

**PERANCANGAN APLIKASI PEMETAAN LALU LINTAS DAERAH  
RAWAN KECELAKAAN POLRES KABUPATEN LOMBOK  
TENGAH BERBASIS ANDROID**

**TUGAS AKHIR**



Disusun oleh :

**Adam Nizar**

**TI172000001**

**PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
(STMIK) LOMBOK  
PRAYA  
2024**

**PERANCANGAN APLIKASI PEMETAAN LALU LINTAS DAERAH  
RAWAN KECELAKAAN POLRES KABUPATEN LOMBOK  
TENGAH BERBASIS ANDROID**

untuk memenuhi persyaratan  
mencapai derajat Sarjana S1  
pada Program Studi Teknik Informatika



Disusun oleh :  
**Adam Nizar**  
**TI17200001**

**PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
(STMIK) LOMBOK  
PRAYA  
2024**

**HALAMAN PERSETUJUAN DOSEN PEMBIMBING**

**TUGAS AKHIR**

**PERANCANGAN APLIKASI PEMETAAN LALU LINTAS DAERAH  
RAWAN KECELAKAAN POLRES KABUPATEN LOMBOK  
TENGAH BERBASIS ANDROID**

yang dipersipakan dan disusun oleh :

**Adam Nizar**

**TI172000001**

telah disetujui oleh Dosen Pembimbing Tugas Akhir  
pada tanggal 1 juli 2024

**Dosen Pembimbing 1**

**Dosen Pembimbing 2**

**Hairul Fahmi, S.Kom.,M.Kom**  
**NIDN. 0815128201**

**Khairul Imtihan, S.Kom.,M.Kom**  
**NIDN. 0826098701**

Mengetahui,  
Ketua Program Studi Teknik Informatika  
STMIK Lombok

**Sofiansyah Fadli, S.Kom.,M.Kom**  
**NIK.90.021217.01**

**HALAMAN PENGESAHAN DOSEN PENGUJI**  
**TUGAS AKHIR**  
**PERANCANGAN APLIKASI PEMETAAN LALU LINTAS DAERAH**  
**RAWAN KECELAKAAN POLRES KABUPATEN LOMBOK**  
**TENGAH BERBASIS ANDROID**

yang disusun oleh

**Adam Nizar**

**TI17200001**

telah disetujui oleh Dosen Penguji Tugas Akhir  
pada tanggal 1 Juli 2024

**Nama Penguji**

Lalu Mutawalli, S.Kom., M.Ikom., M.Kom  
NIDN. 0805078901

**Tanda Tangan**

---

Wire Bagye, S.Kom., M.Kom  
NIDN. 0801018603

---

Maemun Saleh, S.Kom  
NUPN. 9908419990

---

Tugas Akhir ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Sarjana Komputer  
Tanggal

**KETUA STMIK LOMBOK**

Khairul Imtihan, S.Kom.,M.Kom  
NIK. 87.041012.01

## **HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR**

Yang bertanda tangan di bawah ini,

Nama : Adam Nizar

NIM : TI17200001

Tugas Akhir dengan Judul :

### **PERANCANGAN APLIKASI PEMETAAN LALU LINTAS DAERAH RAWAN KECELAKAAN POLRES KABUPATEN LOMBOK TENGAH BERBASIS ANDROID**

Menyatakan bahwa : seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Praya, 10 juli 2024

( Adam Nizar )

## **MOTTO**

“sebaik baiknya manusia adalah yang bermanfaat bagi manusia lain”

**(Husein Ja’far Al Hadar)**

“Jika kamu tidak tahan terhadap lelahnya belajar, maka kamu akan menanggung  
pahitnya kebodohan”

**(Imam Syafi’i)**

”Suara ibumu dirumah adalah nikmat yang harus patut kamu syukuri”

**(BoyCandra)**

“Apapun yang kamu lakukan setiap hari, apapun yang menjadi dunia kamu saat ini, inget baik-baik pesan ini!, jangan pernah sia-siakan setiap 5 menit kesempatan yang diberikan kepada kamu, karena kamu ga akan pernah tau, bisa jadi 5 menit itu justru akan merubah hidup kamu selamanya”

**(Pandji Pragiwaksono)**

## **PERSEMBAHAN**

Penyelesaian tugas akhir ini tidak terlepas dari dukungan, doa, serta bantuan baik secara moral maupun materi dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan rasa terima kasih yang mendalam kepada semua yang telah berkontribusi dalam menyelesaikan skripsi ini. Skripsi ini saya persembahkan kepada:

1. Kepada diri saya sendiri. Terima kasih atas perjuangan dan keteguhan yang tidak pernah menyerah hingga akhir.
2. Kepada kedua orang tua dan keluarga saya yang selalu mendoakan dan memberikan dukungan tanpa henti. Doa-doa kalian adalah kekuatan yang mendorong saya untuk terus maju dan berusaha yang terbaik.
3. Terimkasih kepada sahabat yang telah mendukung saya selama ini baik diwaktu susah maupun senang. Kita tidak akan pernah tau rasa lapar dan perjuangan akan menjadi sebuah kesuksesan di masa depan.
4. Bapak Hairul Fahmi, S.Kom.,M.Kom dan Khairul Imtihan, S.Kom .,M.kom selaku dosen pembimbing. Terima kasih atas kesabaran, waktu, ilmu, pengarahan dan motivasi yang telah diberikan selama penyusunan dan penulisan skripsi.
5. Kepada pasangan saya, Martina Fariza. Terima kasih atas dukungan dan cinta yang telah diberikan.
6. Kepada Lukman Hakim. Terimkasih yang telah meluangkan waktu dan memberikan bimbingan teknis selama proses penggerjaan tugas akhir ini.

## **KATA PENGANTAR**

Puji syukur selalu terpanjatkan kehadirat Allah SWT atas rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir ini meskipun masih banyak terdapat kekurangan.

Tugas Akhir ini ditujukan untuk memenuhi salah satu persyaratan untuk memperoleh gelar Sarjana Komputer di Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Lombok.

Penulis menyadari bahwa tanpa bimbingan, bantuan, dan doa dari berbagai pihak, Tugas Akhir ini tidak akan dapat diselesaikan tepat pada waktunya. Oleh karena itu, penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam proses penyusunan dan penulisan skripsi ini, yaitu kepada :

1. Kedua orang tua dan keluarga tercinta yang telah memberikan dukungan dan doa yang tiada terputus.
2. Bapak Hairul Fahmi, S.Kom.,M.Kom dan Bapak Khairul Imtihan, S.Kom.,M.Kom selaku Dosen Pembimbing.
3. Rekan-rekan seperjuangan dan seluruh pihak, yang tidak bisa saya sebutkan satu per satu.

Akhir kata penulis berharap skripsi ini dapat bermanfaat bagi pihak lain pada umumnya dan rekan-rekan di STMIK Lombok pada khususnya.

Praya, 20 Juni 2024

Penulis

## DAFTAR ISI

<b>HALAMAN SAMPUL.....</b>	<b>i</b>
<b>HALAMAN JUDUL.....</b>	<b>ii</b>
<b>HALAMAN PERSETUJUAN DOSEN PEMBIMBING.....</b>	<b>iii</b>
<b>HALAMAN PENGESAHAN DOSEN PENGUJI .....</b>	<b>iv</b>
<b>HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR .....</b>	<b>v</b>
<b>MOTTO .....</b>	<b>vi</b>
<b>PERSEMBAHAN.....</b>	<b>vii</b>
<b>KATA PENGANTAR.....</b>	<b>viii</b>
<b>DAFTAR ISI.....</b>	<b>ix</b>
<b>DAFTAR TABEL.....</b>	<b>xii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xiii</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>xvi</b>
<b>DAFTAR SINGKATAN.....</b>	<b>xvii</b>
<b>ABSTRAK .....</b>	<b>xviii</b>
<b>ABSTRACT .....</b>	<b>xix</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Sistematika Penulisan.....	5
<b>BAB II LANDASAN TEORI &amp; TINJAUAN PUSTAKA .....</b>	<b>7</b>
2.1.1 Tinjauan Pustaka .....	7
2.2 Landasan Teori .....	11
2.2.1 Sistem.....	11
2.2.2 Informasi.....	13
2.2.3 Sistem Informasi Geografis .....	13

2.2.4 <i>Emulator Android</i> .....	15
2.2.5 <i>Android</i> .....	15
2.2.6 <i>Android Studio</i> .....	16
2.2.7 <i>Unified Modelling Language (UML)</i> .....	16
2.2.8 <i>Waterfall</i> .....	32
2.2.9 <i>Software Development Kit (SDK)</i> .....	33
2.2.10 <i>Database</i> .....	34
2.2.11 <i>Google Maps</i> .....	35
2.2.12 Satuan Lalu Lintas (Satlantas) .....	36
2.2.13 Kepolisian Resor (POLRES) .....	36
1.2.14 <i>System Usability Scale (SUS)</i> .....	38
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>40</b>
3.1 Tempat Dan Jadwal Penelitian .....	40
3.1.1 Tempat Penelitian .....	40
3.1.2 Jadwal Penelitian .....	40
3.2 Alat Dan Bahan Penelitian .....	41
3.2.1 Alat Penelitian.....	41
3.2.2 Bahan Penelitian .....	42
3.3 Alur Penelitian.....	43
3.4 Pengumpulan Data.....	44
3.4.1 Observasi .....	44
3.4.2 Wawancara.....	44
3.5 Perancangan Sistem.....	45
3.5.1 <i>Use Case Diagram</i> .....	45
3.5.2 <i>Activity Diagram</i> .....	46
3.5.3 <i>Sequence Diagram</i> .....	54

3.6 Rancangan Struktur <i>Menu</i> .....	62
3.7 Perancangan <i>Design Interface</i> .....	63
3.8 Metode Testing .....	65
3.8.1 <i>System Usability Scale (SUS)</i> .....	65
<b>BAB IV IMPLEMENTASI DAN PEMBAHASAN .....</b>	<b>66</b>
4.1 <i>Database</i> .....	66
4.1.1 <i>Authentication</i> .....	66
4.1.2 Lokasi Kecelakaan .....	67
4.1.3 Notifikasi .....	68
4.1.4 <i>User Location</i> .....	68
4.2 <i>Interface</i> .....	69
4.2.1 Halaman <i>Login</i> .....	69
4.2.3 Halaman Utama .....	70
4.2.4 Halaman Detail Lokasi .....	71
4.2.5 Halaman <i>Sidebar Profile</i> .....	72
4.2.6 Halaman Notifikasi .....	73
4.2.7 Halaman Hasil Pencarian Lokasi .....	74
4.2.8 Halaman Tambah Data .....	74
4.2.9 Halaman <i>Pop Up Logout</i> .....	75
4.3 Hasil Pengujian .....	76
<b>BAB IV PENUTUP .....</b>	<b>79</b>
5.1 Kesimpulan .....	79
5.2 Saran .....	79
<b>DAFTAR PUSTAKA .....</b>	
<b>LAMPIRAN .....</b>	

## **DAFTAR TABEL**

Tabel 3.1 Jadwal Penelitian.....	41
Tabel 4.1 Hasil Penilaian Responden.....	77

## DAFTAR GAMBAR

Gambar 2.1 <i>Actor/Role</i> .....	17
Gambar 2.2 <i>Use Case</i> .....	18
Gambar 2.3 <i>Subject Boundary</i> .....	18
Gambar 2.4 <i>Association Relationship</i> .....	18
Gambar 2.5 <i>Include Relationship</i> .....	19
Gambar 2.6 <i>Include Relationship</i> .....	19
Gambar 2.7 <i>Generalization Relationship</i> .....	19
Gambar 2.8 <i>Action</i> .....	20
Gambar 2.9 <i>Activity</i> .....	20
Gambar 2.10 <i>Object Node</i> .....	20
Gambar 2.11 <i>Control Flow</i> .....	20
Gambar 2.12 <i>Object Flow</i> .....	21
Gambar 2.13 <i>Symbol Initail</i> .....	21
Gambar 2.14 <i>Symbol Initial Node</i> .....	21
Gambar 2.15 <i>Final-flow Node</i> .....	21
Gambar 2.16 <i>Decision Node</i> .....	22
Gambar 2.17 <i>Merge Node</i> .....	22
Gambar 2.18 <i>Fork Node</i> .....	22
Gambar 2.19 <i>Join Node</i> .....	23
Gambar 2.20 <i>Symbol Swimlane</i> .....	23
Gambar 2.21 <i>Symbol Class</i> .....	24
Gambar 2.22 <i>Attrubute</i> .....	24
Gambar 2.23 <i>Operation</i> .....	24
Gambar 2.24 <i>Generalizaion</i> .....	25
Gambar 2.25 <i>Association</i> .....	25
Gambar 2.26 Tahap <i>multiplicity</i> .....	26
Gambar 2.27 <i>Symbol Aggregation</i> .....	27
Gambar 2.28 <i>Objek aggregation</i> .....	27
Gambar 2.29 <i>Symbol Objek Composition</i> .....	28
Gambar 2.30 <i>Objek Composite</i> .....	28
Gambar 2.31 <i>Symbol Actor</i> .....	28

Gambar 2.32 <i>Symbol Object</i> .....	29
Gambar 2.33 <i>Boundary Class</i> .....	29
Gambar 2.34 <i>Control Class</i> .....	29
Gambar 2.35 <i>Entity Class</i> .....	30
Gambar 2.36 <i>Symbol Lifeline</i> .....	30
Gambar 2.37 <i>Execution Occurrence</i> .....	30
Gambar 2.38 <i>Symbol Message</i> .....	31
Gambar 2.39 <i>Guard Condition</i> .....	31
Gambar 2.40 <i>Object Destruction</i> .....	31
Gambar 2.41 <i>Symbol Frame</i> .....	31
Gambar 2.42 Tahapan Metode <i>Waterfall</i> .....	32
Gambar 2.43 Aturan Menghitung SUS .....	38
Gambar 3.1 Tempat Penelitian.....	40
Gambar 3.2 Tahapan Penelitian .....	43
Gambar 3.3 <i>Use Case Diagram</i> .....	45
Gambar 3.4 Proses <i>Login User</i> .....	46
Gambar 3.5 Proses Melihat Peta .....	47
Gambar 3.6 Proses Melihat Detail Lokasi Kecelakaan.....	48
Gambar 3.7 Proses Pencarian Lokasi Tujuan .....	49
Gambar 3.8 Proses Mendapatkan Notifikasi.....	50
Gambar 3.9 Proses Melihat Notifikasi .....	51
Gambar 3.10 Proses Tambah Data.....	52
Gambar 3.11 Proses <i>Logout</i> .....	53
Gambar 3.12 <i>Sequence Diagram Login</i> .....	54
Gambar 3.13 <i>Sequence Diagram</i> Halaman Utama .....	55
Gambar 3.14 <i>Sequence Diagram</i> Halaman Detail Lokasi Kecelakaan.....	56
Gambar 3.15 <i>Sequence Diagram</i> Mendapatkan Notifikasi.....	57
Gambar 3.16 <i>Sequence Diagram View</i> Notifikasi .....	58
Gambar 3.17 <i>Sequence Diagram Sidebar Profile</i> .....	59
Gambar 3.18 <i>Sequence Diagram</i> Pencarian Lokasi.....	60
Gambar 3.19 <i>Sequence Diagram</i> Tambah Data.....	61
Gambar 3.20 <i>Sequence Diagram Logout</i> .....	62

Gambar 3.21 Rancangan <i>Design Interface</i> .....	63
Gambar 3.22 <i>Form Input Data</i> .....	75
Gambar 4.1 Firebase <i>Realtime Database</i> .....	66
Gambar 4.2 Autentifikasi Dengan Akun Google .....	67
Gambar 4.3 Objek Nama Kecamatan.....	67
Gambar 4.4 Objek Lokasi Kecelakaan.....	68
Gambar 4.5 Objek Notifikasi .....	68
Gambar 4.6 Objek <i>User Location</i> .....	68
Gambar 4.7 Halaman <i>Login</i> .....	69
Gambar 4.8 Halaman Pemilihan <i>Account Google</i> .....	70
Gambar 4.9 Halaman Utama.....	71
Gambar 4. 10 Halaman Detail Lokasi.....	72
Gambar 4.11 Halaman <i>Sidebar Profile</i> .....	73
Gambar 4.12 Halaman Notifikasi .....	73
Gambar 4.13 Halaman Hasil Pencarian .....	74
Gambar 4.14 <i>Pop Up Logout</i> .....	76
Gambar 4.15 Penentuan Hasil Penilaian Dengan Menggunakan <i>Acceptability</i> , <i>Grade Scale</i> ,Dan <i>Adjective Rating</i> .....	78

## **DAFTAR LAMPIRAN**

Lampiran 1 Dokumentasi Wawancara Dan Pengambilan Data.....	1-6
Lampiran 2 Dokumentasi <i>davelopment</i> Aplikasi.....	7-10
Lampiran 3 Kusioner Evaluasi.....	10-12
Lampiran 4 Dokumentasi Pengunaan Aplikasi.....	12-13
Lampiran 5 Penerimaan Notifikasi <i>Smartphone</i> Secara <i>Real-time</i> .....	14

## **DAFTAR SINGKATAN**

1. SIG : Sistem Informasi Geografis
2. *GIS* : *Geographic Information System*
3. *GPS* : *Global Positioning System*
4. *UML* : *Unified Modeling Language*
5. *SDK* : *Software Development Kit*
6. POLRES : Kepolisian Resor
7. *SUS* : *System Usability Scale*

## **ABSTRAK**

Salah satu upaya untuk mengurangi angka kecelakaan tersebut adalah dengan menyediakan informasi mengenai daerah rawan kecelakaan kepada masyarakat. Penelitian ini bertujuan untuk merancang dan membangun sebuah aplikasi berbasis Android yang dapat memetakan daerah rawan kecelakaan di wilayah Polres Kabupaten Lombok Tengah. Aplikasi ini dirancang menggunakan Sistem Informasi Geografis (SIG) yang memanfaatkan data kecelakaan dari tahun 2019, 2022, dan 2023. Pengguna aplikasi dapat menerima notifikasi peringatan ketika mendekati area rawan kecelakaan, sehingga diharapkan dapat meningkatkan kewaspadaan dan mengurangi risiko kecelakaan. Proses pengembangan aplikasi ini menggunakan metode *waterfall*, mulai dari analisis kebutuhan, desain sistem, implementasi, hingga pengujian. Hasil dari penelitian ini menunjukkan bahwa aplikasi yang dibangun mampu menyediakan informasi lokasi kecelakaan secara akurat dan memberikan notifikasi yang efektif kepada pengguna. Dengan demikian, aplikasi ini dapat menjadi alat yang berguna dalam meningkatkan keselamatan berlalu lintas di Kabupaten Lombok Tengah.

Kata Kunci : Aplikasi, Sistem informasi geografis, Google *Maps*, *waterfall*

## **ABSTRACT**

*One effort to reduce the number of accidents is by providing information about accident-prone areas to the public. This research aims to design and build an Android-based application that can map accident-prone areas in the Central Lombok Regency Police area. This application is designed using a Geographic Information System (GIS) which utilizes accident data from 2019, 2022 and 2023. Application users can receive warning notifications when approaching accident-prone areas, so it is hoped that they can increase awareness and reduce the risk of accidents. The application development process uses the waterfall method, starting from requirements analysis, system design, implementation, to testing. The results of this research show that the application developed is able to provide accurate accident location information and provide effective notifications to users. Thus, it is hoped that this application can become a useful tool in improving traffic safety in Central Lombok Regency.*

*Keyword :applications, Geographic information systems, Google Maps, waterfall*

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Perkembangan teknologi informasi saat ini sangat pesat dan berpengaruh terhadap semua aspek, salah satunya teknologi sistem informasi geografis. Sistem informasi geografis (SIG) adalah gabungan tiga unsur pokok yaitu sistem, informasi, dan geografis. Dengan ketiga unsur pokok tersebut sangat membantu dalam memahami sistem informasi geografis. Dari unsur-unsur pokok tersebut, maka sistem informasi geografis dapat di artikan tipe sistem informasi, tetapi dengan unsur “geografis”. Jadi sistem informasi geografis merupakan sistem yang menekankan pada unsur “informasi geografis”. Sistem informasi geografis adalah suatu sistem berbasis *computer* yang digunakan untuk mengelolah dan menyimpan data atau informasi yang berasal dari geografis. SIG sudah digunakan secara luas untuk mengakses informasi tentang suatu lokasi [1].

SIG memiliki fungsi utama untuk memasukkan, menyimpan, mengelola, menganalisis, memodelkan, dan memvisualisasikan data spasial. Kemampuan SIG meliputi pengukuran jarak, luas, dan volume, pemetaan data spasial menjadi peta yang mudah dipahami, pemantauan perubahan lingkungan atau wilayah, serta pemodelan spasial untuk simulasi dan prediksi. Selain itu, SIG mendukung input data, manajemen data, *query* data, analisis dan pemodelan spasial, serta visualisasi dan representasi data dalam format peta, grafik, dan tampilan tiga dimensi. Keunggulan SIG terletak pada kemampuannya dalam mengolah dan mengelola data spasial secara efisien, melakukan analisis spasial yang mendalam, dan menyajikan visualisasi data yang membantu pengambilan keputusan yang lebih baik. SIG digunakan di berbagai bidang seperti perencanaan kota, manajemen sumber daya

alam, mitigasi bencana, dan transportasi, yang semuanya meningkatkan efisiensi, mengurangi biaya, dan memungkinkan keputusan berbasis data yang lebih akurat [2].

Kecelakaan adalah suatu kejadian dimana setidaknya satu kendaraan bermotor terlibat yang terjadi di jalan umum dan mengakibatkan cedera atau luka-luka. Berdasarkan Undang-Undang No.22 Tahun 2009 pasal 1 ayat 24 tentang Lalu Lintas dan Angkutan Jalan yaitu kecelakaan lalu lintas adalah suatu peristiwa di jalan yang tidak diduga dan tidak disengaja melibatkan kendaraan dengan atau pengguna jalan lain yang mengakibatkan korban manusia dan kerugian harta benda [3].

Berdasarkan laporan Kasat Lantas Polres Lombok Tengah IPTU Abdul Rachman, SIK mengungkapkan bahwa berdasarkan data yang masuk di Unit Kecelakaan Lalulintas Sat Lantas Polres Lombok Tengah pada tahun 2023 setiap bulan terjadi Kecelakaan Lalulintas. Berdasarkan *update data* per 29 November 2023 jumlah angka kecelakaan Lalulintas di Unit Laka Lantas Polres Lombok baik yang Meninggal Dunia (MD), Luka Berat (LB) maupun Luka Ringan (LR). Pada bulan januari sampai dengan November terdapat Total jumlah kasus Lakalantas pada tahun 2023 sebanyak 277 dengan rincian Meninggal Dunia (MD) sebanyak 83, Luka Berat (LB) sebanyak 58 dan Luka Ringan (LR) sebanyak 231. penyebab utama kecelakaan ini adalah ketidak patuhan terhadap aturan lalu lintas yang berlaku, kelalaian dalam berkendara, dan kurangnya informasi mengenai daerah rawan kecelakaan [4].

Permasalahan yang dihadapi dalam penelitian ini adalah belum adanya sistem yang mampu memetakan daerah rawan kecelakaan secara efektif di wilayah

Kabupaten Lombok Tengah. Kurangnya informasi mengenai lokasi daerah rawan kecelakaan membuat sulit bagi pihak berwenang untuk mengambil tindakan yang tepat. Hal ini menyebabkan tingginya angka kecelakaan di wilayah tersebut, yang berdampak negatif terhadap keselamatan masyarakat. Penelitian ini bertujuan untuk mengembangkan aplikasi yang dapat memetakan daerah rawan kecelakaan dan memberikan peringatan saat mendekati area rawan kecelakaan di wilayah Kabupaten Lombok Tengah. Untuk menyelesaikan permasalahan penelitian ini, metode yang digunakan adalah metode waterfall. Metode waterfall dipilih karena pendekatannya yang sistematis dan terstruktur, yang memastikan setiap tahap dalam pengembangan aplikasi dilakukan secara berurutan dan lengkap sebelum melanjutkan ke tahap berikutnya [5].

Berdasarkan hasil wawancara yang telah dilakukan peneliti di Satuan Lalu Lintas (Satlantas) Polres Lombok Tengah Terhadap Kepala Bidang Satuan Lalu Lintas IPTU Abdul Rachman Virga Maulidhany Yusuf, S.Tr.K., S.I.K. pada bulan januari 2024 mengatakan bahwa berbagai upaya kepolisian dalam mengurangi angka kecelakaan seperti penyuluhan, pemasangan spanduk, edukasi kepada masyarakat hingga petugas kepolisian terjun langsung ke lokasi titik daerah yang sering terjadi kecelakaan untuk mengimbau kepada masyarakat agar mematuhi peraturan tata tertib berlalu lintas. Meskipun berbagai upaya telah diimplementasikan, pengamatan peneliti menunjukkan bahwa upaya tersebut masih kurang efektif. Oleh karena itu, perlunya pembagunan Sistem Informasi Geografis (SIG) untuk memetakan daerah yang sering rawan kecelakaan dan memberikan notifikasi peringatan secara *rela-time* kepada pengguna saat mendekati daerah

tersebut, serta meningkatkan kesadaran masyarakat di Kabupaten Lombok Tengah akan pentingnya keselamatan berlalu lintas.

### **1.2 Rumusan Masalah**

Dari latar belakang pada penelitian, dapat diuraikan rumusan masalah sebagai berikut:

Bagaimana membuat sistem yang dapat digunakan untuk memberikan peringatan (*warning*) pada area kecelakaan di daerah Kabupaten Lombok Tengah?

