

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Perancangan suatu sistem yang dapat membantu meningkatkan kinerja anggota kepolisian dalam menindaklanjuti sebuah peristiwa kejahatan atau kendala lalu lintas. Dari media cetak atau elektronik seringkali kita mendengar berbagai peristiwa yang dapat menimbulkan ketidaknyamanan dan mengusik rasa aman pada masyarakat. Oleh karena itu dibutuhkan aplikasi yang diharapkan dapat membantu meningkatkan kinerja para anggota kepolisian agar lebih efisien.

Kota Surabaya memiliki luas wilayah 33.306,30 hektar atau sekitar 374,36 km<sup>2</sup> dengan penduduk berjumlah 2,853,661 orang (BPS Kota Surabaya, 2012). Sebagai kota terbesar kedua di Indonesia, Surabaya memiliki berbagai permasalahan yang harus segera ditangani. Saat ini para anggota kepolisian dihubungi melalui nomer kontak telepon yang terdekat lokasinya dengan posisinya saat itu. Hal ini dibutuhkan agar pihak kepolisian dapat dengan cepat bertindak menuju ke lokasi perkara. Akan tetapi kelemahan dari sistem manual ini adalah harus mengetahui dengan tepat dan benar nomer kontak telepon tersebut. Maka dibutuhkan sebuah sistem yang bisa memberikan informasi lokasi tempat kejadian perkara.

Sistem manual dengan menggunakan teknologi telepon masih belum bisa menentukan dengan akurat dimana lokasi realtime tempat kejadian perkara yang akan dituju. Maka para anggota kepolisian membutuhkan sebuah sistem aplikasi yang membantu anggota dalam melacak tempat kejadian perkara dengan menggunakan GPS.

Dengan demikian berdasarkan latar belakang yang telah dijelaskan dan masalah yang dipaparkan, sistem aplikasi dibangun untuk membantu anggota kepolisian dalam meningkatkan kinerja dalam menangani masalah.

## **1.2 Rumusan Masalah**

Berdasarkan uraian latar belakang di atas, maka dapat dirumuskan bagaimana merancang sistem aplikasi pusat komando pemanggilan darurat kejadian kriminal atau kendala lalu lintas di kota Surabaya.

## **1.3 Batasan Masalah**

Pada penulisan tugas akhir ini ada beberapa hal yang menjadi batasan masalah, antara lain sebagai berikut:

- a) Jangkauan berada di wilayah hukum Kepolisian Sektor Rungkut, Surabaya.
- b) Sistem pusat komando ini dibuat dengan menggunakan bahasa pemrograman Android dan MySQL sebagai databasenya.
- c) Kejadian kriminal meliputi: miras, transaksi gelap, perjudian, pencurian, penemuan mayat, dan tawuran.
- d) Kendala lalu lintas meliputi: kemacetan atau kecelakaan pada lalu lintas.
- e) Peta dasar memanfaatkan dari Bing Maps.
- f) Aplikasi berjalan dari 2 sisi, yaitu pengguna publik (masyarakat umum) dan pengguna polisi.
- g) Fitur-fitur yang akan dibuat:
  - Menampilkan informasi kejadian kriminal atau kendala lalu lintas.
  - Memantau seluruh kejadian melalui sebuah website oleh kepolisian.
  - Pemetaan untuk user polisi.
  - Menampilkan notifikasi kepada user dalam radius 1000 m dari tempat kejadian perkara.

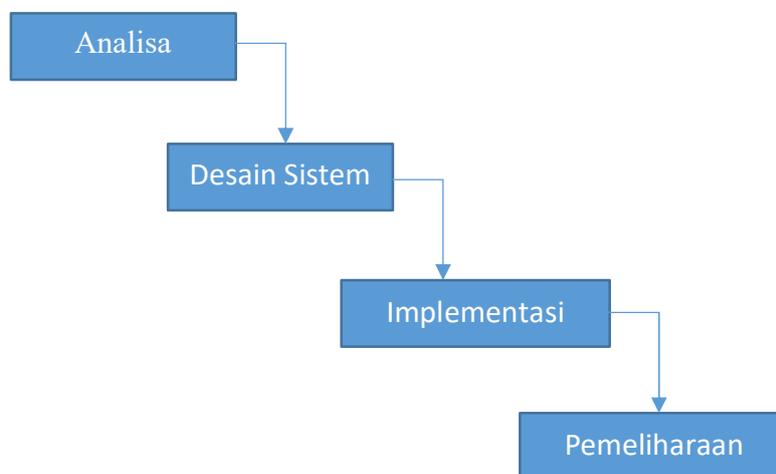
## **1.4 Tujuan Penelitian**

Penelitian ini bertujuan untuk membangun sebuah Sistem Aplikasi Pusat Komando Pemanggilan Darurat Kejadian Kriminal atau Kendala lalu lintas di Kota Surabaya.

## 1.5 Metodologi Penelitian

Penelitian ini menggunakan metode *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

(Haryanti & Irianto, 2003) Inti dari metode *waterfall* adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear. Jadi setiap tahap harus diselesaikan terlebih dahulu secara penuh sebelum diteruskan ke tahap berikutnya untuk menghindari terjadinya pengulangan tahapan. Secara garis besar metode *waterfall* mempunyai langkah - langkah sebagai berikut: Analisa Sistem, Desain Sistem, Implementasi Sistem, Pemeliharaan Sistem



Gambar 1.1 Diagram Alir Secara Waterfall

### 1. Analisa Sistem

Tahap analisa sistem dilakukan dengan cara menganalisa sistem yang ada untuk menemukan kelemahan-kelemahan sehingga dapat diusulkan perbaikan dari sistem yang ada saat ini masih manual yaitu pemanggilan manual melalui nomer telepon.

### 2. Desain Sistem

Melakukan perancangan sistem dengan membuat desain model yang digambarkan dengan *Data Flow Diagram* (DFD). Mendesain *database* dilakukan dengan cara mengidentifikasi kriteria - kriteria yang diperlukan oleh sistem.

### **3. Implementasi Sistem**

Pada bagian ini akan dilakukan pembuatan sistem dari desain yang telah dibuat, meliputi pembuatan database dan coding program. Pada bagian ini juga dilakukan pengujian sistem dan mendokumentasikan sistem dan memastikan bahwa pengguna yang terlibat dapat mengoperasikan sistem. Serta memastikan bahwa sistem berjalan dengan baik dan benar.

### **4. Pemeliharaan Sistem**

Melakukan kegiatan pemeliharaan dan pembaruan informasi terhadap data yang berada di sistem dan harus dilakukan secara rutin.

#### **1.6 Sistematika Penulisan**

Sistematika penulisan yang akan dibuat dalam tugas akhir ini disusun beberapa bab, yang dijelaskan sebagai berikut:

#### **BAB I PENDAHULUAN**

Bab ini menjelaskan permasalahan yang ada beserta solusi yang digunakan untuk menyelesaikan permasalahan tersebut. Penjelasan tersebut akan di bagi-bagi menjadi sub bab yaitu latar belakang masalah, perumusan masalah, batasan masalah, tujuan dari pembuatan tugas akhir, metodologi penelitian, dan sistematika penulisan beserta jadwal pelaksanaan.

#### **BAB II TINJAUAN PUSTAKA**

Bab ini membahas tentang teori-teori yang berhubungan dengan fakta atau kasus yang sedang dibahas. Disamping itu juga dapat disajikan mengenai berbagai asas atau pendapat yang berhubungan dan benar-benar bermanfaat sebagai bahan untuk melakukan analisis terhadap fakta atau kasus yang sedang diteliti pada bab IV.

### **BAB III LANDASAN TEORI**

Bab ini membahas tentang dasar teori yang digunakan dalam penelitian tugas akhir ini. Adapun yang dibahas pada bab ini adalah teori yang berkaitan dengan pembangunan "*Aplikasi Pusat Komando Pemanggilan Darurat Kejadian Kriminal Atau Kendala Lalu Lintas Di Kota Surabaya*"

### **BAB IV ANALISIS DAN PERANCANGAN SISTEM**

Bab ini membahas tentang flowchart sistem, analisa sistem yang dibuat, rancangan sistem, dan fitur-fitur pada sistem.

### **BAB V IMPLEMENTASI SISTEM**

Bab ini merupakan kesimpulan dari pembuatan sistem dan saran yang diambil dari pengalaman pembuatan sistem.

### **BAB VI PENGUJIAN SISTEM**

Bab ini menjelaskan tentang percobaan dan stabilitas terhadap sistem.

### **BAB VII PENUTUP**

Bab ini menjelaskan tentang hasil dari percobaan sistem secara umum dan pemberian saran untuk pengembangan selanjutnya.

- **Bagian Akhir :**
  1. Daftar Pustaka
  2. Lampiran

### 1.7 Jadwal Penelitian

Tabel 1.1 Jadwal Penelitian

NO	KEGIATAN	Waktu Pelaksanaan																											
		1				2				3				4				5				6							
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
1.	Analisa Sistem	■	■	■	■	■	■	■	■																				
2.	Desain Sistem					■	■	■	■	■	■	■	■																
3.	Implementasi Sistem									■	■	■	■	■	■	■	■	■	■	■	■								
4.	Pemeliharaan Sistem																	■	■	■	■	■	■	■	■	■	■	■	■

## BAB II

### TINJAUAN PUSTAKA

Adapun jurnal atau penelitian yang berhubungan dengan laporan tugas akhir diantara lain:

Hendra Nugraha Lengkong, Alicia A.E. Sinsuw, ST., MT., Arie S.M Lumenta, ST., MT., 2015. *Perancangan Penunjuk Rute Pada Kendaraan Pribadi Menggunakan Aplikasi Mobile GIS Berbasis Android Yang Terintegrasi Pada Google Maps*. Berkunjung ke tempat baru dalam kota maupun luar kota merupakan hal yang sangat wajar. Permasalahannya adalah terkadang beberapa orang susah mengingat kembali letak suatu tempat ataupun arah rute menuju tempat tersebut. Sehingga meningkatnya mobilitas masyarakat berbanding lurus dengan meningkatnya kebutuhan informasi berupa peta digital. Dalam penelitian ini aplikasi Penunjuk Rute pada kendaraan pribadi menggunakan aplikasi MobileGis berbasis Android yang terintegrasi pada Google Maps diharapkan dapat memenuhi kebutuhan masyarakat untuk melihat rute penunjuk arah sesuai kebutuhan pengguna (user). Dengan bantuan Global Positioning System (GPS) yang berfungsi sebagai penunjuk lokasi, Location Based Service (LBS) yang menyediakan informasi berdasarkan letak geografis perangkat mobile, melalui visualisasi Google Maps, maka aplikasi ini akan mudah digunakan. Aplikasi ini dibangun dengan pemrograman java Android menggunakan software ADT Bundle yang di dalamnya terdapat Eclipse sebagai editor bahasa pemrograman java, ADT sebagai plugin untuk Eclipse, dan SDK untuk kepentingan development aplikasi berbasis Android. Hasil akhir dari penelitian ini adalah berupa aplikasi penanda lokasi peta digital berbasis mobile GIS pada smartphone android. Aplikasi ini memiliki beberapa fitur utama seperti input data, menampilkan list data tersimpan, menampilkan rute pada peta dan membackup dan mengimpor data.

Chintya Nermelita Mandalahi, Simon Siregar, Duddy Soegiarto, 2015. *Rancang Bangun Pelacak Bus Untuk Smart City*. Sistem pelacak kendaraan berbasis GPS menggunakan komunikasi GPRS yang merupakan sebuah sistem

dimana posisi suatu kendaraan dapat diketahui secara pasti. Sistem pelacak menggunakan GPS (Global Positioning System) untuk menentukan posisi tersebut serta GPRS (General Packet Radio Service) sebagai jembatan pengiriman data dari mikrokontroler ke server. Modul GPS dan GPRS menggunakan SIMCOM SIM900. Sistem GPS menggunakan NMEA-0183 protocol untuk berkomunikasi dengan mikrokontroler ATmega 328-P. Mikrokontroler digunakan sebagai penghubung sistem GPS (menggunakan NMEA-0183 Protocol) dan GPRS melalui serial TTL (UART). Mikrokontroler bertugas mengirimkan data GPS untuk kemudian dikirimkan kepada server melalui komunikasi GPRS menggunakan protokol AT+Command untuk memerintahkan GSM mengirimkan data kedalam server. Pengiriman data tersebut menggunakan protokol HTTP dengan metode \$GET. Koordinat lokasi kendaraan yang tersimpan pada server akan ditampilkan pada halaman web yang telah dilengkapi dengan peta (menggunakan google maps) dan didalamnya disertai marker posisi kendaraan. Halaman web tersebut akan ditampilkan oleh mini computer (RADXA Board) dan akan muncul pada layar monitor. Terdapat tabel informasi memungkinkan untuk pencarian lokasi dan jam.

Muslim Amrullah, Herry Hermawan, Siska Dewi Lestari, 2016. *Membangun Sistem Navigasi Kota Palangkaraya Menggunakan Google Maps API*. Perancangan sistem navigasi kota Palangkaraya dilakukan dengan menggunakan Google Maps API untuk menampilkan peta lokasi di Palangkaraya. Google Maps merupakan suatu peta yang dapat dilihat dengan menggunakan suatu browser. Cara membuat peta untuk ditampilkan pada suatu web atau blog yaitu dengan membutuhkan pengetahuan mengenai JavaScript, serta koneksi internet yang sangat stabil. Website sistem navigasi dirancang dan dikembangkan dengan menggunakan bahasa pemrograman Javascript dan PHP dengan menggunakan database dari Google Maps API. Dengan Google Maps API kita memiliki lokasi dan halaman WEB yang memungkinkan pengguna memasukkan alamat awal dan alamat akhir untuk menampilkan navigasi kota Palangkaraya.

Arief Susanto, Ahmad Kharis, Tutik Khotimah, 2016. *Sistem Informasi Geografis Pemetaan Lahan Pertanian Dan Komoditi Hasil Panen Kabupaten Kudus*. Seiring dengan kemajuan teknologi dan informasi komputer saat ini, GIS (Geographic Sistem Informasi) teknologi sedang dikembangkan baik desktop

maupun online. Banyak pemerintahan lembaga dan perusahaan yang memanfaatkan teknologi GIS untuk mengumpulkan data pada data regional dan rekaman asetnya, seperti data tanah, wilayah data dan lain-lain. Salah satu perusahaan atau Departemen mensyaratkan bahwa Dinas Pertanian, Perikanan dan Kehutanan Kabupaten Kudus, karena tidak ada pemetaan lahan pertanian dan pengumpulan data tentang tanaman komoditas di daerah tersebut. Dengan sistem, Dinas Pertanian, Perikanan dan Kehutanan Kabupaten Kudus menjadi lebih banyak efisien, cepat dan lengkap untuk mencatat hasil panen komoditas dan memiliki data pemetaan lokal di Kabupaten Kudus, dan dengan sistem komputerisasi online, maka Departemen akan lebih mudah catat area dengan lebih cepat dan di mana saja. Metode yang digunakan untuk merancang sistem itu adalah menggunakan metode Waterfall oleh Sammerville Ian. Hasil penelitian yang akan dihasilkan oleh sistem tersebut untuk memfasilitasi Dinas Pertanian, Perikanan dan Kehutanan Kabupaten Kudus untuk mengetahui hasil Komoditas di suatu kecamatan atau desa.

