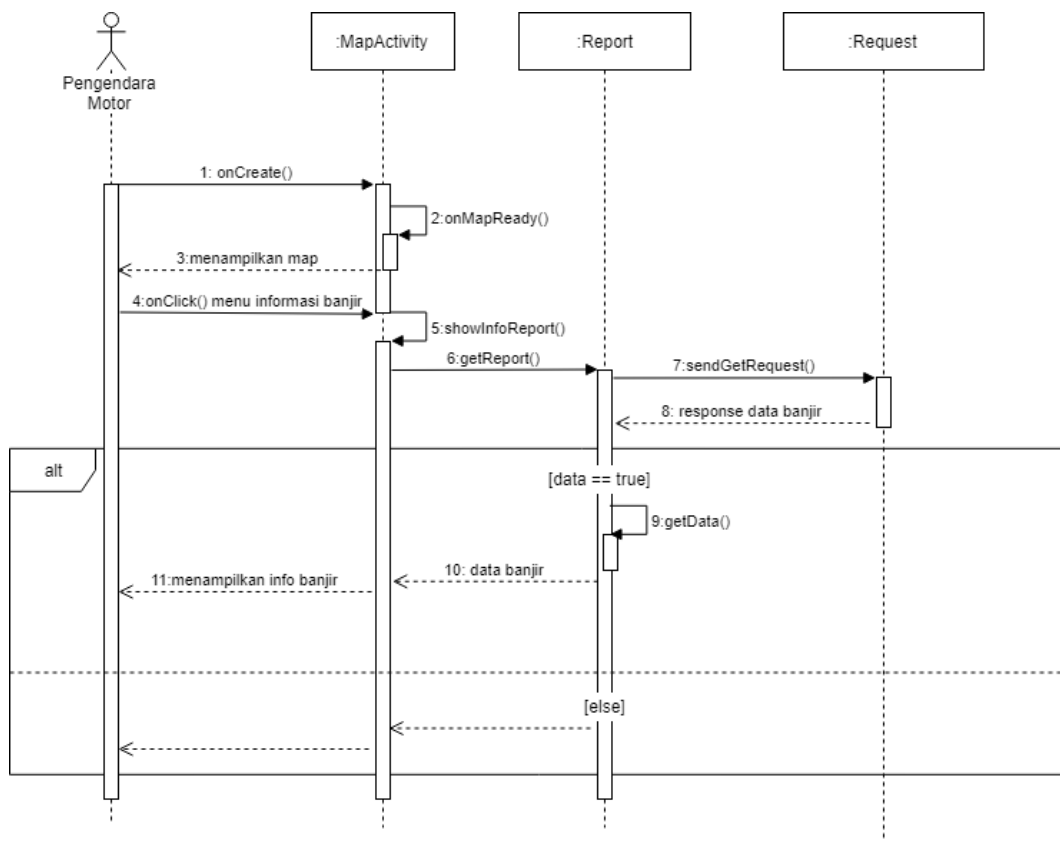
Berikut adalah skenario *sequence diagram* mendapatkan info prediksi cuaca yang ditunjukkan pada gambar 3.34



**Gambar 3.34 Sequence Diagram Mendapatkan Info Prediksi Cuaca**

### 9. Sequence Diagram Menampilkan Informasi Banjir Terkini

Berikut adalah skenario *sequence diagram* menampilkan informasi banjir terkini yang ditunjukkan pada gambar 3.35