### **1.3 Batasan Masalah**

Berdasarkan perumusan masalah yang diuraikan sebelumnya, maka masalah akan dibatasi pada :

1. Penelitian ini menggunakan data *Blackspot* tahun 2019, 2022 dan 2023.
2. Penelitian ini berfokus pada pengembangan aplikasi untuk pengguna jalan di kabupaten lombok tengah.

### **1.4 Tujuan Penelitian**

Tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Merancang dan membangun sistem daerah rawan kecelakaan.
2. Menyediakan informasi daerah rawan kecelakaan kepada masyarakat melalui sistem geografis.

### **1.5 Manfaat Penelitian**

Manfaat dari penelitian ini antara lain :

1. Memberikan informasi tentang daerah rawan kecelakaan dan notifikasi lewat *smartphone* untuk meningkatkan kewaspadaan masyarakat agar lebih berhati-hati dalam berkendara.

2. Mengurangi angka kecelakaan dengan membangun sistem informasi geografis berbasis android untuk menyajikan informasi.

### **1.6 Sistematika Penulisan**

Penelitian ini terdiri dari tiga bab dengan sistematika penulisan sebagai berikut:

#### **HALAMAN JUDUL**

#### **HALAMAN PENGESAHAN TUGAS AKHIR**

#### **DAFTAR ISI**

#### **DAFTAR TABEL**

#### **DAFTAR GAMBAR**

#### **DAFTAR LAMPIRAN**

#### **DAFTAR SINGKATAN**

#### **BAB I PENDAHULUAN**

Pada bab I berisi tentang latar belakang, perumusan masalah, batasan penelitian, tujuan, manfaat penelitian, dan sistematika penulisan.

#### **BAB II LANDASAN TEORI & TINJAUAN PUSTAKA**

Pada bab ini memuat dasar-dasar teori pendukung penelitian, Tinjauan Pustaka berisikan tinjauan pustaka (literatur *review*) penelitian terdahulu maksimal 5 tahun terakhir,

#### **BAB III METODOLOGI PENELITIAN**

Berisikan uraian secara terperinci tentang jadwal dan kegiatan penelitian, langkah-langkah penelitian yang dilaksanakan. Menjelaskan secara rinci mengenai metode yang digunakan dalam penelitian.

#### **BAB IV HASIL DAN PEMBAHASAN**

Bab ini menjelaskan tentang hasil penelitian yang telah didapatkan.

Bagian ini memuat hasil penelitian dan pembahasan yang sifatnya terpadu dan tidak dipecah menjadi sub judul tersendiri.

## **BAB V PENUTUP**

Pada bab ini berisikan tentang kesimpulan dari penulisan skripsi yang telah dibuat dan penulis memberikan saran-saran yang sekiranya dapat bermanfaat

## **DAFTAR PUSTAKA**

## **LAMPIRAN**

## BAB II

### LANDASAN TEORI & TINJAUAN PUSTAKA

#### 2.1 Tinjauan Pustaka

Penelitian ini dilakukan setelah membaca dan mempelajari karya ilmiah terdahulu yang memiliki persamaan kasus dengan judul skripsi yang di angkat oleh peneliti. Karya ilmiah yang telah dikumpulkan kemudian dikaitkan dengan satu sama lain untuk menemukan persamaan, sehingga peneliti dapat mencari hal baru pada penelitian yang akan dilakukan.

Penelitian dengan judul Pembuatan Aplikasi Sistem Peringatan Daerah Rawan Kecelakaan Berbasis Android Di Daerah Istimewa Yogyakarta dilakukan oleh [6]. Data yang digunakan adalah data riwayat kecelakaan tahun 2019 dan 2020. Proses yang dilakukan adalah pengolahan data, pembuatan desain tatap muka, pembuatan sistem aplikasi, dan pengujian usabilitas. Pengolahan data dilakukan untuk menentukan titik yang akan digunakan sebagai titik rawan yang memenuhi kriteria terdapat lebih dari 10 riwayat kecelakaan terjadi dalam rentang waktu 2 tahun di sekitar area tersebut. Selanjutnya tahapan pembuatan sistem dilakukan dengan menyiapkan bagian server kemudian membangun sistem di perangkat android yang bertindak sebagai *client*. Data disimpan dalam *web server* yang dapat diakses oleh perangkat android. Untuk mengevaluasi hasil kegiatan ini, aplikasi yang telah dibuat kemudian dievaluasi menggunakan uji usabilitas yang dilakukan dengan metode kuisioner secara *online* dengan mengambil beberapa responden. Hasil dari penelitian ini adalah sebuah aplikasi yang dapat memberikan informasi area titik rawan kecelakaan yang ada di Daerah Istimewa Yogyakarta serta dapat memberikan peringatan apabila pengguna mendekati area tersebut. Aplikasi

berjalan normal dalam *Android* versi 7.0 ke atas. Persebaran area *blackspot* atau titik rawan disajikan dalam peta dengan sumber *Google Maps*. Hasil analisis melalui kuisioner pengujian aplikasi menunjukkan bahwa aplikasi berjalan dengan baik sesuai dengan fungsinya.

Penelitian dengan judul Sistem Informasi Geografis Titik Rawan Kecelakaan Daerah Sumatera Barat Berbasis *Web* yang dilakukan oleh [7]. Menurut data yang dimiliki Direktorat Lalu Lintas Polda Sumbar, jumlah daerah rawan kecelakaan di Sumatera Barat terdapat 48 titik. Hal ini menyebabkan dibutuhkannya sebuah Sistem Informasi Geografis daerah rawan kecelakaan di Sumatera Barat berbasis *web* untuk menggambarkan peta lokasi daerah rawan beserta informasi yang dibutuhkan. Aplikasi Sistem Informasi Geografis Daerah Rawan Kecelakaan Berbasis *Web*, dengan bahasa pemograman (JavaScript dan PHP), MySql sebagai basis data dan *Google Maps Api*. Hasil akhir dari penelitian ini adalah terciptanya aplikasi SIG berbasis *web* yang dapat menyajikan informasi lokasi daerah rawan kecelakaan di Sumatera Barat secara terinci dan dapat diakses dengan mudah oleh masyarakat.

Penelitian yang dilakukan oleh [8] dengan judul Perancangan Sistem Informasi Geografis Perlengkapan Jalan Berbasis Daerah Rawan Kecelakaan. Untuk meningkatkan pelayanan pada lalu lintas yang ada, salah satu upaya yang dilakukan adalah peningkatan kualitas perlengkapan jalan. Permasalahan yang terjadi data dan sistem manajemen pada Dinas Perhubungan Kabupaten Balangan saat ini dapat dikatakan buruk dan tidak bersifat *up to date*. Disamping itu banyak terdapat jalan yang belum di lengkapi dengan rambu dan pada daerah rawan kecelakaan belum memiliki perlengkapan jalan yang memadai. Tujuan penelitian ini untuk analisis

kebutuhan rambu pada jalan di Kabupaten Balangan serta kebutuhan perlengkapan jalan pada daerah rawan kecelakaan. Hasil analisis tersebut ditampilkan menggunakan metode Sistem Informasi Geografis pada aplikasi ArcGIS 10.3 yang hasilnya untuk memudahkan Dinas Perhubungan dalam melakukan perawatan dan pengambilan keputusan dalam perencanaan perlengkapan jalan sebagai upaya peningkatan pelayanan dan keselamatan transportasi.

Penelitian dengan judul Perancangan Sistem Informasi Geografis Statistik Kepolisian (Sigap) Kabupaten Gorontalo oleh [9]. Pada saat ini kebutuhan akan data yang cepat, tepat dan akurat sangatlah harus dilakukan dengan mudah menggunakan teknologi informasi yang memadai. Tujuan dari penelitian ini yakni membuat suatu inovasi tentang kepolisian baik untuk administrasi kepolisian, data kejahatan, rawan kecelakaan serta data kepolisian lainnya baik dalam bentuk data terstruktur maupun bentuk statistik. Dalam sistem ini juga diberikan kemudahan dalam visualisasi menggunakan peta *Geographic Information System (GIS)* yang menampilkan informasi secara geografis data kejahatan dan daerah rawan kecelakan sehingga memberikan informasi data kepada kepolisian dan masyarakat pada khususnya. Metode pengembangan sistem yang digunakan yakni *research and development*. Metode ini sangat baik dalam mengembangkan sistem secara sistematis dari awal sampai akhir pada tahapan maintenance. Hasil dari penelitian ini selain memberikan tampilan visual geografis juga memberikan kemudahan dari segi informasi maupun statistik kepolisian di Kabupaten Gorontalo.

Penelitian terdahulu dengan judul Pengembangan Sistem Informasi Geografis Pemetaan Lokasi Kecelakaan Berbasis *Webgis* (Studi Kasus: Daerah Operasional Polres Kota Batu) oleh [10]. Saat ini belum terdapat sistem informasi yang

digunakan untuk menangani masalah kecelakaan di Kota Batu. Salah satu solusi yang dapat digunakan adalah membangun sebuah sistem informasi geografis (SIG) yang dapat mengidentifikasi lokasi-lokasi yang paling tinggi tingkat kecelakaannya. Data yang digunakan dalam pengembangan sistem ini adalah data koordinat, data-data yang berhubungan dengan kecelakaan lalu-lintas yang didapatkan dari Polres Kota Batu, dan data jaringan jalan. Data yang dikumpulkan dalam periode tahun 2013-2016. SIG dibangun akan memanfaatkan metode *Street Profile Analysis* untuk mengetahui perkembangan kecelakaan lalu-lintas pada segmen jalan tertentu. Sedangkan untuk menentukan tingkat kecelakaan lalu-lintas menggunakan *Equivalent Accident Number (EAN)*. Hasil dari SIG ini adalah sebuah WebGIS yang menampilkan lokasi-lokasi kecelakaan beserta identifikasi lokasi yang memiliki tingkat kecelakaan tertinggi. Dengan sistem ini lokasi yang memiliki tingkat kecelakaan lalu-lintas tertinggi berhasil diidentifikasi terdapat pada Jalan Raya Waturejo. Sedangkan hasil pemetaan lokasi kecelakaan lalu-lintas ditemukan sejumlah 26 titik selama tahun 2013-2016. Hasil pengujian *black box testing* sebesar 100%. Kemudian hasil *compatibility testing*, sistem tidak dapat berjalan pada *browser Internet Explorer* saja. Sedangkan hasil pengujian UAT mendapatkan hasil hampir 88% untuk pengguna.

Berdasarkan 5 (lima) karya ilmiah yang telah diuraikan, penelitian-penelitian sebelumnya dalam proses authentifikasi masih secara manual karena pengguna masih registrasi *account* dan menginputkan *username* dan *password* untuk *login* kedalam sistem dan tingkat keamanan pengguna masih rentan, sedangkan pada penilitian ini terdapat inovasi baru dimana proses autentifikasi dilakukan dengan menggunakan akun Google untuk *login* kedalam sistem. Pendekatan ini tidak hanya

mempercepat proses masuk ke dalam sistem, tetapi juga meningkatkan keamanan karena lokasi penyimpanan menggunakan server google dan dengan memanfaatkan fitur-fitur keamanan yang disediakan oleh Google, seperti autentikasi dua faktor, pemantauan aktivitas *login* yang mencurigakan, dan manajemen izin aplikasi. Hal ini dapat mengurangi risiko akses yang tidak sah dan memberikan pengalaman pengguna yang lebih aman dan nyaman.

Penelitian ini mengembangkan aplikasi dengan fitur pengiriman notifikasi peringatan secara *real-time* kepada pengguna saat mendekati area rawan kecelakaan dan pencarian *rute* lokasi tujuan untuk memudahkan pengguna dalam menentukan perjalanan. antarmuka aplikasi dirancang untuk lebih *user-friendly* dengan navigasi yang mudah untuk membantu pengguna baru memahami fitur-fitur aplikasi. Sedangkan pada uraian lima penelitian sebelumnya fitur belum diintegrasikan kedalam aplikasi.

## 2.2 Landasan Teori

### 2.2.1 Sistem

Pengertian sistem dapat ditelusuri dari sisi asal katanya, yakni dari Bahasa Inggris yang disebut sebagai *system*. Seluruh Kamus Inggris-Indonesia menerjemahkan kata *system* sebagai susunan. Misalnya yang terdapat dalam kata sistem syaraf berarti susunan syaraf, sistem jaringan berarti susunan jaringan, dan lain sebagainya. Sistem secara sederhana didefinisikan sebagai himpunan dari sekelompok elemen-elemen yang mempunyai keterkaitan dan keterhubungan satu sama lainnya dan semuanya itu membentuk satu kesatuan yang utuh. Secara formal, memberi batasan sistem sebagai sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Batasan sistem tersebut

sesuai untuk suatu organisasi atau perusahaan maupun suatu bidang fungsional tertentu. Organisasi terdiri dari sejumlah sumber daya, dan sumber daya tersebut bekerja menuju tercapainya suatu tujuan tertentu yang ditentukan oleh pemilik atau *level* manajemen/pimpinan. Mendefinisikan sistem sebagai suatu kesatuan usaha yang terdiri dari bagian-bagian yang berkaitan satu sama lain yang berusaha untuk mencapai suatu tujuan dalam suatu lingkungan yang kompleks. Pengertian tersebut mencerminkan adanya beberapa bagian dan hubungan antarbagian, yang menunjukkan kompleksitas dari sistem yang meliputi kerja sama antara bagian yang interdependen satu sama lain. Selain itu, dapat dilihat bahwa sistem berusaha mencapai tujuan. Pencapaian tujuan dimaksud menyebabkan timbulnya dinamika, perubahan yang terus-menerus perlu dikembangkan dan dikendalikan. Mengacu pada definisi atau pengertian sistem sebagaimana yang telah diuraikan, dapat dinyatakan bahwa sistem sebagai gugus dari elemen-elemen yang saling berinteraksi secara teratur. Interaksi tersebut adalah dalam rangka mencapai tujuan atau subtujuan. Sistem juga terdiri dari berbagai macam, antara lain sistem terbuka, sistem tertutup, dan sistem dengan umpan-balik (*feedback*). Berdasarkan definisi sistem, kita dapat memahami bahwa sesuatu dapat dinyatakan sebagai sistem apabila memenuhi syarat-syarat sebagai berikut [11].

1. Sistem harus dibentuk untuk menyelesaikan tujuan.
2. Elemen sistem harus mempunyai rencana yang ditetapkan.
3. Adanya hubungan di antara elemen sistem.
4. Unsur dasar dari proses (arus informasi, energi, dan material) lebih penting dari pada elemen sistem.
5. Tujuan organisasi lebih penting dari pada tujuan elemen.

### **2.2.2 Informasi**

Informasi adalah sekumpulan data fakta yang diorganisasi atau diolah dengan cara tertentu sehingga mempunyai arti bagi penerima. Data yang telah diolah menjadi sesuatu yang berguna bagi si penerima maksudnya yaitu dapat memberikan keterangan atau pengetahuan. Informasi sangat penting pada suatu organisasi / instansi. Informasi (*information*) dapat didefinisikan sebagai berikut “Mendefinisikan informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seorang yang menggunakan data tersebut” [12].

### **2.2.3 Sistem Informasi Geografis**

Sebuah sistem informasi geografis (SIG) mengintegrasikan perangkat keras, perangkat lunak, dan data untuk perolehan, pengelolaan, analisa, dan menampilkan berbagai bentuk informasi berreferensi geografis. Sistem informasi geografis menurunkan berbagai informasi dari dunia nyata di muka bumi. Dalam perkembangannya saat ini, Sistem informasi geografis menempatkan basis data informasi pada suatu ruang waktu geografis untuk mempertajam analisis dan pengambilan keputusan. Sistem informasi geografis lebih lanjut banyak diaplikasikan secara *online* baik sebagai dasar pengambilan keputusan ataupun sebagai media penyebaran informasi kepada masyarakat. Teknologi sistem informasi geografis digunakan untuk mengatur dan memanfaatkan data geografis, dan secara luas dikenal sebagai alat dalam berbagai bidang seperti pengelolaan kehutanan, perencanaan perkotaan, teknik sipil, pengelolaan permukiman, bisnis, dan studi lingkungan hidup. Beberapa contoh aplikasi dalam sistem informasi geografis adalah sebagai berikut [13] :

a. Pemetaan Lokasi

Fungsi ini adalah fungsi yang paling mendasar dan umum. Fungsi ini diwujudkan sebagai suatu proses pemetaan digital yang ada dalam SIG. Secara umum dikenal berbagai teknik digitasi dan pemrosesan data dasar lainnya hingga proses *layouting*.

b. Pemetaan informasi kuantitas

Fungsi ini mengaplikasikan data-data kuantitatif untuk memunculkan informasi tematik tertentu, seperti informasi tertinggi, terluas, terbesar atau sebaliknya. Fungsi ini juga mencakup tentang keterkaitan data dengan data yang lainnya dalam konteks ruang.

c. Pemetaan kepadatan

Fungsi ini menekankan pada model konsentrasi sebaran data. Informasi ditampilkan dalam bentuk diagram, warna ataupun tekstur pada peta poligon, garis, ataupun titik.

d. Pencarian informasi data spasial

Fungsi ini memberikan kemudahan dalam pencarian data yang memenuhi kriteria tertentu, baik terhadap informasi yang ada dalam satu data spasial ataupun lebih. Proses dalam fungsi ini banyak dikembangkan sebagai model analisis spasial dalam SIG.

e. Monitoring Perubahan

Data temporal dapat digunakan dalam analisis trend atau perubahan antar waktu. Analisis ini didasarkan pada perubahan kuantitas data ataupun perubahan spasialnya.

#### 2.2.4 Emulator Android

*Emulator android* merupakan sebuah perangkat lunak yang mendukung komputer, baik laptop maupun perangkat komputer lainnya untuk dapat tetap mengakses *android* dalam penggunaan komputer dengan cara mensimulasikan *android* ke perangkat komputer. Dalam memilih *emulator android* seringkali timbul permasalahan untuk menggunakan *emulator* mana yang terbaik dan sesuai dengan spesifikasi komputer yang dimiliki, hingga untuk pengkondisian ini, harus dilakukan percobaan *emulator* satu-persatu pada upaya memastikan *emulator* mana yang berkualitas dan maksimal dalam mensimulasikan *android* pada komputer. Sistem Pendukung Keputusan dapat menjadi sebuah pendekatan yang baik dalam pemilihan *emulator android*, dengan beberapa kriteria dalam menentukan *emulator android* terbaik, maka digunakan metode VIKOR karena memiliki konsep dasar multikriteria. Metode VIKOR menjadi pendekatan dalam melakukan perengkingan pada alternatif untuk mendapatkan solusi ideal dan terbaik dengan kriteria-kriteria yang telah dimiliki dan ditetapkan. Dalam pengujian dengan mengelola data, didapatkan hasil bahwa *emulator genymotion* memiliki hasil akhir yaitu 0 dan menjadi *emulator* terbaik. Dengan penerapan metode VIKOR, maka para pengguna android atau para pencari *emulator*, yaitu para komunitas pecinta *games* atau *gamer*, dan para pecinta android akan terbantu untuk menentukan *emulator android* yang ideal sesuai dengan spesifikasi dan kualitas *emulator* terbaik [14].

#### 2.2.5 Android

Pada saat ini hampir semua orang di dunia menggunakan *system operasi mobile* ini. Berbagai macam gadget menggunakan android sebagai *system*

operasinya. Android menawarkan pendekatan yang menyeluruh pada pengembangan aplikasi, dimana satu aplikasi android yang dibangun dapat berjalan di berbagai perangkat yang menggunakan *system operasi android*, baik yang digunakan *smartphone*, *smartwatch*, *tablet*, dan perangkat lainnya. Dalam bahasa inggris istilah android diartikan “Robot yang menyerupai manusia”, dapat dilihat dari *icon* android yang melambangkan robot berwarna hijau yang memiliki sepasang kaki dan sepasang tangan. Sebagai *system operasi*, android mempunyai fungsi sebagai penghubung antara perangkat keras dengan pengguna (*user*). Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh dari telepon pintar dan komputer *tablet* daya pikat android sebagai *system operasi* yang sangat popular saat ini adalah terletak pada *platform opensource* yang membuka peluang besar pada pengembang aplikasi untuk pengembangan teknologi terbaru [15].

### **2.2.6 Android Studio**

*Android studio* ini adalah lingkungan pengembangan baru dan terintegrasi dengan penuh, yang telah di rilis oleh google untuk sistem operasi *Android* dan dirancang untuk menjadi peralatan baru dalam pengembangan aplikasi dan memberi alternaif selain *Eclips* yang saat ini menjadi IDE (*Integrated Development Environment*) yang banyak di pakai. atau dalam artian lain adalah sebuah lingkungan pengembangan terintegrasi resmi yang memang di rancang khusus untuk pengembangan sistem operasi android [16].

### **2.2.7 Unified Modelling Language (UML)**

*Unified Modeling Language (UML)* adalah suatu teknik untuk memodelkan sistem. UML ditemukan oleh Grady Booch, Ivar Jacobson, dan James Rumbaugh.

*UML* versi terbaru, yaitu versi 2.5, terdiri dari lima belas diagram. Diagram-diagram tersebut dibagi ke dalam dua kelompok, yaitu *structure diagram* dan *behavior diagram*. *Structure diagram* menggambarkan data dan hubungan statis dalam suatu sistem informasi. *Structure diagram* terdiri dari *package*, *object*, *component*, *class*, *deployment*, *composite structure*, dan *profile diagram*. *Behavior diagram* menggambarkan hubungan dinamis diantara objek yang mewakili sistem informasi bisnis. *Behavior diagram* terdiri dari *sequence*, *timing*, *interaction overview*, *activity*, *use case*, *protocol state machine*, *communication* dan *behavior state machine diagram* [17].

### 1) *Use case Diagram*

*Use case* digunakan untuk menggambarkan fungsi dasar dari sebuah sistem informasi. *Use case* mendeskripsikan cara sistem bisnis berinteraksi dengan lingkungannya [18]. Berikut adalah elemen-elemen dari *use case diagram*:

#### a) *Actor/Role*



Gambar 2.1 *Actor/Role*

*Actor/role* adalah orang atau sistem lain yang berinteraksi dengan sistem saat ini.

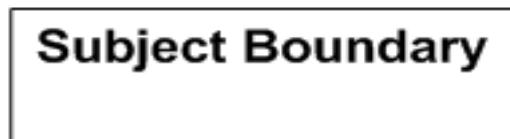
b) *Use Case*



Gambar 2.2 *Use Case*

*Use case* adalah bagian utama dari fungsionalitas sistem. Bisa *extend* (memperluas) *use case* lainnya. Ditempatkan di dalam *system boundary* (batasan sistem). Dilabeli dengan kata kerja – frase kata benda.

c) *Subject Boundary*



Gambar 2.3 *Subject Boundary*

Berisi nama dari sistem yang diletakkan di dalam atau di bagian atas *boundary*. Mewakili ruang lingkup sistem. *Actor* berada di luar ruang lingkup sistem.

d) *Association Relationship*



Gambar 2.4 *Association Relationship*

Menghubungkan *actor* dengan *use case*. Menunjukkan komunikasi dua arah (Menunjukkan komunikasi satu arah jika menggunakan tanda panah). Tanda \* untuk keragaman dari asosiasi (*multiplicity of the association*). Namun umumnya hanya digambarkan garis saja.

e) *Include Relationship*



Gambar 2.5 *Include Relationship*

Memasukkan satu *use case* dalam *use case* lainnya. Perilaku (*behavior*) yang harus terpenuhi agar sebuah *event* dapat terjadi, di mana kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya. Tanda panah mengarah dari *base use case* (pusat) menuju ke *use case* yang di-*include*.

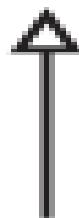
f) *Extend Relationship*



Gambar 2.6 *Include Relationship*

Memperluas *use case* untuk memasukkan perilaku opsional. Tanda panah mengarah dari *use case* tambahan ke *base use case* (pusat).

g) *Generalization Relationship*



Gambar 2.7 *Generalization Relationship*

Mewakili *use case* khusus untuk *use case* yang lebih umum. Tanda panah mengarah dari *use case* khusus (*specialized*) ke *use case* yang lebih umum.

h) *Activity Diagram*

*Activity diagram* mengilustrasikan kegiatan utama dan hubungan di antara kegiatan dalam suatu proses [18]. Berikut adalah elemen-elemen dari *activity* diagram:

1. *Action*



Gambar 2.8 *Action*

Perilaku yang sederhana dan tidak dapat diuraikan. Dilabeli dengan namanya.

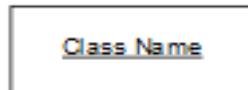
2. *Activity*



Gambar 2.9 *Activity*

Digunakan untuk mewakili sekumpulan tindakan (*action*). Dilabeli dengan Namanya.

3. *Object Node*



Gambar 2.10 *Object Node*

Digunakan untuk mewakili sebuah objek yang terhubung ke sekumpulan arus objek. Dilabeli dengan nama *classnya*.

4. *Control Flow*



Gambar 2.11 *Control Flow*

Menunjukkan urutan eksekusi.

5. *Object Flow*



Gambar 2.12 *Object Flow*

Menunjukkan aliran suatu objek dari satu aktivitas (atau tindakan) ke aktivitas (atau tindakan) lain.

6. *Initial*



Gambar 2.13 *Symbol Initail*

Menggambarkan awal dari serangkaian tindakan atau kegiatan.

7. *Initial Node*



Gambar 2.14 *Symbol Initial Node*

Digunakan untuk menghentikan semua arus kontrol dan arus objek dalam suatu aktivitas (atau tindakan).