Khilliyatul Lubbi, 2016. *Sistem Informasi Manajemen Warung Internet (WARNET) Berbasis Google Maps API (Studi Kasus: Kabupaten Karanganyar)*. Dinas Perhubungan Komunikasi dan Informatika Karanganyar (Dishubkominfo) sebagai salah satu dinas yang membantu urusan pemerintah memiliki banyak tugas, salah satunya adalah mengelola perizinan pendirian warnet. Dalam mengelola perizinan pendirian warnet tersebut mulai dari pengajuan rekomendasi perizinan pendirian warnet hingga keluar terbitan rekomendasi izin pendirian, Dishubkominfo masih menggunakan cara manual, sehingga semua data warnet pun tersimpan dalam Microsoft Excel, Microsoft Word, atau bahkan print out. Pengelolaan tugas dengan cara seperti itu menyebabkan ketidakefektifan dan kesulitan pada saat pengarsipan. Dengan adanya permasalahan tersebut, maka dibangun sebuah sistem untuk mengelola perizinan pendirian warnet di Kabupaten Karanganyar yang dilengkapi peta persebaran lokasi warnet dengan menggunakan Google Maps API. Dengan adanya sistem ini, pemohon hanya perlu datang membawa berkas ke Dishubkominfo ketika berkasnya sudah dipastikan lengkap dan benar oleh pihak Dishubkominfo melalui sistem. Selain itu, Sistem dapat memudahkan tugas mulai dari pengajuan rekomendasi warnet oleh pemohon hingga keluar terbitan rekomendasi izin pendirian dari Dishubkominfo. Sistem

yang dibuat diuji dengan menggunakan dua metode, yaitu metode blackbox dan menggunakan tools GT Metrix. Dengan menggunakan metode blackbox, sistem memiliki tingkat kesuksesan (fungsional yang diterima) dengan kisaran 98%, dan tidak sukses sekitar 2 %. Sedangkan pengujian menggunakan GT Metrix menunjukkan rata-rata performa Page Speed Score sekitar 40% dan performa YSlow sekitar 78%.

Gema Ilham Muttaqien, Prof. Dr. Budi Murtiyasa, M.Kom, Dedi Gunawan, S.T.,M.T., 2016. *Aplikasi Pemetaan Rumah Kos Di Sekitar Ums Berbasis Google Map Api*. Pada umumnya mahasiswa dalam mencari rumah kos terutama mahasiswa baru menggunakan tehnik konvensional, seperti bertanya kepada teman, biro kos, ataupun bertanya langsung kepada pemilik rumah kos. Ketika mereka bertanya pada pihak tersebut pada umumnya mereka bertanya tentang fasilitas rumah kos, biaya rumahkos tiap bulan / tahun , dan jarak tempuh antara rumah kos dengan tempat kuliah. Sistem yang dibangun nantinya menggunakan google map API java script sebagai aplikasi pemetaan. Ditambahkan bahasa pemrograman PHP dan data base MySQL untuk menampung data-data beserta detail dan lokasi tersebut. Aplikasi tersebut nantinya dapat diakses melalui browser yang ada di komputer maupun browser yang ada di smart phone. Adanya aplikasi pemetaan menggunakan google map API yang dapat membantu mahasiswa dalam menentukan pilihan rumah kos. Aplikasi pemetaan ini dilengkapi dengan gambar disertai keterangan dari rumah-rumah kos tersebut, sehingga menunjang keputusan mahasiswa tersebut..

Adityo Suryo Nugroho, 2017. *Aplikasi Stasiun Pemberhentian Kereta Api Prameks Menggunakan Google Map Berbasis Android*. Transportasi umum di era ini sangat dibutuhkan oleh masyarakat karena tuntutan mobilitas yang tinggi, karena itu dibutuhkan transportasi umum yang cepat dan efisien. Dalam hal ini Kereta api Prameks adalah salah satu pilihan transportasi umum bagi masyarakat sekitar Yogyakarta sebagai moda transportasi yang bebas macet dan murah. Sehingga kereta api Prameks sangat populer dan banyak penumpang baru setiap harinya. Untuk penumpang yang sudah sering naik kereta api Prameks sangat mudah mengenali di mana stasiun pemberhentian kereta api Prameks. Namun untuk pelanggan baru banyak yang belum mengetahui di mana stasiun pemberhentian

kereta api Prameks. Untuk membantu penumpang baru ini maka perlu sebuah aplikasi di perangkat mobile yang dapat memberikan informasi stasiun pemberhentian kereta api Prameks dan pemberitahuan jika sudah sampai di stasiun tujuan penumpang. Penumpang tinggal menggunakan aplikasi di perangkat smartphonenya untuk melihat di mana stasiun pemberhentian kereta api Prameks berupa peta dan penanda stasiun tujuan. Terdapat juga informasi jarak dan perkiraan waktu tempuh. Jika waktu tempuh dirasa masih lama, penumpang bisa menyimpan smartphonenya dan nanti jika sudah sampai di stasiun tujuan maka akan ada alarm berbunyi untuk memberitahukan bahwa kereta sudah sampai.

Denny Setia Putra, 2017. *Sistem Informasi Geografis Tempat Wisata Edukasi di DKI Jakarta Berbasis Android*. Jakarta merupakan salah satu tujuan pariwisata di Indonesia. Berbagai tempat wisata edukasi dapat ditemui di DKI Jakarta seperti Monas, TMII, Kebun Binatang Ragunan, Planetarium, dan lain-lain. Berbagai tempat tersebut sangat bermanfaat untuk menambah wawasan bagi para pengunjung. Umumnya pengunjung tempat wisata mencari informasi melalui internet dengan smartphone mereka. Banyak diantaranya yang membutuhkan informasi tentang tempat wisata edukasi di DKI Jakarta sebelum berkunjung. Penelitian ini nantinya akan dapat membantu masyarakat dalam mencari alamat dan lokasi geografis tempat wisata edukasi di DKI Jakarta. Dalam mengembangkan sistem informasi geografis ini peneliti menggunakan bahasa pemrograman Java dan XML, dimana Java digunakan untuk membangun sistemnya dan XML digunakan untuk membangun user interface atau tampilan antar mukanya, sedangkan basis data yang digunakan yaitu MySQL. Metodologi pengembangan sistem yang digunakan adalah metodologi Waterfall. Aplikasi Sistem Informasi Geografis berbasis Android ini merupakan suatu aplikasi pembantu yang dapat diakses oleh user untuk mendapatkan informasi geografis tentang tempat wisata edukasi di DKI Jakarta. Dengan menggunakan sistem informasi geografis dengan bahasa pemrograman Java dan MySQL dalam pengolahan data dan penyajian informasi, diharapkan dapat menghasilkan informasi yang lebih cepat, tepat waktu dan akurat dibanding dengan sistem yang lainnya.

Alfien Rahmenda, Moehammad Awaluddin, Arief Laila Nugraha, 2017. *Pembuatan Aplikasi Sebaran Lokasi Kos Berbasis WebGIS Menggunakan Google*

*Map API (Studi Kasus: Area Kampus Universitas Diponegoro)*. Tempat Kos merupakan sarana tempat tinggal yang sangat penting bagi mahasiswa pendatang atau perantau yang berasal dari lain kota, provinsi maupun negara. Tempat kos selama ini di cari oleh mahasiswa dengan cara manual yaitu bertanya kepada beberapa orang di sekitar. Di area kampus Universitas Diponegoro Semarang data mengenai tempat indekost belum tersimpan dengan baik, kalau pun ada itu masih belum terkini serta belum berbentuk pelayanan umum, dengan memanfaatkan sistem informasi geografis berbasis web atau yang biasa disebut WebGIS ini mahasiswa dapat mengetahui informasi mengenai kos. Untuk mengetahui informasi mengenai tempat kos di sekitar kampus Universitas Diponegoro, diperlukan data spasial berupa koordinat lokasi tempat kos serta data non spasial yang berisi informasi mengenai tempat kos seperti nama kos, alamat, jumlah kamar, serta fasilitas-fasilitas yang di sediakan oleh pemilik tempat kos, kemudian dalam pembuatannya website ini menggunakan struktur website HTML, bahasa pemrograman (Javascript, dan PHP), MySQL sebagai pembuat database, serta menggunakan peta dasar Google Map. Hasil dari penelitian ini adalah sebuah aplikasi WebGIS yang berisi data informasi tempat kos serta website dengan alamat [tembalangkost.com](http://tembalangkost.com) yang berisi fitur-fitur berupa daftar kos beserta informasinya, fitur pencarian kos, fitur peta kos dan jalur navigasi menuju tempat kos dan fitur download peta persebaran kos yang telah di integrasi dengan Google Map API sehingga dapat dilihat oleh semua orang. Setelah dilakukan pengujian usability pada web tersebut, didapatkan hasil untuk komponen efektivitas sebesar 3,25, komponen kemudahan pengguna sebesar 3,415, dan komponen kepuasan pengguna 3,386 dari rentang skala 1-5 sehingga mendapatkan kriteria “cukup baik”.

Tri Afiebbawa Exactanaya, Arief Laila Nugraha, Andri Suprayogi, 2018. *Desain Pengembangan Aplikasi Sebaran Pendidikan Berbasis WebGIS Di Kecamatan Demak Kabupaten Demak*. Informasi pendidikan adalah sebuah jasa yang menginformasikan tentang pendidikan umum dan keagamaan. Tempat pendidikan selama ini dicari oleh orang dengan cara bertanya kepada beberapa orang yang pernah bersekolah atau masih bersekolah di lokasi tersebut. Bahkan tidak jarang pula ada informasi yang disajikan di internet. Namun, informasi tentang pendidikan ini biasanya hanya menyajikan data-data sekolah umum. Maka

dari itu hal utama yang melatar belakangi pembuatan web ini adalah perlunya informasi pendidikan yang bertujuan untuk memudahkan seseorang untuk mencari informasi pendidikan keagamaan dan umum. Penelitian ini berupa aplikasi Sistem Informasi Geografis (SIG) tentang tempat pendidikan berbasis web dengan wilayah penelitian di area kecamatan Demak. WebGIS digunakan karena dalam penyampaian dan tampilan sistem informasi geografis lebih menarik serta mempresentasikan kondisi sebenarnya. Aplikasi ini dibuat menggunakan struktur website HTML, bahasa pemrograman (javascript dan PHP), MySQL sebagai pembuat database, serta menggunakan peta dasar dari Google Maps API. Hasil penelitian ini berupa aplikasi WebGIS informasi pendidikan di area kecamatan Demak yang menggunakan Google Map API sebagai penyedia peta gratis yang diintegrasikan ke dalam web. Data sekolah yang ditampilkan dalam web sejumlah 237 sekolah, yang terdiri dari sekolah umum dan keagamaan. Hasil uji usability efektifitas aplikasi “cukup efektif”, kemudahan pengguna “cukup mudah”, dan kepuasan pengguna “Cukup puas”.

Berdasarkan dengan penelitian sebelumnya dari uraian jurnal atau penelitian tersebut, persamaan dengan sistem yang akan dibuat adalah sama sama merupakan sebuah sistem informasi berbasis web atau android. Sedangkan perbedaan dengan sistem yang akan dibuat adalah dengan menggunakan Bing Maps API. Dan sistem ini nanti akan dibuat dalam bentuk *mobile* yang dimaksudkan untuk mempermudah masyarakat dalam melaporkan suatu kejadian dan membantu anggota polisi dalam pelacakan lokasi tempat kejadian perkara dan dalam bentuk *web* yang dimaksudkan untuk pengelolaan user dan data.

Tabel 2.1 Perbandingan Tinjauan Pustaka

No.	Nama Peneliti	Judul Penelitian	Fokus Penelitian
1.	Hendra Nugraha Lengkong, Alicia A.E. Sinsuw, ST., MT., Arie S.M Lumenta, ST., MT. (2015)	Perancangan Penunjuk Rute Pada Kendaraan Pribadi Menggunakan Aplikasi Mobile GIS Berbasis Android Yang Terintegrasi Pada Google Maps	<p>Permasalahan: Terkadang beberapa orang susah mengingat kembali letak suatu tempat ataupun arah rute menuju tempat tersebut.</p> <p>Menggunakan: Google Map API</p> <p>Hasil Akhir: Aplikasi penanda lokasi peta digital berbasis mobile GIS pada smartphone android. Aplikasi ini memiliki beberapa fitur utama seperti input data, menampilkan list data tersimpan, menampilkan rute pada peta, dan membackup dan mengimpor data.</p>
2.	Chintya Nermelita Mandalahi, Simon Siregar, Duddy Soegiarto (2015)	Rancang Bangun Pelacak Bus Untuk Smart City.	<p>Permasalahan: Orang-orang lebih memilih sarana transportasi bus karena lebih praktis dan ekonomis. Karena hal tersebut, halte bus atau sering disebut terminal bus kini menjadi ramai dan penuh dengan orang yang berdesak-desakan untuk menggunakan transportasi ini. Salah satu akibat antrian tersebut karena masyarakat tidak tahu secara pasti kapan kedatangan bus, sehingga orang-orang harus datang lebih dahulu untuk menunggu bus yang akan datang. Keadaan tersebut dirasa kurang efektif dan dari sistem transportasi bus belum memberikan sistem yang informatif kepada masyarakat.</p> <p>Menggunakan: Google Map API</p> <p>Hasil Akhir: Sistem pelacak bus yang informatif berbasis web dengan memanfaatkan GPS dan GPRS.</p>