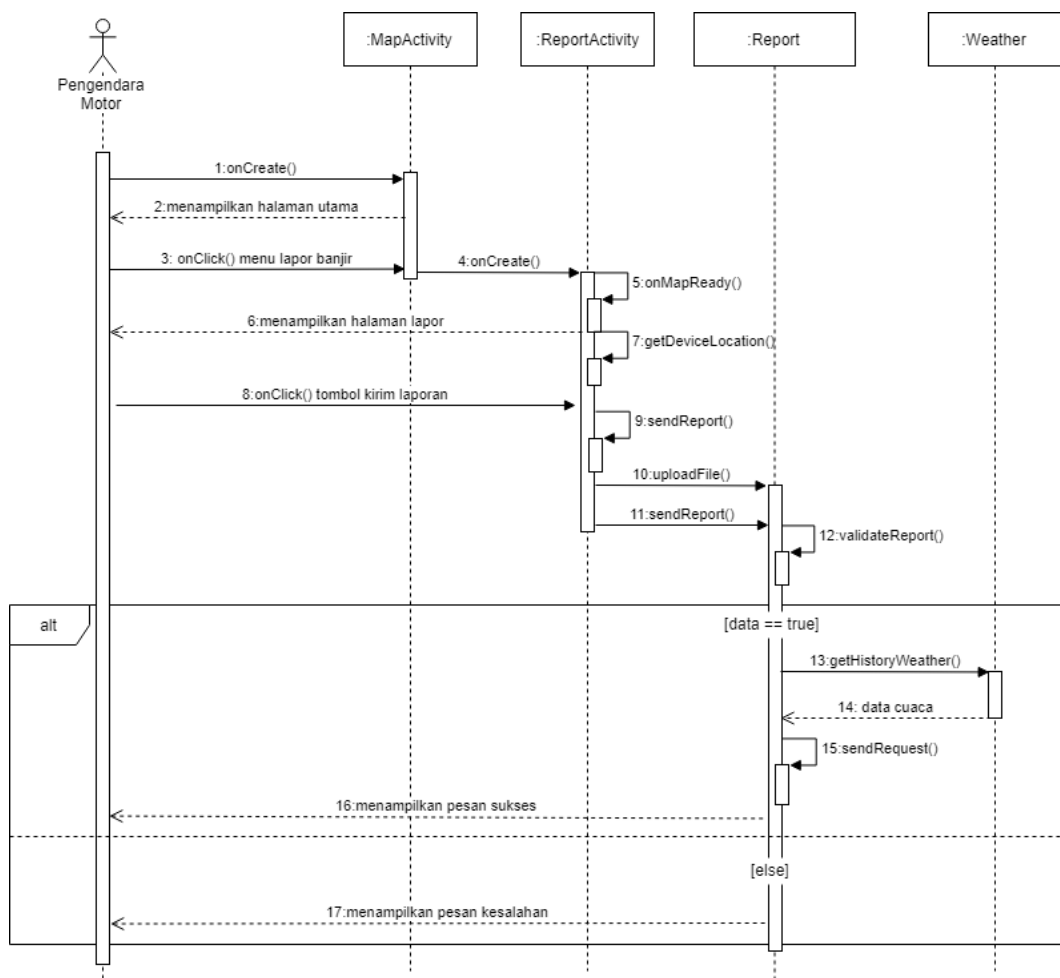


**Gambar 3.35 Sequence Diagram Menampilkan Informasi Banjir Terkini**



### 10. Sequence Diagram Melaporkan Daerah Banjir

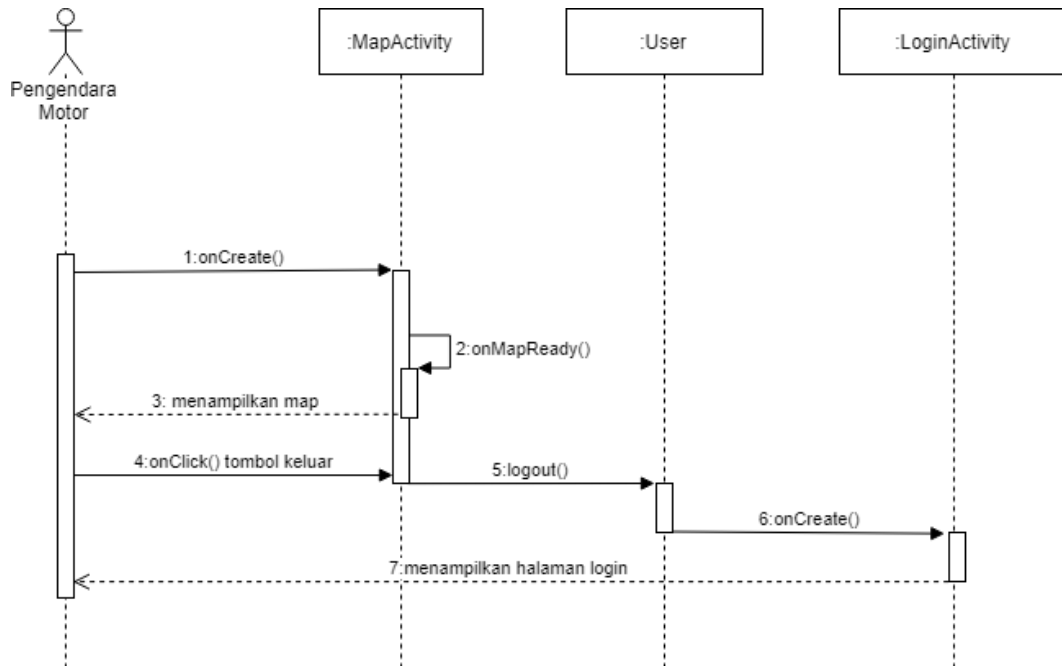
Berikut adalah skenario *sequence diagram* melaporkan daerah banjir yang ditunjukkan pada gambar 3.36



**Gambar 3.36** Sequence Diagram Melaporkan Daerah Banjir

### 11. Sequence Diagram Logout

Berikut adalah skenario *sequence diagram* logout pengguna yang ditunjukkan pada gambar 3.37



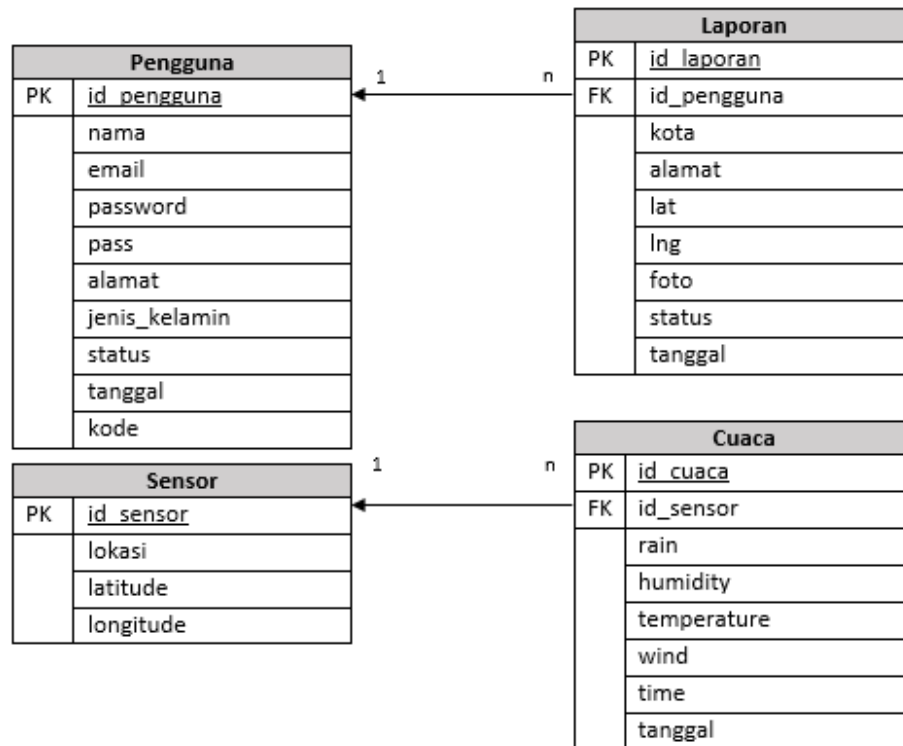
**Gambar 3.37 Sequence Diagram Logout**

## 3.2 Perancangan Sistem

Perancangan sistem dilakukan setelah tahapan analisis sistem telah dilakukan. Perancangan dapat diartikan sebagai suatu teknik untuk mendefinisikan suatu perangkat, suatu proses maupun sistem secara mendetail untuk kemudian direalisasikan. Adapun perancangan sistem yang dilakukan meliputi perancangan skema relasi, perancangan struktur tabel, perancangan struktur menu, perancangan antarmuka dan perancangan jaringan semantik.