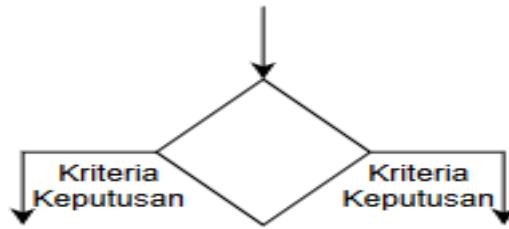
8. *Final-flow Node*



Gambar 2.15 *Final-flow Node*

Digunakan untuk menghentikan aliran kontrol atau aliran objek tertentu.

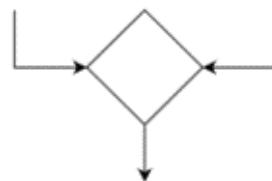
9. *Decision Node*



Gambar 2.16 *Decision Node*

Digunakan untuk mewakili kondisi pengujian untuk memastikan bahwa aliran kontrol atau aliran objek hanya turun satu jalur. Dilabeli dengan kriteria keputusan untuk melanjutkan ke jalur tertentu.

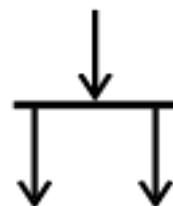
#### 10. *Merge Node*



Gambar 2.17 *Merge Node*

Digunakan untuk menyatukan kembali berbagai jalur keputusan yang dibuat menggunakan simpul keputusan.

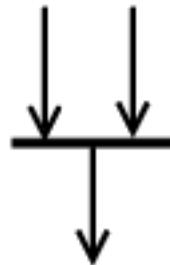
#### 11. *Fork Node*



Gambar 2.18 *Fork Node*

Digunakan untuk membagi perilaku menjadi seperangkat aktivitas yang paralel atau bersamaan dari aktivitas (atau tindakan).

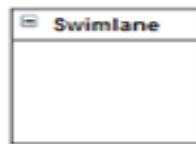
### 12. Join Node



Gambar 2.19 *Join Node*

Digunakan untuk menyatukan kembali serangkaian arus aktivitas (atau tindakan) yang paralel atau bersamaan.

### 13. Swimlane



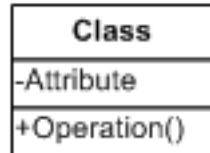
Gambar 2.20 *Symbol Swimlane*

Digunakan untuk memecah diagram aktivitas menjadi baris dan kolom untuk menetapkan kegiatan individu (atau tindakan) kepada individu atau objek yang bertanggung jawab untuk melaksanakan aktivitas (atau tindakan). Dilabeli dengan nama individu atau objek yang bertanggung jawab.

### 14. Class Diagram

*Class diagram* adalah model statis yang menunjukkan kelas dan hubungan di antara kelas yang tetap konstan dalam sistem dari waktu ke waktu. *Class diagram* menggambarkan kelas, yang meliputi perilaku dan keadaan, dengan hubungan antar kelas [18]. Berikut adalah elemen-elemen dari class diagram:

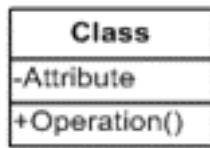
a. *Class*



Gambar 2.21 *Symbol Class*

Mewakili orang, tempat, atau hal-hal yang dibutuhkan sistem untuk menangkap dan menyimpan informasi. Memiliki nama yang diketik dengan huruf tebal dan berpusat di bagian atas kotak. Memiliki daftar atribut di kotak tengahnya. Memiliki daftar operasi di kotak bawahnya. Tidak secara eksplisit menunjukkan operasi yang tersedia untuk semua kelas.

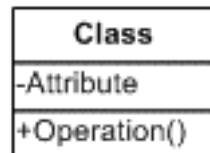
b. *Attribute*



Gambar 2.22 *Attribute*

Merupakan properti yang menggambarkan keadaan suatu objek. Dapat diturunkan dari atribut lain, ditampilkan dengan menempatkan garis miring sebelum nama atribut.

c. *Operation*



Gambar 2.23 *Operation*

Mewakili tindakan atau fungsi yang dapat dilakukan oleh kelas. Dapat diklasifikasikan sebagai konstruktur, permintaan, atau operasi pembaruan.

Termasuk tanda kurung yang mungkin berisi parameter atau informasi yang diperlukan untuk melakukan operasi.

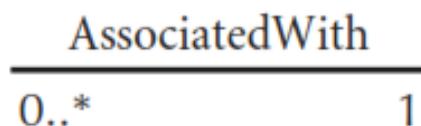
d. *Generalization*



Gambar 2.24 *Generalizaion*

Mewakili hubungan sejenis di antara beberapa kelas.

e. *Association*

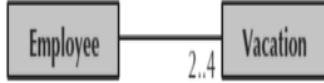


Gambar 2.25 *Association*

Mewakili hubungan antara beberapa kelas atau kelas dan dirinya sendiri.

Dilabeli menggunakan frasa kata kerja atau nama peran, yang lebih baik mewakili hubungan. Dapat ada di antara satu atau beberapa kelas. Berisi simbol multiplisitas, yang mewakili waktu minimum dan maksimum *instance* kelas dapat dikaitkan dengan *instance* kelas terkait. Tujuan utama dari *class* diagram adalah untuk menunjukkan relasi, atau asosiasi, yang *class* miliki dengan *class* lainnya. Relasi juga memiliki *multiplicity*, yang mendokumentasikan bagaimana *class* bisa diasosiasikan dengan *class* lainnya. Format *multiplicity* Angka minimal : Angka Maksimal. Penjelasan lebih detail mengenai *multiplicity* bisa dilihat pada table dibawah ini.

Tabel 2.2 *Multiplicity*

Hanya satu	1		Dalam satu departemen hanya memiliki satu bos
Satu atau lebih	1..*		Satu bos bertanggungjawab untuk satu atau lebih karyawan
Nol atau satu	0..1		Seorang karyawan bisa saja belum/tidak menikah (memiliki 0 pasangan) atau menikah (memiliki satu pasangan)
Range yang spesifik	2..4		Seorang karyawan bisa mengambil libur (cuti) mulai dari 2 sampai 4 hari setiap tahunnya
Multiple	1..3,5		Seorang karyawan adalah anggota dari satu sampai 3 atau 5 komite

Untuk menentukan kardinalitas (derajat relasi) antar dua *entitas/class*, cukup ambil angka maksimal dari tiap *multiplicity*. Penjelasannya bisa dilihat pada gambar dibawah ini.

Gambar 2.26 Tahap *multiplicity*

kardinalitas dari gambar di atas adalah 1..M. Membaca dari kiri ke kanan: menunjukkan sa *user* menginput minimal 1, maksimal banyak transaksi. Membaca

dari kanan ke kiri: 1 Transaksi hanya bisa diinput oleh 1 orang. Jika ada relasi 1..M (di sebelah kiri) dan 1..M (di sebelah kanan), maka kardinalitasnya M..N (*many to many*).

#### f. *Aggregation*



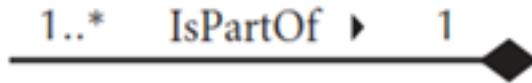
Gambar 2.27 *Symbol Aggregation*

Merupakan hubungan sebagian (*a part of*) yang logis diantara beberapa kelas atau suatu kelas dengan kelas itu sendiri. *Aggregation* terbuat dari objek-objek yang bisa dibagi atau ditukar. Angka di sebelah kanan boleh 1, boleh juga banyak (\*). Contoh dari *aggregation* bisa dilihat pada gambar di bawah ini.



Gambar 2.28 Objek *aggregation*

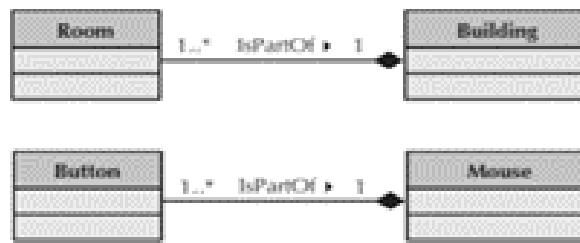
g. *Composition*



Gambar 2.29 *Symbol Objek Composition*

Mewakili hubungan fisik antara beberapa kelas atau kelas itu sendiri.

*Composition* terdiri dari objek-objek yang tidak bisa dibagi atau ditukar dan hidup selama *composite* objek hidup. Angka di sebelah kanan hanya boleh. Contoh dari *composition* bisa dilihat pada gambar di bawah ini.



Gambar 2.30 Objek Composite

h. *Sequence Diagram*

*Sequence* diagram menunjukkan pesan yang lewat di antara objek untuk *use case* tertentu dari waktu ke waktu. *Sequence* diagram mengilustrasikan objek-objek yang berpartisipasi di dalam suatu *use case*. Berikut adalah elemen-elemen dari *sequence* diagram:

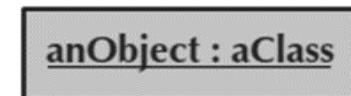
a) *Actor*



Gambar 2.31 *Symbol Actor*

Adalah orang atau sistem yang memperoleh manfaat dari dan berada di luar sistem. Berpartisipasi dalam suatu urutan dengan mengirim dan / atau menerima pesan. Ditempatkan di bagian atas diagram.

b) *Object*



Gambar 2.32 *Symbol Object*

Berpartisipasi dalam suatu urutan dengan mengirim dan / atau menerima pesan. Ditempatkan di bagian atas diagram. Adapun penggambaran lain dari *object* bisa dilihat seperti di bawah ini:



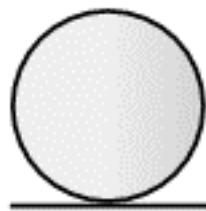
Gambar 2.33 *Boundary Class*

*Boundary Class*: Menggambarkan tampilan program.



Gambar 2.34 *Control Class*

*Control Class*: Menggambarkan *controller*.



Gambar 2.35 *Entity Class*

*Entity Class:* Menggambarkan *class*.

c) *Lifeline*



Gambar 2.36 *Symbol Lifeline*

Menunjukkan kehidupan suatu objek selama suatu urutan. Berisi X pada titik di mana kelas tidak lagi berinteraksi.

d) *Execution Occurrence* (Kejadian Eksekusi)

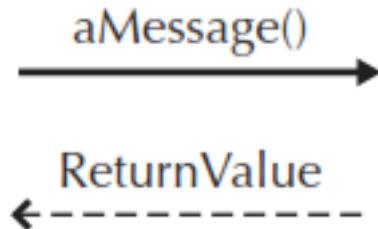


Gambar 2.37 *Execution Occurrence*

Merupakan persegi panjang sempit panjang yang ditempatkan di atas *lifeline*.

Menunjukkan kapan suatu objek mengirim atau menerima pesan.

e) *Message*



Gambar 2.38 *Symbol Message*

Menyampaikan informasi dari satu objek ke objek lainnya. Pemanggilan operasi diberi label dengan pesan yang dikirim dan panah padat, sedangkan pengembalian diberi label dengan nilai yang dikembalikan dan ditampilkan sebagai tanda panah putus-putus.

f) *Guard Condition*



Gambar 2.39 *Guard Condition*

Merupakan tes yang harus dipenuhi untuk pesan yang akan dikirim.

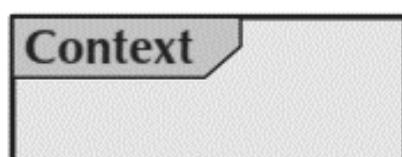
g) *Object Destruction*



Gambar 2.40 *Object Destruction*

X ditempatkan di ujung *lifeline* objek untuk menunjukkan bahwa objek tersebut akan keluar dari eksistensi.

h) *Frame*

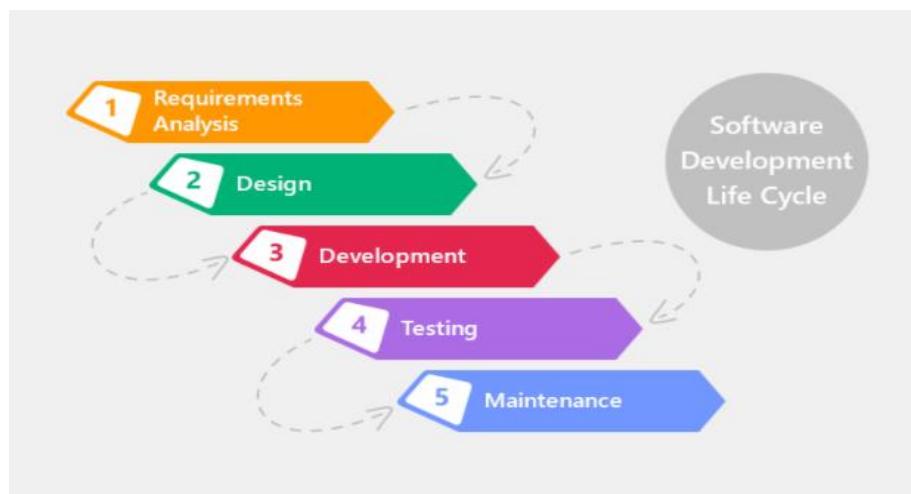


Gambar 2.41 *Symbol Frame*

Menunjukkan konteks *sequence* diagram.

### 2.2.8 Waterfall

*Waterfall* adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear. Jadi jika langkah satu belum dikerjakan maka tidak akan bisa melakukan pengerjaan langkah 2, 3 dan seterusnya. Secara otomatis tahapan ke-3 akan bisa dilakukan jika tahap ke-1 dan ke-2 sudah dilakukan. Secara garis besar metode *waterfall* mempunyai langkah-langkah sebagai berikut : Analisa, *Design*, *Code* dan *Testing*, Penerapan dan Pemeliharaan [19].



Gambar 2.42 Tahapan Metode *Waterfall*

Berikut Penjelasan Tahapan Metode *Waterfall* :

1. Analisis

Langkah ini merupakan analisa terhadap kebutuhan *system*. Pengumpulan data dalam tahap ini bisa malakukan sebuah penelitian, wawancara atau *study literature*. Tahapan ini akan menghasilkan dokumen *user requirment* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan sistem. Dokumen ini lah yang akan menjadi acuan sistem analis untuk menterjemahkan ke dalam Bahasa pemrograman.

## 2. *Design*

Proses desain akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada : struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*. Dokumen inilah yang akan digunakan programmer untuk melakukan aktivitas pembuatan sistemnya.

## 3. *Coding & Testing*

Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, *design* dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*.

## 4. Penerapan

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (periperal atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

### **2.2.9 Software Development Kit (SDK)**

Adalah kumpulan alat pengembangan perangkat lunak dalam satu paket yang dapat diinstal. Mereka memfasilitasi pembuatan aplikasi dengan memiliki kompiler, *debugger*, dan terkadang kerangka perangkat lunak . Mereka biasanya spesifik untuk platform perangkat keras dan kombinasi sistem operasi. Untuk membuat aplikasi dengan fungsionalitas tingkat lanjut seperti iklan, pemberitahuan *push*, dll; sebagian besar pengembang perangkat lunak aplikasi menggunakan perangkat pengembangan perangkat lunak tertentu. Beberapa *SDK* diperlukan

untuk mengembangkan aplikasi khusus platform. Misalnya, pengembangan aplikasi Android pada *platform Java* memerlukan Java *Development Kit*. Untuk aplikasi iOS (aplikasi) ini diperlukan. Untuk *Platform Windows Universal*, *.NET Framework SDK* mungkin digunakan. Ada juga *SDK* yang menambahkan fitur tambahan dan dapat dipasang di aplikasi untuk menyediakan analisis, data tentang aktivitas aplikasi, dan opsi monetisasi. Beberapa pencipta terkemuka dari jenis *SDK* ini termasuk Google, Smaato, InMobi, dan Facebook [20].

#### **2.2.10 Database**

Pengertian *database* adalah sekumpulan data yang dikelola berdasarkan ketentuan tertentu yang saling berkaitan sehingga memudahkan dalam pengelolaannya. Dihimpun dari berbagai sumber, secara sederhana, *database* atau basis data merupakan sekumpulan data atau informasi yang tersimpan secara sistematis. *Database* memiliki peran penting dalam perangkat untuk mengumpulkan informasi, data, atau *file* secara terintegrasi. *Database* berwujud tabel yang terdiri dari kolom dan baris yang memuat atribut dan nilai tertentu. Adapun jumlah kolom dan baris dalam suatu *database* tergantung pada jumlah kategori atau jenis informasi yang perlu disimpan [21]. Dan berikut enam fungsi *Database* :

- 1) Mengelompokan data dan informasi.
- 2) Memudahkan dalam identifikasi data.
- 3) Memudahkan proses akses, menyimpan, pembaharuan, dan penghapusan
- 4) Menjadi alternatif terkait masalah penyimpanan ruang dalam suatu aplikasi.
- 5) Menjaga kualitas data yang diakses sesuai input.
- 6) Menunjang kinerja aplikasi yang memerlukan penyimpanan data.

### 2.2.11 Google Maps

Google *Maps* adalah layanan pemetaan web yang dikembangkan oleh Google. Layanan ini memberikan citra satelit, peta jalan, panorama 360°, kondisi lalu lintas, dan perencanaan *rute* untuk berpergian dengan berjalan kaki, mobil, sepeda (versi beta), atau angkutan umum. Google *Maps* dimulai sebagai program desktop C++, dirancang oleh Lars dan Jens Eilstrup Rasmussen pada *Where 2 Technologies*. Pada Oktober 2004, perusahaan ini diakuisisi oleh Google, yang diubah menjadi sebuah aplikasi web. Setelah akuisisi tambahan dari perusahaan visualisasi data geospasial dan analisis lalu lintas, Google *Maps* diluncurkan pada Februari 2005. Layanan ini menggunakan JavaScript, XML, dan AJAX. Google *Maps* menawarkan API yang memungkinkan peta untuk dimasukkan pada situs *web* pihak ketiga, dan menawarkan penunjuk lokasi untuk bisnis perkotaan dan organisasi lainnya di berbagai negara di seluruh dunia. Google *Map Maker* memungkinkan pengguna untuk bersama-sama mengembangkan dan memperbarui pemetaan layanan di seluruh dunia. Tampilan satelit Google *Maps* adalah "*top-down*". Sebagian besar citra resolusi tinggi dari kota adalah foto udara yang diambil dari pesawat pada ketinggian 800 sampai 1.500 kaki (240–460 meter), sementara sebagian besar citra lainnya adalah dari satelit. Sebagian besar citra satelit yang tersedia adalah tidak lebih dari tiga berusia tahun dan diperbarui secara teratur. Google *Maps* menggunakan varian dekat dari proyeksi *Mercator*, dan karena itu Google *Maps* tidak dapat secara akurat menunjukkan daerah di sekitar kutub. Google *Maps* untuk seluler dirilis pada bulan September 2008. Pada Agustus 2013, Google *Maps* bertekad untuk menjadi aplikasi yang paling populer di dunia untuk ponsel cerdas, dengan lebih dari 54% dari pemilik ponsel cerdas di seluruh dunia [22].

### **2.2.12 Satuan Lalu Lintas (Satlantas)**

Satuan Lalu Lintas yang selanjutnya disingkat Satlantas adalah unsur pelaksana tugas pokok fungsi lalu lintas pada tingkat Polres yang berada di bawah Kapolres. Satlantas bertugas melaksanakan Turjawali lalu lintas, pendidikan masyarakat lalu lintas (Dikmaslantas), pelayanan registrasi dan identifikasi kendaraan bermotor dan pengemudi, penyidikan kecelakaan lalu lintas dan penegakan hukum di bidang lalu lintas [23]. Dalam melaksanakan tugas, Satlantas menyelenggarakan fungsi:

1. pembinaan partisipasi masyarakat melalui kerja sama lintas sektoral, Dikmaslantas, dan pengkajian masalah di bidang lalu lintas.
2. pelaksanaan operasi kepolisian bidang lalu lintas dalam rangka penegakan hukum dan keamanan, keselamatan, ketertiban, kelancaran lalu lintas (Kamseltibcarlantas).
3. pelayanan administrasi registrasi dan identifikasi kendaraan bermotor serta pengemudi.
4. pelaksanaan patroli jalan raya dan penindakan pelanggaran serta penanganan kecelakaan lalu lintas dalam rangka penegakan hukum, serta menjamin Kamseltibcarlantas di jalan raya.
5. pengamanan dan penyelamatan masyarakat pengguna jalan dan perawatan dan pemeliharaan peralatan dan kendaraan.

### **2.2.13 Kepolisian Resor (POLRES)**

Polres merupakan satuan organisasi Polri yang berkedudukan di ibukota kabupaten/kota di daerah hukum masing-masing. Polres bertugas menyelenggarakan tugas pokok Polri dalam memelihara keamanan dan ketertiban

masyarakat, menegakkan hukum, serta memberikan perlindungan, pengayoman, dan pelayanan kepada masyarakat dan melaksanakan tugas-tugas Polri lainnya dalam daerah hukum Polres, sesuai dengan ketentuan peraturan perundang-undangan. Polres menyelenggarakan fungsi [24]:

- 1) pemberian pelayanan kepolisian kepada masyarakat, dalam bentuk penerimaan dan penanganan laporan/pengaduan, pemberian bantuan dan pertolongan termasuk pengamanan kegiatan masyarakat dan instansi pemerintah, dan pelayanan surat izin/keterangan, serta pelayanan pengaduan atas tindakan anggota Polri sesuai dengan ketentuan peraturan perundang-undangan.
- 2) pelaksanaan fungsi intelijen dalam bidang keamanan guna terselenggaranya deteksi dini (*early detection*) dan peringatan dini (*early warning*).
- 3) penyelidikan dan penyidikan tindak pidana, fungsi identifikasi dan fungsi laboratorium forensik lapangan dalam rangka penegakan hukum, serta pembinaan, koordinasi, dan pengawasan Penyidik Pegawai Negeri Sipil (PPNS).
- 4) pembinaan masyarakat, yang meliputi pemberdayaan masyarakat melalui perpolisian masyarakat, pembinaan dan pengembangan bentuk-bentuk pengamanan swakarsa dalam rangka peningkatan kesadaran dan ketaatan warga masyarakat terhadap hukum dan ketentuan peraturan perundangundangan, terjalinnya hubungan antara Polri dengan masyarakat, koordinasi dan pengawasan kepolisian khusus.
- 5) pelaksanaan fungsi Sabhara, meliputi kegiatan pengaturan, penjagaan pengawalan, patroli (Turjawali) serta pengamanan kegiatan masyarakat dan pemerintah, termasuk penindakan tindak pidana ringan (Tipiring), pengamanan

unjuk rasa dan pengendalian massa, serta pengamanan objek vital, pariwisata dan *Very Important Person (VIP)*.

- 6) pelaksanaan fungsi lalu lintas, meliputi kegiatan Turjawali lalu lintas, termasuk penindakan pelanggaran dan penyidikan kecelakaan lalu lintas serta registrasi dan identifikasi kendaraan bermotor dalam rangka penegakan hukum dan pembinaan keamanan, keselamatan, ketertiban, dan kelancaran lalu lintas.
- 7) pelaksanaan fungsi kepolisian perairan, meliputi kegiatan patroli perairan, penanganan pertama terhadap tindak pidana perairan, pencarian dan penyelamatan kecelakaan di wilayah perairan, pembinaan masyarakat perairan dalam rangka pencegahan kejahatan, dan pemeliharaan keamanan.

#### **1.2.14 System Usability Scale (SUS)**

*System Usability Scale* adalah alat sederhana terdiri dari sepuluh item yang digunakan untuk memberikan penilaian subjektif terhadap kegunaan suatu sistem. *SUS* dikembangkan sebagai skala *Likert* yang dirancang untuk mengukur kemudahan penggunaan sistem dengan cepat dan efisien dalam konteks industri. Setelah responden menggunakan sistem yang dievaluasi, mereka mencatat respons mereka terhadap setiap item pada skala dari 1 hingga 5. Skor *SUS* dihitung dengan menjumlahkan kontribusi skor dari setiap item, kemudian mengalikannya dengan 2,5 untuk mendapatkan nilai total yang berkisar antara 0 hingga 100. *SUS* telah terbukti sebagai alat yang andal dan efektif dalam mengevaluasi kegunaan berbagai sistem, mendukung perbaikan berkelanjutan dalam desain produk [25].

$$\begin{array}{lcl} \bar{x} & = & \text{skor rata-rata} \\ \sum x & = & \text{jumlah skor SUS} \\ n & = & \text{jumlah responden} \end{array} \quad (2.1 \text{ Rumus Menghitung SUS})$$

Aturan menghitung sus Setelah melakukan pengumpulan data dari responden, kemudian data tersebut dihitung. Dalam cara menggunakan *System Usability Scale* (*SUS*) ada beberapa aturan dalam perhitungan skor *SUS*. Berikut ini aturan-aturan saat perhitungan skor pada kuesionernya:

1. Setiap pertanyaan bermomor ganjil, skor setiap pertanyaan yang didapat dari skor pengguna akan dikurangi 1.
2. Setiap pertanyaan bermomor genap, skor akhir didapat dari nilai 5 dikurangi skor pertanyaan yang didapat dari pengguna.
3. Skor *SUS* didapat dari hasil penjumlahan skor setiap pertanyaan yang kemudian dikali 2,5.

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Tempat Dan Jadwal Penelitian

##### 3.1.1 Tempat Penelitian

Satuan lalu lintas (Satlantas) dipimpin oleh Kasat Lantas yang bertanggung jawab kepada Kapolres dan dalam pelaksanaan tugas sehari-hari di bawah kendali WakaPolres Kasat Lantas bertugas melaksanakan Turjawali lalu lintas, pendidikan masyarakat lalu lintas (Dikmaslantas), pelayanan registrasi dan identifikasi kendaraan bermotor dan pengemudi, penyidikan kecelakaan lalu lintas dan penegakan hukum di bidang lalu lintas. Struktur Organisasi Satuan Lalu Lintas (Satlantas).