3.	Muslim Amrullah, Herry Hermawan, Siska Dewi Lestari (2016)	Membangun Sistem Navigasi Kota Palangkaraya Menggunakan Google Maps API	<p>Permasalahan: Palangkaraya merupakan pusat bisnis, perdagangan, industry dan pendidikan di Kalimantan Tengah. Salah satu cara untuk penyajian informasi daerah di kota Palangkaraya melalui visualisasi dalam bentuk data dan atau informasi yang dikaitkan dengan kondisi geografis suatu wilayah.</p> <p>Menggunakan: Google Map API</p> <p>Hasil Akhir: Aplikasi berbasis <i>website</i> Sistem Navigasi Kota Palangkaraya yang menyajikan informasi tentang rute-rute perjalanan dari inputan alamat awal dan alamat akhir, pencarian tempat dan petunjuk arah dan sarana untuk mempermudah masyarakat mendapatkan informasi tentang kota Palangkaraya dengan mudah dan cepat.</p>
4.	Arief Susanto, Ahmad Kharis, Tutik Khotimah (2016)	Sistem Informasi Geografis Pemetaan Lahan Pertanian Dan Komoditi Hasil Panen Kabupaten Kudus	<p>Permasalahan: Salah satu perusahaan atau Departemen mensyaratkan bahwa Dinas Pertanian, Perikanan dan Kehutanan Kabupaten Kudus, karena tidak ada pemetaan lahan pertanian dan pengumpulan data tentang tanaman komoditas di daerah tersebut.</p> <p>Menggunakan: Google Map API</p> <p>Hasil Akhir: Hasil penelitian yang akan dihasilkan oleh sistem tersebut untuk memfasilitasi Dinas Pertanian, Perikanan dan Kehutanan Kabupaten Kudus untuk mengetahui hasil Komoditas di suatu kecamatan atau desa.</p>
5.	Khiliyatul Lubbi (2016)	Sistem Informasi Manajemen Warung Internet (WARNET) Berbasis Google Maps API (Studi Kasus : Kabupaten Karanganyar)	<p>Permasalahan: Dishubkominfo masih menggunakan cara manual dalam mengelola perizinan pendirian warnet, sehingga semua data warnet pun tersimpan dalam Microsoft Excel, Microsoft Word, atau bahkan print out.</p>

			<p>Pengelolaan tugas dengan cara seperti itu menyebabkan ketidakefektifan dan kesulitan pada saat pengarsipan.</p> <p>Menggunakan: Google Map API</p> <p>Hasil Akhir: Sistem untuk mengelola perizinan pendirian warnet di Kabupaten Karanganyar yang dilengkapi peta persebaran lokasi warnet dengan menggunakan Google Maps API. Dengan adanya sistem ini, pemohon hanya perlu datang membawa berkas ke Dishubkominfo ketika berkasnya sudah dipastikan lengkap dan benar oleh pihak Dishubkominfo melalui sistem. Selain itu, Sistem dapat memudahkan tugas mulai dari pengajuan rekomendasi warnet oleh pemohon hingga keluar terbitan rekomendasi izin pendirian dari Dishubkominfo.</p>
6.	Gema Ilham Muttaqien, Prof. Dr. Budi Murtiyasa, M.Kom, Dedi Gunawan, S.T.,M.T. (2016)	Aplikasi Pemetaan Rumah Kos Di Sekitar Ums Berbasis Google Map Api	<p>Permasalahan: Pada umumnya mahasiswa dalam mencari rumah kos terutama mahasiswa baru menggunakan teknik konvensional, seperti bertanya kepada teman, biro kos, ataupun bertanya langsung kepada pemilik rumah kos.</p> <p>Menggunakan: Google Map API</p> <p>Hasil Akhir: Aplikasi dapat diakses melalui browser yang ada di komputer maupun browser yang ada di smartphone. Adanya aplikasi pemetaan menggunakan google map API yang dapat membantu mahasiswa dalam menentukan pilihan rumah kos. Aplikasi pemetaan ini dilengkapi dengan gambar disertai keterangan dari rumahrumah kos tersebut, sehingga menunjang keputusan mahasiswa tersebut.</p>

7.	Adityo Suryo Nugroho (2017)	Aplikasi Stasiun Pemberhentian Kereta Api Prameks Menggunakan Google Map Berbasis Android	<p>Permasalahan: Untuk penumpang yang sudah sering naik kereta api Prameks sangat mudah mengenali di mana stasiun pemberhentian kereta api Prameks. Namun untuk pelanggan baru banyak yang belum mengetahui di mana stasiun pemberhentian kereta api Prameks.</p> <p>Menggunakan: Google Map API</p> <p>Hasil Akhir: Aplikasi mobile yang dapat memberikan informasi stasiun pemberhentian kereta api Prameks dan pemberitahuan jika sudah sampai di stasiun tujuan penumpang.</p>
8.	Denny Setia Putra (2017)	Sistem Informasi Geografis Tempat Wisata Edukasi di DKI Jakarta Berbasis Android	<p>Permasalahan: Umumnya pengunjung tempat wisata mencari informasi melalui internet dengan smartphone mereka. Banyak diantaranya yang membutuhkan informasi tentang tempat wisata edukasi di DKI Jakarta sebelum berkunjung.</p> <p>Menggunakan: Google Map API</p> <p>Hasil Akhir: Aplikasi Sistem Informasi Geografis berbasis Android ini merupakan suatu aplikasi pembantu yang dapat diakses oleh user untuk mendapatkan informasi geografis tentang tempat wisata edukasi di DKI Jakarta.</p>
9.	Alfien Rahmenda, Moehammad Awaluddin, Arief Laila Nugraha (2017)	Pembuatan Aplikasi Sebaran Lokasi Kos Berbasis WebGIS Menggunakan Google Map API (Studi Kasus : Area Kampus Universitas Diponegoro)	<p>Permasalahan: Tempat kos selama ini di cari oleh mahasiswa dengan cara manual yaitu bertanya kepada beberapa orang di sekitar. Di area kampus Universitas Diponegoro Semarang data mengenai tempat indekost belum tersimpan dengan baik, kalau pun ada itu masih belum terkini serta belum berbentuk pelayanan umum.</p> <p>Menggunakan: Google Map API</p> <p>Hasil Akhir: Hasil dari penelitian ini adalah sebuah aplikasi WebGIS yang berisi data informasi tempat kos serta website dengan alamat <a href="http://tembalangkost.com">tembalangkost.com</a> yang</p>

10.	Tri Afiebbawa Exactanaya, Arief Laila Nugraha, Andri Suprayogi (2018)	Desain Pengembangan Aplikasi Sebaran Pendidikan Berbasis WebGIS Di Kecamatan Demak Kabupaten Demak	<p>berisi fitur-fitur berupa daftar kos beserta informasinya, fitur pencarian kos, fitur peta kos dan jalur navigasi menuju tempat kos dan fitur download peta persebaran kos yang telah di integrasi dengan Google Map API sehingga dapat dilihat oleh semua orang.</p> <p>Permasalahan: Tempat pendidikan selama ini dicari oleh orang dengan cara bertanya kepada beberapa orang yang pernah bersekolah atau masih bersekolah di lokasi tersebut. Bahkan tidak jarang pula ada informasi yang disajikan di internet. Namun, informasi tentang pendidikan ini biasanya hanya menyajikan data-data sekolah umum.</p> <p>Menggunakan: Google Map API</p> <p>Hasil Akhir: Hasil dari penelitian ini adalah sebuah aplikasi WebGIS informasi pendidikan di area kecamatan Demak yang menggunakan Google Map API sebagai penyedia peta gratis yang diintegrasikan ke dalam web. Data sekolah yang ditampilkan dalam web sejumlah 237 sekolah, yang terdiri dari sekolah umum dan keagamaan.</p>
-----	---	--	--

## BAB III

### LANDASAN TEORI

#### 3.1 Definisi Aplikasi Mobile

Aplikasi mobile adalah aplikasi yang bisa digunakan secara berpindah-pindah tempat (*mobile*) sehingga bisa menikmati aplikasi kapan pun dan di mana pun pada *smartphone* sambil menjalankan aktifitas rutin.

Android adalah salah satu platform sistem operasi yang digemari masyarakat karena sifatnya yang open source sehingga memungkinkan pengguna untuk melakukan pengembangan. Android merupakan generasi baru platform mobile berbasis linux yang mencakup sistem operasi, middleware, dan aplikasi (Safaat, 2012).

Arsitektur Android terdiri dari bagian-bagian seperti berikut:

- a. *Applications* dan *Widgets*: layer (lapisan) dimana pengguna hanya berhubungan dengan aplikasi saja.
- b. *Applications Framework*: lapisan dimana para pengembang melakukan pembuatan aplikasi yang akan dijalankan di sistem operasi Android dengan komponen-komponennya meliputi *views*, *contents provider*, *resource manager*, *notification manager*, *activity manager*.
- c. *Libraries*: lapisan dimana fitur-fitur android berada yang berada diatas kernel meliputi library C/C++ inti seperti Libc dan SSL.
- d. *Android Run Time*: lapisan yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan implementasi Linux yang terbagi menjadi dua bagian yaitu *Core Libraries* dan *Dalvik virtual Machine*.
- e. *Linux Kernel*: Layer yang berisi file-file system untuk mengatur *processing*, *memory*, *resource*, *driver*, dan sistem operasi android lainnya.

Sistem operasi yang mendasari Android dilisensikan dibawah GNU, GPLv2 (*General Public License verse 2*) yang sering dikenal dengan istilah *copyleft*. Pendistribusian Android dibawah lisensi dari Apache Software yang memungkinkan untuk distribusi kedua dan seterusnya.

### **3.2 Pusat Komando**

Komando adalah kewenangan untuk memberikan perintah, mengkoordinasikan, mengendalikan, memantau dan mengevaluasi kegiatan tanggap darurat (BNPB, 2013).

Pusat Komando merupakan sistem yang mengumpulkan data, memproses data untuk menghasilkan informasi yang berguna bagi pihak-pihak terkait dalam rangka penanggulangan kejadian kriminal dan kendala lalu lintas pada masa tanggap darurat. Pada penelitian ini pelapor memberikan informasi mengenai suatu peristiwa darurat melalui aplikasi dan mengirimkannya, polisi yang menerima informasi mengenai suatu peristiwa yang telah dilaporkan oleh pelapor akan melakukan penelusuran terhadap informasi yang berada di dalam laporan peristiwa tersebut dan segera mengevaluasi kegiatan tanggap darurat.

### **3.3 Definisi Kriminal dan Kendala Lalu Lintas**

(KBBI, 2012) Kriminal adalah kejahatan atau pelanggaran hukum yang dapat dihukum menurut undang-undang. Contoh kejadian kriminal diantaranya adalah: transaksi gelap, pembunuhan, judi, dan lain sebagainya.

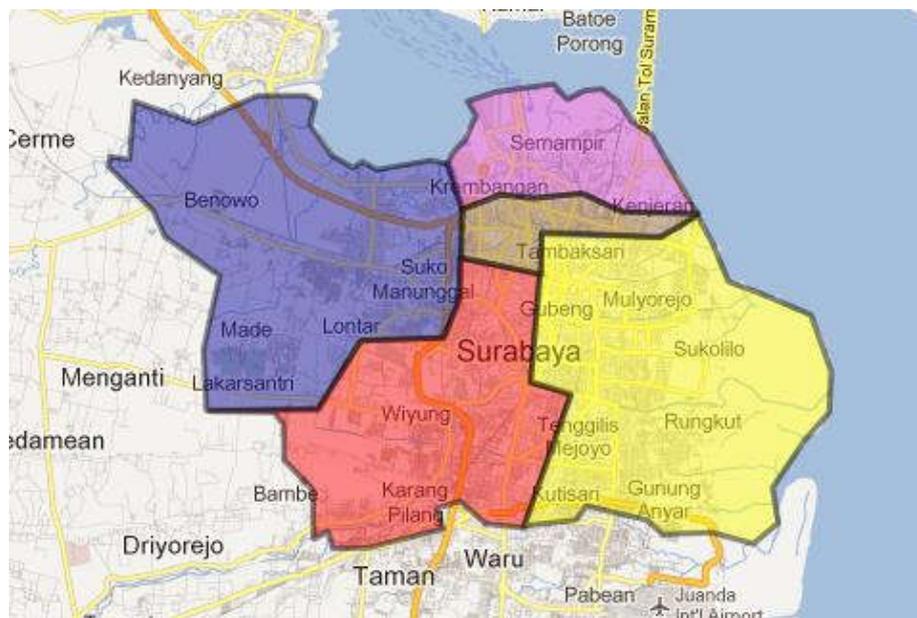
Kendala lalu lintas adalah situasi atau keadaan tersendatnya arus lalu lintas yang ditandai dengan menurunnya kecepatan perjalanan dari kecepatan yang seharusnya atau bahkan terhentinya lalu lintas yang dapat disebabkan oleh beberapa hal, diantaranya ialah banyaknya jumlah lalu lintas kendaraan yang melebihi kapasitas jalan dan terjadinya kecelakaan.

Adapun aktifitas kriminal pada penelitian ini adalah ditemukannya aktifitas perjudian, miras, transaksi gelap (produk pasar gelap / illegal), pencurian, penemuan mayat, dan tawuran (perkelahian yang meliputi banyak orang) yang terjadi di dalam jangkauan wilayah hukum kepolisian sektor Rungkut.

Adapun kendala lalu lintas pada penelitian ini adalah terjadinya kemacetan pada arus lalu lintas atau jika terjadinya peristiwa atau tragedi kecelakaan pada arus lalu lintas yang berada di dalam jangkauan wilayah hukum kepolisian sektor Rungkut.

### 3.4 Kota Surabaya

Kota Surabaya memiliki luas wilayah 33.306,30 hektar atau sekitar 374,36 km<sup>2</sup> dengan penduduk berjumlah 2,853,661 orang (BPS Kota Surabaya, 2012). Surabaya adalah sebuah kota metropolitan terbesar yang berada di dalam provinsi Jawa Timur. Sebagai kota terbesar kedua di Indonesia, Surabaya memiliki berbagai permasalahan yang harus segera ditangani diantaranya adalah rawan terjadi tindak kriminal atau terjadinya kendala lalu lintas



Gambar 3.1 Peta Surabaya

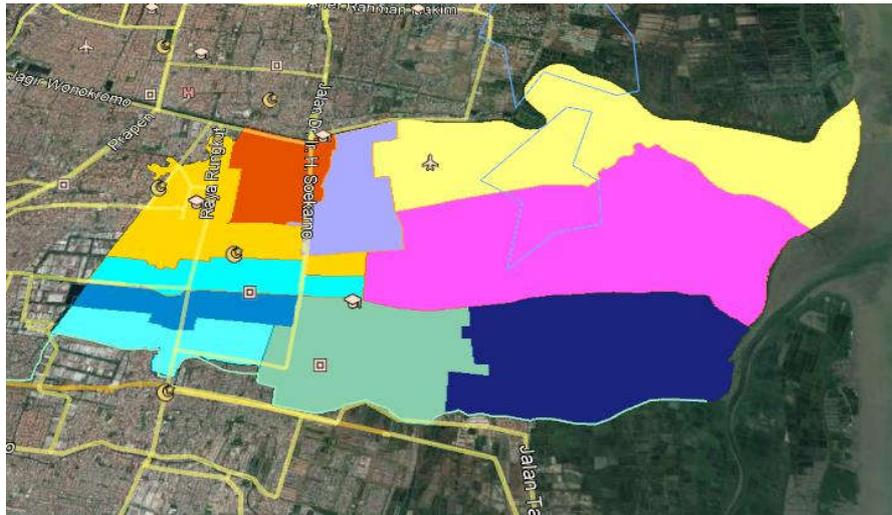
Wilayah hukum kepolisian sektor Rungkut merangkul 2 (dua) kecamatan yaitu kecamatan Rungkut dan kecamatan Gunung Anyar.