### 3.2.1 Skema Relasi

Skema relasi merupakan suatu rangkaian relasi dari dua buah tabel atau lebih pada sistem basis data. Berikut merupakan skema relasi yang ada pada penelitian ini ditunjukkan pada gambar 3.38



**Gambar 3.38 Skema Relasi**

Perancangan struktur tabel dilakukan untuk mengidentifikasi atribut – atribut dari setiap table yang ada pada skema relasi. Adapun perancangan struktur tabel yang telah dilakukan yaitu sebagai berikut

1. Tabel Pengguna

**Tabel 3.34 Struktur Tabel Pengguna**

<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>	<i>Null</i>	<i>Reference</i>
id_pengguna	int	11	Primary Key	No	-
nama	varchar	30	-	No	-
email	varchar	50	Unique	No	-
password	varchar	100	-	No	-
pass	varchar	100	-	No	-
alamat	varchar	100	-	No	-
jenis_kelamin	varchar	30	-	No	-
status	varchar	30	-	No	-
tanggal	date	-	-	No	-
Kode	varchar	10	-	No	-

## 2. Tabel Laporan

**Tabel 3.35 Struktur Tabel Laporan**

<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>	<i>Null</i>	<i>Reference</i>
id_laporan	<i>int</i>	11	<i>Primary Key</i>	No	-
id_pengguna	<i>int</i>	11	<i>Foreign Key</i>	No	Pengguna(id_pengguna)
kota	<i>varchar</i>	30	-	No	-
alamat	<i>varchar</i>	100	-	No	-
lat	<i>varchar</i>	20	-	No	-
lng	<i>varchar</i>	20	-	No	-
foto	<i>varchar</i>	60	-	No	-
status	<i>varchar</i>	20	-	No	-
tanggal	<i>datetime</i>	-	-	No	-

## 3. Tabel Sensor

**Tabel 3.36 Struktur Tabel Sensor**

<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>	<i>Null</i>	<i>Reference</i>
id_sensor	<i>varchar</i>	30	<i>Primary Key</i>	No	-
lokasi	<i>varchar</i>	100	-	No	-
latitude	<i>varchar</i>	30	-	No	-
longitude	<i>varchar</i>	30	-	No	-

## 4. Tabel Cuaca

**Tabel 3.37 Struktur Tabel Cuaca**

<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>	<i>Null</i>	<i>Reference</i>
id_cuaca	<i>int</i>	11	<i>Primary Key</i>	No	-
id_sensor	<i>varchar</i>	30	<i>Foreign Key</i>	No	Sensor(id_sensor)
rain	<i>varchar</i>	10	-	No	-
humidity	<i>varchar</i>	10	-	No	-
temperature	<i>varchar</i>	10	-	No	-
wind	<i>varchar</i>	10	-	No	-
time	<i>varchar</i>	60	-	No	-
tanggal	<i>datetime</i>	-	-	No	-

**3.2.2 Perancangan Struktur Menu**

Perancangan struktur menu merupakan gambaran alur menu pada aplikasi sehingga dapat memberikan pemahaman dalam penggunaan aplikasi. Adapun Perancangan struktur menu yang dilakukan meliputi sistem yang ada pada *platform mobile*. Berikut struktur menu sistem yang terdapat pada *platform mobile* ditunjukkan pada gambar 3.39



**Gambar 3.39 Struktur Menu Aplikasi**

### 3.2.3 Perancangan Antarmuka

Perancangan antarmuka bertujuan untuk memberi gambaran tentang sistem yang akan dibangun pada level antarmuka sehingga akan memberikan gambaran jelas bagaimana sistem akan diimplementasikan. Adapun Perancangan antarmuka sistem yang dibangun pada *platform mobile* yaitu sebagai berikut.

#### 1. Perancangan Antarmuka Login Pengendara

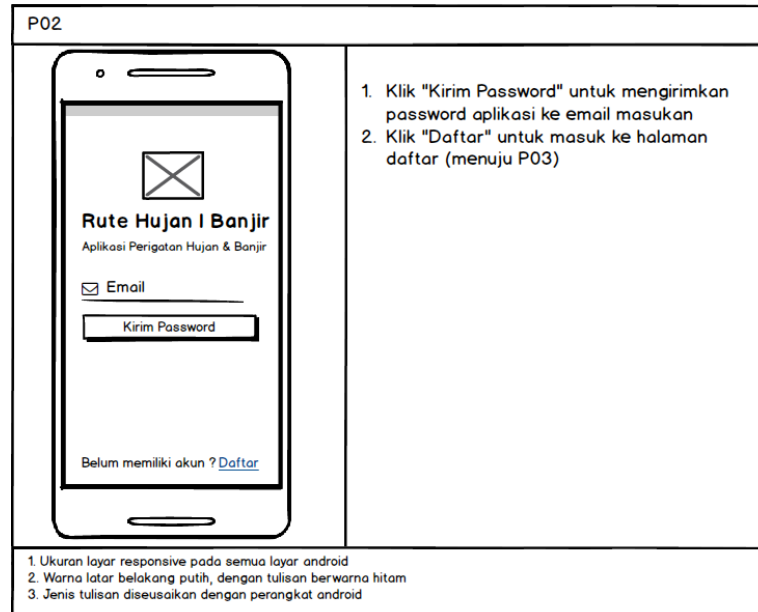
Berikut adalah perancangan antarmuka untuk halaman login pengendara ditunjukkan pada gambar 3.40