Gambar 3.1 Tempat Penelitian

##### 3.1.2 Jadwal Penelitian

Penelitian dilakukan di Kantor Dinas Satlantas Kabupaten Lombok Tengah, Provinsi Nusa Tenggara Barat. Beralamatkan di Jln. Basuki Rahmat No. 12 Praya Kabupaten Lombok Tengah, Provinsi Nusa Tenggara Barat 83518 Indonesia.

Adapun jadwal penelitian yang akan dilakukan seperti yang terlihat pada tabel 3.1 dibawah ini.

Tabel 3.1 Jadwal Penelitian

No	Nama Kegiatan	Bulan											
		Mei			Juni				Juli				
		1	2	3	1	2	3	4	1	2	3	4	
1	Identifikasi Masalah												
2	Pengumpulan Data												
3	Analisis Data												
4	Perancangan												
5	Implementasi												
6	Pengujian												
7	Hasil												

## 3.2 Alat Dan Bahan Penelitian

### 3.2.1 Alat Penelitian

Alat penelitian yang digunakan oleh peneliti mencakup *hardware* dan *software* antara lain sebagai berikut:

#### 1. *Hardware*

*Hardware* yang digunakan pada penelitian ini antara lain :

A. Laptop Thinkpad X270 dan spesifikasinya sebagai berikut :

a. *Device name*: DESKTOP-3GTGPFM

b. *Processor* : Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz 2.50 GHz

c. *Installed RAM* : 8.00 GB (7.86 GB *usable*)

- d. *Device ID* : 683EE6BB-1A7C-4AA1-A566-15E01778113D
  - e. *Product ID* : 00330-50323-39137-AAOEM
  - f. *System type* : 64-bit operating system, x64-based processor
  - g. *Edition* : Windows 10 Pro
  - h. *Version* : 22H2
  - i. OS *build* : 19045.3930
- B. Handphone Xiaomi Redmi 7 dan spesifikasinya sebagai berikut :
- a. Model Perangkat : Redmi 7
  - b. Versi MIUI : MIUI Global 11.0.2 Stabil 11.0.2.0 (QFLMIXM)
  - c. Versi *Android* : 20 QKQ1.191008.001
  - d. RAM : 3,00GB
  - e. Model : M1810F6LG
  - f. CPU : Octa-Core Max. 1.80GHz
  - g. Penyimpanan Internal : 32.00GB
2. *Software*

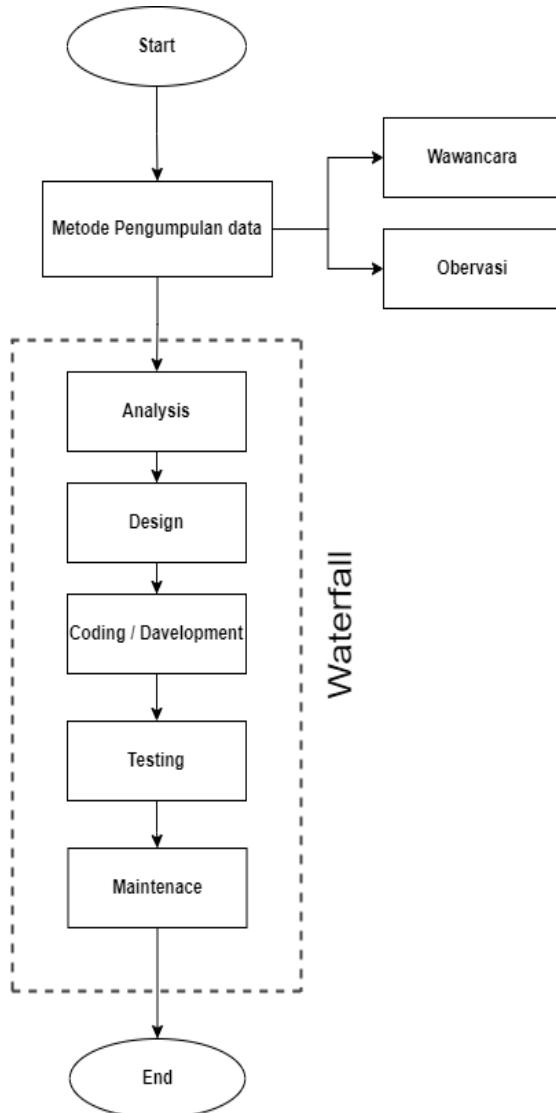
*Software* yang digunakan pada penelitian ini antara lain:

- a. *Android Studio iguana*
- b. *Firebase*
- c. *Google Cloud*
- d. *Maps SDK For Android*

### **3.2.2 Bahan Penelitian**

- 1. Hasil Wawancara
- 2. Hasil Observasi

### 3.3 Alur Penelitian



Gambar 3.2 Tahapan Penelitian

Berikut uraian alur penelitian yang telah dirancang oleh peneliti :

1. Pengumpulan data merupakan kegiatan mencari data dan informasi di lapangan yang akan digunakan untuk menjawab permasalahan penelitian. Pada penelitian ini peneliti mengumpulkan data menggunakan metode observasi dan wawancara.
2. Peneliti melakukan wawancara Kepala Bidang Bidang Satuan Lalu Lintas IPTU Abdul Rachman Virga Maulidhany Yusuf, S.Tr.K., S.I.K, guna untuk

mengetahui Upaya Pihak Kepolisian dalam mencegah terjadinya kecelakaan lalu lintas di Lombok Tengah yang saat ini masih sering terjadi.

3. Peniliti juga melakukan oberservasi pada daerah yang sering terjadi kecelakaan yang ada di Kabupaten Lombok Tengah.
4. Pada tahap ini dilakukan metode pengerjaan sistem secara berurutan dengan metode *Waterfall* yang memiliki 5 tahapan yaitu tahapan *Analisis, design, coding, testing dan maintenance.*

### **3.4 Pengumpulan Data**

#### **3.4.1 Observasi**

Observasi dilakukan secara langsung pada berbagai lokasi yang sering mengalami kecelakaan di Kabupaten Lombok Tengah. termasuk jalan-jalan utama, persimpangan, dan daerah dengan tingkat kecelakaan yang tinggi untuk mendapatkan situasi lalu lintas yang sedang terjadi. Pengamatan meliputi kepatuhan pengendara terhadap rambu lalu lintas, penggunaan alat keselamatan seperti helm dan sabuk pengaman, kondisi fisik jalan termasuk penerangan dan rambu-rambu keselamatan, serta kepadatan arus lalu lintas. Hasil pengamatan peniliti menunjukkan banyak pengendara yang tidak mematuhi aturan lalu lintas dan banyak kondisi serta bahu jalan yang rusak

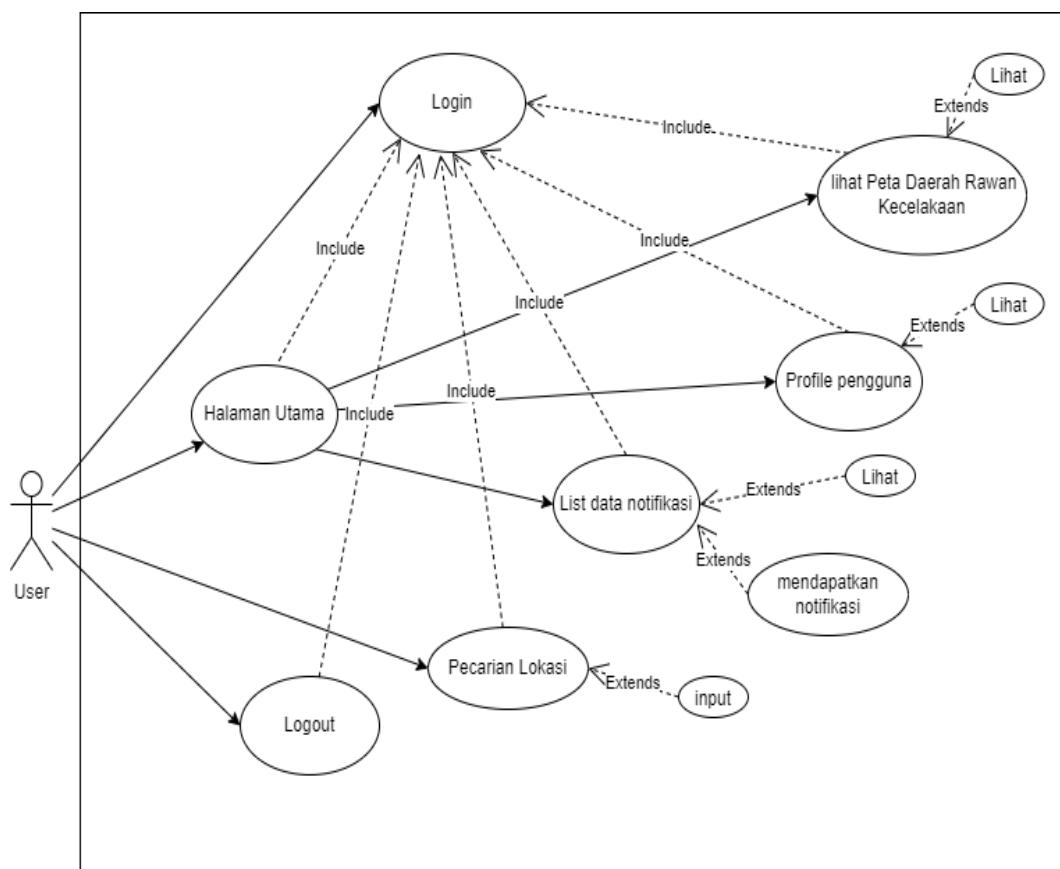
#### **3.4.2 Wawancara**

Wawancara dilakukan dengan Kepala Bidang Satuan Lalu Lintas IPTU Abdul Rachman Virga Maulidhany Yusuf S.Tr.K. S.I.K, sebagai narasumber untuk mengetahui upaya pihak kepolisian dalam mencegah terjadinya kecelakaan lalu lintas di daerah Kabupaten Lombok Tengah. Hasil dari wawancara tersebut adalah informasi upaya kepolisian dalam mengimbau masyarakat seperti penyuluhan,

pemasangan spaduk, pemasangan rambu-rambu lalu lintas di setiap jalan, edukasi kepada masyarakat dan hingga pihak kepolisian terjun langsung ke lokasi daerah yang sering terjadi kecelakaan. Oleh sebab itu perlunya dibuat Aplikasi yang bisa untuk memetakan titik daerah yang sering terjadi rawan kecelakaan serta memberikan peringatan notifikasi melalui *smartphone* kepada pengguna jalan untuk menanggulangi kecelakaan di wilayah Kabupaten Lombok Tengah.

### 3.5 Perancangan Sistem

#### 3.5.1 Use Case Diagram



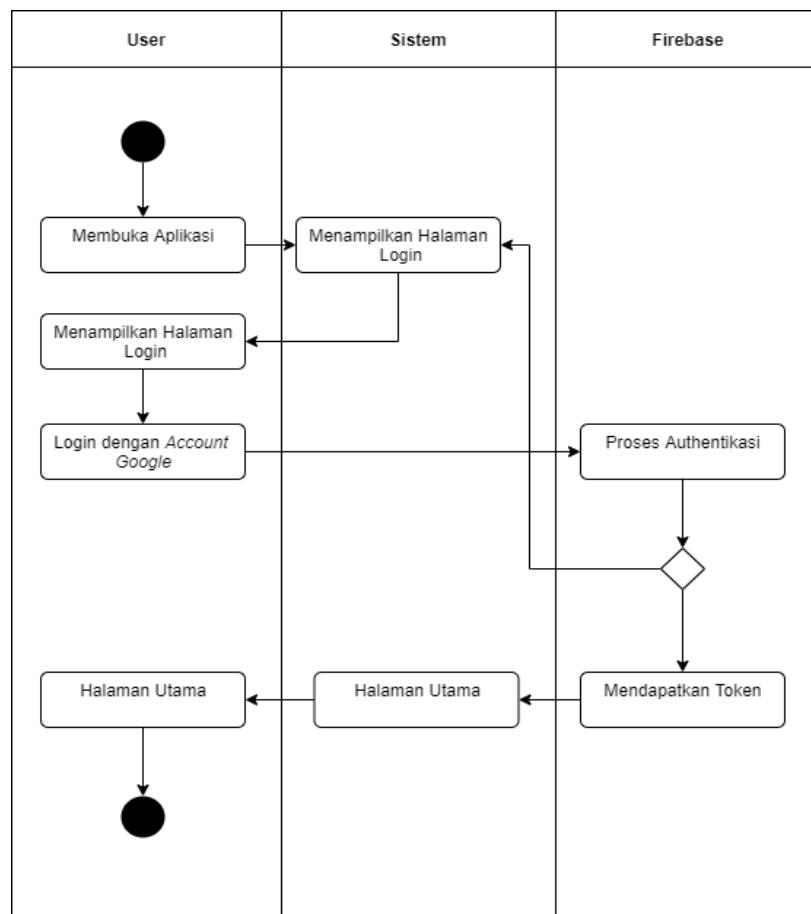
Gambar 3.3 Use Case Diagram

Berikut adalah penjelasan mengenai *use case* diagram yang menggambarkan interaksi *User* dalam sistem informasi geografis (SIG) untuk pemetaan daerah rawan kecelakaan.

- a. *User* yang mengakses Aplikasi untuk melihat informasi tentang daerah rawan kecelakaan.
- b. Bisa melakukan *login*, melihat peta daerah rawan kecelakaan, melihat detail lokasi, mencari lokasi tujuan, mendapatkan notifikasi, melihat notifikasi, melihat *profile* dan keluar dari aplikasi.

### 3.5.2 Activity Diagram

#### 1. Proses *Login User*



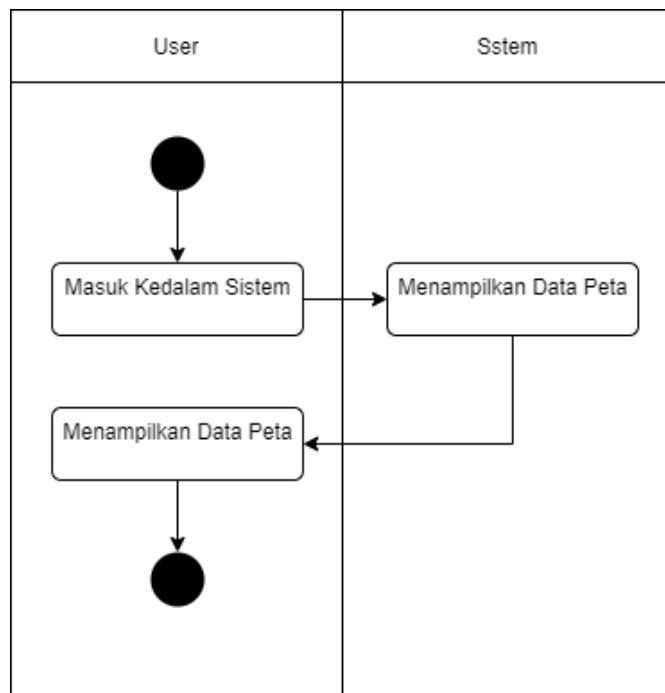
Gambar 3.4 Proses *login user*

Berikut merupakan uraian dari Proses *Login User* :

- 1) *User* membuka aplikasi daerah rawan Kecelakaan Kabupaten Lombok Tengah.
- 2) Sistem menampilkan halaman *login* untuk *user*.

- 3) *User* masuk dengan *account Google* yang tersedia di perangkat.
- 4) Sistem akan melakukan proses authentifikasi dari *account Google* yang dipilih oleh *user*.
- 5) Firebase secara otomatis akan membuat *token* dari *account google* yang diterima.
- 6) Namun jika *token* berhasil dibuat maka sistem akan menampilkan halaman utama dari aplikasi daerah rawan kecelakaan dan jika tidak berhasil maka akan kembali ke halaman *login*.

## 2. Proses melihat peta daerah rawan kecelakaan

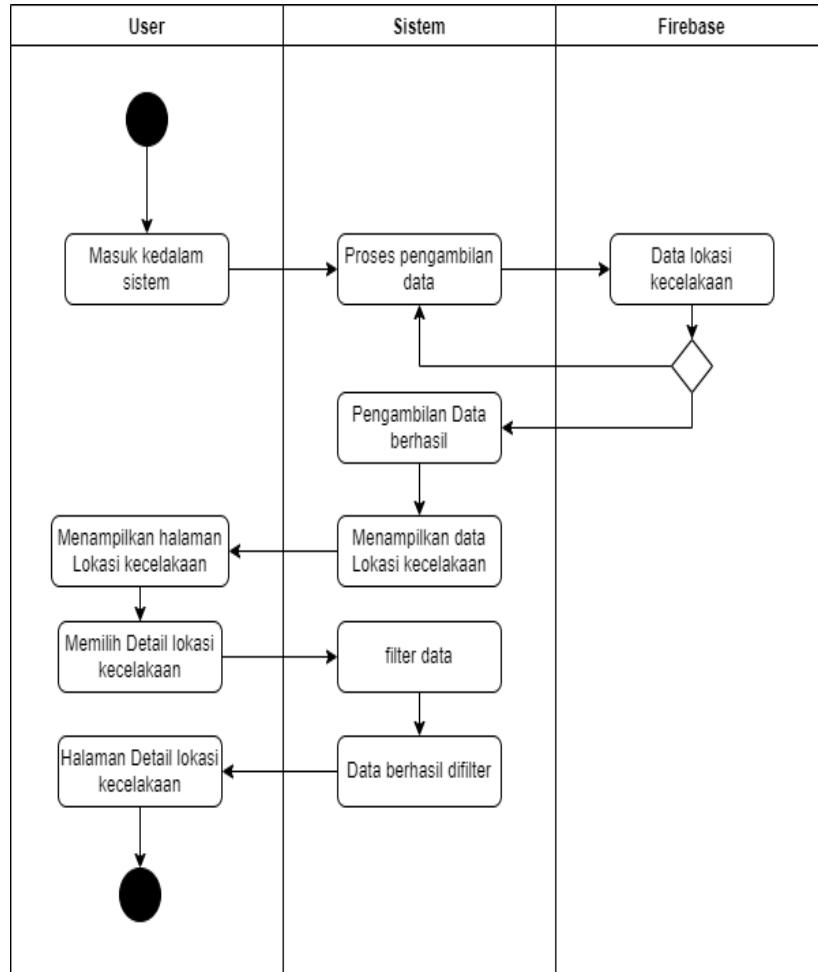


Gambar 3.5 Proses melihat peta

Berikut merupakan uraian dari Proses melihat peta :

- 1) *User* masuk kedalam sistem.
- 2) Sistem akan menampilkan peta Google *maps*.
- 3) Kemudian Peta ditampilkan kepada *user*.

### 3. Proses melihat detail lokasi kecelakaan



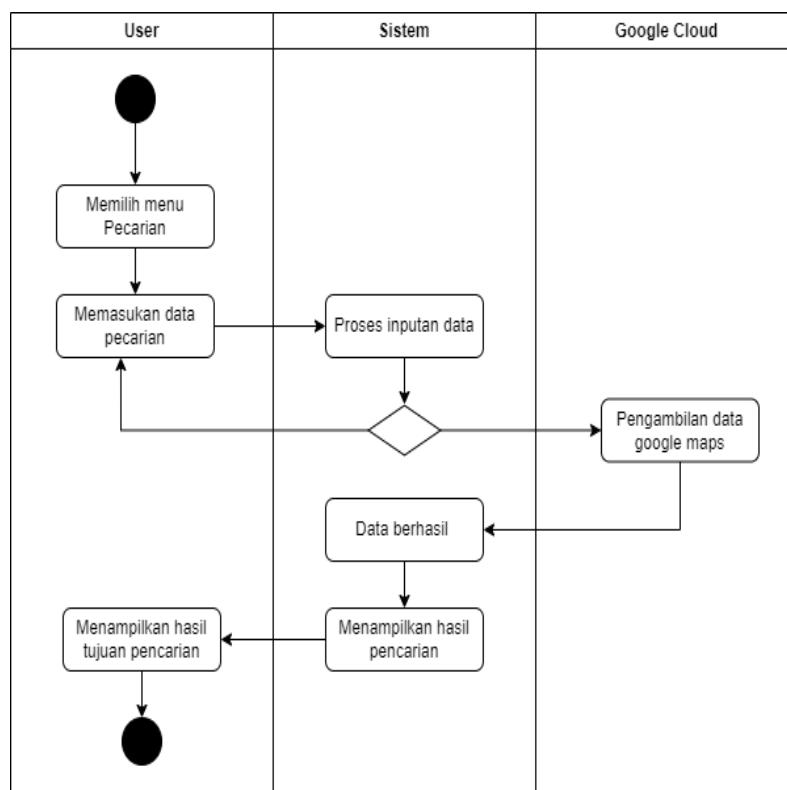
Gambar 3.6 Proses melihat detail lokasi kecelakaan

Berikut merupakan uraian dari Proses melihat detail lokasi kecelakaan.

- 1) *User* masuk kedalam sistem.
- 2) Sistem akan melakukan proses pengambilan data lokasi daerah rawan kecelakaan didalam Firebase dan jika gagal maka akan dilakukan proses ulang dan jika berhasil akan menampilkan halaman data lokasi daerah kecelakan berdasarkan kecamatan.

- 3) Setelah sistem berhasil menampilkan seluruh data rawan kecelakaan. *User* akan memilih salah satu data yang ingin dilihat detail lokasi daerah rawan kecelakaan.
- 4) Setelah diproses oleh Google *cloud* maka akan Kembali diolah oleh *system* kemudian akan menampilkan hasil pencarian alamat tujuan.
- 5) Dan sistem akan menampilkan hasil pencarian alamat tujuan kepada *user* dan menampilkan dengan *rute Google maps*.

#### 4. Proses pencarian lokasi tujuan



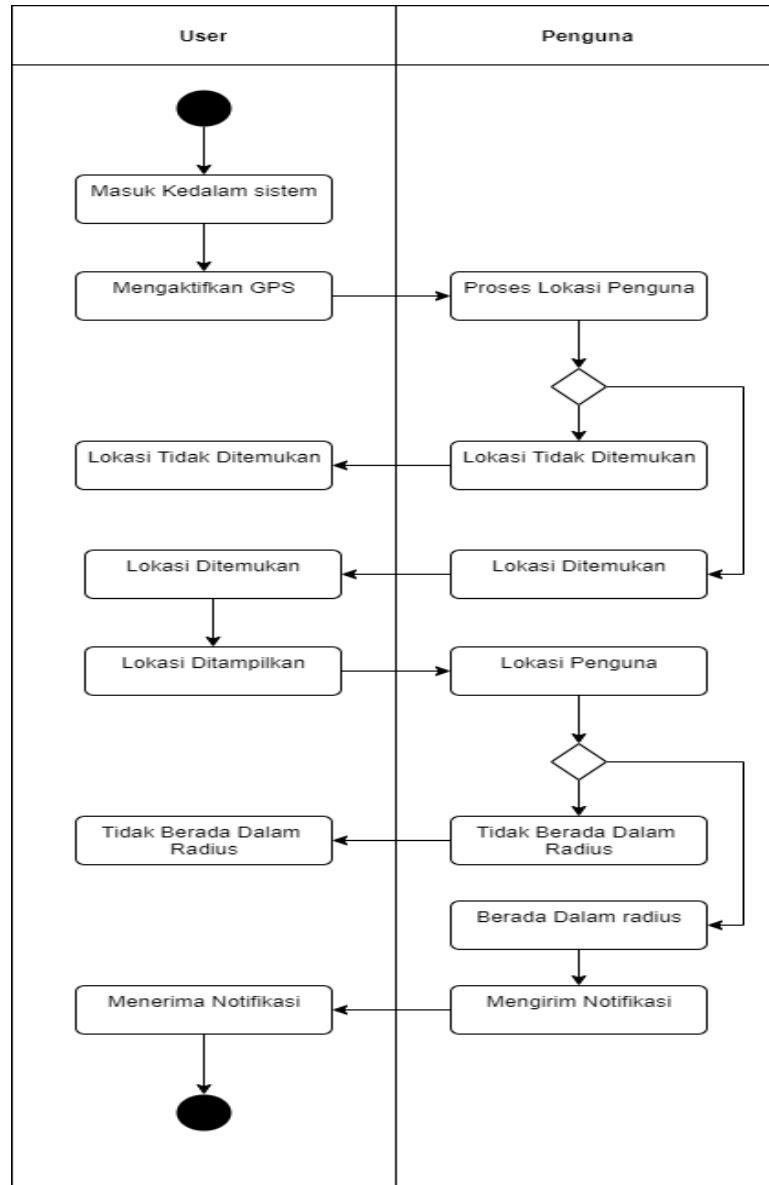
Gambar 3.7 proses pencarian lokasi tujuan

Berikut merupakan uraian dari proses pencarian lokasi tujuan

- 1) *User* masuk kedalam sistem.
- 2) *User* memilih menu atau kolom pecarian yang terdapat di paling atas.
- 3) *User* menginputkan data lokasi yang ingin dituju.