Kecamatan Rungkut memiliki 6 (enam) kelurahan, sedangkan kecamatan Gunung Anyar memiliki 4 (empat) kelurahan. Wilayah hukum kepolisian rungkut diantaranya, yaitu:

1. Kelurahan Kedungbaruk
2. Kelurahan Wonorejo
3. Kelurahan Medokanayu
4. Kelurahan Rungkut Kidul
5. Kelurahan Kali Rungkut
6. Kelurahan Penjaringansari

7. Kelurahan Gunung Anyar
8. Kelurahan Gunung Anyar Tambak
9. Kelurahan Rungkut Tengah
10. Kelurahan Rungkut Menanggal

Penampakan dari jangkauan wilayah hukum kepolisian sektor Rungkut adalah sebagai berikut:



Gambar 3.2 Wilayah Hukum Kepolisian Sektor Rungkut

## **BAB IV**

### **ANALISA DAN PERANCANGAN SISTEM**

#### **4.1 Analisa Permasalahan**

Langkah-langkah yang dilakukan dalam perancangan perangkat lunak adalah sebagai berikut:

##### **4.1.1 Analisis Masalah**

Selama ini di kepolisian sektor Rungkut banyak menggunakan sistem manual ketika adanya pelaporan, misalnya jika terdapat kecelakaan seringkali orang melapor dengan melalui teknologi telepon. Untuk pelapor sendiri akan kesulitan dalam hal melapor jika tidak mengetahui nomer telepon yang akan dituju untuk dilakukannya pelaporan. Dengan menggunakan teknologi telepon masih belum bisa menentukan informasi dengan akurat dimana lokasi tempat terjadinya perkara yang akan dituju.

Dengan adanya masalah diatas, maka dibutuhkan suatu aplikasi sistem informasi yang dapat melakukan pelaporan tanpa harus mengetahui nomer telepon untuk pelaporan yang dapat memberikan informasi lokasi dan melacak tempat kejadian perkara dengan menggunakan teknologi GPS melalui latitude dan longitude yang terdeteksi pada device yang terinstall aplikasi sistem informasi tersebut.

##### **4.1.2 Analisis Data**

Data yang digunakan dalam penulisan pada penelitian ini adalah data pemetaan jangkauan di dalam wilayah hukum kepolisian sektor Rungkut yang berupa latitude dan longitude.

#### **4.2 Fungsi Utama Perangkat Lunak**

Mengacu pada hasil kajian terhadap sistem pelaporan secara manual melalui teknologi telepon, maka program perlu mengimplementasikan fungsi-fungsi utama sebagai berikut:

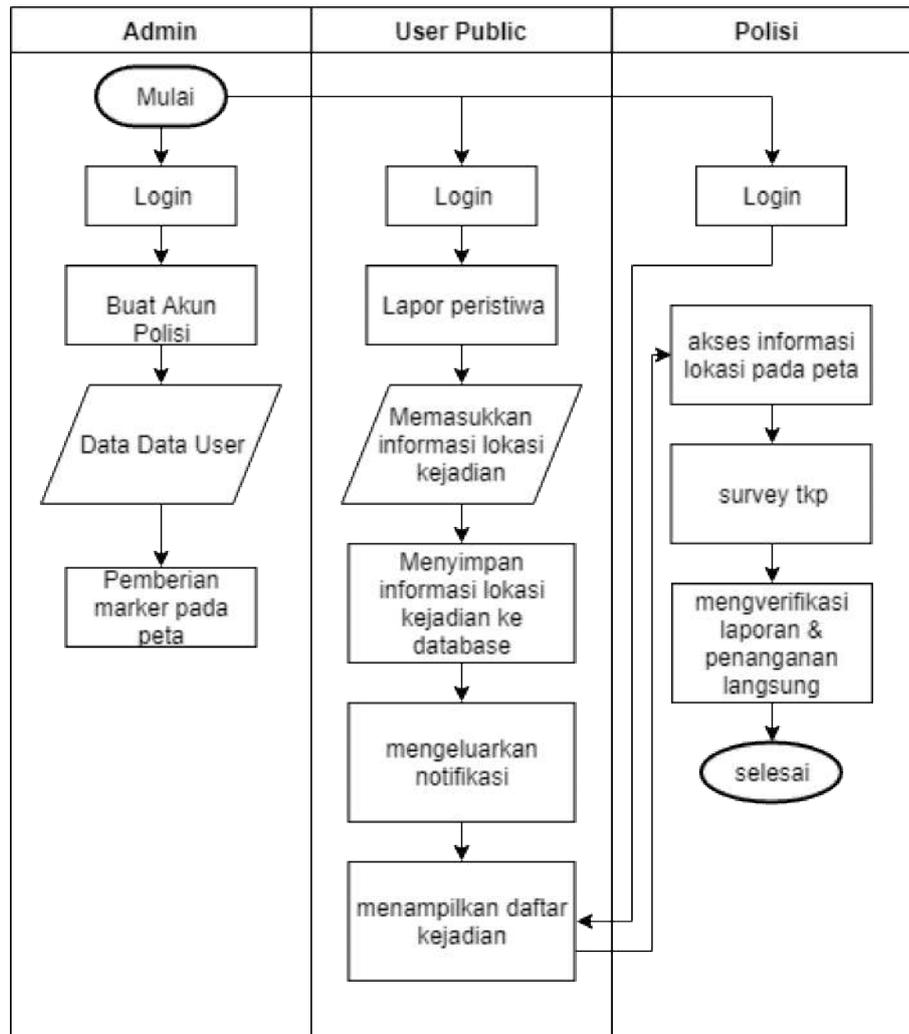
1. Program yang mampu menyimpan data secara detail dan real.
2. Program mampu menyimpan informasi yang dilaporkan oleh user yaitu masyarakat.
3. Program mampu memberikan kemudahan untuk user dalam melakukan pelaporan.
4. Program mampu memberikan informasi yang real dan detail.
5. Program mampu menampilkan pemetaan area berdasarkan kasus.
6. Program mampu memberikan rute jalan menuju lokasi peristiwa.
7. Program mampu mengirimkan notifikasi kepada user masyarakat yang berada dalam radius 1000 m dari lokasi kejadian.

### **4.3 Perancangan Sistem**

Tujuan utama dari perancangan sistem adalah memberikan gambaran perencanaan sistem yang akan dibangun atau dikembangkan, secara umum untuk memahami alur informasi dan proses dalam sistem. Berikut ini adalah tahapan-tahapan atau langkah-langkah yang akan dilakukan dalam perancangan sistem.

#### **4.3.1 Flowchart Sistem**

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. Flowchart menolong analisis dan programmer untuk memecahkan masalah kedalam segmen yang lebih kecil dan menolong menganalisis alternatif-alternatif lain dalam pengoperasian. Flowchart biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.



Gambar 4.1 Sistem Flowchart

Berikut adalah penjelasan dari Flowchart diatas:

1. Terdapat 3 (tiga) jenis user, yaitu: admin, user public (masyarakat), dan polisi.
2. User admin untuk mengelola data pada sistem informasi berbasis web, sedangkan User Public dan Polisi menjalan aplikasi berbasis *mobile*.

#### 4.4 DFD (Data Flow Diagram)

DFD (Data Flow Diagram) merupakan alat perancangan sistem yang digunakan untuk penggambaran analisis maupun rancangan sistem yang berorientasi pada alur data, yaitu suatu proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut.

#### 4.4.1 DFD Level 0

Konteks diagram merupakan gambaran secara umum dari sistem aplikasi, konteks diagram pada sistem aplikasi penelitian ini dapat dilihat pada gambar 4.2 dibawah ini:



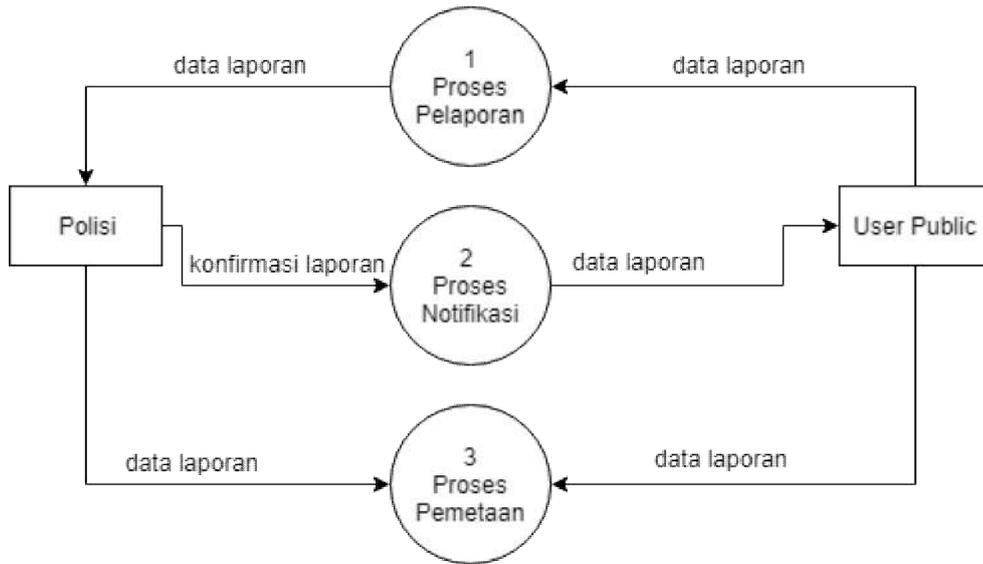
Gambar 4.2 DFD Level 0

Berikut adalah penjelasan dari DFD Level 0 diatas:

1. User Public melakukan proses input data judul laporan, gambar, isi laporan, tempat dan jenis kejadian.
2. Polisi menerima informasi dari data laporan, kemudian meninjau lokasi tempat kejadian perkara (TKP). Polisi mengonfirmasi data laporan dan memublish laporan.
3. User Public menerima data laporan yang telah di konfirmasi oleh Polisi.

#### 4.4.2 DFD Level 1

DFD Level 1 merupakan penjabaran dari data flow diagram level 0, DFD Level 1 memiliki beberapa proses yaitu proses pelaporan, proses notifikasi, proses pemetaan yang digambarkan pada gambar 4.3 dibawah ini:



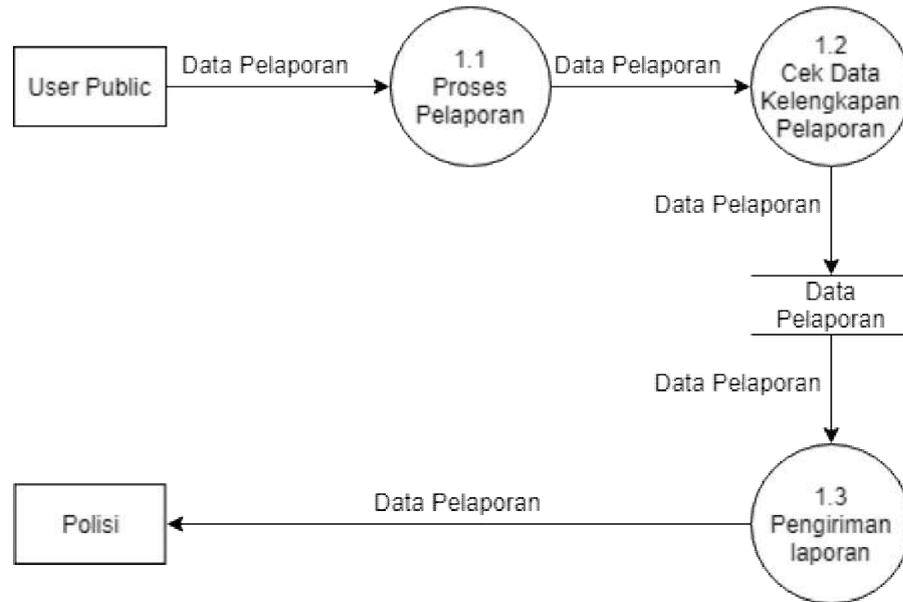
Gambar 4.3 DFD Level 1

Penjelasan dari DFD Level 1 diatas adalah sebagai berikut:

1. User Public melakukan proses submit data laporan ke dalam aplikasi, kemudian Polisi menerima data laporan yang telah dikirimkan oleh User Public.
2. Polisi melakukan peninjauan ke lokasi tempat kejadian perkara (TKP) dan melakukan proses konfirmasi dan mempublish data laporan kepada User Public. Sehingga User Public akan menerima data laporan yang telah dikonfirmasi oleh Polisi.
3. Polisi melihat pemetaan area yang rawan terjadinya kendala lalu lintas dan kriminalitas.

#### 4.4.3 DFD Level 2 Proses 1

DFD Level 2 Proses 1 merupakan penjabaran dari data flow diagram level 1 proses 1, DFD Level 2 proses 1 memiliki beberapa proses yaitu proses pelaporan, proses cek data kelengkapan laporan, dan proses tampilkan data pelaporan yang digambarkan pada gambar 4.4 dibawah ini:



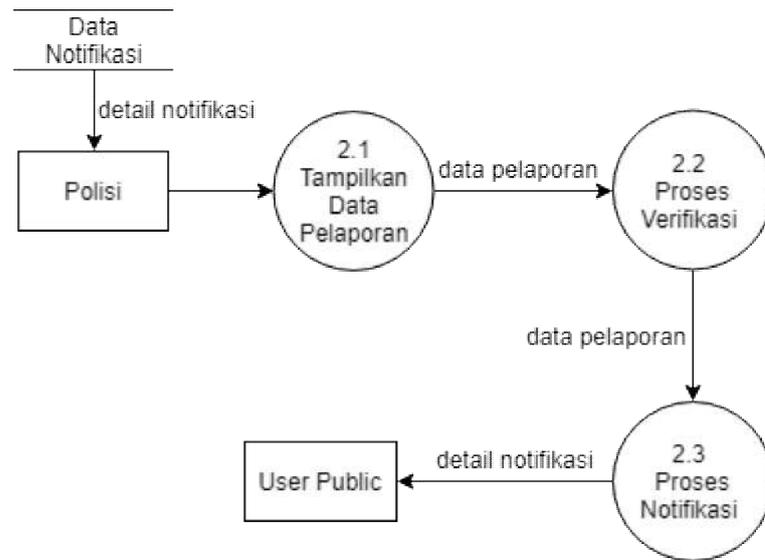
Gambar 4.4 DFD Level 2 Proses 1

Penjelasan dari DFD Level 2 Proses 1 diatas adalah sebagai berikut:

1. User Public melakukan proses pengisian data pelaporan pada form pelaporan dalam aplikasi.
2. User Public memeriksa kembali kelengkapan data data pelaporan yang diperlukan, sesuai dengan kolom form pelaporan kemudian mengirimkannya ke dalam sistem yang akan diteruskan kepada polisi berupa notifikasi.
3. Polisi mendapatkan informasi data pelaporan berupa notifikasi.