P01	
	<ol style="list-style-type: none"> <li>1. Klik form email untuk memasukkan data email</li> <li>2. Klik form password untuk memasukkan data password</li> <li>3. Klik tombol "Masuk" untuk melakukan login pengendara (menuju P04)</li> <li>4. Klik tombol "Daftar" untuk menampilkan halaman pendaftaran (menuju P03)</li> <li>5. Klik "Lupa Password Untuk menuju halaman kirim password(P02)</li> </ol>
<ol style="list-style-type: none"> <li>1. Ukuran layar responsive pada semua layar android</li> <li>2. Warna latar belakang putih, dengan tulisan berwarna hitam</li> <li>3. Jenis tulisan disesuaikan dengan perangkat android</li> </ol>	

**Gambar 3.40 Perancangan Antarmuka Login Pengendara**

## 2. Perancangan Antarmuka Lupa Password

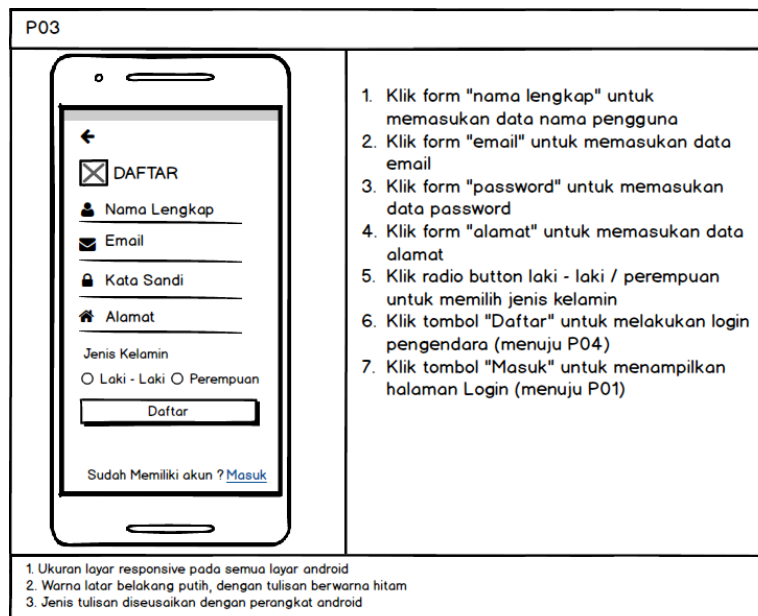
Berikut adalah perancangan antarmuka untuk halaman lupa password pengendara ditunjukkan pada gambar 3.41



**Gambar 3.41 Perancangan Antarmuka Daftar**

## 3. Perancangan Antarmuka Daftar

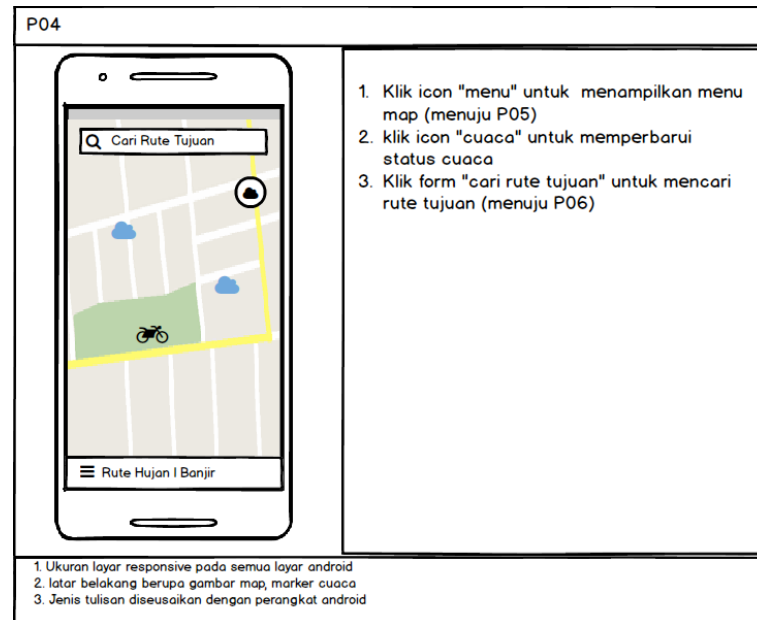
Berikut adalah perancangan antarmuka untuk halaman pendaftaran pengendara ditunjukkan pada gambar 3.42



**Gambar 3.42 Gambar 3.43 Perancangan Antarmuka Daftar**

#### 4. Perancangan Antarmuka Halaman Utama

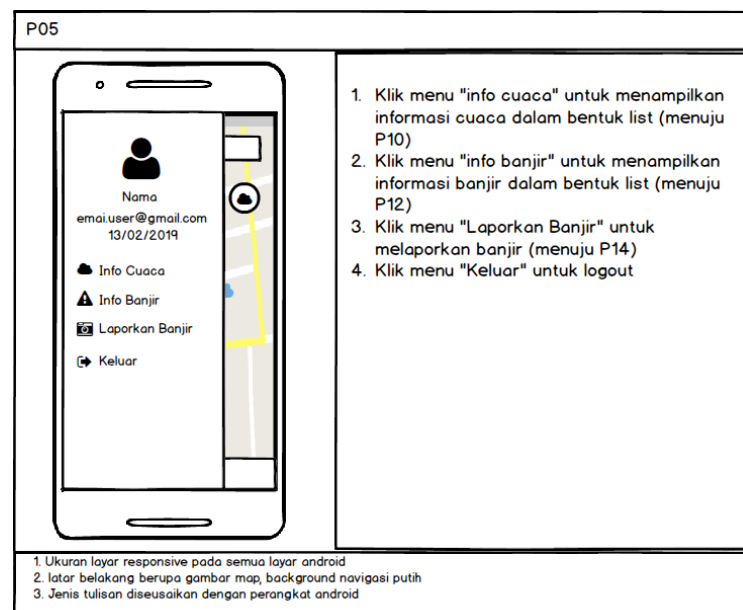
Berikut adalah perancangan antarmuka untuk halaman utama aplikasi ditunjukkan pada gambar 3.43