- 4) Kemudian sistem akan mengolah data tersebut kemudian akan diproses ke Google *cloud* sebagai penyedia layanan Google *maps*

## 5. Proses mendapatkan notifikasi



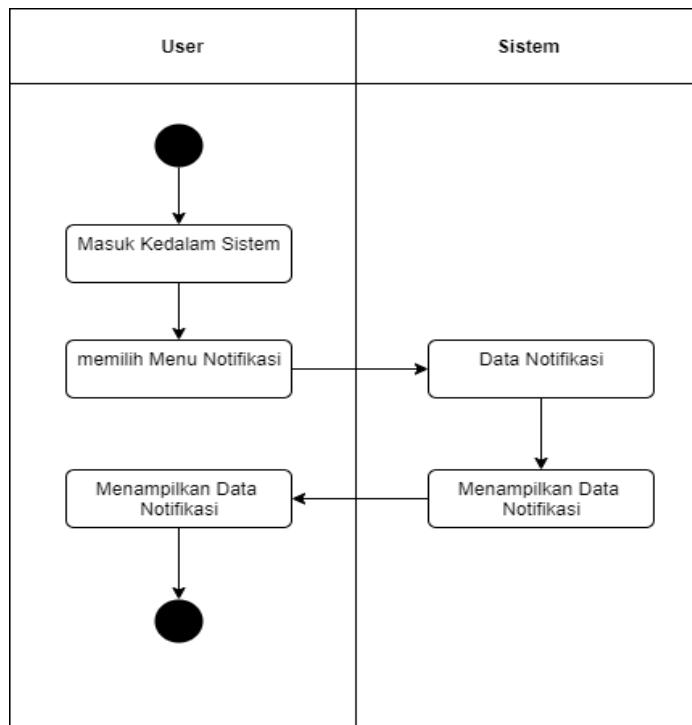
Gambar 3.8 Proses Mendapatkan Notifikasi

Berikut merupakan uraian dari Proses Penguna Mendapatkan Notifikasi :

- 1) *User* masuk kedalam Aplikasi Daerah Rawan Kecelakaan.
- 2) *User* mengaktifkan *GPS* pada perangkat *smartphone* yang digunakan.
- 3) Kemudian sistem akan mendeteksi lokasi *user* saat ini.

- 4) Jika lokasi *user* tidak ditemukan maka lokasi *user* tidak ditampilkan di peta jika sebaliknya lokasi ditemukan akan ditampilkan lokasi *user* yang berada saat ini pada peta Google *maps*.
- 5) Sistem akan mendeteksi lokasi *user* berada , jika lokasi lokasi *user* berada dalam radius daerah rawan kecelakaan sistem akan mengirim notifikasi.

## 6. Proses Melihat Notifikasi

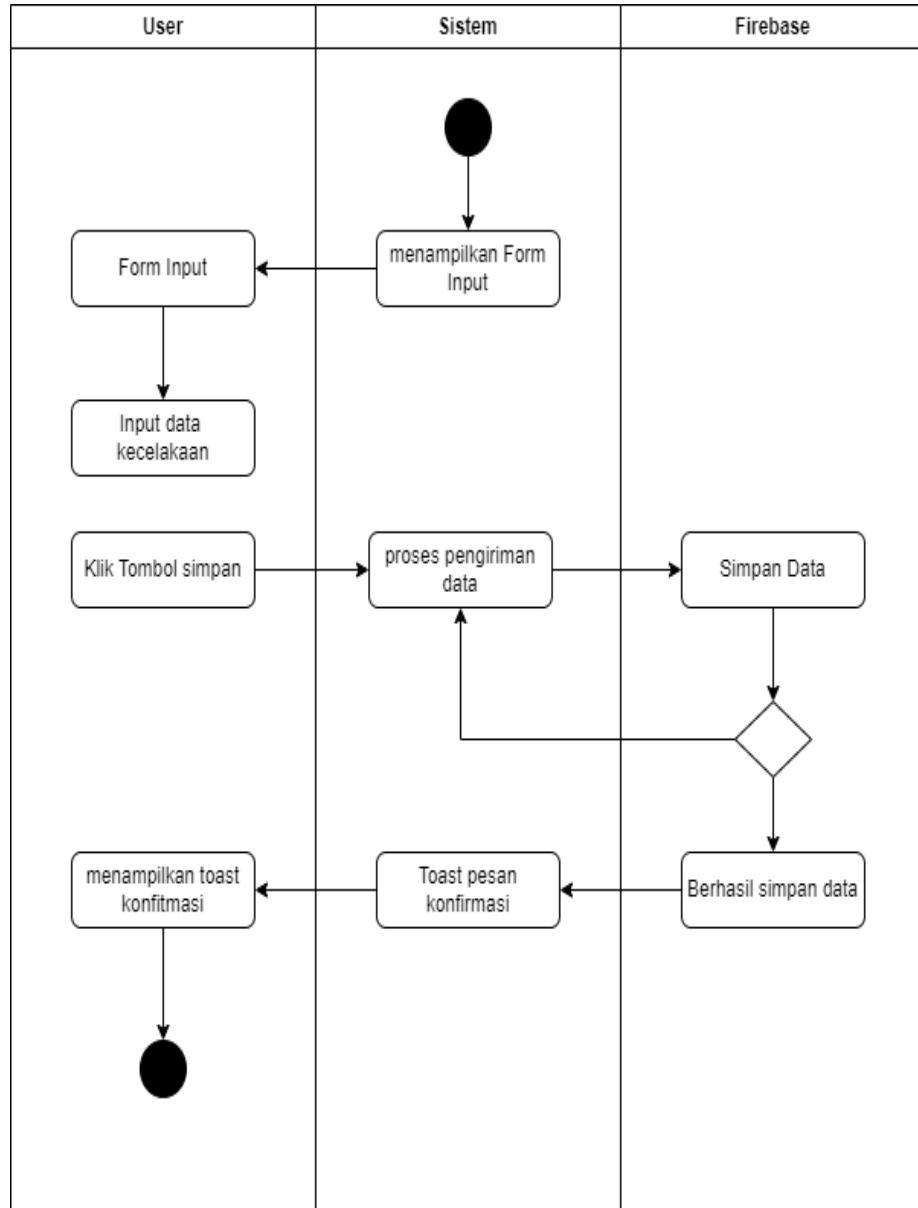


Gambar 3.9 Proses Melihat Notifikasi

Berikut merupakan uraian dari Proses melihat notifikasi :

- 1) *User* masuk kedalam Aplikasi Daerah Rawan Kecelakaan.
- 2) Kemudian pada *menu navigation view* terdapat fitur notifikasi.
- 3) *User* memilih fitur notifikasi, sistem akan menampilkan halaman seluruh *history* notifikasi yang sudah diterima.

## 7. Proses Tambah Data



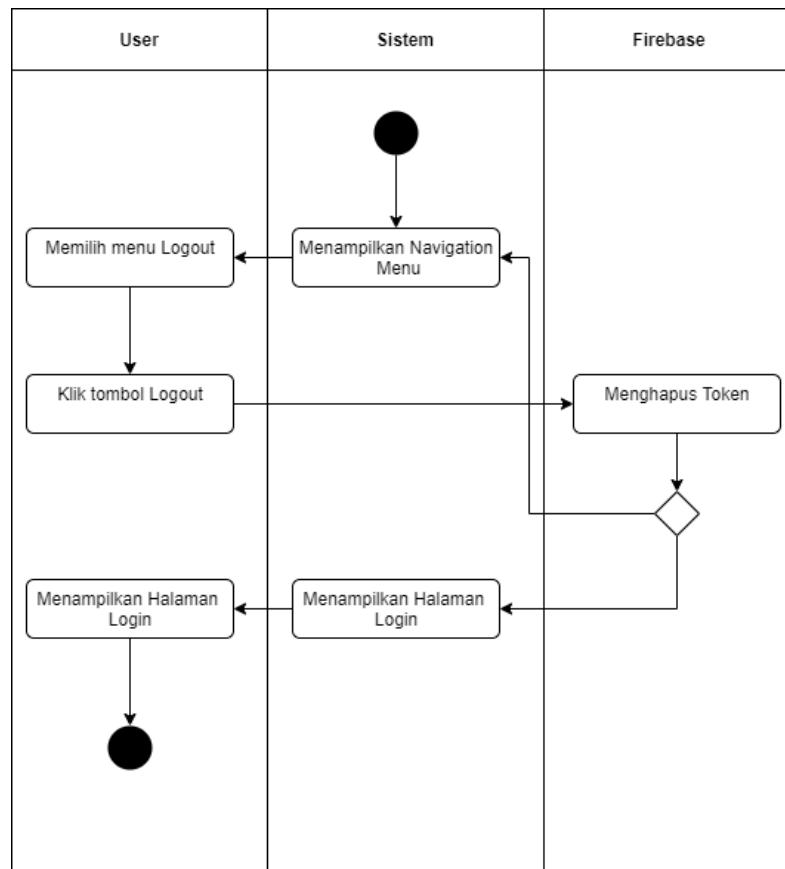
Gambar 3.10 Proses Tambah Data

Berikut merupakan uraian dari Proses Tambah Data :

- 1) Sistem menampilkan antarmuka *form* input kepada *User*.
- 2) *User* memasukkan data kecelakaan ke dalam *form*.
- 3) *User* memasukan data kecelakaan pada *form* input
- 4) *User* mengklik tombol simpan untuk mengirimkan data

- 5) Setelah pengguna mengklik tombol simpan, sistem memproses pengiriman data.
- 6) Firebase menerima data dari sistem dan menyimpannya ke dalam *real-time database*.
- 7) Sistem menyiapkan pesan *toast* konfirmasi untuk ditampilkan.

## 8. Proses Logout



Gambar 3.11 Proses Logout

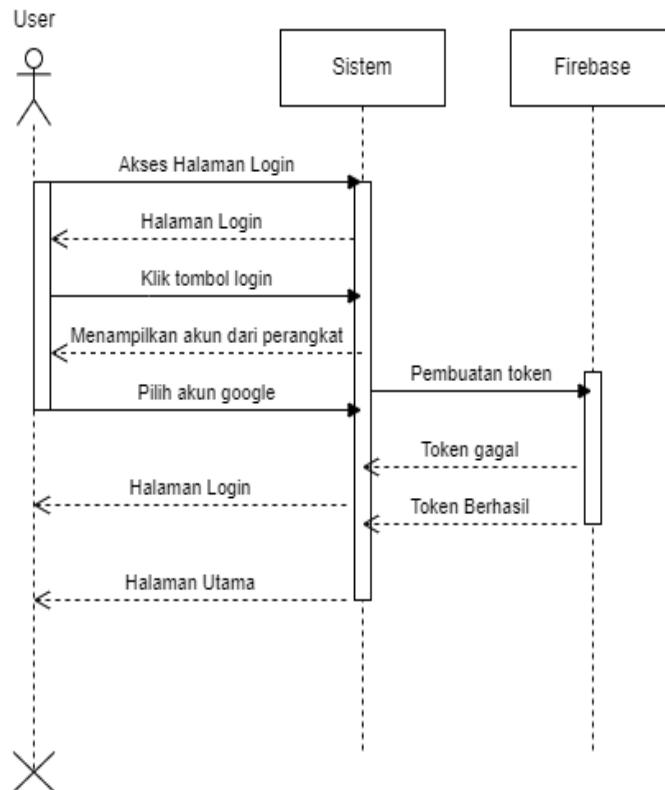
Berikut merupakan uraian dari Proses Logout :

- 1) Sistem menampilkan *menu navigation view*
- 2) *User* memilih tombol *logout* yang berada di paling bawah.
- 3) sistem akan menghapus *token* yang dibuat oleh *account Google* pengguna yang tersimpan di *database real-time*.

- 4) Jika *token* tidak berhasil dihapus akan kembali ke halaman utama , namun jika *token* berhasil dihapus sistem akan menampilkan halaman *login*.

### 3.5.3 Sequence Diagram

#### 1) Login



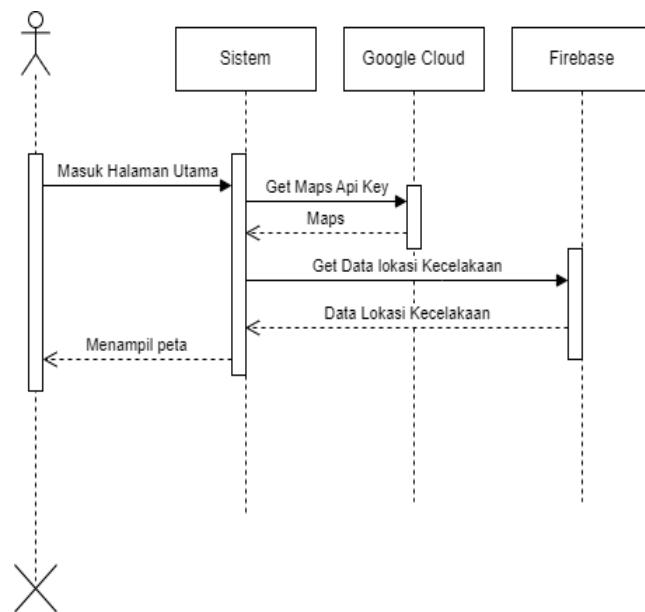
Gambar 3.12 Sequence Diagram *Login*

Berikut uraian penjelasan dari *Sequence Diagram Login* yang mengambarkan alur proses:

- User* mengakses halaman *login*.
- Sistem menampilkan halaman *login*.
- User* mengklik tombol login yang tersedia pada halaman *login*.
- Sistem Menampilkan akun dari perangkat dan menampilkan pilihan akun Google yang tersimpan di perangkat *user*.

- e. *User* memilih akun google yang tersedia atau memasukan akun google baru untuk masuk kedalam sistem.
- f. Menampilkan akun dari perangkat: Sistem menampilkan pilihan akun Google yang tersimpan di perangkat *user*.
- g. Menampilkan akun dari perangkat: Sistem menampilkan pilihan akun Google yang tersimpan di perangkat *user*.
- h. Menampilkan akun dari perangkat: Sistem menampilkan pilihan akun Google yang tersimpan di perangkat *user*.

## 2) Halaman Utama



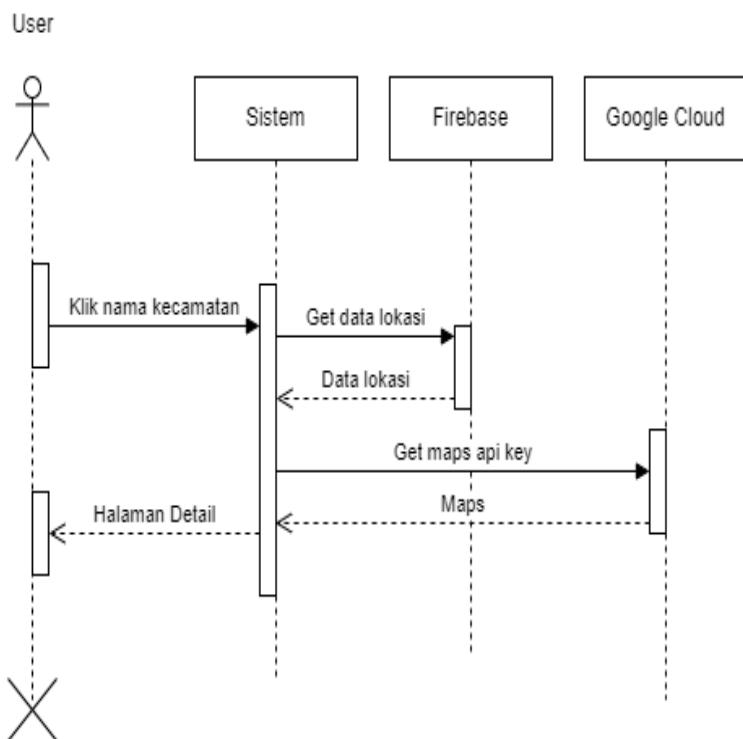
Gambar 3.13 Sequence Diagram Halaman Utama

Berikut uraian penjelasan dari *sequence* diagram halaman utama yang mengambarkan alur proses:

- a. Sistem melalukan *request* dan *get data maps* menggunakan *maps api key* yang ada didalam sistem ke Google *cloud*.
- b. Google *cloud* mengirimkan hasil *request data google maps* ke sistem.

- c. Sistem akan melakukan *get* data kedalam Firebase *real-time database*.
- d. Firebase akan mengembalikan data berupa *JSON*.
- e. Kemudian sistem akan menampilkan *maps* dan data titik daerah rawan kecelakaan berupa marker berdasarkan data yang diambil dari Firebase.

### 3) Halaman Detail Lokasi Kecelakaan



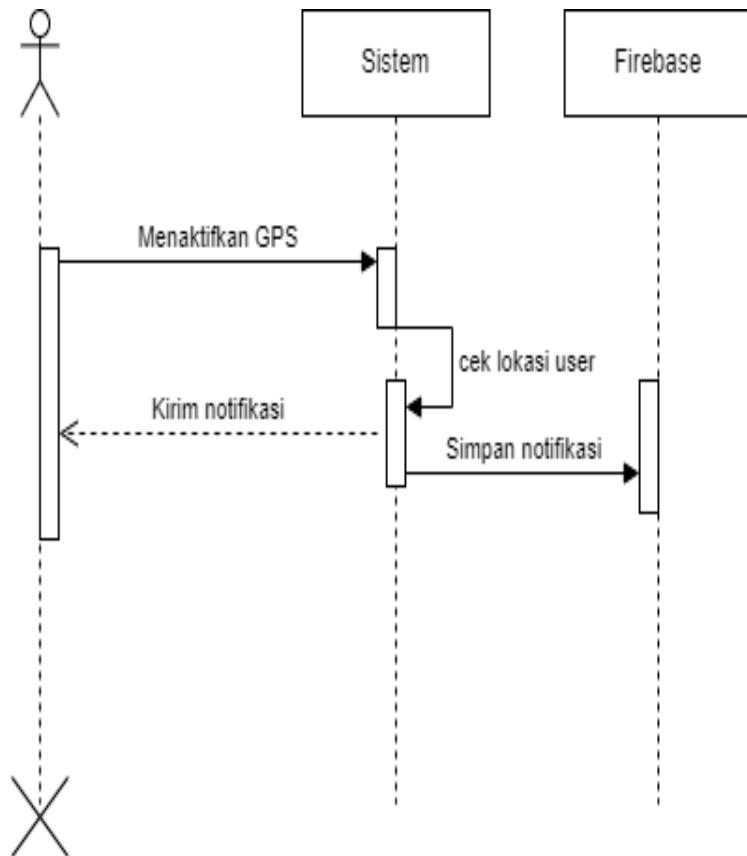
Gambar 3.14 Sequence Diagram Halaman Detail Lokasi Kecelakaan

Berikut uraian penjelasan dari *sequence* diagram halaman detail lokasi kecelakaan yang mengambarkan alur proses:

- a. *User* akan mengklik salah nama kecamatan yang berisi titik rawan kecelakaan pada daerah tersebut.
- b. Sistem akan melakukan *get* data kedalam firebase *real-time database*.
- c. Firebase akan mengembalikan data berupa *JSON*.

- d. Sistem akan melalukan *request* dan *get* data menggunakan *maps api key* ke Google *cloud*.
- e. Kemudian sistem akan menampilkan peta *maps* yang berisi titik daerah rawan kecelakaan berdasarkan nama kecamatan yang dipilih.

#### 4) Mendapatkan Notifikasi



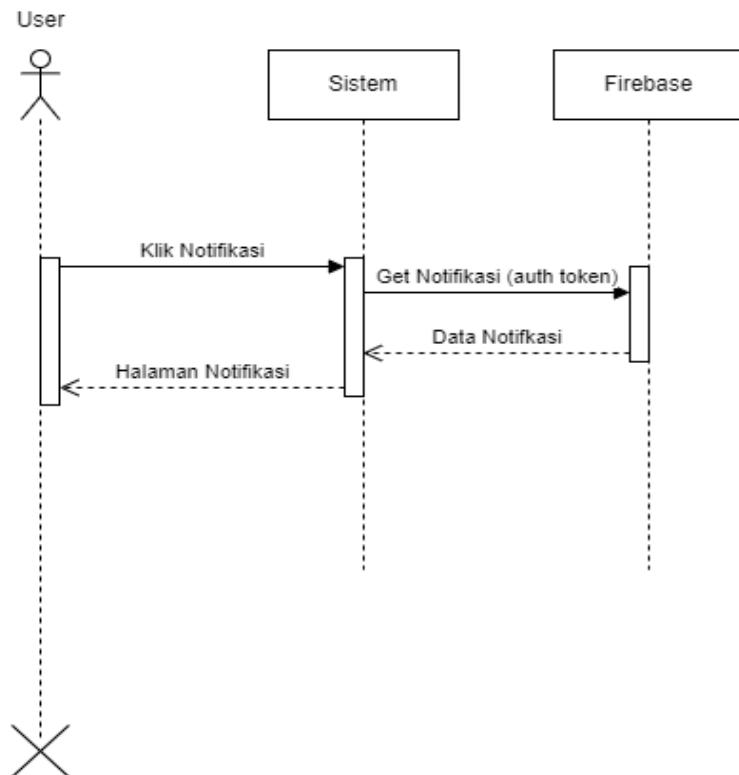
Gambar 3.15 Sequence Diagram Mendapatkan Notifikasi

Berikut uraian penjelasan dari *sequence diagram* mendapatkan notifikasi Kecelakaan yang mengambarkan alur proses:

- a. *User* mengaktifkan *Global Positioning System (GPS)* pada perangkat.
- b. Sistem akan melalukan pengecekan lokasi *user* saat ini, apakah berada dalam radius daerah rawan kecelakaan.

- c. Apabila *user* berada dalam radius tersebut sistem akan mengirimkan notifikasi dan menyimpan notifikasi kedalam Firebase *real-time database*.

### 5) View Notifikasi

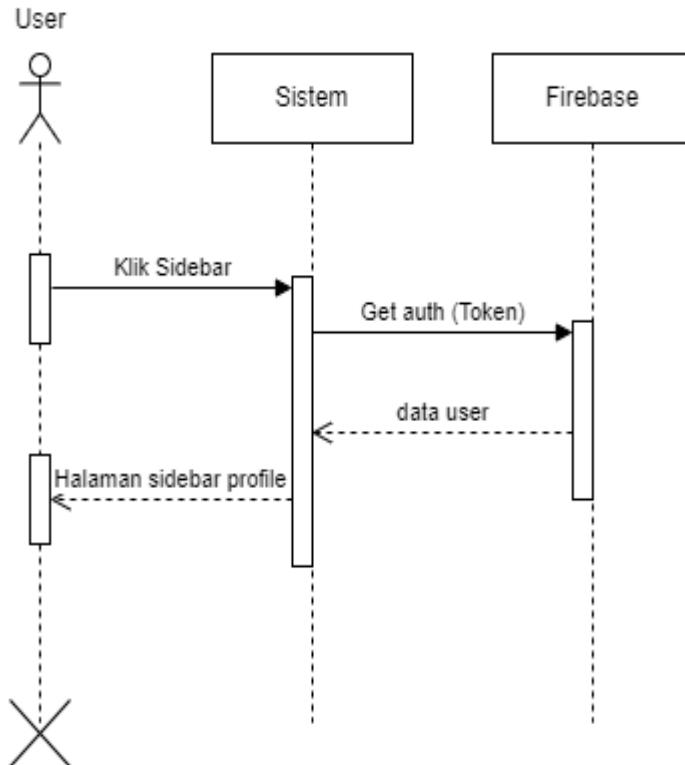


Gambar 3.16 Sequence Diagram View Notifikasi

Berikut uraian penjelasan dari Sequence Diagram *view* notifikasi yang mengambarkan alur proses:

- a. *User* mengklik menu notifikasi yang terdapat di *sidebar*.
- b. Sistem akan melakukan *get data* berdasarkan *token authentikasi* akun Google ke dalam Firebase *real-time database*.
- c. Firebase akan mengambilkan data berupa JSON ke sistem.
- d. Sistem akan menampilkan data yang berhasil di ambil dari Firebase *real-time database*.
- e. Sistem menampilkan halaman notifikasi kepada *user*.

### 6) Sidebar Profile

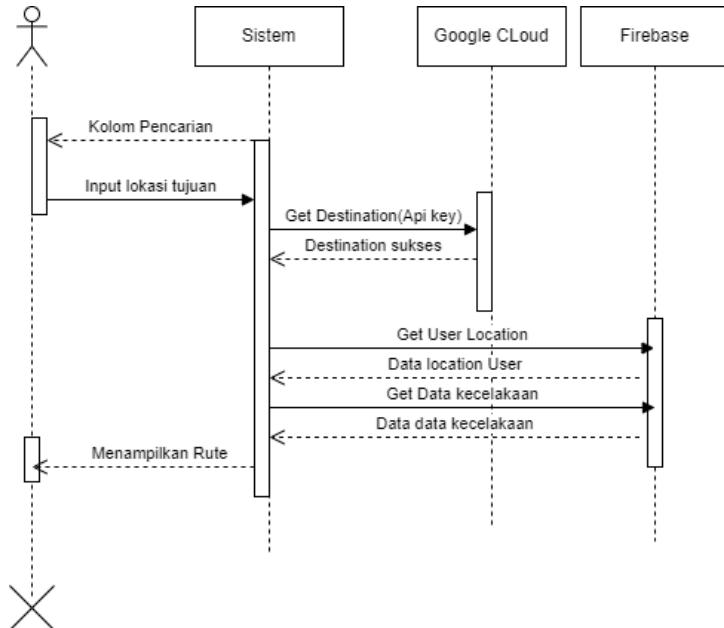


Gambar 3.17 Sequence Diagram *Sidebar Profile*

Berikut uraian penjelasan dari *Sequence Diagram Login* yang mengambarkan alur proses:

- User* mengklik *menu titik tiga* yang berada di pojok kanan atas aplikasi untuk menampilkan *sidebar profile*.
- Sistem akan melalukan *get data* berdarkan *token authentifikasi* kedalam *firebase authentication*.
- Sistem akan menampilkan halam *sidebar profile* yang berisi nama *user*, nama email, dan *foto profile* yang digunakan pada akun Google.