#### 4.4.4 DFD Level 2 Proses 2

DFD Level 2 Proses 2 merupakan penjabaran dari data flow diagram level 1 proses 2, DFD Level 2 proses 2 memiliki beberapa proses yaitu proses tampilan data pelaporan, proses verifikasi, dan proses notifikasi yang digambarkan pada gambar 4.4 dibawah ini:



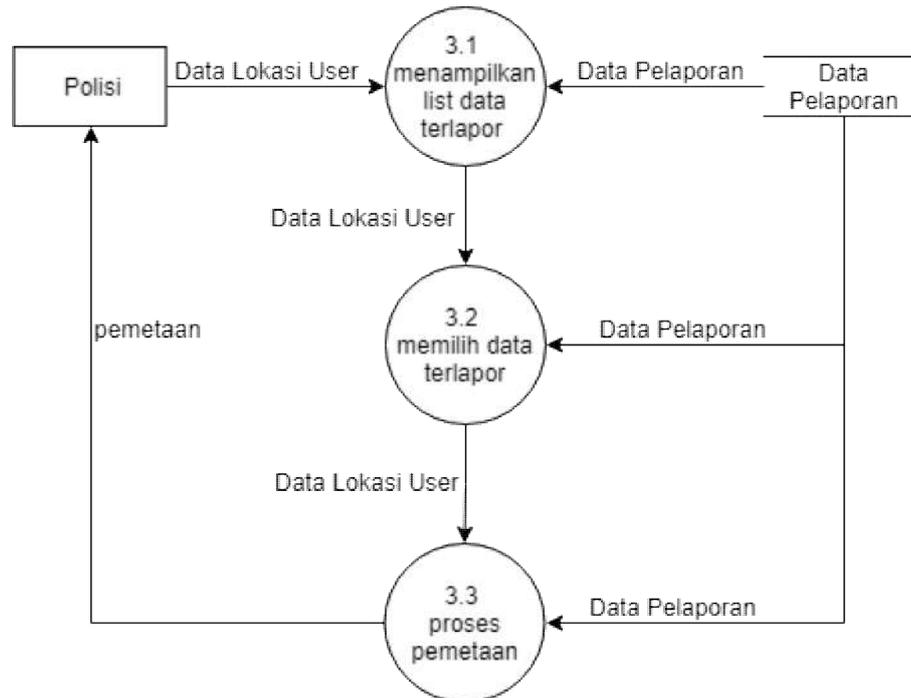
Gambar 4.5 DFD Level 2 Proses 2

Penjelasan dari DFD Level 2 Proses 2 diatas adalah sebagai berikut:

1. Polisi menerima notifikasi dari pelaporan yang dilakukan oleh user.
2. Polisi menampilkan detail dari laporan yang dikirimkan oleh user. Kemudian memberikan verifikasi jika memang benar terjadi kejadian/peristiwa yang dilaporkan oleh user
3. User Public mendapatkan notifikasi dan informasi mengenai laporan yang telah diverifikasi oleh Polisi.

#### 4.4.5 DFD Level 2 Proses 3

DFD Level 2 Proses 3 merupakan penjabaran dari data flow diagram level 1 proses 3, DFD Level 2 proses 3 memiliki beberapa proses yaitu proses menampilkan list data terlapor, proses memilih data terlapor, dan proses pemetaan yang digambarkan pada gambar 4.5 dibawah ini:



Gambar 4.6 DFD Level 2 Proses 3

Penjelasan dari DFD Level 2 Proses 3 diatas adalah sebagai berikut:

1. Polisi menampilkan list data laporan yang telah dilaporkan oleh User Public. Kemudian memilih data laporan mana yang akan ditinjau detail dari lokasi kejadian.
2. Polisi akan mendapatkan sebuah rute jalan dari lokasi user Polisi saat ini menuju lokasi kejadian berdasarkan koordinat yang tertangkap pada laporan tersebut.

#### 4.5 ERD (Entity Relationship Diagram)

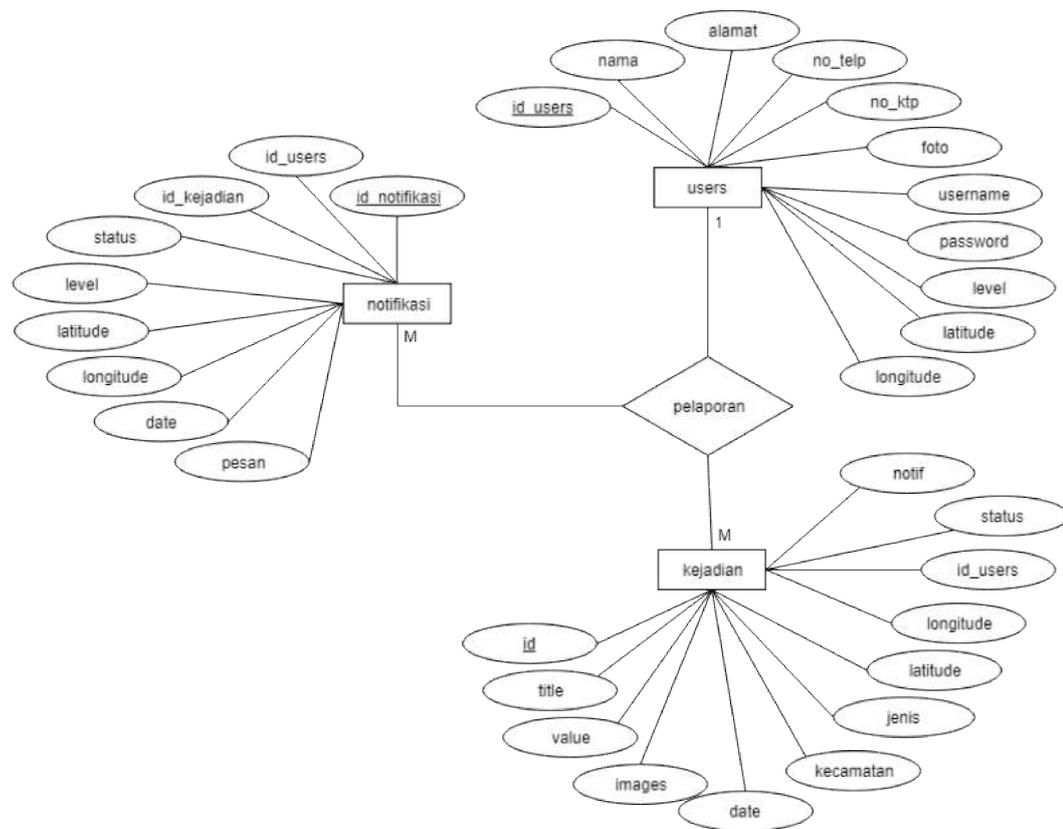
ERD (Entity Relationship Diagram) adalah representasi grafis dari sistem informasi yang menunjukkan hubungan antar tabel di dalam sebuah sistem. ERD menerangkan relasi antara atribut dimana atribut mempunyai fungsi untuk mendeskripsikan karakteristik dari entitas tersebut, isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain.

Sistem informasi ini mempunyai 3 tabel yang ada di dalam database yaitu sebagai berikut:

1. Tabel kejadian
2. Tabel notifikasi

### 3. Tabel users

Untuk lebih jelasnya dapat dilihat pada gambar 4.7.



Gambar 4.7 ERD

Berikut penjelasan dari ERD diatas:

1. Entitas users; berisi data id\_users, nama, alamat, no\_telp, no\_ktp, foto, username, password, level, latitude, dan longitude.
2. Entitas notifikasi; berisi data id\_notifikasi, id\_users, id\_kejadian, status, level, latitude, longitude, date, dan pesan.
3. Entitas kejadian; berisi data id, title, value, images, date, kecamatan, jenis, latitude, longitude, id\_users, status, dan notif.

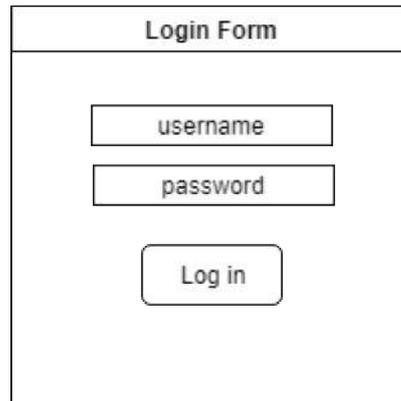
### 4.6 Rancangan Antarmuka Program

Rancangan antarmuka merupakan desain antarmuka program yang akan diimplementasikan. Rancangan yang dibuat antara lain:

## 1. Perancangan Web Monitoring Admin

### a. Login Form

Halaman login form menampilkan kolom isian username dan kolom isian password, serta tombol login. Halaman login form digunakan user Admin untuk melakukan login ke dalam web aplikasi dengan memasukkan username dan password.

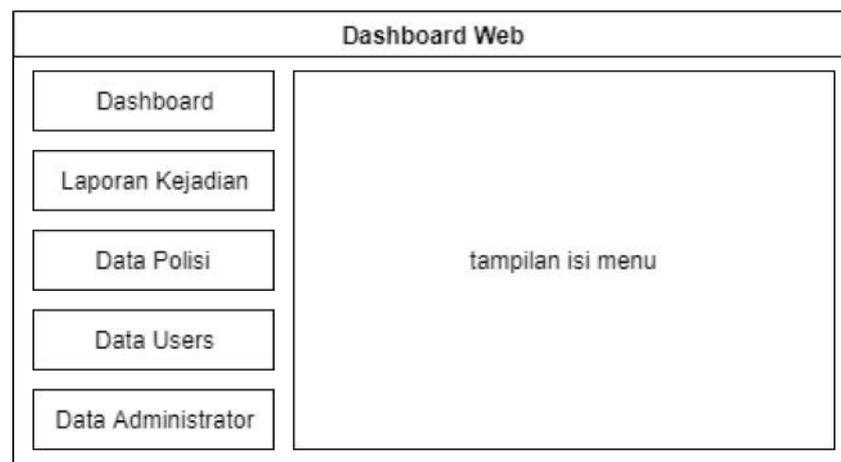


The diagram shows a rectangular box titled "Login Form". Inside the box, there are three vertically stacked elements: a rectangular input field labeled "username", another rectangular input field labeled "password", and a rounded rectangular button labeled "Log in".

Gambar 4.8 Login Form

### b. Dashboard Web Admin

Halaman dashboard web admin menampilkan beberapa menu di sisi kiri diantaranya adalah: menu Dashboard, menu Laporan Kejadian, menu Data Polisi, menu Data Users, dan menu Data Administrator. Sedangkan, pada sisi kanan menampilkan isi dari menu yang dipilih.



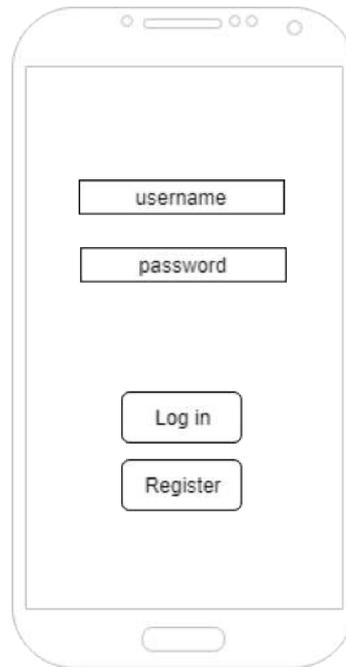
The diagram shows a rectangular box titled "Dashboard Web". On the left side, there is a vertical list of five menu items, each in a rectangular box: "Dashboard", "Laporan Kejadian", "Data Polisi", "Data Users", and "Data Administrator". To the right of this list is a large rectangular area labeled "tampilan isi menu", representing the content area for the selected menu item.

Gambar 4.9 Dashboard Web

## 2. Perancangan Mobile User Public

### a. Login Form

Halaman login form pada aplikasi *mobile* menampilkan kolom isian username dan kolom isian password. Pada halaman ini juga menampilkan 2 (dua) button, yaitu button log in dan button register.

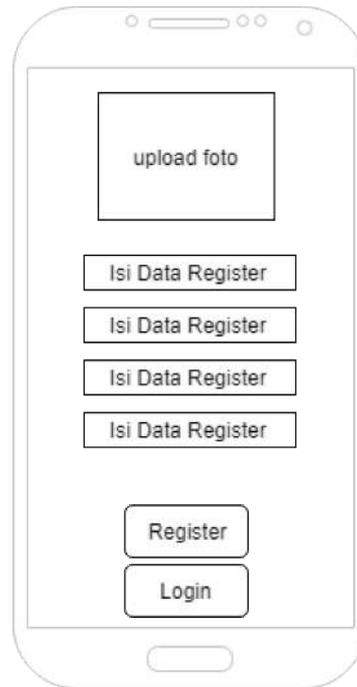


Gambar 4.10 Login Form User Public

### b. Register Form

Halaman register form menampilkan kolom kolom isian untuk pengisian data user yang akan melakukan registrasi dan dilengkapi dengan 2 (dua) button, yaitu: button register dan button login.

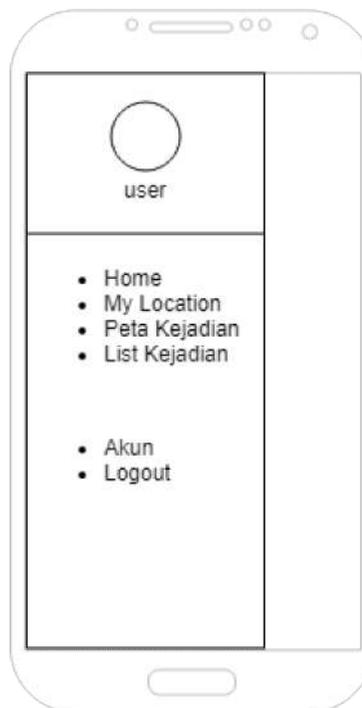
Halaman ini digunakan oleh user masyarakat untuk melakukan proses registrasi dengan cara mengirimkan beberapa data informasi dari user yang akan melakukan proses registrasi.