**Gambar 3.44 Perancangan Antarmuka Halaman Utama**

#### 5. Perancangan Antarmuka Menu Utama

Berikut adalah perancangan antarmuka untuk halaman menu utama ditunjukkan pada gambar 3.44



**Gambar 3.45 Perancangan Antarmuka Menu Utama**

## 6. Perancangan Antarmuka Rute Hujan dan Banjir

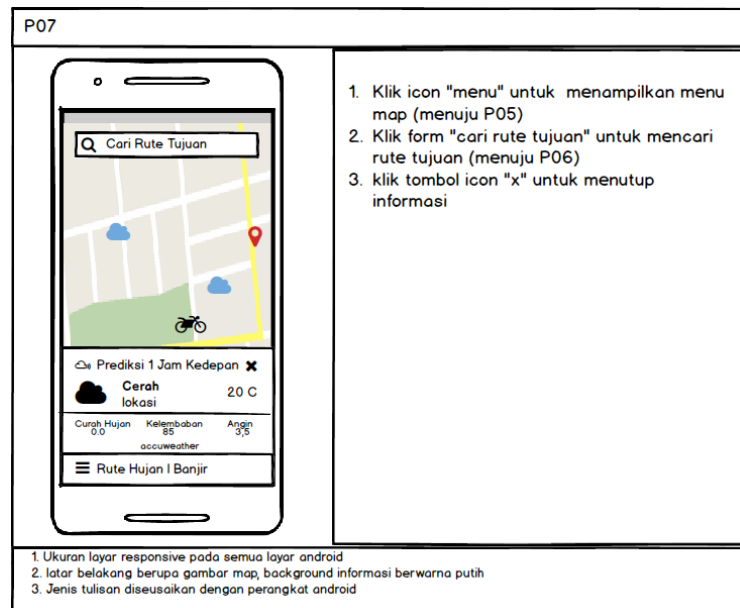
Berikut adalah perancangan antarmuka untuk halaman rute hujan dan banjir pengendara ditunjukkan pada gambar 3.45



**Gambar 3.46 Perancangan Antarmuka Rute Hujan dan Banjir**

## 7. Perancangan Antarmuka Prediksi Cuaca

Berikut adalah perancangan antarmuka untuk halaman prediksi cuaca ditunjukkan pada gambar 3.46

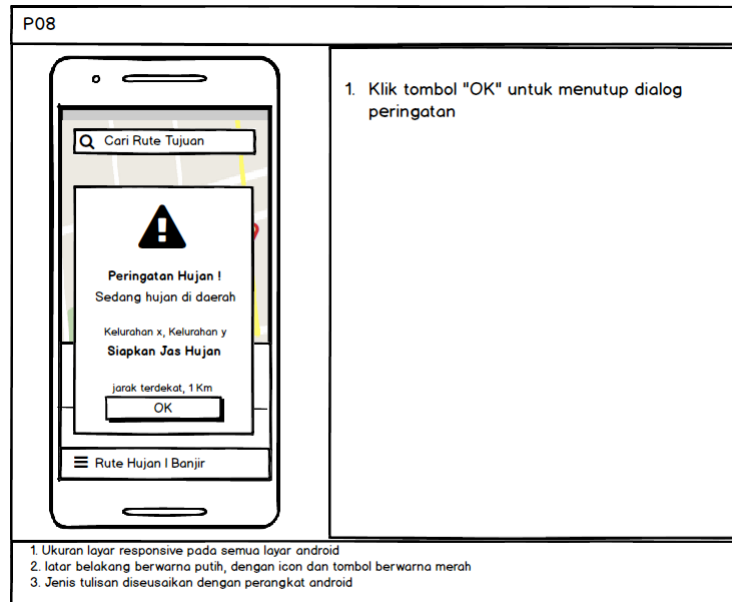


**Gambar 3.47 Perancangan Antarmuka Prediksi Cuaca**



## 8. Perancangan Antarmuka Peringatan Hujan

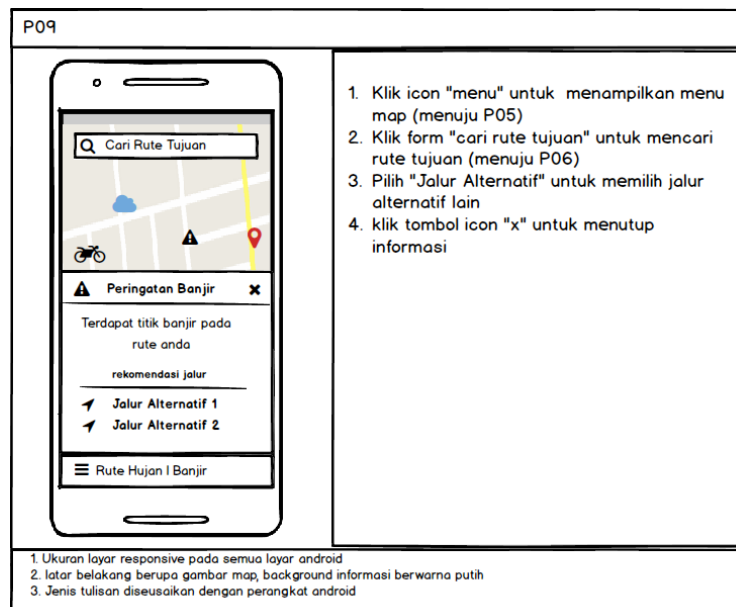
Berikut adalah perancangan antarmuka untuk halaman peringatan hujan bagi pengendara ditunjukkan pada gambar 3.47