## 7) Pencarian Lokasi



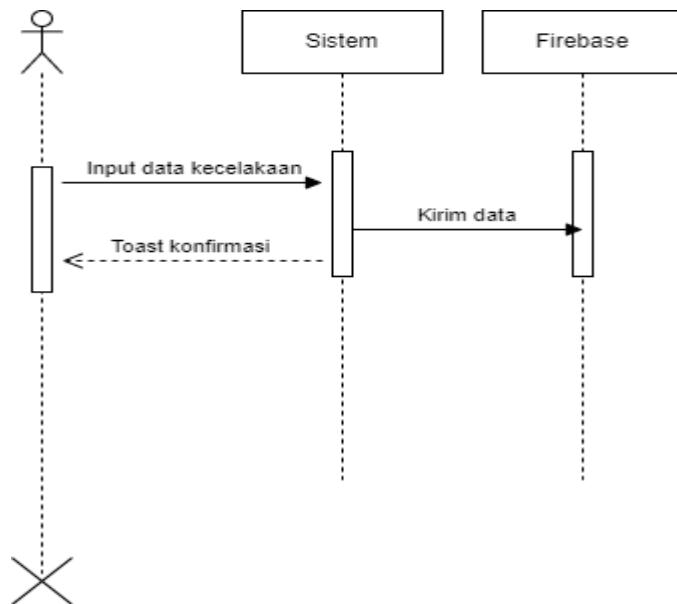
Gambar 3.18 Sequence Diagram Pencarian Lokasi

Berikut uraian penjelasan dari *Sequence Diagram Login* yang mengambarkan alur proses:

- Sistem menampilkan kolom pencarian.
- User* akan menginputkan nama lokasi tujuan pada kolom pencarian.
- Sistem akan melalukan *request get destination* data menggunakan *maps api key* ke google *cloud*.
- Google akan mengirimkan hasil *request* oleh sistem berupa *JSON*.
- Kemudian sistem akan melakukan *get data location user* kedalam Firebase *real-time database*.
- Firebase mengambilkan data *location user* berupa *JSON*.
- Sistem akan melaukan *get data lokasi kecelakan* ke dalam Firebase *real-time database*.
- Firbase mengembalikan data lokasi kecelakaan berupa format *JSON*.

- i. Sistem akan menampilkan *rute* dar lokasi *user* ke lokasi tujuan yang dinputkan oleh *user*.

## 8) Tambah Data

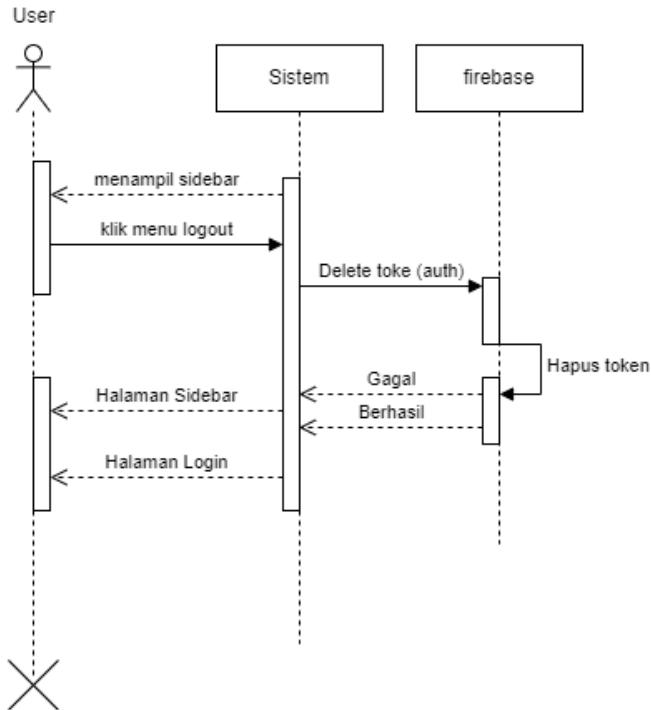


Gambar 3.19 Sequence Diagram Tambah Data

Berikut uraian penjelasan dari *Sequence Diagram Login* yang mengambarkan alur proses:

- a. *User* memasukkan data kecelakaan ke dalam sistem. Sistem menerima data dari pengguna dan mengirimkan data tersebut ke Firebase.
- b. Firebase menerima data dari sistem dan memproses penyimpanannya.
- c. Setelah menerima konfirmasi dari Firebase bahwa data berhasil disimpan, sistem mengirimkan pesan konfirmasi (*toast*) kembali kepada *user*.

### 9) Logout



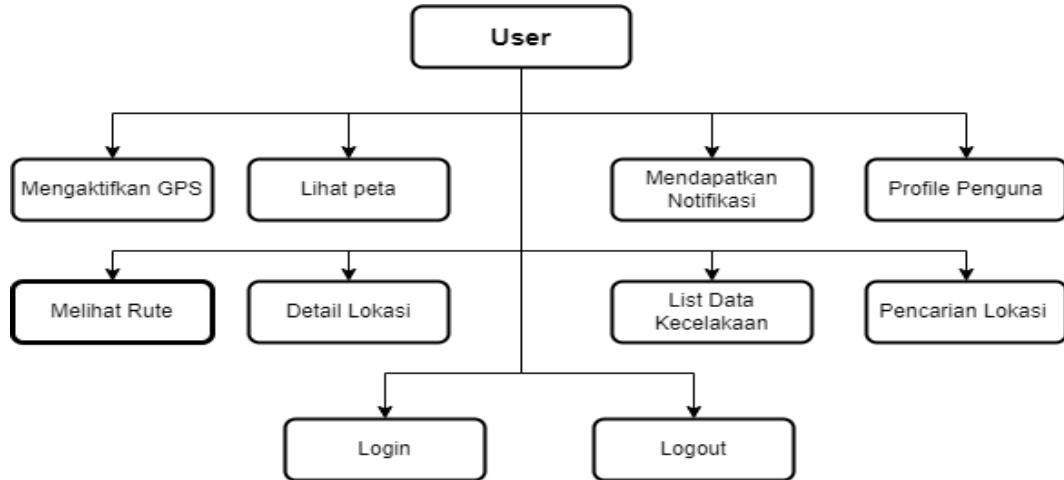
Gambar 3.20 Sequence Diagram Logout

Berikut uraian penjelasan dari *Sequence Diagram Login* yang mengambarkan alur proses:

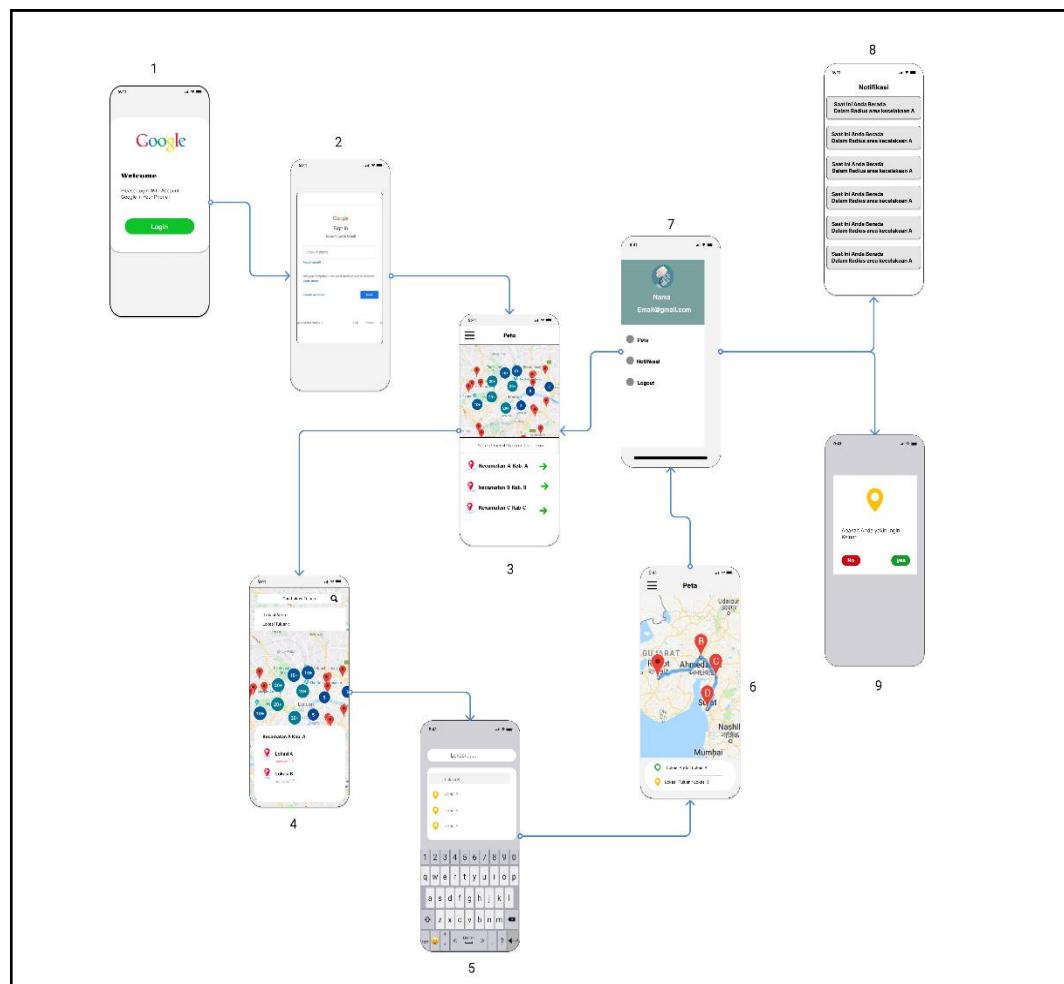
- Sistem menampilkan *sidebar*.
- User* memilih *menu logout*.
- Sistem akan melakukan proses *delete token* yang ada didalam *firebase authentication*.
- Apabila *token* gagal terhapus akan di alihkan ke halaman *sidebar profile*, dan jika berhasil dihapus akan alihkan ke halaman *login*.

### 3.6 Rancangan Struktur Menu

Struktur menu adalah proses pengorganisasian dan pengelompokan item *menu* atau fitur pada antarmuka pengguna (UI) agar mudah diakses dan dipahami oleh pengguna aplikasi.

Gambar 3.21 Struktur *Menu*

### 3.7 Perancangan *Design Interface*

Gambar 3.22 Rancangan *Design Interface*

Berikut merupakan uraian penjelasan dari Gambar rancangan *design interface* :

1. Halaman *login* dengan akun Google. *User* akan menekan tombol "*Login*" untuk melanjutkan.
2. Diarahkan ke halaman Google *Sign-In* untuk memasukkan akun Google atau registrasi akun Google.
3. Setelah berhasil *login*, diarahkan ke halaman utama aplikasi yang menampilkan peta dengan titik lokasi arean rawan kecelakaan. *User* dapat memilih lokasi yang diinginkan dari daftar nama kecamatan di bawah peta.
4. Kemudian aplikasi akan memapilkan halaman detail lokasi arean rawan kecelakaan berdasarkan nama kecamatan yang dipilih oleh *user*.
5. *User* dapat melakukan pencarian lokasi tujuan dengan memasukkan nama kedalam kolom pencarian.
6. Kemudian aplikasi menampilkan Hasil pencarian berupa *route* dari lokasi *user* saat ke lokasi tujuan.
7. Terdapat tombol titik tiga di kanan pojok atas. Ketika di klik akan menampilkan *profile user* berdasarkan akun Google yang dimasukkan dan terdapat tiga *menu* untuk berpindah halaman, *menu* peta akan mengambilkan ke halaman utama, *menu* notifikasi akan menampilkan halaman *history* notifikasi dan *menu logout* akan menampilkan *pop up logout*.
8. Halaman ini menampilkan history notifikasi yang pernah diterima oleh *user*.
9. *Pop up logout* diberikan opsi untuk mengkonfirmasi lokasi yang dipilih dengan menekan tombol "Ya" atau "Tidak".

### 3.8 Metode Testing

Metode testing dalam pengembangan perangkat lunak adalah proses yang digunakan untuk mengevaluasi fungsi dan kualitas dari aplikasi. Berikut adalah metode dan teknik yang digunakan dalam pengujian Aplikasi Daerah rawan kecelakaan Kabupaten Lombok Tengah:

#### 3.8.1 *System Usability Scale (SUS)*

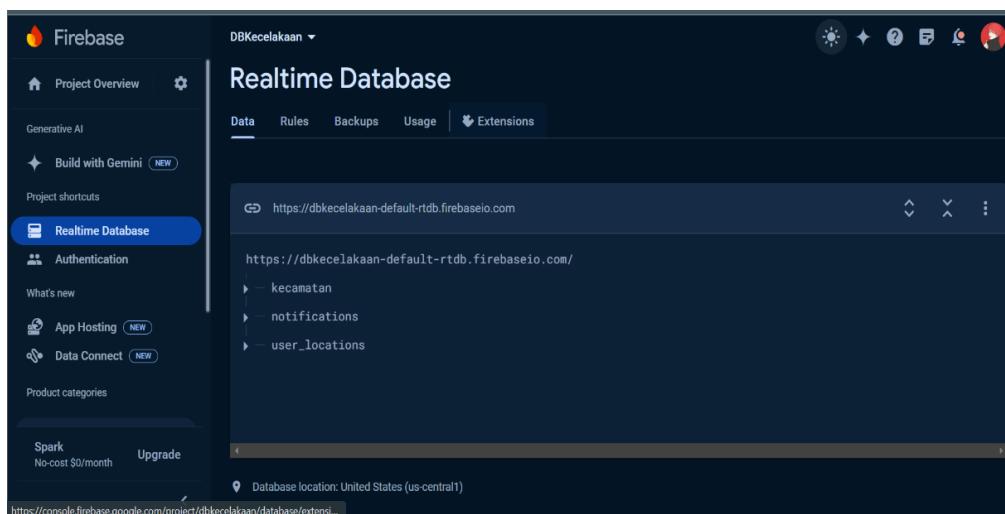
*System Usability Scale (SUS)* adalah kuesioner yang berisi 10 butir pertanyaan. SUS memberikan penilaian subjektif terhadap *usability* sebagai sejauh mana kemudahan yang didapatkan pengguna mencapai tujuan dalam menggunakan aplikasi. Kemudahan tersebut meliputi aspek *effectiveness*, *effeciency*, dan *satisfaction* yang dirasakan pengguna. Efektif merupakan ukuran keakuratan dan kelengkapan tertentu untuk dapat mencapai sasaran diukur dari berapa besar usaha yang diperlukan untuk mencapai tujuan dan menyelesaikan tugas dalam menggunakan aplikasi. Efisiensi diukur dari mencapai tujuan dalam menggunakan aplikasi. Sedangkan kepuasan dapat diukur dari opini subjektif yang dirasakan pengguna setelah mencapai tujuan dalam menggunakan aplikasi.

## BAB IV

### IMPLEMENTASI DAN PEMBAHASAN

#### 4.1 Database

Pada tahap implementasi digunakan Firebase sebagai tempat penyimpanan data karena Firebase *Realtime Database* adalah layanan *database cloud* yang disediakan oleh Google sebagai bagian dari *platform* Firebase. Ini adalah *database* NoSQL yang dapat menyimpan dan sinkronisasi data secara *real-time* antara berbagai klien, seperti aplikasi *web*, *mobile*, atau perangkat IoT. Firebase *Realtime Database* menggunakan model JSON untuk menyimpan data, yang membuatnya sangat fleksibel dan mudah diakses.



Gambar 4.1 Firebase *Realtime Database*

Adapun penjelasan terkait penyimpanan data kedalam firebase *realtime database* yaitu sebagai berikut:

##### 4.1.1 Authentication

Autentikasi dengan akun Google merupakan salah satu metode autentikasi yang disediakan oleh Firebase *Authentication*. Dengan menggunakan autentikasi

ini, pengguna dapat masuk ke dalam aplikasi menggunakan akun Google yang sudah ada didalam perangkat atau membuat akun baru.

The screenshot shows the Firebase Authentication interface under the 'Users' tab. It displays a table of four users, each with their email address, provider (Google), creation date, last sign-in date, and user UID. The columns are Identifier, Providers, Created, Signed In, and User UID. The users listed are adamnizam122@gmail.com, adminnizamwekaweka37..., megatari253@gmail.com, and dagulli24434@gmail.co... . The interface includes a search bar at the top and pagination controls at the bottom.

Identifier	Providers	Created	Signed In	User UID
adamnizam122@gmail.com		Jan 22, 2024	Jun 4, 2024	4ZXOX3KaVmZ97vQsDlxbdU...
adminnizamwekaweka37...		Jan 18, 2024	May 27, 2024	4qqlyVz8JxNFRB3rfnzspX1CY...
megatari253@gmail.com		Jan 17, 2024	Jun 2, 2024	MMsSwzTKSXPECc0Cm4E8jZ...
dagulli24434@gmail.co...		Jan 17, 2024	Mar 22, 2024	FP4w1l6vcZdVzWK94T7i0vXA...

Gambar 4.2 Autentifikasi Dengan Akun Google

#### 4.1.2 Lokasi Kecelakaan

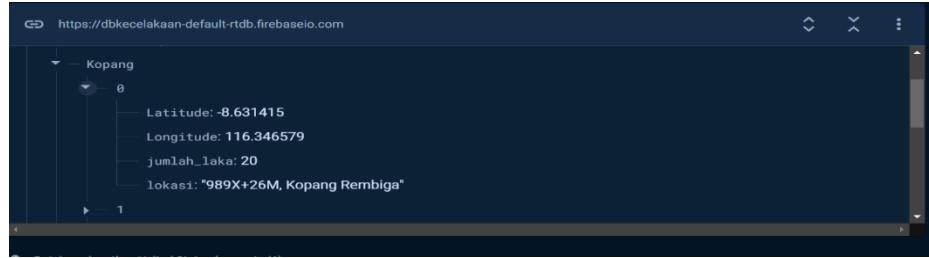
Objek lokasi kecelakaan merupakan data Lokasi kejadian yang disimpan kedalam Firebase *real-time database* berdasarkan objek nama kecamatan yang mempunyai turunan lokasi kecelakaan untuk memetakan titik-titik lokasi daerah rawan kecelakaan.

The screenshot shows the Firebase Real-Time Database interface with a URL of https://dbkecelakaan-default-rtbd.firebaseio.com/. It displays a hierarchical database structure under the 'kecamatan' key, which contains five child nodes: Batukliang, Kopang, Praya\_Barat, Praya\_Tengah, and Praya\_Timur.

```

https://dbkecelakaan-default-rtbd.firebaseio.com/
  - kecamatan
    - Batukliang
    - Kopang
    - Praya_Barat
    - Praya_Tengah
    - Praya_Timur
  
```

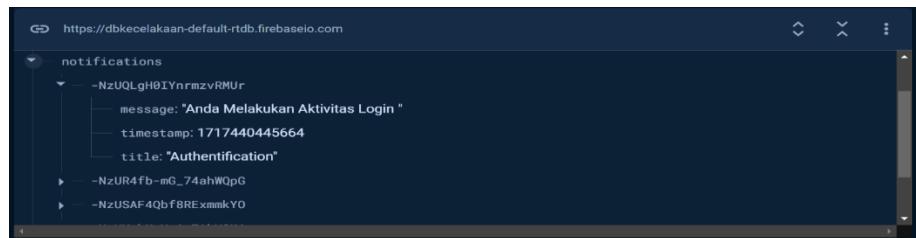
Gambar 4.3 Objek Nama Kecamatan



Gambar 4.4 Objek Lokasi Kecelakaan

#### 4.1.3 Notifikasi

Objek *notifications* kecelakaan adalah informasi yang disimpan dalam Firebase dan digunakan untuk menyimpan pemberitahuan yang dikirimkan kepada pengguna ketika mendekati area kecelakaan.



Gambar 4.5 Objek Notifikasi

#### 4.1.4 User Location

Objek *user location* digunakan untuk menyimpan data lokasi pengguna yang masuk kedalam dalam aplikasi, seperti melacak posisi pengguna secara *real-time* dan memberikan notifikasi berbasis lokasi. Berikut adalah rincian mengenai struktur dan fungsi dari objek *user location*:



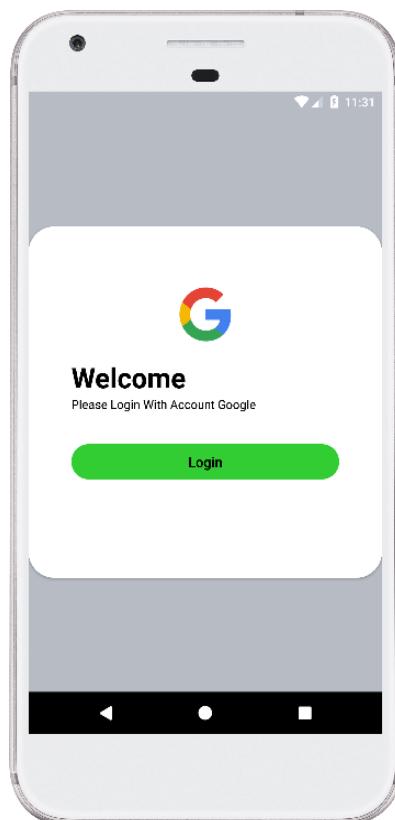
Gambar 4.6 Objek User Location

## 4.2 Interface

*Interface* adalah halaman yang disediakan oleh aplikasi sebagai sarana interaksi antara pengguna dan aplikasi. *Interface* sering disebut tampilan halaman dari aplikasi. Hasil implementasi pembuatan *interface* atau tampilan aplikasi yaitu sebagai berikut:

### 4.2.1 Halaman *Login*

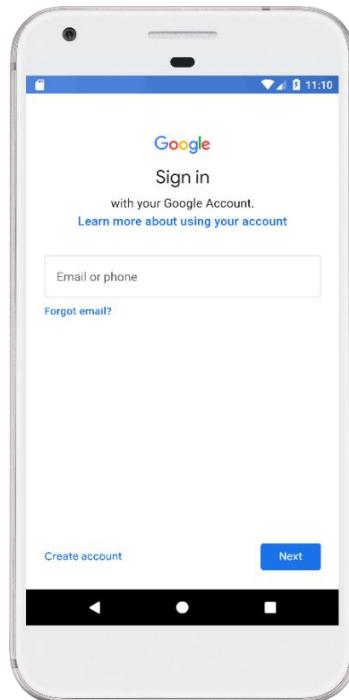
Halaman *login* merupakan halaman yang ditampilkan oleh aplikasi ketika *user* pertama kali membuka aplikasi. Halaman *login* digunakan sebagai proses *authentication* akun sebagai syarat untuk masuk kedalam aplikasi. Proses *authentication* dilakukan menggunakan *account Google* yang terdapat di dalam perangkat.



Gambar 4.7 Halaman *Login*

#### 4.2.2 Halaman Memasukan *Account Google*

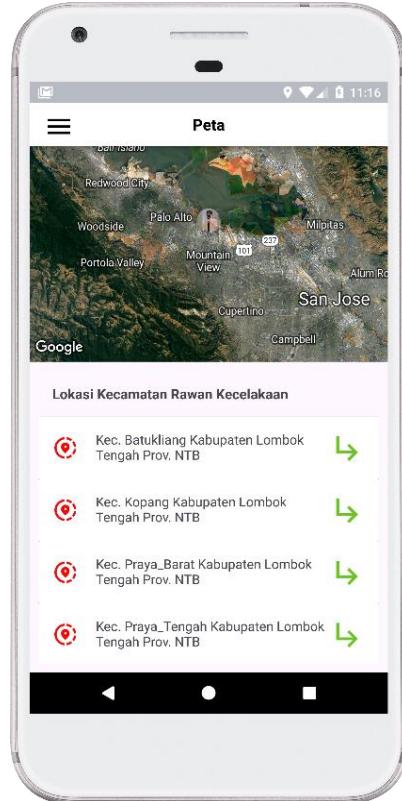
Halaman ini adalah halaman yang ditampilkan ketika belum ada akun Google yang terhubung ke dalam aplikasi atau jika pengguna ingin menggunakan akun Google baru untuk masuk ke dalam aplikasi. Pada halaman ini, pengguna diberikan opsi untuk melakukan proses masuk atau registrasi menggunakan akun Google. Halaman ini bertujuan untuk memudahkan pengguna dalam proses masuk ke dalam aplikasi dengan menyediakan opsi untuk menggunakan akun Google yang sudah ada atau membuat akun baru jika diperlukan.



Gambar 4.8 Halaman Pemilihan *Account Google*

#### 4.2.3 Halaman Utama

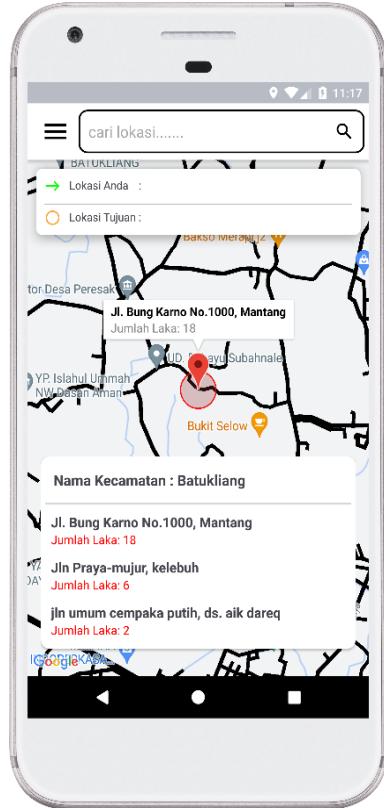
Halaman utama merupakan halaman yang ditampilkan oleh aplikasi ketika *user* berhasil *login* menggunakan *account* Google. *user* akan ditampilkan peta (*maps*) dan terdapat marker atau pin sebagai penanda untuk wilayah lokasi kejadian rawan kecelakaan berdasarkan nama kecamatan.