Gambar 4.11 Register Form User

c. Navigation Drawer

Halaman navigation drawer menampilkan foto user dan menampilkan beberapa menu yang diantaranya yaitu: menu home, menu my location, menu peta kejadian, menu list kejadian, menu akun, dan menu logout.

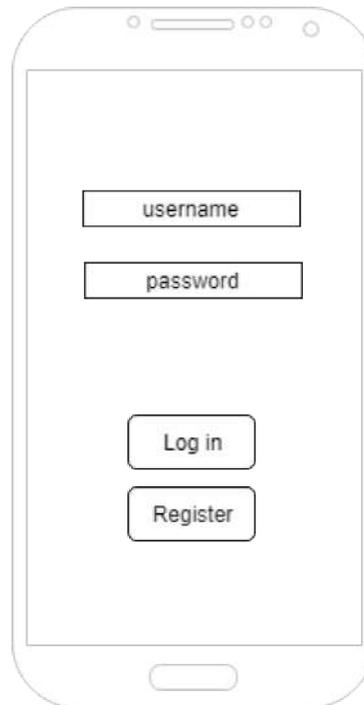


Gambar 4.12 Navigation Drawer User Public

### 3. Perancangan Mobile User Police

#### a. Login Form

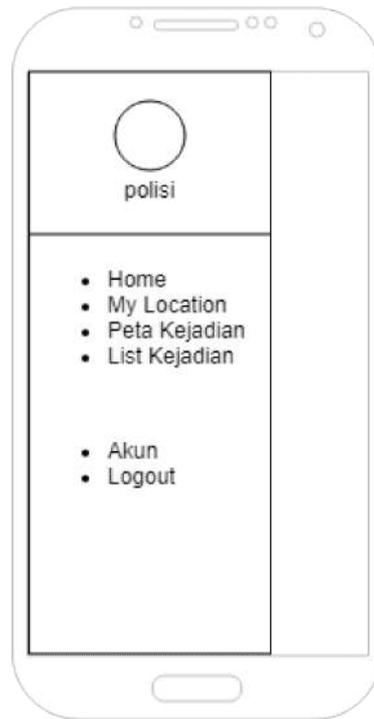
Halaman login form pada aplikasi *mobile* menampilkan kolom isian username dan kolom isian password. Pada halaman ini juga menampilkan 2 (dua) button, yaitu button log in dan button register.



Gambar 4.13 Login Form User Police

#### b. Navigation Drawer

Halaman navigation drawer menampilkan foto user dan menampilkan beberapa menu yang diantaranya yaitu: menu home, menu my location, menu peta kejadian, menu list kejadian, menu akun, dan menu logout.



Gambar 4.14 Navigation Drawer User Police

## **BAB V**

### **IMPLEMENTASI SISTEM**

#### **5.1 Spesifikasi Perangkat Keras**

Perangkat keras yang digunakan dalam pembuatan aplikasi ini adalah:

1. Intel® Core™ i5-8250U 1.60GHz - 1.80GHz
2. Memori DDR4 8 GB
3. Harddisk 1 TB
4. LCD 14.0 HD

#### **5.2 Spesifikasi Perangkat Lunak**

Perangkat lunak yang digunakan dalam pembuatan aplikasi ini adalah:

1. Android Studio
2. Sublime Text Editor
3. XAMPP
4. Chrome Browser
5. SQLyog
6. MEmu Android Emulator

#### **5.3 Batasan Implementasi Sistem**

Tahapan implementasi pada sistem ini merupakan kelanjutan dari tahapan perancangan sistem yang telah di uraikan sebelumnya pada bab IV, pada bab ini juga akan diuraikan implementasi proses sistem dari perancangan sistem yang telah dibuat sebelumnya.

#### **5.4 Implementasi Sistem**

Implementasi pada sistem ini memiliki beberapa proses yang sudah di uraikan pada bab VI, dan terbagi menjadi beberapa proses, dan form. Koding dari implementasi sistem ini cantumkan pada bab ini.

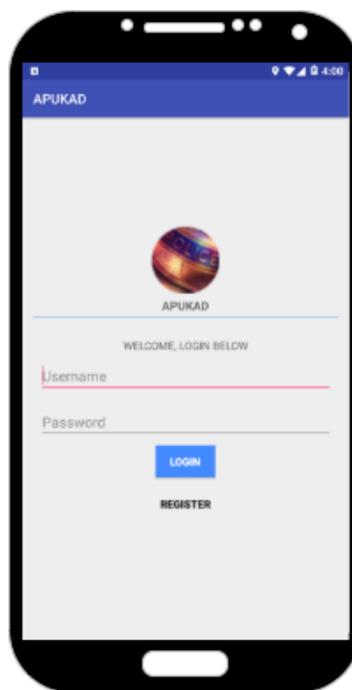
## 5.5 Implementasi Antarmuka

Implementasi antarmuka merupakan tampilan sistem secara keseluruhan yang akan dijelaskan seperti dibawah ini:

### 5.5.1 Aplikasi Polisi

#### 5.5.1.1 Halaman Login

Halaman login berfungsi untuk membedakan akun polisi satu dengan yang lainnya, jadi setiap polisi memiliki akun yang berbeda beda.



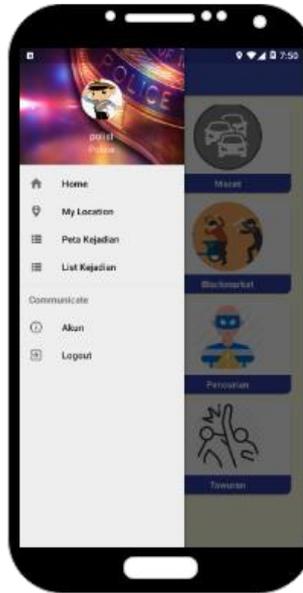
Gambar 5.1 Desain Halaman Login.

Berikut ini adalah fungsi form dari halaman login:

- A. Kolom isian Username digunakan untuk mengisi username dari user yang akan melakukan login.
- B. Kolom isian Password digunakan untuk mengisi password dari user tersebut.
- C. Tombol Login berfungsi untuk melakukan proses login setelah mengisi kolom isian Username dan kolom isian Password.
- D. Tombol Register berfungsi untuk menampilkan form Register yang digunakan untuk proses registrasi user public.

### 5.5.1.2 Navigation Drawer

Navigation Drawer berfungsi untuk menampilkan beberapa menu yang disediakan oleh aplikasi.



Gambar 5.2 Desain Navigation Drawer.

Berikut adalah fungsi dari menu-menu pada navigation drawer:

- A. Menu Home digunakan untuk menampilkan beberapa menu kejadian kriminal dan kendala lalu lintas.
- B. Menu My Location digunakan untuk menampilkan lokasi user pada saat ini.
- C. Menu Peta Kejadian digunakan untuk menampilkan pemetaan area wilayah hukum kepolisian sektor Rungkut berdasarkan jenis kejadian.
- D. Menu List Kejadian digunakan untuk menampilkan laporan yang dilaporkan oleh user public.
- E. Menu Akun digunakan untuk menampilkan data akun user.
- F. Logout berfungsi untuk mengeluarkan akun dan menampilkan halaman login.

### 5.5.1.3 Halaman Home

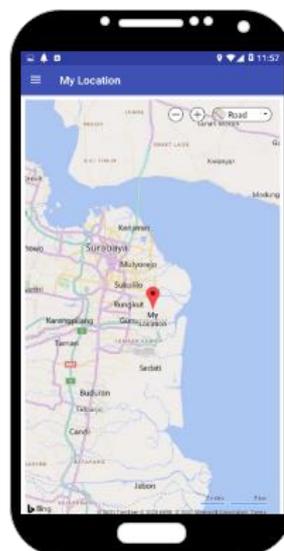
Halaman home berfungsi untuk melihat daftar kejadian yang terkonfirmasi yang dibedakan berdasarkan jenis kejadiannya.



Gambar 5.3 Desain Halaman Home

#### 5.5.1.4 Halaman My Location

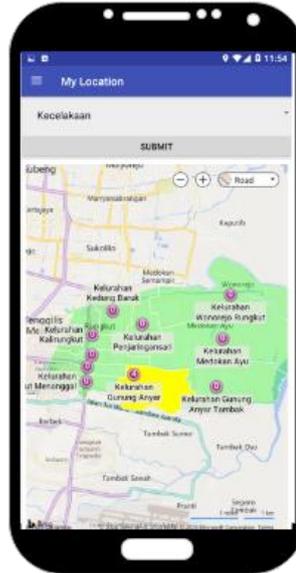
Halaman my location berfungsi untuk melihat informasi dari keberadaan posisi user pada saat ini.



Gambar 5.4 Desain Halaman My Location

#### 5.5.1.5 Halaman Peta Kejadian

Halaman peta kejadian berfungsi untuk menampilkan peta teritorial wilayah hukum kepolisian sektor Rungkut dengan dilengkapi jumlah total kejadian pada masing-masing area berdasarkan jenis kejadian.



Gambar 5.5 Desain Halaman Peta Kejadian

Pada halaman ini terdapat:

- A. Menu option untuk menampilkan peta dari masing masing jenis kejadian, yang terdiri dari: kecelakaan, macet, miras, transaksi gelap, perjudian, pencurian, penemuan mayat, dan tawuran.
- B. Tombol Submit untuk menampilkan peta dari jenis kejadian yang dipilih.

Berikut adalah source code pembuatan layer peta:

```
<?
$jns=$_GET[jenis];
include "koneksi.php";
?>
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <script type='text/javascript'>
    function GetMap() {
      var map = new Microsoft.Maps.Map('#myMap', {
        credentials: 'AtXR66yKokeuAlmaBjD7ZF7HU7ZB9xi0hswKD4-
EtpEyOD4ulFcyEvroutYlgAJ5',
        center: new Microsoft.Maps.Location(-7.326800,
112.790102),
        zoom: 13
      });

      var myGeoJson = {
        "type": "Polygon",
        "coordinates": [[
          [112.788016,-7.332108],[112.787067,-
7.331913],[112.784871,-7.331828],[112.782495,-
7.331767],[112.781695,-7.331692],[112.780896,-
```

```

7.331617],[112.780851,-7.332089],[112.78066,-7.332686],[112.780568,-
7.333174],[112.780577,-7.333646],[112.78072,-7.334161],[112.780669,-
7.335184],[112.78045,-7.335418],[112.780422,-7.335437],[112.780353,-
7.335437],[112.780329,-7.335457],[112.779158,-
7.335329],[112.778581,-7.335279],[112.778598,-
7.335434],[112.778261,-7.337068],[112.777924,-
7.338703],[112.777743,-7.339475],[112.77765,-7.339874],[112.777516,-
7.340444],[112.777441,-7.340516],[112.777455,-
7.340831],[112.777384,-7.341356],[112.777339,-
7.341765],[112.777341,-7.341955],[112.777368,-
7.342066],[112.777497,-7.342164],[112.778344,-
7.342385],[112.778856,-7.342496],[112.779324,-
7.342603],[112.779378,-7.342669],[112.779382,-
7.342789],[112.779347,-7.343278],[112.779301,-
7.343971],[112.779292,-7.344349],[112.779319,-
7.344606],[112.779377,-7.344695],[112.779493,-
7.344775],[112.779675,-7.344828],[112.77992,-7.34485],[112.781556,-
7.345165],[112.78253,-7.345354],[112.783372,-7.345517],[112.783805,-
7.345557],[112.784317,-7.345579],[112.784638,-
7.345561],[112.784834,-7.345468],[112.78502,-7.345266],[112.785198,-
7.345164],[112.78539,-7.345119],[112.785564,-7.345146],[112.785809,-
7.345195],[112.785965,-7.34519],[112.786107,-7.345101],[112.786254,-
7.344901],[112.786374,-7.344746],[112.786459,-
7.344679],[112.786633,-7.344612],[112.787404,-
7.344372],[112.788143,-7.344153],[112.788709,-
7.343993],[112.788914,-7.343971],[112.789243,-
7.344037],[112.789609,-7.344148],[112.790197,-7.34441],[112.790888,-
7.344698],[112.791182,-7.344849],[112.791468,-
7.345169],[112.791722,-7.345449],[112.792519,-7.346011],[112.79296,-
7.346207],[112.793366,-7.346393],[112.793553,-
7.346455],[112.793945,-7.346441],[112.794582,-
7.346459],[112.795046,-7.346485],[112.795416,-
7.346547],[112.795621,-7.346565],[112.795781,-7.346516],[112.79587,-
7.346378],[112.79599,-7.34608],[112.796066,-7.345924],[112.796502,-
7.345691],[112.796672,-7.34554],[112.79681,-7.345402],[112.797006,-
7.345322],[112.797377,-7.345243],[112.797424,-
7.344985],[112.797474,-7.344707],[112.797512,-
7.343856],[112.797486,-7.343214],[112.797655,-
7.341564],[112.797681,-7.340555],[112.801892,-
7.340998],[112.801958,-7.34088],[112.801996,-7.339596],[112.800422,-
7.339571],[112.800132,-7.338025],[112.800106,-
7.337502],[112.799987,-7.336532],[112.798937,-
7.336559],[112.798976,-7.335223],[112.79992,-7.335118],[112.799696,-
7.332734],[112.795431,-7.333351],[112.792016,-
7.333072],[112.789194,-7.332442],[112.788016,-7.332108]
]
];
//Load the GeoJson Module.
Microsoft.Maps.loadModule('Microsoft.Maps.GeoJson', function
() {
    //Gunung Anyar
    var id1 = Microsoft.Maps.GeoJson.read(myGeoJson, {
        polygonOptions: {
            <?
            $query = mysql_query("SELECT COUNT(kecamatan) AS jml FROM kejadian
            WHERE jenis='$jns' AND kecamatan='Gunung Anyar'");
            while ($row = mysql_fetch_array($query)){
                $jml1=$row['jml'];
                if ($row['jml']=="0") { ?>
                    fillColor: 'rgba(0, 255, 0, 0.3)',
                <? } elseif ($row['jml']<="10") { ?>
                    fillColor: 'rgba(255,255,0,3)',
                <? } elseif ($row['jml']>="11") { ?>
                    fillColor: 'rgba(255, 0, 0, 0.8)',
                <? } } ?>
        }
    });
}
);

```

```

        strokeColor: 'white',
        strokeThickness: 1
    }
    });
var pins1 = new Microsoft.Maps.Pushpin(new Microsoft.Maps.Location(-
7.334501, 112.783152), {title: 'Kelurahan Gunung Anyar', text:'<?
echo $jml1; ?>'});

        var rings = [id1, id2, id3, id4, id5, id6,
id7, id8, id9, id10, pins1, pins2, pins3, pins4, pins5, pins6,
pins7, pins8, pins9, pins10 ];
        //Add the shape to the map.
        map.entities.push(rings);
    });
}
</script>
<script type='text/javascript'
src='http://www.bing.com/api/maps/mapcontrol?callback=GetMap' async
defer></script>
</head>
<body>
    <div id="myMap"
style="position:relative;width:100%;100%;"></div>
</body>
</html>

```

Source code diatas berfungsi untuk menampilkan pemetaan teritorial dari wilayah hukum kepolisian sektor Rungkut yang dapat menampilkan seberapa rawan area tersebut dengan 3 (tiga) kondisi warna dengan parameter sebagai berikut:

1. Warna hijau, jika dalam area tersebut terdapat 0 (nihil) pelaporan.
2. Warna kuning, jika dalam area tersebut terdapat 1 hingga 10 pelaporan.
3. Warna merah, jika dalam area tersebut terdapat 11 atau lebih pelaporan.