**Gambar 3.48 Perancangan Antarmuka Peringatan Hujan**

## 9. Perancangan Antarmuka Rute Alternatif Banjir

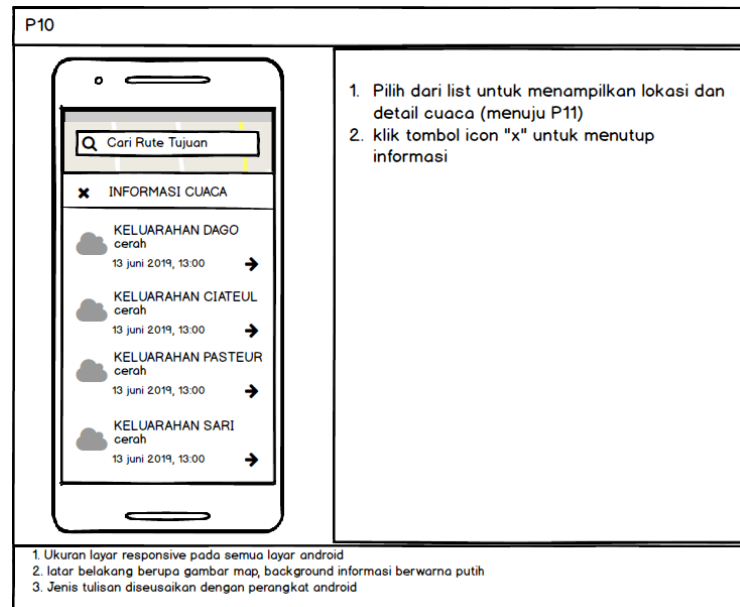
Berikut adalah perancangan antarmuka untuk halaman rute alternatif banjir pengendara ditunjukkan pada gambar 3.48



**Gambar 3.49 Perancangan Antarmuka Rute Alternatif Banjir**

## 10. Perancangan Antarmuka Menu Informasi Cuaca

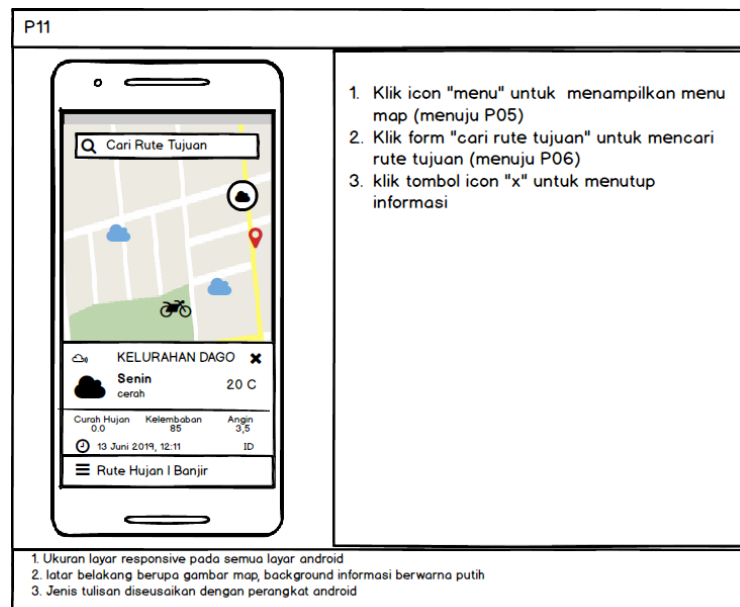
Berikut adalah perancangan antarmuka untuk halaman menu informasi cuaca ditunjukkan pada gambar 3.49



**Gambar 3.50 Perancangan Antarmuka Menu Informasi Cuaca**

## 11. Perancangan Antarmuka Detail Informasi Cuaca

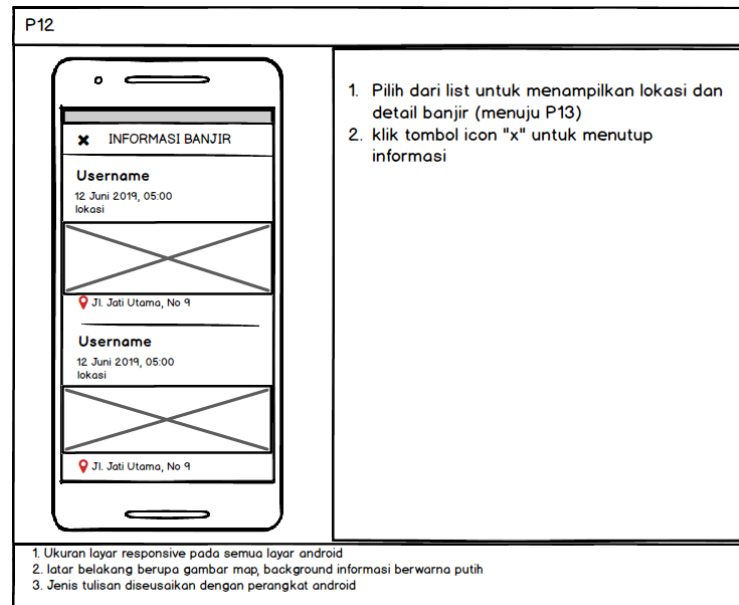
Berikut adalah perancangan antarmuka untuk halaman detail informasi cuaca ditunjukkan pada gambar 3.50