Gambar 4.9 Halaman Utama

#### 4.2.4 Halaman Detail Lokasi

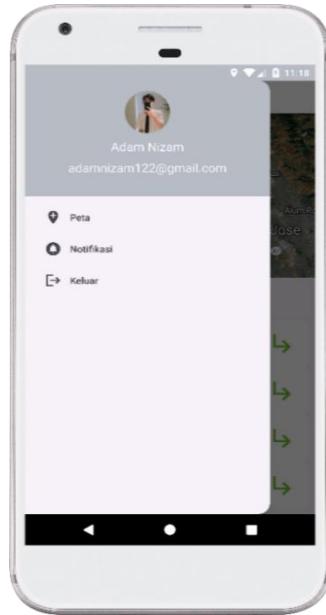
Halaman detail ini merupakan halaman yang menampilkan detail lokasi kejadian rawan kecelakaan berdasarkan kecamatan yang dipilih oleh *user*. Halaman ini menampilkan lokasi titik rawan kecelakaan yang ada di daerah tersebut. Dan terdapat form pencarian lokasi tujuan ini digunakan melihat *rute* dari lokasi *user* saat ke lokasi tujuan hanya dengan mengisikan nama lokasi pada *form* pencarian, aplikasi akan segera memproses hasil inputan tersebut dan menampilkan *rute* kedalam peta dan apabila proses pencarian *rute* gagal, aplikasi akan menampilkan *toast* kesalahan.



Gambar 4.10 Halaman Detail Lokasi

#### 4.2.5 Halaman *Sidebar Profile*

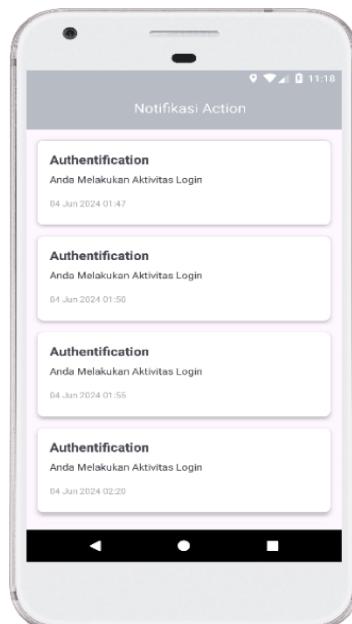
Halaman *sidebar profile* adalah halaman yang menampilkan *profile account* google yang berhasil masuk kedalam aplikasi dan terdapat *foto profile*, nama pengguna dan nama email yang dimasukkan oleh *user*. Halaman *sidebar profile* terdapat tiga *menu* yaitu menu peta untuk kembali ke halaman utama atau hanya mengeser ke kiri untuk kembali ke halaman utama, *menu notifikasi* untuk melihat notifikasi yang pernah diterima oleh *user* dan *menu logout* untuk keluar dari sistem.



Gambar 4.11 Halaman Sidebar Profile

#### 4.2.6 Halaman Notifikasi

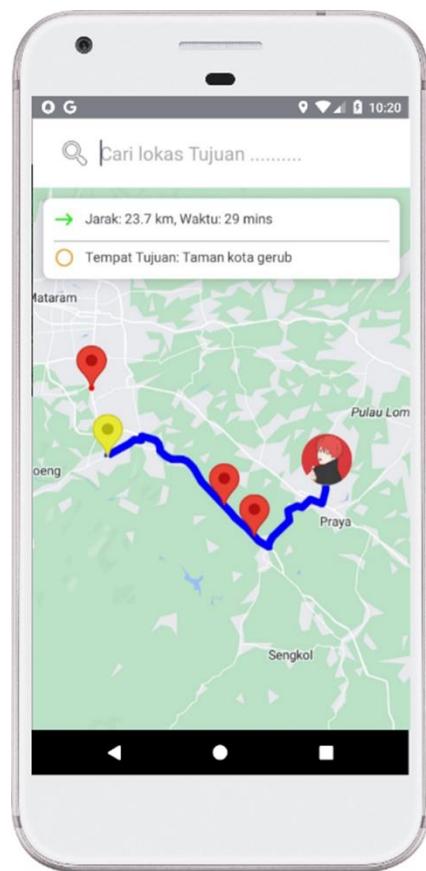
Halaman notifikasi merupakan halaman yang menampilkan *history* notifikasi yang pernah diterima oleh *user*. Pada halaman terdapat *title* atau nama judul notifikasi notifikasi, keterangan notifikasi berupa teks keterangan dan tanggal penerimaan notifikasi.



Gambar 4.12 Halaman Notifikasi

#### 4.2.7 Halaman Hasil Pencarian Lokasi

Halaman ini dirancang untuk menampilkan *rute* perjalanan dan estimasi waktu tempuh dari lokasi pengguna saat ini ke lokasi tujuan yang diinginkan. Pengguna dapat memasukkan alamat atau titik tujuan melalui *form* input pencarian yang tersedia. Setelah memasukkan lokasi tujuan, sistem akan secara otomatis menghitung *rute* terbaik dan menampilkan jalur perjalanan dengan warna biru tua di peta.

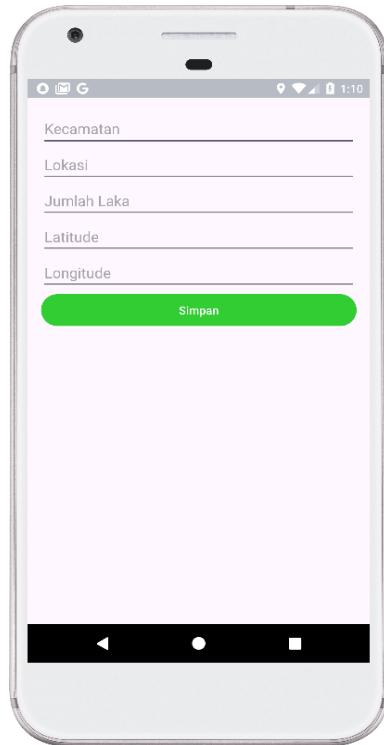


Gambar 4.13 Halaman Hasil Pencarian

#### 4.2.8 Halaman Tambah Data

*Form* input data adalah data yang dikumpulkan melalui elemen-elemen input dalam *form* di dalam aplikasi. *Form* ini digunakan untuk mengumpulkan data

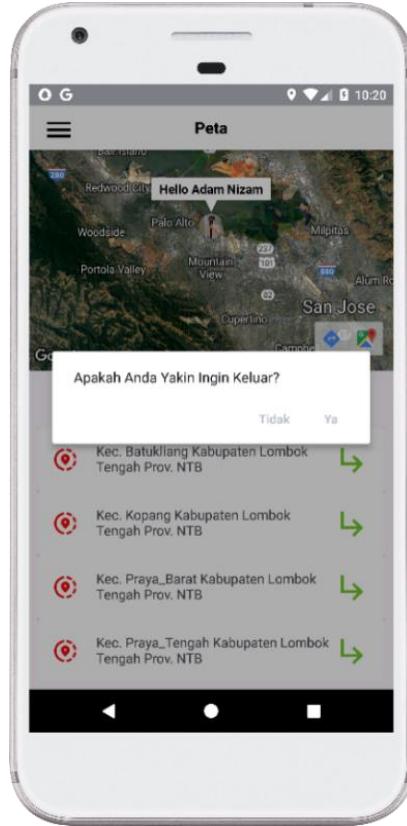
informasi daerah rawan kecelakaan dari *user*, seperti nama kecamatan, nama lokasi, jumlah kecelakaan, *latitude* dan *longitude*.



Gambar 3.23 *Form Input Data*

#### 4.2.9 Halaman *Pop Up Logout*

*Pop-up logout* adalah pesan konfirmasi yang muncul untuk memastikan apakah pengguna ingin keluar dari aplikasi. Jika pengguna memilih "Ya", maka *token* autentikasi pengguna akan dihapus dari Firebase, sehingga pengguna akan keluar dari sesi aplikasi saat ini dan harus melakukan login ulang jika ingin mengakses kembali. Jika pengguna memilih "Tidak", *token* autentikasi akan tetap tersimpan di Firebase, dan pengguna akan tetap berada dalam sesi aktif, memungkinkan *user* untuk terus menggunakan aplikasi tanpa perlu *login* ulang.



Gambar 4.14 Pop Up Logout

### 4.3 Hasil Pengujian

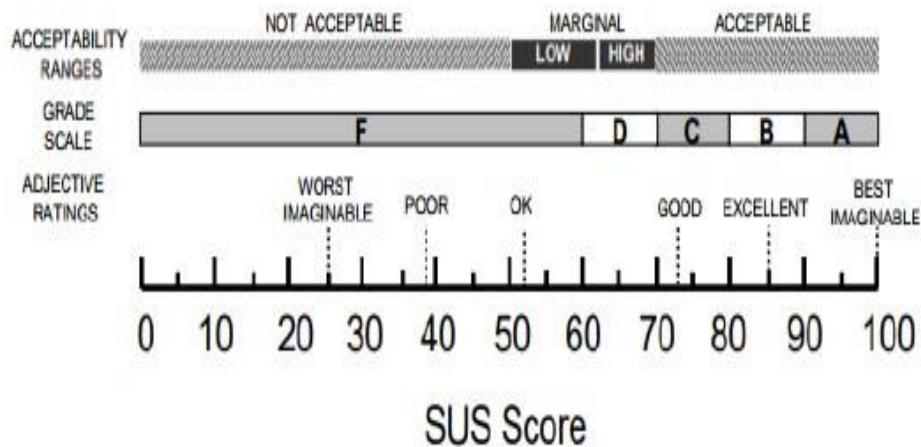
Hasil pengujian dan evaluasi menggunakan metode *System Usability Scale* (*SUS*) dengan melibatkan 20 responden pengguna jalan di Kabupaten Lombok Tengah. Setelah melakukan beberapa tugas yang diberikan kepada responden untuk pengujian *task scenario* antara lain *login* kedalam aplikasi menggunakan akun Google, mencari lokasi tujuan, menampilkan marker pada peta, mendapatkan notifikasi, melihat *profile*, melihat *history* notifikasi, dan *logout* dari aplikasi. Responden diminta mengisi kuesioner *SUS*, dan hasilnya dirangkum dalam Tabel 4.1 yang menunjukkan tingkat *usability* aplikasi berdasarkan persepsi pengguna.

Tabel 4.1 Hasil Penilaian Responden

R	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8	P 9	P 10	Jumlah	Skor SUS (Jumlah x 2.5)
R1	4	4	4	3	4	3	3	2	4	1	32	80
R2	3	4	3	3	4	2	3	4	4	1	31	77.5
R3	0	1	4	4	4	1	3	4	4	2	27	67.5
R4	4	2	4	3	3	3	3	2	4	2	30	75
R5	4	3	4	3	4	3	3	3	2	1	30	75
R6	3	4	2	2	4	1	3	2	3	1	25	62.5
R7	4	3	3	4	4	4	0	3	4	2	31	77.5
R8	4	4	3	2	4	3	4	2	4	4	34	85
R9	4	3	4	4	4	4	1	4	3	3	34	85
R10	2	2	3	1	2	4	3	3	4	4	28	70
R11	3	3	4	1	2	3	4	4	4	4	32	80
R12	3	4	0	4	4	4	4	4	4	4	35	87.5
R13	4	4	4	4	4	4	4	4	4	4	40	100
R14	4	4	4	4	4	3	3	2	4	1	33	82.5
R15	3	4	0	0	2	2	3	4	2	1	21	52.5
R16	0	2	4	2	4	3	4	4	4	3	30	75
R17	0	1	4	1	3	0	4	4	2	1	20	50
R18	2	3	4	1	2	1	3	4	4	4	28	70
R19	0	3	2	2	3	1	3	2	0	2	18	45
R20	2	4	2	2	0	4	2	1	0	0	17	42.5

Jumlah Responden	20
Total Skor SUS	1440
Nilai Rata-Rata	72

Berdasarkan hasil penilaian dari 20 responden diperoleh total nilai Skor SUS sebesar 1440 dengan nilai rata-rata yang dihasilkan adalah 72 Setelah mendapatkan hasil akhir dari penilaian responden. langkah selanjutnya adalah menentukan *grade* hasil penilaian dengan menggunakan dua cara yang dapat digunakan. Penentuan *grade* pertama dilihat dari sisi penerimaan pengguna dengan menggunakan metode *Acceptability*, *Grade Scale*, *Adjective Rating*. Penentuan *grade* kedua dilihat dari sisi *percentile range* (SUS Skor) yang memiliki penilaian yang terdiri dari A,B,C,D,E dan F. berdasarkan kedua cara penentuan hasil penilaian tersebut diperoleh hasil sebagai berikut.



Gambar 4.15 Penentuan Hasil Penilaian Dengan Menggunakan *Acceptability*,

*Grade Scale,Dan Adjective Rating*

Berdasarkan gambar di atas, Penentuan hasil penilaian Aplikasi Daerah Rawan Kecelakaan Pada Polres Kabupaten Lombok Tengah Berbasis Android yaitu:

1. Tingkat *Acceptability Range* pengguna terhadap aplikasi termasuk ke dalam kategori *Acceptable*.
2. Tingkat *Grade Scale* pengguna terhadap aplikasi termasuk ke dalam kategori C.
3. Tingkat *Adjective Rating* pengguna aplikasi termasuk ke dalam kategori *GOOD*.

## **BAB IV**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil pengujian yang telah dilakukan oleh peneliti, maka diperoleh kesimpulan sebagai berikut:

1. Aplikasi pemetaan daerah rawan kecelakaan berhasil memberikan notifikasi *real-time* kepada pengguna saat mendekati daerah rawan kecelakaan, yang dapat meningkatkan kewaspadaan pengguna dan membantu mengurangi angka kecelakaan.
2. Aplikasi ini menyediakan informasi mengenai daerah rawan kecelakaan yang dapat diakses dengan mudah melalui perangkat Android. Hal ini mempermudah masyarakat dalam mendapatkan informasi penting terkait keselamatan berlalu lintas.
3. Aplikasi ini dikategorikan cepat karena membutuhkan durasi 3.902 detik.

#### **5.2 Saran**

Beberapa saran yang dapat diberikan untuk pengembangan selanjutnya dari aplikasi ini adalah:

1. Penambahan fitur dari sisi *administrator* berbasis Android maupun *website* karena pada penelitian berfokus pada pengguna.
2. Belum adanya versi *web service* agar aplikasi semakin lengkap dan mudah dikelola.
3. Integrasi dengan layanan darurat seperti ambulans untuk respons cepat bila terjadi kecelakaan.

## **DAFTAR PUSTAKA**

- [1] S. Pakpahan, H. Indy, and F. Lubis, “Sistem Informasi Geografis Titik Rawan Kecelakaan Lalu-Lintas Berbasis Android ( Studi Kasus : Lintas Sumatera Medan-Lubuk Pakam ),” vol. 04, no. 02, pp. 103–109, 2022.
- [2] S. Hussein, “Fungsi dan Keunggulan SIG Beserta Kelemahannya,” 23 april. [Online]. Available: <https://geospasialis.com/keunggulan-sig/>
- [3] R. ROHANI, H. HASYIM, and Z. PIRJATURROFI, “Analisis Penentuan Lokasi Rawan Kecelakaan Lalu Lintas Dengan Metode Batas Kontrol Atas (Bka) Berbasis Geographic Information System (Gis) Di Kabupaten Lombok Tengah,” *Ganec Swara*, vol. 18, no. 2, p. 975, 2024, doi: 10.35327/gara.v18i2.885.
- [4] Admin, “Data Lakalantas Polres Lombok Tengah Tahun 2023, Berikut Rinciannya,” Nasional Informasi Terbaru dan Viral. [Online]. Available: <https://www.gonasional.com/2024/02/data-lakalantas-polres-lombok-tengah.html>
- [5] Y. Ikawati and P. B. Santosa, “Perancangan dan Evaluasi Aplikasi Peringatan Rawan Kecelakaan Lalu Lintas berbasis Android,” *JGISE J. Geospatial Inf. Sci. Eng.*, vol. 6, no. 1, p. 47, 2023, doi: 10.22146/jgise.86034.
- [6] “Pembuatan Aplikasi Sistem Peringatan Daerah Rawan Kecelakaan Berbasis Android di Daerah Istimewa Yogyakarta YENI IKAWATI, Dr. Eng. Purnama Budi Santosa, S.T., M.App.Sc.,” 2022.

- [7] H. Saputra, I. Stephane, K. Karfindo, and S. Jelita, “Sistem Informasi Geografis Pemetaan Titik Rawan Kecelakaan Daerah Sumatera Barat Berbasis Web,” *Ilk. J. Ilm.*, vol. 10, no. 2, pp. 225–231, 2018, doi: 10.33096/ilkom.v10i2.312.225-231.
- [8] S. Handayani, D. Angga, and P. Pasa, “Perancangan Sistem Informasi Geografis Perlengkapan Jalan Berbasis Daerah Rawan Kecelakaan,” *J. Keselam. Transp. Jalan (Indonesian J. Road Safety)*, vol. 9, no. 1, pp. 21–28, 2022, doi: 10.46447/ktj.v9i1.418.
- [9] M. I. Abas and R. Lamusu, “Perancangan Sistem Informasi Geografis Statistik Kepolisian (Sigap) Kabupaten Gorontalo,” *J. Ilmu Komput.*, vol. 1, no. 2, p. 30, 2021, doi: 10.31314/juk.v1i2.1099.
- [10] F. Wahabi, F. Ramdani, and S. A. Wicaksono, “pengembangan sistem informasi geografis pemetan lokasi kecelakaan berbasis web GIS ( studi kasus : daerah operasional polres kota batu,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 9, pp. 2990–2999, 2018.
- [11] M. S. Janry Haposan U. P. Simanungkalit, S.Si., “KONSEP DASAR SISTEM INFORMASI ( Review ),” *Lect. Notes Sist. Inf.*, pp. 1–10, 2012.
- [12] Fithrie Soufiftri, “Konsep Sistem Informasi,” *J. Adm. Pendidik.*, vol. 3, pp. 1–14, 2005, [Online]. Available: <https://ejournal.upi.edu/index.php/JAPSPs/article/viewFile/6095/4116>
- [13] R. Ariana, “Sistem\_Informasi\_Geografis\_Pengertian\_Da,” *J. Pendidik.*, pp. 1–23, 2019.

- [14] R. P. Sari and M. Susanti, “Penerapan Metode VIKOR (Visekriterijumsko Kompromisno Rangiranje) dalam Pengambilan Keputusan Pemilihan Emulator Android pada Komputer,” *J. Media Inform. Budidarma*, vol. 6, no. 3, p. 1746, 2022, doi: 10.30865/mib.v6i3.4205.
- [15] F. Firly, I. P. Dewi, L. Mursyida, and A. D. Samala, “Dasar-dasar Android Studio Dan Membuat Aplikasi Mobile Sederhana,” *Angew. Chemie Int. Ed. 6(11), 951–952.*, vol. 10, pp. 1–239, 2021.
- [16] [Www.kemkes.go.id](http://www.kemkes.go.id), “ANALISA DAN PERANCANGAN SISTEM INFORMASI TEXT CHATTING BERBASIS ANDROID WEB VIEW,” *Satukan Tekad Menuju Indones. Sehat*, vol. 8, no. 2, pp. 1–5, 2020.
- [17] Dennis Alan, *Object-Oriented Systems Analysis and Design: An MDA Approach with UML*. 2013. [Online]. Available: <https://books.google.com/books?hl=en&lr=&id=rbLrBgAAQBAJ&oi=fnd&pg=PR5&dq=analytical+hierarchy+process&ots=YK5Pv1Mqgl&sig=0APVYULjIENCA4W4nVtc20vcKAM%0Ahttps://www.academia.edu/download/55050518/1DataStorageDesign.pdf>
- [18] S. Indriyani Fintri, Yunita, Muthia Dinda A, Surniandari Artika, “20. Buku-Ajar-APSI\_2,” 2019.
- [19] B. Hartono, *Cara Mudah dan Cepat Sistem Informasi*. 2021.
- [20] Wikipedia, “Software development kit.” [Online]. Available: [https://en.wikipedia.org/wiki/Software\\_development\\_kit](https://en.wikipedia.org/wiki/Software_development_kit)
- [21] HUSEN MULACHELA, “Database Adalah: Pengertian dan Jenisnya,”

- Katadata.com. [Online]. Available:  
<https://katadata.co.id/digital/teknologi/61c04e3f62f5b/database-adalah-pengertian-dan-jenisnya>
- [22] Wikipedia, “Google Maps.” [Online]. Available:  
[https://id.wikipedia.org/wiki/Google\\_Maps#Alternatif](https://id.wikipedia.org/wiki/Google_Maps#Alternatif)
- [23] K. resor kota Yogyakarta, “Satlantas.” [Online]. Available:  
<https://www.polresjogja.com/p/satlantas.html#:~:text=Satuan%20Lalu%20Lintas%20yang%20selanjutnya,yang%20berada%20di%20bawah%20Kapolres>.
- [24] P. Sumbar, “Struktur Organisasi.” [Online]. Available:  
<https://50kota.sumbar.polri.go.id/index.php/struktur-organisasi/>
- [25] J. Brooke, “SUS: A ‘Quick and Dirty’ Usability Scale,” *Usability Eval. Ind.*, no. November 1995, pp. 207–212, 2020, doi: 10.1201/9781498710411-35.

## LAMPIRAN

### Lampiran 1 Dokumentasi Wawancara dan Pengambilan data

Foto wawancara



Foto Pengambilan data *blackspot* daerah kecelakaan pada petugas kepolisian



## **PERTANYAAN WAWANCARA**

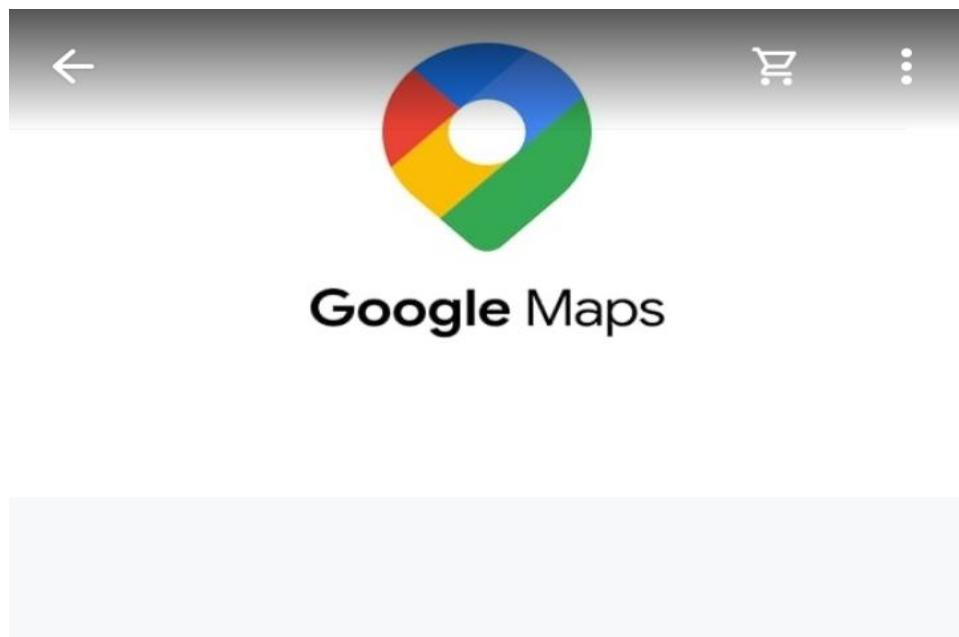
1. Apa saja upaya pihak kepolisian dalam mencegah terjadinya kecelakaan lalu lintas di Lombok Tengah?
2. Bagaimana kondisi fisik jalan dan kepatuhan pengendara terhadap rambu lalu lintas di daerah yang sering terjadi kecelakaan?
3. Apakah ada inovasi atau teknologi yang digunakan untuk membantu mengurangi kecelakaan lalu lintas?
4. Bagaimana respon masyarakat terhadap upaya-upaya yang telah dilakukan oleh pihak kepolisian?
5. Apa kendala utama yang dihadapi dalam upaya mengurangi kecelakaan lalu lintas?
6. Apakah Anda bisa menjelaskan lebih detail mengenai penyuluhan yang dilakukan oleh pihak kepolisian?

## **Jawaban Narasumber**

- 1) Pihak kepolisian melakukan berbagai upaya untuk mencegah terjadinya kecelakaan lalu lintas. Upaya-upaya tersebut meliputi penyuluhan kepada masyarakat, pemasangan spanduk, pemasangan rambu-rambu lalu lintas di setiap jalan, edukasi kepada masyarakat, dan terjun langsung ke lokasi daerah yang sering terjadi kecelakaan.
- 2) Hasil pengamatan menunjukkan banyak pengendara yang tidak mematuhi aturan lalu lintas dan kondisi jalan serta bahu jalan banyak yang rusak. Ini termasuk kurangnya penerangan dan rambu-rambu keselamatan.
- 3) saat ini masih belum ada, Perlunya dibuat aplikasi yang bisa memetakan titik daerah rawan kecelakaan serta memberikan peringatan notifikasi lewat smartphone kepada pengguna jalan untuk menangulangi kecelakaan di wilayah Kabupaten Lombok Tengah.
- 4) Respon masyarakat cukup positif terhadap upaya-upaya yang dilakukan. Namun, masih ada kendala dalam hal kepatuhan terhadap aturan lalu lintas, yang memerlukan edukasi dan sosialisasi lebih lanjut
- 5) Penyuluhan dilakukan secara berkala di berbagai komunitas dan sekolah. Materi penyuluhan mencakup pentingnya mematuhi rambu lalu lintas, penggunaan alat keselamatan seperti helm dan sabuk pengaman, serta dampak negatif dari perilaku berkendara yang tidak aman. Kami juga menggunakan media sosial dan radio lokal untuk menjangkau lebih banyak orang.