#### 5.5.1.6 Halaman List Kejadian

Halaman list kejadian berfungsi untuk menampilkan laporan kejadian baru yang telah dilaporkan oleh *user public*.



Gambar 5.6 Desain Halaman List Kejadian

### 5.5.1.7 Halaman Akun

Halaman akun berfungsi untuk menampilkan informasi identitas dari akun pengguna.



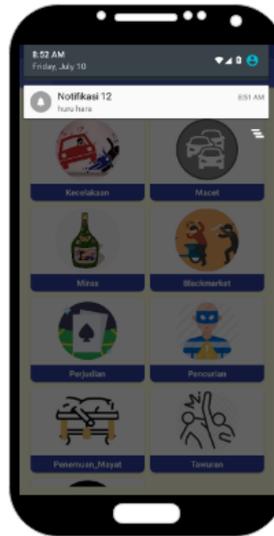
Gambar 5.7 Desain Halaman Akun

## 5.5.2 Aplikasi User Public

### 5.5.2.1 Notifikasi

Notifikasi berfungsi untuk memberikan informasi terbaru mengenai laporan dari suatu kejadian. Ketika terdapat laporan baru dari pengguna maka Polisi akan

menerima notifikasi dan yang bersangkutan akan memberikan verifikasi terhadap laporan tersebut untuk menghindari laporan fiktif. Setelah laporan terverifikasi maka dengan otomatis pengguna yang berada dalam radius 1000m dari lokasi kejadian akan menerima notifikasi laporan tersebut.



Gambar 5.8 Notifikasi.

Berikut adalah source code dari notifikasi:

#### A. Source Code Java

```
private void showCHAT(){
    JSONObject jsonObject = null;
    ArrayList<HashMap<String,String>> list = new ArrayList<HashMap<String,
String>>();
    try {
        jsonObject = new JSONObject(JSON_STRING);
        JSONArray result = jsonObject.getJSONArray(Server.TAG_JSON_ARRAY);
        mCartItemCount=result.length();
        for(int i = 0; i<result.length(); i++){
            JSONObject jo = result.getJSONObject(i);
            String Tid = jo.getString(Server.TAG_ID);
            String Tidnot = jo.getString(Server.TAG_IDNOT);
            String Tjns = jo.getString(Server.TAG_JENIS);
            String NOTIFICATION_CHANNEL_ID = "channel_androidnotif";
            Context context = this.getApplicationContext();
            NotificationManager notificationManager = (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);
            if (android.os.Build.VERSION.SDK_INT >=
android.os.Build.VERSION_CODES.O) {
                String channelName = "Android Notif Channel";
                int importance = NotificationManager.IMPORTANCE_HIGH;
                NotificationChannel mChannel = new
NotificationChannel(NOTIFICATION_CHANNEL_ID, channelName, importance);
                notificationManager.createNotificationChannel(mChannel);
            }
            if ("Police".equals(ID_LEVEL)) {
```

```

        Intent mIntent = new Intent(this, DetailLaporan.class);
        mIntent.putExtra(Server.TAG_ID2,Tid);
        mIntent.putExtra(Server.TAG_IDNOT2,Tidnot);
        PendingIntent pendingIntent = PendingIntent.getActivity(this,
i, mIntent, PendingIntent.FLAG_UPDATE_CURRENT);
        NotificationCompat.Builder builder = new
NotificationCompat.Builder(this,NOTIFICATION_CHANNEL_ID);
        builder.setContentIntent(pendingIntent)
            .setSmallIcon(R.drawable.ic_bell)

        .setLargeIcon(BitmapFactory.decodeResource(this.getResources(),
R.drawable.ic_bell))

            .setTicker("notif starting")
            .setAutoCancel(true)
            .setVibrate(new long[]{1000, 1000, 1000, 1000, 1000})
            .setLights(Color.RED, 3000, 3000)
            .setDefaults(Notification.DEFAULT_SOUND)
            .setContentTitle("Notifikasi
"+jo.getString(Server.TAG_ID))
                .setContentText(jo.getString(Server.TAG_TITLE));
        notificationManager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
        notificationManager.notify(i, builder.build());
    } else if ("Users".equals(ID_LEVEL)) {
        Intent mIntent = new Intent(this, DetailKejadian.class);
        mIntent.putExtra(Server.TAG_ID2,Tid);
        mIntent.putExtra(Server.TAG_IDNOT2,Tidnot);
        PendingIntent pendingIntent = PendingIntent.getActivity(this,
i, mIntent, PendingIntent.FLAG_UPDATE_CURRENT);
        NotificationCompat.Builder builder = new
NotificationCompat.Builder(this,NOTIFICATION_CHANNEL_ID);
        builder.setContentIntent(pendingIntent)
            .setSmallIcon(R.drawable.ic_bell)

        .setLargeIcon(BitmapFactory.decodeResource(this.getResources(),
R.drawable.ic_bell))

            .setTicker("notif starting")
            .setAutoCancel(true)
            .setVibrate(new long[]{1000, 1000, 1000, 1000, 1000})
            .setLights(Color.RED, 3000, 3000)
            .setDefaults(Notification.DEFAULT_SOUND)
            .setContentTitle("Notifikasi
"+jo.getString(Server.TAG_ID))
                .setContentText(jo.getString(Server.TAG_TITLE));
        notificationManager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
        notificationManager.notify(i, builder.build());
    }
    }
} catch (JSONException e) {
    e.printStackTrace();
}
}

@Override
protected void onResume() {
    super.onResume();
    startTimer();
}
}

```

Source code diatas berfungsi untuk menampilkan laporan kejadian pada user Polisi dan user Public yang berisi pengaturan mengenai notifikasi seperti ikon notifikasi, vibrate notifikasi, getar notifikasi, dan lain lain.

## B. Source Code PHP

```
<?php

//Mendapatkan Nilai Dari Variable ID Pegawai yang ingin ditampilkan
$lvl = $_GET['lvl'];
$idlog = $_GET['idlog'];
$lati = $_GET['lati'];
$longi = $_GET['longi'];
//Import File Koneksi Database
include 'koneksi.php';

$query1 = mysql_query("UPDATE users SET `lat` = '$lati', `lng` = '$longi' WHERE id_users = '$idlog' ");

//Mendapatkan Hasil GROUP BY pesan.id_ke WHERE users.level NOT IN ('$id')
//$row = mysql_fetch_array($query);
//$r = mysqli_query($con,$sql);

//Membuat Array Kosong
$result = array();
if ($lvl=='Police'){
    $query = mysql_query("SELECT * FROM notifikasi
    INNER JOIN kejadian ON notifikasi.id_kejadian=kejadian.id
    WHERE notifikasi.status=1 AND notifikasi.id_users='$idlog' AND
    notifikasi.level='$lvl'");
    while($row = mysql_fetch_array($query)){
        $query1 = mysql_query("UPDATE `db_apukad`.`notifikasi` SET
        `status` = '2' WHERE `notifikasi`.`id_notifikasi`
        =$row[id_notifikasi] ");
        //Memasukkan Nama dan ID kedalam Array Kosong yang telah
        dibuat
        array_push($result,array(
            "id"=>$row['id'],
            "idnot"=>$row['id_notifikasi'],
            "title"=>$row['title'],
            "jenis"=>$row['jenis']

        ));
    }
} elseif ($lvl=='Users'){
    $query = mysql_query("SELECT id_notifikasi, id_kejadian,
    latitude, longitude,pesan,
    (6367 * acos(cos(radians('$lati'))
    * cos(radians(latitude))
    * cos(radians(longitude)
    - radians('$longi')) + sin(radians('$lati'))
    * sin(radians(latitude)))) AS nilaiJarak FROM
    notifikasi WHERE date=curdate() AND status='4' AND `id_users` =
    '$idlog'");
    while($row = mysql_fetch_array($query)){
        if ($row['nilaiJarak'] <= 1) {
            $query1 = mysql_query("UPDATE
            `db_apukad`.`notifikasi` SET `status` = '2' WHERE
            `notifikasi`.`id_notifikasi` =$row[id_notifikasi] ");
        }
    }
}
}
```

```

//Memasukkan Nama dan ID kedalam Array Kosong yang
telah dibuat
array_push($result,array(
    "id"=>$row['id_kejadian'],
    "idnot"=>$row['id_notifikasi'],
    "title"=>$row['pesan'],
    "jenis"=>$row['pesan']
));
}
else {}
}
//Menampilkan Array dalam Format JSON
echo json_encode(array('result'=>$result));
?>

```

Source code diatas berfungsi untuk mengirimkan notifikasi kepada user polisi dan mengirimkan notifikasi kepada user public dari laporan yang terverifikasi oleh polisi yang berada dalam radius 1000m.

### 5.5.2.2 Halaman Pelaporan

Halaman ini berfungsi sebagai form pelaporan yang digunakan oleh user public (masyarakat) untuk melaporkan suatu kejadian.



Gambar 5.9 Halaman Pelaporan

Berikut adalah source code dari halaman pelaporan:

```

public class InsertLapor extends AppCompatActivity {
    private static final String IMAGE_DIRECTORY = "/image";
    private int GALLERY = 1, CAMERA = 2;
    private Spinner spinner1, spinner2;
    Button buttonChoose;
    FloatingActionButton buttonUpload;
    Toolbar toolbar;
    ImageView imageView;
    EditText txt_name,txt_judul;
    Bitmap bitmap, decoded;
    int PICK_IMAGE_REQUEST = 1;
    int bitmap_size = 60; // range 1 - 100
    private String myLat = "hello";
    private String myLong= "hello";
    ProgressDialog pDialog;
    Button btn_send;
    EditText txt_isi;
    Intent intent;
    int success;
    ConnectivityManager conMgr;
    private String url = Server.URL + "insert_lapor.php";
    private static final String TAG =
InsertLapor.class.getSimpleName();
    private String UPLOAD_URL =
"http://10.0.2.2/android/upload_image/upload.php";
    private static final String TAG_SUCCESS = "success";
    private static final String TAG_MESSAGE = "message";
    private String KEY_IMAGE = "image";
    String tag_json_obj = "json_obj_req";
    LocationManager locationManager;
    LocationListener locationListener;
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if(grantResults.length > 0 && grantResults[0]==
PackageManager.PERMISSION_GRANTED){
            if(ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION)==
PackageManager.PERMISSION_GRANTED) {

locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0,
0, locationListener);
                }
            }
        }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_insert);
        addItemOnSpinner2();
        buttonChoose = (Button) findViewById(R.id.buttonChoose);
        requestMultiplePermissions();
        imageView = (ImageView) findViewById(R.id.imageView);
        buttonChoose.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showFileChooser();
            }
        });
        conMgr = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
        {
            if (conMgr.getActiveNetworkInfo() != null

```

```

        && conMgr.getActiveNetworkInfo().isAvailable()
        && conMgr.getActiveNetworkInfo().isConnected() {
    } else {
        Toast.makeText(getApplicationContext(), "No Internet
Connection",
            Toast.LENGTH_LONG).show();
    }
}
btn_send = (Button) findViewById(R.id.btn_send);
txt_judul = (EditText) findViewById(R.id.txt_judul);
txt_isi = (EditText) findViewById(R.id.txt_isi);
spinner1 = (Spinner) findViewById(R.id.spinner1);
spinner2 = (Spinner) findViewById(R.id.spinner2);
locationManager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);
locationListener = new LocationListener() {
    @Override
    public void onLocationChanged(Location location) {
        Log.i("LOCATION", location.toString());

        myLat = location.getLatitude() + "";
        myLong = location.getLongitude() + "";
    }
    @Override
    public void onStatusChanged(String provider, int status,
Bundle extras) {
    }
    @Override
    public void onProviderEnabled(String provider) {
    }
    @Override
    public void onProviderDisabled(String provider) {
    }
};

if(ContextCompat.checkSelfPermission(this,Manifest.permission.ACCESS_FI
NE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
    ActivityCompat.requestPermissions(this,new String
[] {Manifest.permission.ACCESS_FINE_LOCATION},1);
} else {
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0,
10, locationListener);
}
btn_send.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        String judul = txt_judul.getText().toString();
        String isi = txt_isi.getText().toString();
        String lati = myLat;
        String longi = myLong;
        String idu = ID_PUBLIC;
        String kec =
String.valueOf(spinner1.getSelectedItem());
        String kejadian =
String.valueOf(spinner2.getSelectedItem());
        if (conMgr.getActiveNetworkInfo() != null
            && conMgr.getActiveNetworkInfo().isAvailable()
            && conMgr.getActiveNetworkInfo().isConnected())
        {
            checkRegister(judul, isi, lati, longi, idu, kec,
kejadian );
        } else {
            Toast.makeText(getApplicationContext(), "No

```

```

Internet Connection", Toast.LENGTH_SHORT).show();
        }
    }
});
}
// add items into spinner dynamically
public void addItemOnSpinner2() {
    spinner2 = (Spinner) findViewById(R.id.spinner2);
    List<String> list = new ArrayList<String>();
    list.add("Kecelakaan");
    list.add("Macet");
    list.add("Miras");
    list.add("Blackmarket");
    list.add("Perjudian");
    list.add("Pencurian");
    list.add("Penemuan Mayat");
    list.add("Tawuran");
    ArrayAdapter<String> dataAdapter = new
ArrayAdapter<String>(this,
        android.R.layout.simple_spinner_item, list);

dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dro
pdown_item);
    spinner2.setAdapter(dataAdapter);
}
public String getStringImage(Bitmap bmp) {
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    bmp.compress(Bitmap.CompressFormat.JPEG, bitmap_size, baos);
    byte[] imageBytes = baos.toByteArray();
    String encodedImage = Base64.encodeToString(imageBytes,
Base64.DEFAULT);
    return encodedImage;
}
private void showFileChooser() {
    showPictureDialog();
}
@Override
public void onActivityResult(int requestCode, int resultCode,
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == this.RESULT_CANCELED) {
        return;
    }
    if (requestCode == GALLERY) {
        if (data != null) {
            Uri contentURI = data.getData();
            try {
                Bitmap bitmap =
MediaStore.Images.Media.getBitmap(this.getContentResolver(),
contentURI);

                String path = saveImage(bitmap);
                setToImageView(getResizedBitmap(bitmap, 512));
            } catch (IOException e) {
                e.printStackTrace();
                Toast.makeText(InsertLapor.this, "Failed!",
Toast.LENGTH_SHORT).show();
            }
        }
    }
    else if (requestCode == CAMERA) {
        Bitmap thumbnail = (Bitmap) data.getExtras().get("data");
        setToImageView(getResizedBitmap(thumbnail, 512));
        saveImage(thumbnail);
    }
}
private void kosong() {