**Gambar 3.51 Perancangan Antarmuka Detail Informasi Cuaca**

## 12. Perancangan Antarmuka Menu Informasi Banjir

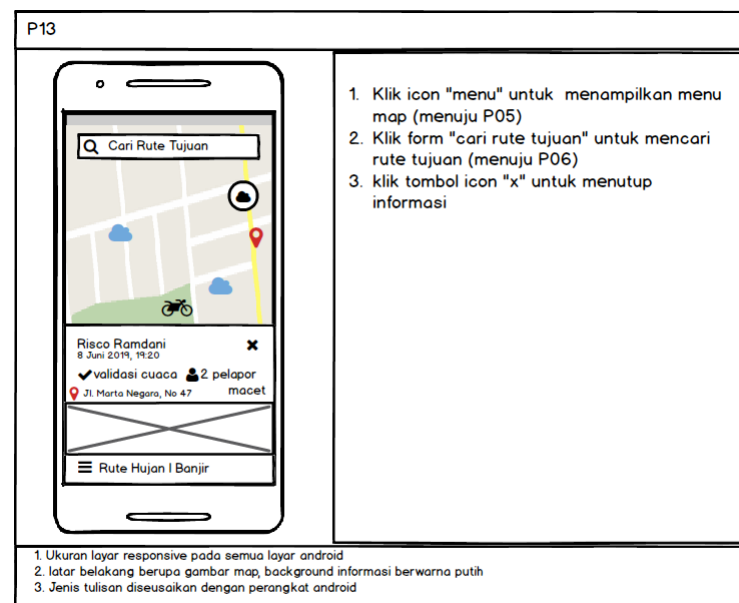
Berikut adalah perancangan antarmuka untuk halaman menu informasi banjir ditunjukkan pada gambar 3.51



**Gambar 3.52 Perancangan Antarmuka Halaman Menu Informasi Banjir**

## 13. Perancangan Antarmuka Detail Informasi Banjir

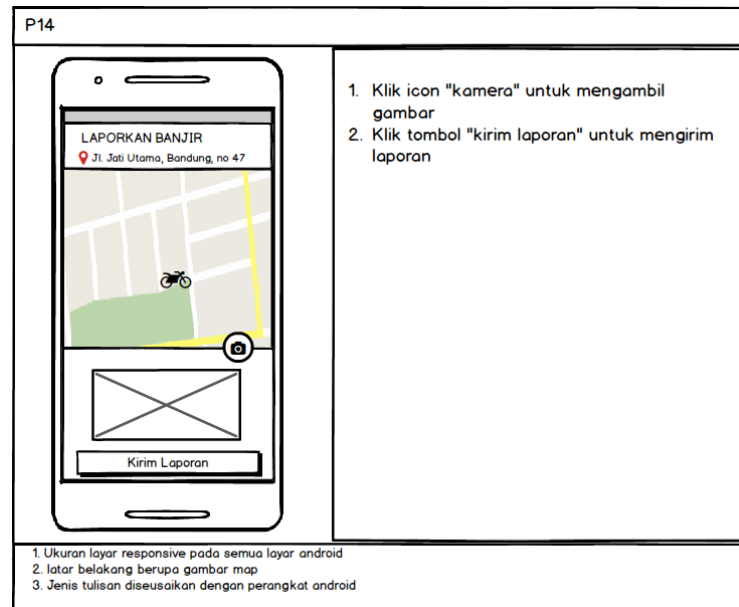
Berikut adalah perancangan antarmuka untuk halaman detail informasi banjir ditunjukkan pada gambar 3.52



**Gambar 3.53 Perancangan Antarmuka Halaman Detail Informasi Banjir**

#### 14. Perancangan Antarmuka Pelaporan Banjir

Berikut adalah perancangan antarmuka untuk halaman pelaporan banjir ditunjukkan pada gambar 3.53



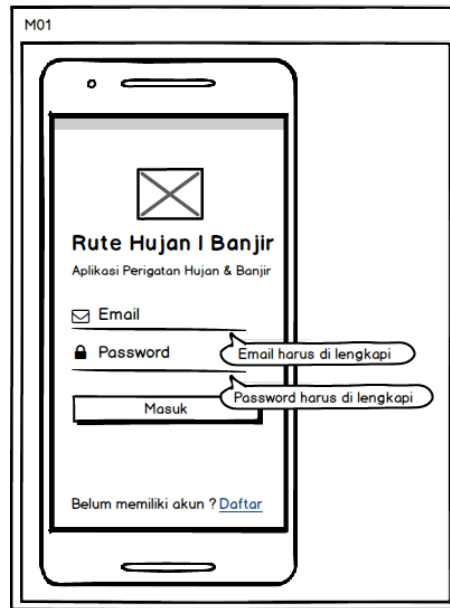
**Gambar 3.54 Perancangan Antarmuka Halaman Pelaporan Banjir**

#### 3.2.4 Perancangan Pesan

Perancangan pesan merupakan perancangan yang dilakukan untuk memberikan gambaran pesan apa saja yang akan ditampilkan pada saat sistem dijalankan oleh pengguna dalam melaksanakan perintah – perintah yang ada pada suatu menu. Berikut merupakan perancangan pesan yang dilakukan pada sistem yang akan dibangun pada *platform mobile* ditunjukkan pada tabel

##### 1. Pesan kesalahan data kurang lengkap halaman login

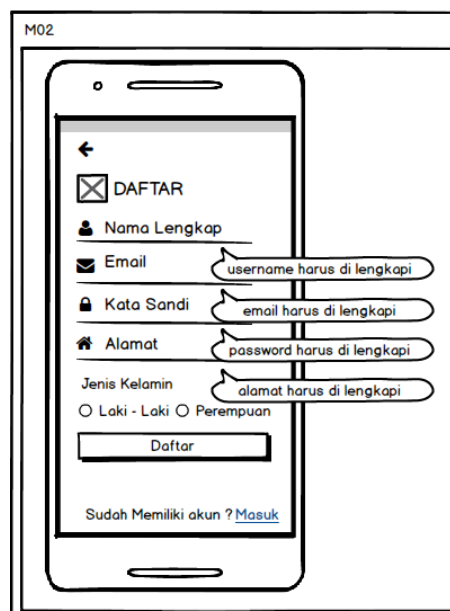
Pesan ini akan tampil ketika pengendara tidak mengisi masukan yang ada pada halaman login secara lengkap, ataupun terdapat masukan yang masih kosong pada form login. Berikut pesan kesalahan tersebut dapat dilihat pada gambar 3.54