## Lampiran 1 Dokumentasi *development* Aplikasi

Penyewaan *Api Key*



### Google Maps API Keys 30 Hari

Rp 130.000,00

#### Google Maps API Keys Tersedia

Maps Elevation API  
Maps Embed API  
Maps JavaScript API  
Maps SDK for Android  
Maps SDK for iOS  
Maps Static API  
Roads API  
Places API  
Places API (New)  
D... [Baca selengkapnya](#)

[Tambah ke permintaan pesanan](#)

## Testing get api key google maps menggunakan postman

The screenshot shows a Postman interface with a GET request to `https://maps.googleapis.com/maps/api/directions/json?`. The parameters include `origin=-8.698557116.2827827&destination=-8.6911398,116.2497192&mode=driving&key=AlzaSyDjlqC7oks1oDDhsqewHtoPVYXgEsdWKxI`. The response body shows an error message: "The provided API key is expired.".

## Limit pengunaan dan request api key

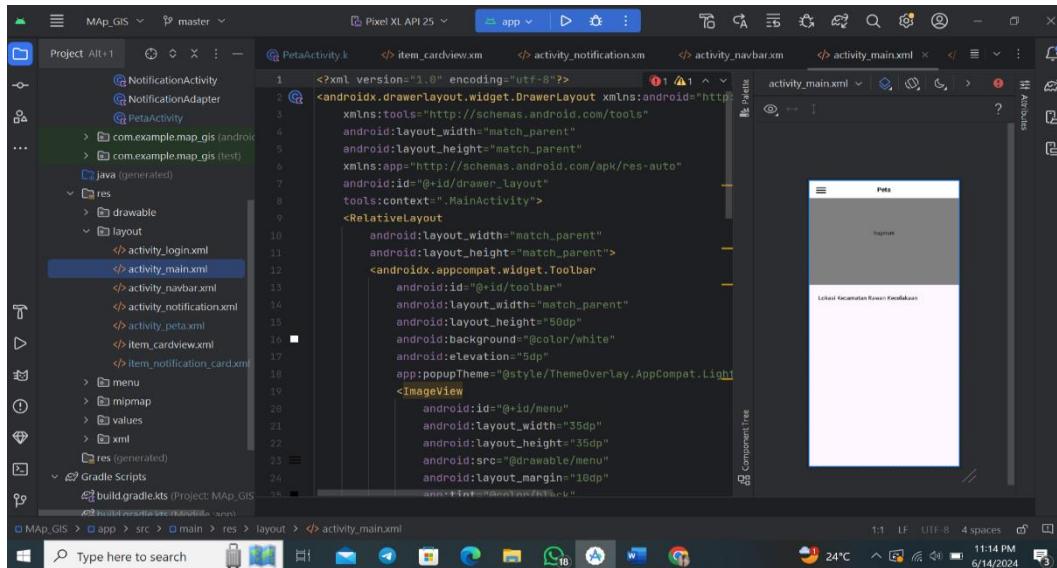
The screenshot shows a Grafana dashboard with a query for "Consumed API - Request count". The results table shows two rows:

response_code_class	Name (from credential_id)	version	Value
2xx	2xx apikey:AlzaSyDjlqC7oks1oDDhsqewHtoPVYXgEsdWKxI v1	v1	110
4xx	4xx apikey:AlzaSyDjlqC7oks1oDDhsqewHtoPVYXgEsdWKxI v1	v1	251

## Menjalankan kode menggunakan emulator pada android studio

The screenshot shows the Android Studio interface with the `PetaActivity.kt` file open. The code is related to initializing Places and handling location data. To the right, a Pixel XL API 25 emulator is running, showing a "Welcome" screen with a "Login" button.

## Source code activity main

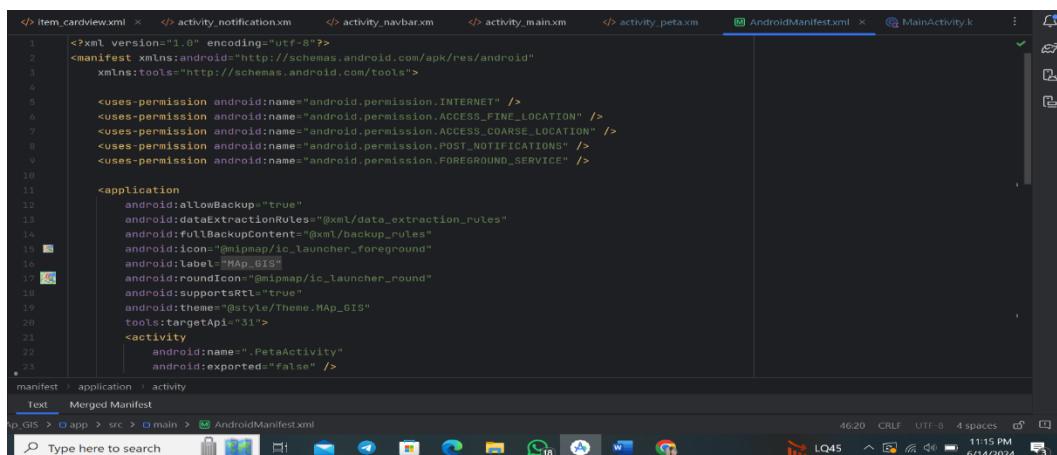


The screenshot shows the Android Studio interface with the project navigation bar at the top. Below it, the file tree shows a folder named 'com.example.map\_gis' containing Java files like 'PetActivity.kt' and 'NotificationAdapter.kt', and XML files such as 'activity\_main.xml', 'activity\_notification.xml', and 'activity\_peta.xml'. The main editor window displays the XML code for 'activity\_main.xml'. The code defines a DrawerLayout with a Toolbar and an ImageView. A preview window on the right shows a dark-themed mobile application screen with a toolbar and an image view. The bottom status bar shows system information like battery level and time.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout xmlns:android="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/drawer_layout"
    tools:context=".MainActivity">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="50dp"
            android:background="@color/white"
            android:elevation="10dp"
            app:popupTheme="@style/ThemeOverlay.AppCompat.Light"/>
        <ImageView
            android:id="@+id/menu"
            android:layout_width="35dp"
            android:layout_height="35dp"
            android:src="@drawable/menu"
            android:layout_margin="10dp"/>
    

```

## Source code manifest

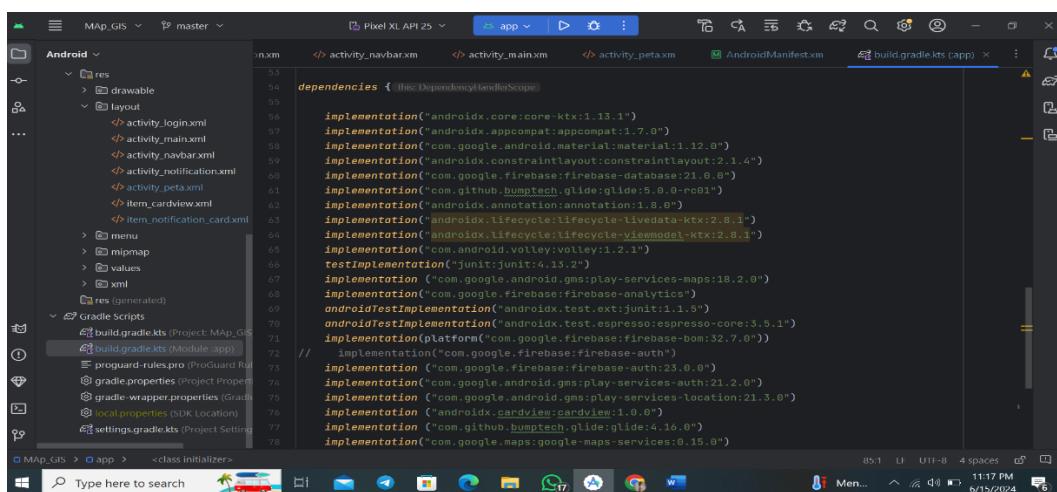


The screenshot shows the Android Studio interface with the project navigation bar at the top. Below it, the file tree shows a folder named 'com.example.map\_gis' containing Java files like 'PetActivity.kt' and 'NotificationAdapter.kt', and XML files like 'activity\_main.xml'. The main editor window displays the XML code for 'AndroidManifest.xml'. The code defines a manifest with a single application tag containing an activity named 'PetActivity'. The activity is marked with 'android:exported="false"'. The bottom status bar shows system information like battery level and time.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.POST_NOTIFICATIONS" />
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher_foreground"
        android:label="Map_GIS"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.MAP_GIS"
        tools:targetApi="31">
        <activity
            android:name=".PetActivity"
            android:exported="false" />
    


```

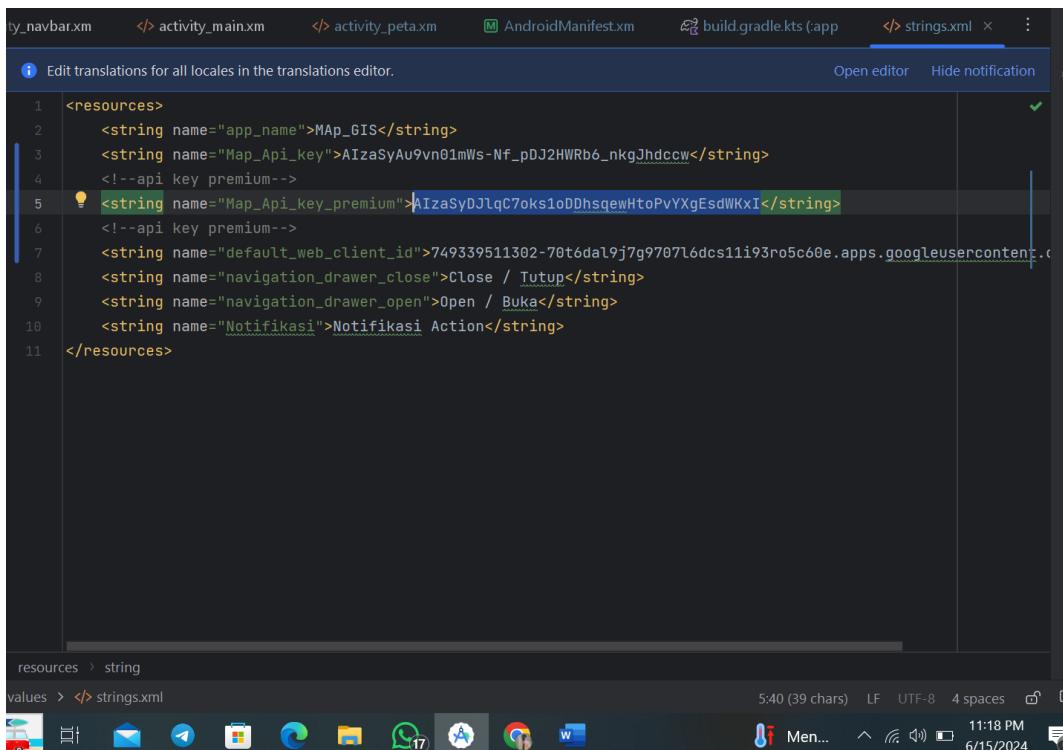
## Konfigurasi library level App



The screenshot shows the Android Studio interface with the project navigation bar at the top. Below it, the file tree shows a folder named 'com.example.map\_gis' containing Java files like 'PetActivity.kt' and 'NotificationAdapter.kt', and XML files like 'activity\_main.xml'. The main editor window displays the Kotlin-based build script for the app module. The script includes dependencies for various Android libraries such as androidx.core, androidx.appcompat, com.google.android.material, androidx.constraintlayout, androidx.lifecycle, com.google.firebase.firebaseio, com.github.bumptech.glide, androidx.annotation, androidx.lifecycle.livedata, androidx.lifecycle.viewmodel, com.android.volley, junit, com.google.android.gms.maps, com.google.firebase.firebaseio.analytics, androidx.test.espresso, androidx.test.junit, com.google.firebase.firebaseio.auth, com.google.firebase.firebaseio.database, com.google.android.gms.play-services-auth, com.google.android.gms.play-services-auth-base, com.google.android.gms.play-services-location, androidx.cardview, com.github.bumptech.glide, and com.google.maps.google-maps-services. The bottom status bar shows system information like battery level and time.

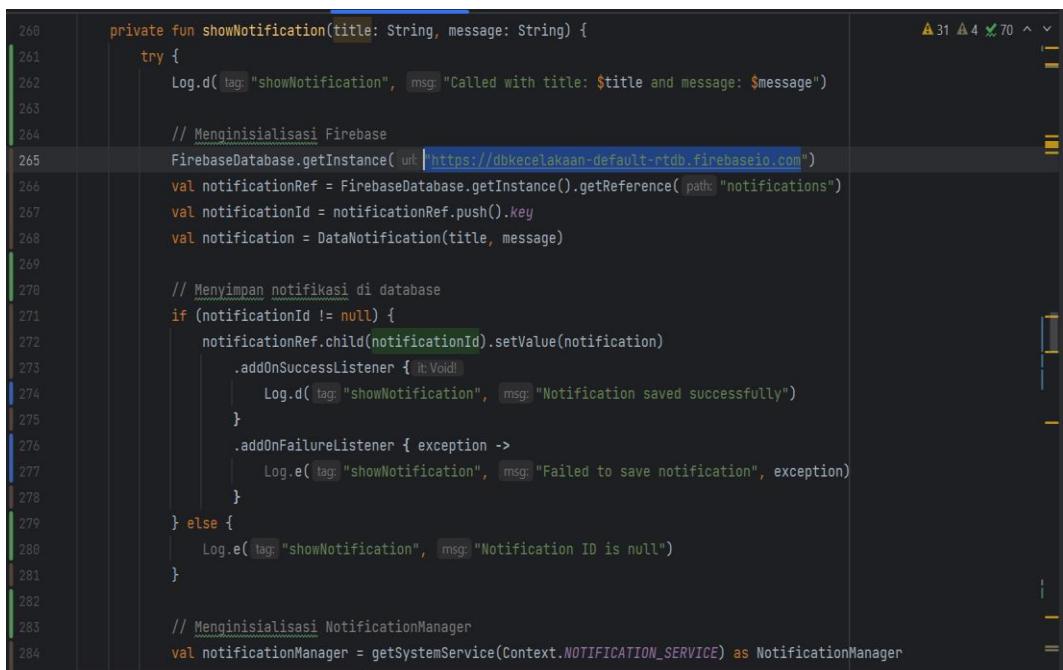
```
dependencies {
    implementation("androidx.core:core-ktx:1.13.1")
    implementation("androidx.appcompat:appcompat:1.7.0")
    implementation("com.google.android.material:material:1.12.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    implementation("com.google.firebase:firebase-database:21.0.0")
    implementation("com.github.bumptech.glide:glide:5.0.0-rc01")
    implementation("androidx.annotation:annotation:1.8.0")
    implementation("androidx.lifecycle:lifecycle-livedata-ktx:2.8.1")
    implementation("androidx.lifecycle:lifecycle-viewmodel-ktx:2.8.1")
    implementation("com.android.volley:volley:1.2.1")
    testImplementation("junit:junit:1.13.2")
    implementation("com.google.android.gms:play-services-maps:18.2.0")
    implementation("com.google.firebase:firebase-analytics")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
    implementation(platform("com.google.firebase:firebase-bom:32.7.0"))
    implementation("com.google.firebase:firebase-auth")
    implementation("com.google.firebaseio.firebaseio-admin")
    implementation("com.google.firebaseio.firebaseio-auth")
    implementation("com.google.android.gms:play-services-auth:21.2.0")
    implementation("com.google.android.gms:play-services-auth-base:21.3.0")
    implementation("androidx.cardview:cardview:1.0.0")
    implementation("com.github.bumptech.glide:glide:4.16.0")
    implementation("com.google.maps.google-maps-services:0.15.0")
}
```

## Penyimpanan Direktori Api Key



```
<resources>
    <string name="app_name">MAP_GIS</string>
    <string name="Map_Api_key">AIzaSyAu9vn01mWs-Nf_pDJ2HWrb6_nkgJhdccw</string>
    <!--api key premium-->
    <string name="Map_Api_key_premium">AIzaSyDJlqC7oks1oDDhsqewHtoPvYXgEsdWKxI</string>
    <!--api key premium-->
    <string name="default_web_client_id">749339511302-70t6dal9j7g9707l6dcsl1i93ro5c60e.apps.googleusercontent.com</string>
    <string name="navigation_drawer_close">Close / Tutup</string>
    <string name="navigation_drawer_open">Open / Buka</string>
    <string name="Notifikasi">Notifikasi Action</string>
</resources>
```

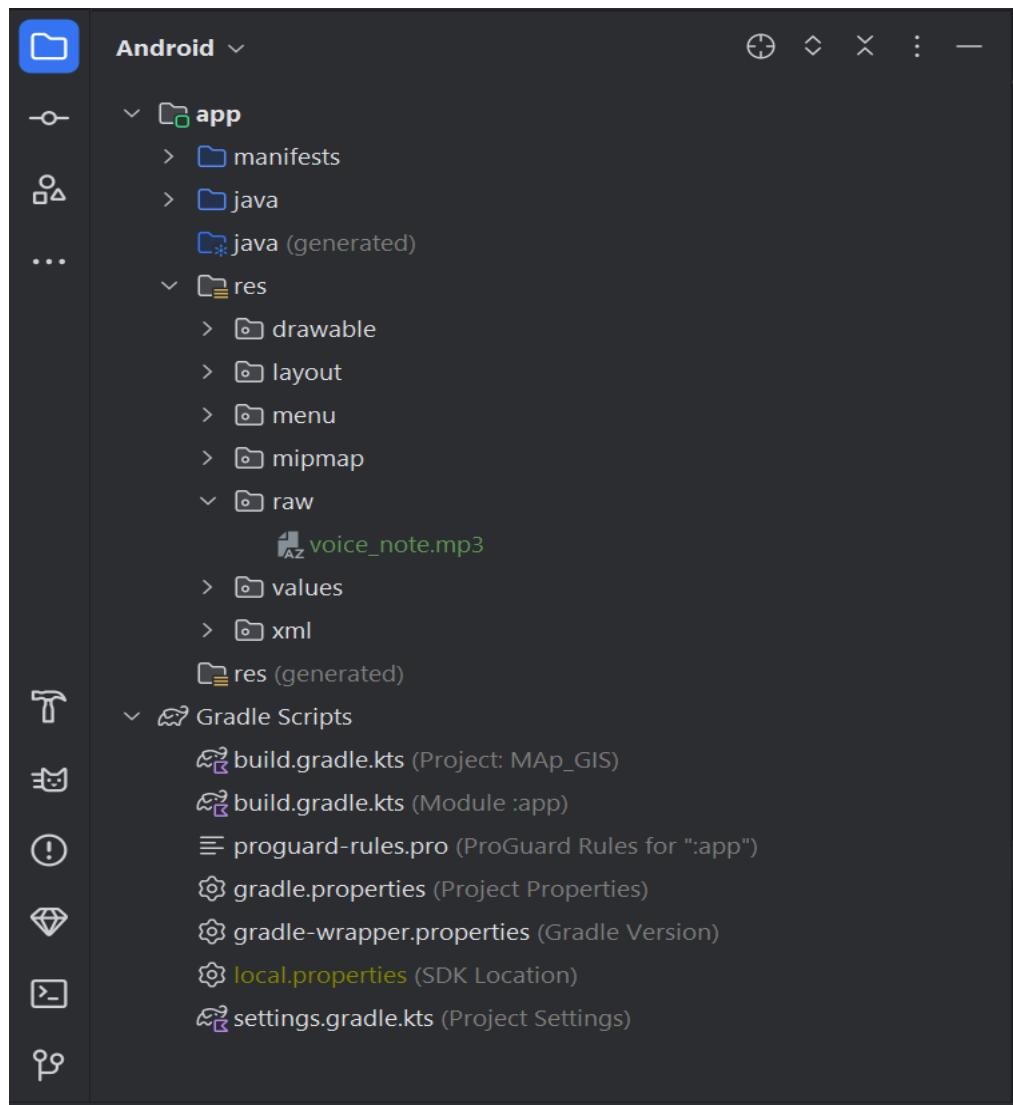
## Source code fungsi mengirimkan notifikasi



```
260     private fun showNotification(title: String, message: String) {
261         try {
262             Log.d("showNotification", "Called with title: $title and message: $message")
263
264             // Menginisialisasi Firebase
265             FirebaseDatabase.getInstance().url = "https://dbkecelakaan-default.firebaseio.com"
266             val notificationRef = FirebaseDatabase.getInstance().getReference("notifications")
267             val notificationId = notificationRef.push().key
268             val notification = DataNotification(title, message)
269
270             // Menyimpan notifikasi di database
271             if (notificationId != null) {
272                 notificationRef.child(notificationId).setValue(notification)
273                     .addOnSuccessListener { it: Void? ->
274                         Log.d("showNotification", "Notification saved successfully")
275                     }
276                     .addOnFailureListener { exception ->
277                         Log.e("showNotification", "Failed to save notification", exception)
278                     }
279             } else {
280                 Log.e("showNotification", "Notification ID is null")
281             }
282
283             // Menginisialisasi NotificationManager
284             val notificationManager = getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager
```

```
286  
287  
288  
289     val channelName = "Channel Name"  
290     val customSoundUri = Uri.parse("file:///android_asset/" + packageName + "/" + R.raw.voice_note)  
291  
292     Log.d("showNotification", "Custom sound URI: $customSoundUri")  
293  
294     val notificationBuilder = NotificationCompat.Builder(context, channelId)  
295         .setContentTitle(title)  
296         .setContentText(message)  
297         .setSmallIcon(R.drawable.notification)  
298         .setPriority(NotificationCompat.PRIORITY_HIGH)  
299         .setDefaults(Notification.DEFAULT_SOUND) // Set default sound  
300         .setSound(customSoundUri)  
301  
302     Log.d("showNotification", "Building notification")  
303  
304     // Mengirim notifikasi  
305     notificationManager.notify(0, notificationBuilder.build())  
306  
307     Log.d("showNotification", "Notification sent")  
308  
309     // Memutar suara secara langsung untuk pengujian  
310     val mediaPlayer = MediaPlayer.create(context, customSoundUri)  
311     mediaPlayer?.start()  
312  
313 } catch (e: Exception) {  
    Log.e("showNotification", "Error showing notification", e)
```

Lokasi Path Penyimpanan suara Voice Note



## Lampiran 2 Kusioner Evaluasi

**Tabel 1.** Hasil Jawaban Responden

No	NAMA	Alamat	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1	Gilang Pratama	Desa Ganti kec. Praya timur	5	1	5	2	5	2	4	3	5	4
2	Baiq Sinar Rara Arian	Desa mantang kec Batukling	4	1	4	2	5	3	4	1	5	4
3	Melinda	Bodak	1	4	5	1	5	4	4	1	5	3
4	Alamulloh Istiqlal	Desa kumbung kec kuripan	5	3	5	2	4	2	4	3	5	3
5	Safwan Hafiz	Desa pengejek kec. Jonggat	5	2	5	2	5	2	4	2	3	4
6	M Ansori	Rancak , praya	4	1	3	3	5	4	4	3	4	4
7	Rosalina Aryanti	Desa barebali , mertak mas	5	2	4	1	5	1	1	2	5	3
8	Sahril nizam	Barebali batukliang	5	1	4	3	5	2	5	3	5	1
9	Martina Fariza	Serengat , RW 01	5	2	5	1	5	1	2	1	4	2
10	Bunga Tribuana	Praya timur , Mujur	3	3	4	4	3	1	4	2	5	1
11	Leni Junika	Danggah	4	2	5	4	3	2	5	1	5	1
12	Ahmad Irfan Mawali	Paoq Dandak, desa durian	4	1	1	1	5	1	5	1	5	1
13	Murizal	Kerembong	5	1	5	1	5	1	5	1	5	1
14	M Saipul Fahmi	Stanggor	5	1	5	1	5	2	4	3	5	4
15	Muhamad Sadri	Kopang	4	1	1	5	3	3	4	1	3	4
16	Rumawan	Bolor	1	3	5	3	5	2	5	1	5	2
17	Muhammad Mulliadi	Pen,gadang	1	4	5	4	4	5	5	1	3	4
18	Rery Ayu Rastami	Leneng	3	2	5	4	3	4	4	1	5	1
19	M Saipul Akrom	Sembalun	1	2	3	3	4	4	4	3	1	3
20	Nanang Mashud	Praya timur	3	1	3	3	1	1	3	4	1	5

P1 = Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini?

P2 = Saya berpikir akan menggunakan sistem ini lagi?

P3 = Saya merasa sistem ini rumit untuk digunakan?

P4 = Saya merasa sistem ini mudah digunakan?

P5 = Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini ?

P6 = Saya merasa fitur-fitur sistem ini berjalan dengan semestinya?

P7 = Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistemini)?

P8 = Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat?

P9 = Saya merasa sistem ini membingungkan?

P10 = Saya merasa tidak ada hambatan dalam menggunakan sistem ini?

Link Kusioner :

[https://docs.google.com/forms/d/e/1FAIpQLSfHvGM3r6dMuwyXlvpnDlqT3g9hSGTxv7WNEK-aqzIya8A/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLSfHvGM3r6dMuwyXlvpnDlqT3g9hSGTxv7WNEK-aqzIya8A/viewform?usp=sf_link)

### Lampiran 3 Dokumentasi Penggunaan Aplikasi

Testing aplikasi pemetaan daerah rawan kecelakaan



## Uji Coba aplikasi pada pengguna jalan



### Lampiran 4 Penerimaan Notifikasi Smartphone Secara Real-time

#### Penerimaan notifikasi