**Gambar 3.55 Perancangan Pesan M01 Data Login Kurang Lengkap**

2. Pesan kesalahan data kurang lengkap halaman daftar

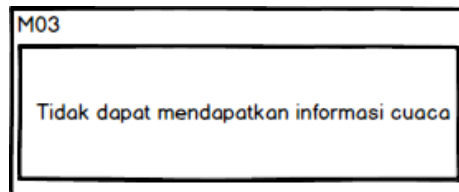
Pesan ini akan tampil ketika pengendara tidak mengisi masukan yang ada pada halaman daftar secara lengkap. Berikut pesan kesalahan tersebut dapat dilihat pada gambar 3.55



**Gambar 3.56 Perancangan Pesan M02 Data Pengguna Kurang Lengkap**

3. Pesan kesalahan informasi cuaca tidak ditemukan

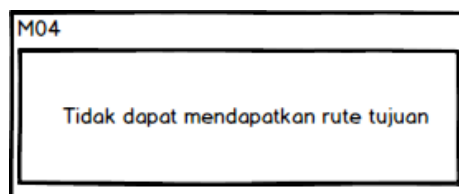
Pesan kesalahan ini akan tampil ketika pengendara hendak mendapatkan informasi cuaca yang ada pada aplikasi namun data tidak didapatkan. Berikut pesan kesalahan tersebut dapat dilihat pada gambar 3.56



**Gambar 3.57 Perancangan Pesan M03 Gagal Mendapatkan Informasi Cuaca**

4. Pesan kesalahan rute tujuan

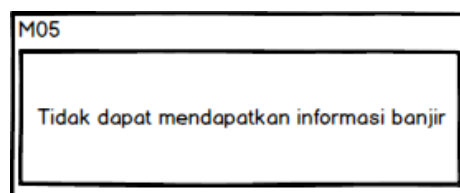
Pesan kesalahan ini akan tampil ketika sistem pengendara hendak mendapatkan rute tujuan namun sistem tidak dapat mendapatkan rute lokasi tujuan pengendara. Berikut pesan kesalahan tersebut dapat dilihat pada gambar 3.57



**Gambar 3.58 Perancangan Pesan M04 Gagal Mendapatkan Rute Tujuan**

5. Pesan kesalahan informasi banjir

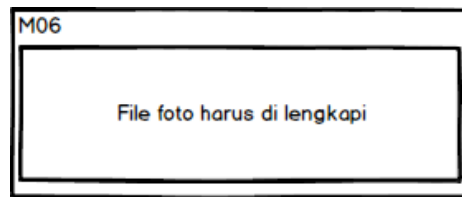
Pesan kesalahan ini akan tampil ketika pengendara hendak mendapatkan informasi banjir yang ada pada aplikasi namun data tidak didapatkan. Berikut pesan kesalahan tersebut dapat dilihat pada gambar 3.58



**Gambar 3.59 Perancangan Pesan M05 Gagal Mendapatkan Informasi Banjir**

6. Pesan kesalahan data foto harus dilengkapi

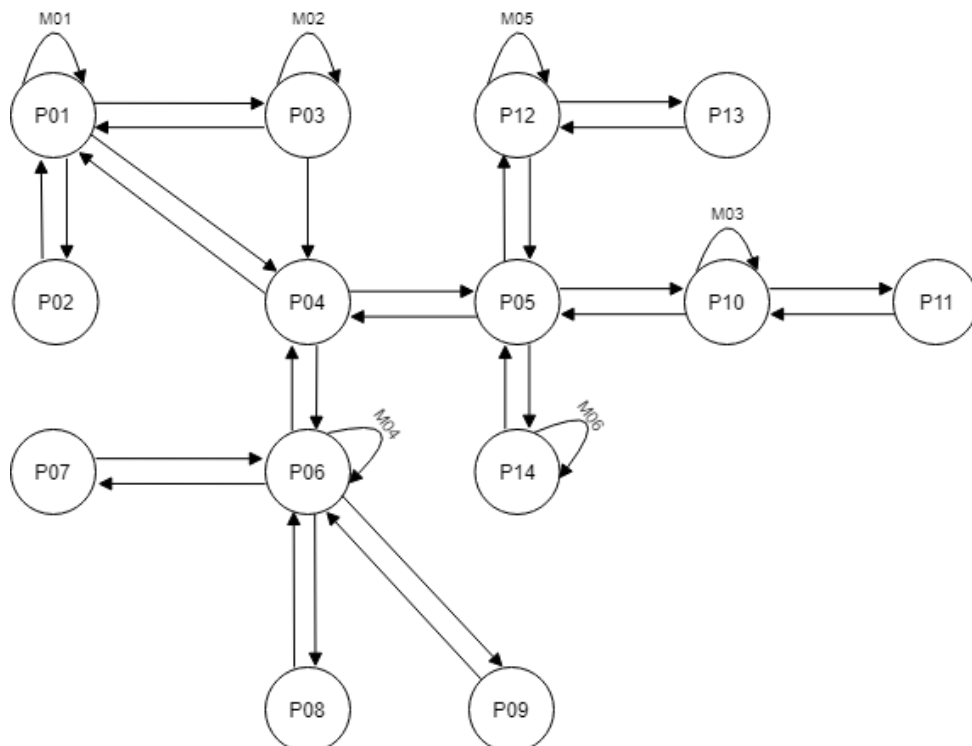
Pesan kesalahan ini akan tampil ketika pengendara tidak melengkapi masukan foto pada halaman pelaporan banjir. Berikut pesan kesalahan tersebut dapat dilihat pada gambar 3.59



**Gambar 3.60 Perancangan Pesan M05 Data Masukan Foto Kurang Lengkap**

### 3.2.5 Jaringan Semantik

Berikut perancangan jaringan semantik pada pembangunan perangkat lunak berbasis *mobile* yaitu sebagai berikut ditunjukkan pada gambar 3.60



**Gambar 3.61 Jaringan Semantik Aplikasi Pada Platform Mobile**