```

```

        imageView.setImageResource(0);
        txt_name.setText(null);
    }
    private void setToImageView(Bitmap bmp) {
        //compress image
        ByteArrayOutputStream bytes = new ByteArrayOutputStream();
        bmp.compress(Bitmap.CompressFormat.JPEG, bitmap_size, bytes);
        decoded = BitmapFactory.decodeStream(new
        ByteArrayInputStream(bytes.toByteArray()));
        //menampilkan gambar yang dipilih dari camera/gallery ke
        ImageView
        imageView.setImageBitmap(decoded);
    }
    // fungsi resize image
    public Bitmap getResizedBitmap(Bitmap image, int maxSize) {
        int width = image.getWidth();
        int height = image.getHeight();
        float bitmapRatio = (float) width / (float) height;
        if (bitmapRatio > 1) {
            width = maxSize;
            height = (int) (width / bitmapRatio);
        } else {
            height = maxSize;
            width = (int) (height * bitmapRatio);
        }
        return Bitmap.createScaledBitmap(image, width, height, true);
    }
    private void checkRegister(final String judul,final String
    isi,final String lati,final String longi,final String idu,final String
    kec,final String kejadian) {
        pDialog = new ProgressDialog(this);
        pDialog.setCancelable(false);
        pDialog.setMessage("Kirim laporan ...");
        showDialog();
        StringRequest strReq = new StringRequest(Request.Method.POST,
        url, new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                Log.e(TAG, "Register Response: " +
                response.toString());
                hideDialog();
                try {
                    JSONObject jsonObj = new JSONObject(response);
                    success = jsonObj.getInt(TAG_SUCCESS);
                    // Check for error node in json
                    if (success == 1) {
                        Log.e("Successfully Register!",
                        jsonObj.toString());
                        Toast.makeText(getApplicationContext(),
                        jsonObj.getString(TAG_MESSAGE),
                        Toast.LENGTH_LONG).show();
                        txt_judul.setText("");
                        txt_isi.setText("");
                        imageView.setImageResource(0);
                    } else {
                        Toast.makeText(getApplicationContext(),
                        jsonObj.getString(TAG_MESSAGE),
                        Toast.LENGTH_LONG).show();
                    }
                } catch (JSONException e) {
                    // JSON error
                    e.printStackTrace();
                }
            }
        }, new Response.ErrorListener() {

```

```

        @Override
        public void onErrorResponse(VolleyError error) {
            Log.e(TAG, "Login Error: " + error.getMessage());
            Toast.makeText(getApplicationContext(),
                error.getMessage(), Toast.LENGTH_LONG).show();
            hideDialog();
        }
    }) {
        @Override
        protected Map<String, String> getParams() {
            // Posting parameters to login url
            Map<String, String> params = new HashMap<String,
String>();

            params.put(KEY_IMAGE, getStringImage(decoded));
            params.put("idu", idu);
            params.put("judul", judul);
            params.put("isi", isi);
            params.put("lati", lati);
            params.put("longi", longi);
            params.put("kec", kec);
            params.put("kejadian", kejadian);
            return params;
        }
    };
    // Adding request to request queue
    ApplicationController.getInstance().addToRequestQueue(strReq,
tag_json_obj);
}
private void showDialog() {
    if (!pDialog.isShowing())
        pDialog.show();
}
private void hideDialog() {
    if (pDialog.isShowing())
        pDialog.dismiss();
}
private void showPictureDialog(){
    AlertDialog.Builder pictureDialog = new
AlertDialog.Builder(this);
    pictureDialog.setTitle("Select Action");
    String[] pictureDialogItems = {
        "Select photo from gallery",
        "Capture photo from camera" };
    pictureDialog.setItems(pictureDialogItems,
        new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int
which) {

                switch (which) {
                    case 0:
                        choosePhotoFromGallary();
                        break;
                    case 1:
                        takePhotoFromCamera();
                        break;
                }
            }
        });
    pictureDialog.show();
}
public void choosePhotoFromGallary() {
    Intent galleryIntent = new Intent(Intent.ACTION_PICK,
android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    startActivityForResult(galleryIntent, GALLERY);
}

```

```

    }
    private void takePhotoFromCamera() {
        Intent intent = new
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
        startActivityForResult(intent, CAMERA);
    }
    public String saveImage(Bitmap myBitmap) {
        ByteArrayOutputStream bytes = new ByteArrayOutputStream();
        myBitmap.compress(Bitmap.CompressFormat.JPEG, 90, bytes);
        File wallpaperDirectory = new File(
Environment.getExternalStorageDirectory() +
IMAGE_DIRECTORY);
        // have the object build the directory structure, if needed.
        if (!wallpaperDirectory.exists()) {
            wallpaperDirectory.mkdirs();
        }
        try {
            File f = new File(wallpaperDirectory,
Calendar.getInstance()
.getTimeInMillis() + ".jpg");
            f.createNewFile();
            FileOutputStream fo = new FileOutputStream(f);
            fo.write(bytes.toByteArray());
            MediaScannerConnection.scanFile(this,
new String[]{f.getPath()},
new String[]{"image/jpeg"}, null);
            fo.close();
            Log.d("TAG", "File Saved:--->" + f.getAbsolutePath());
            return f.getAbsolutePath();
        } catch (IOException e1) {
            e1.printStackTrace();
        }
        return "";
    }
    private void requestMultiplePermissions(){
        Dexter.withActivity(this)
.withPermissions(
Manifest.permission.CAMERA,
Manifest.permission.WRITE_EXTERNAL_STORAGE,
Manifest.permission.READ_EXTERNAL_STORAGE)
.withListener(new MultiplePermissionsListener() {
@Override
public void
onPermissionsChecked(MultiplePermissionsReport report) {
// check if all permissions are granted
if (report.areAllPermissionsGranted()) {
Toast.makeText(getApplicationContext(),
"All permissions are granted by user!", Toast.LENGTH_SHORT).show();
}
// check for permanent denial of any permission
if (report.isAnyPermissionPermanentlyDenied())
{
// show alert dialog navigating to Settings
//openSettingsDialog();
}
}
}
public void
onPermissionRationaleShouldBeShown(List<PermissionRequest> permissions,
PermissionToken token) {
token.continuePermissionRequest();
}
}).
withErrorListener(new PermissionRequestErrorListener()
{
@Override

```

```

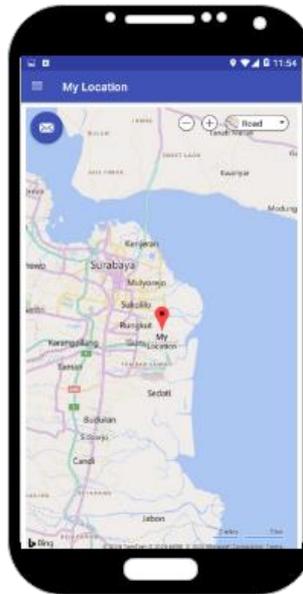
        public void onError(DexterError error) {
            Toast.makeText(getApplicationContext(), "Some
Error! ", Toast.LENGTH_SHORT).show();
        }
    })
    .onSameThread()
    .check();
}
}

```

Source code diatas berfungsi untuk form pengisian data data laporan dan pengambilan otomatis data latitude dan longitude dari lokasi pengirim laporan kejadian.

### 5.5.2.3 Halaman My Location

Halaman my location berfungsi untuk melihat informasi dari keberadaan posisi user pada saat ini. Dilengkapi dengan tombol yang berfungsi untuk melaporkan kejadian berdasarkan lokasi user saat ini.

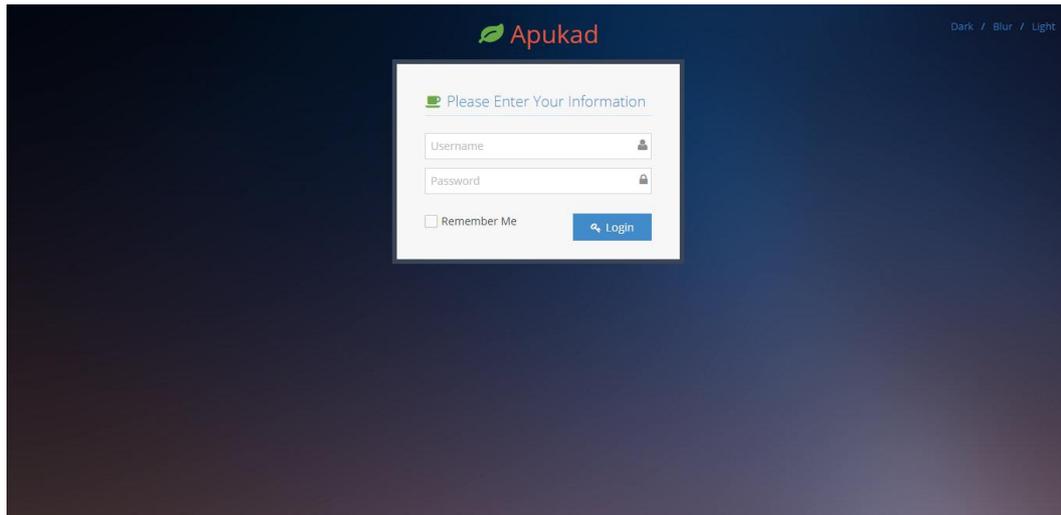


Gambar 5.10 Desain Halaman My Location

## 5.5.3 Aplikasi Administrator

### 5.5.3.1 Halaman Login

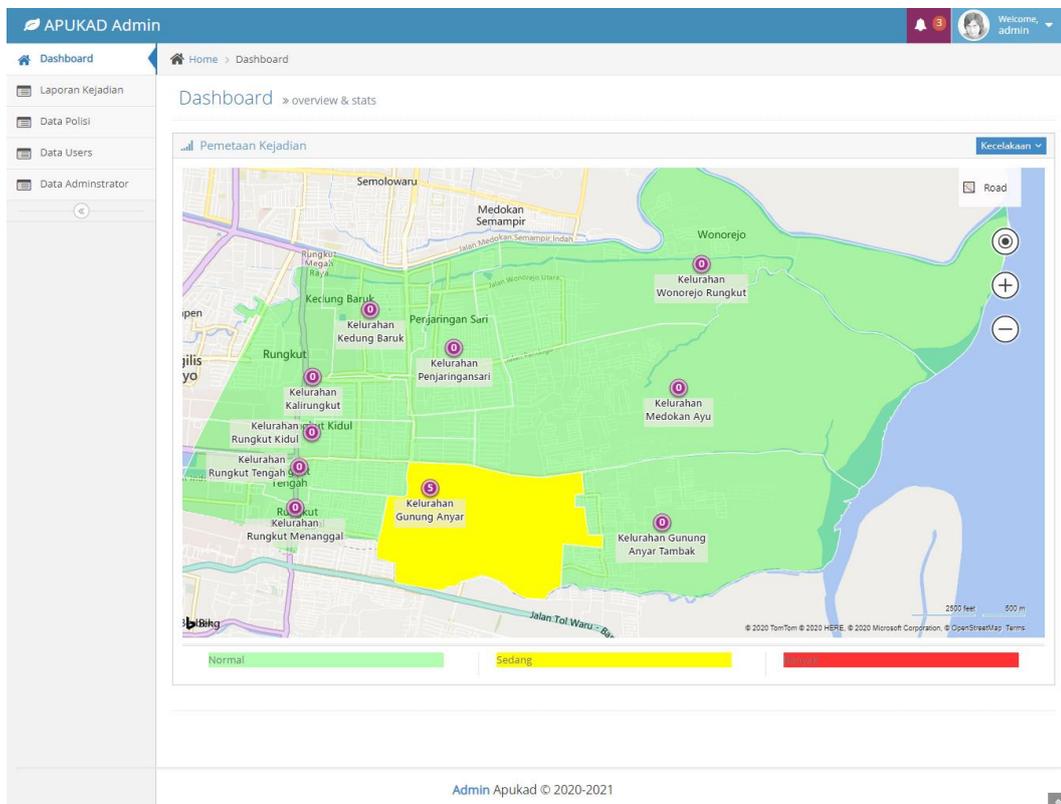
Halaman login berfungsi untuk melakukan proses login untuk user administrator dengan memasukkan username dan password.



Gambar 5.11 Desain Halaman Login

### 5.5.3.2 Halaman Dashboard

Halaman dashboard berfungsi untuk menampilkan peta area wilayah hukum kepolisian sektor Rungkut dilengkapi dengan informasi jumlah dan warna berdasarkan jenis kejadian.



Gambar 5.12 Desain Halaman Dashboard

### 5.5.3.3 Halaman Laporan Kejadian

Halaman laporan kejadian berfungsi untuk menampilkan seluruh informasi dari laporan kejadian yang telah dilaporkan oleh user public / masyarakat.

APUKAD Admin

Home > Tables > Laporan Kejadian

Tables > Laporan Kejadian

Results for "Laporan Kejadian"

Display 10 records Search:

<input type="checkbox"/>	Pengirim	Judul	Jenis	Tanggal	Status	
<input type="checkbox"/>	agus	test	Macet	06 July 2020 00:00:00	Baru	
<input type="checkbox"/>	Choirun Nisa	macet	Macet	06 July 2020 00:00:00	Baru	
<input type="checkbox"/>	Choirun Nisa	teslagi	Macet	06 July 2020 00:00:00	Baru	
<input type="checkbox"/>	agus	oke	Kecelakaan	03 July 2020 00:00:00	Publish	
<input type="checkbox"/>	agus	sa	Kecelakaan	03 July 2020 00:00:00	Publish	
<input type="checkbox"/>	agus	oke	Kecelakaan	03 July 2020 00:00:00	Publish	
<input type="checkbox"/>	agus	lpor	Kecelakaan	03 July 2020 00:00:00	Publish	
<input type="checkbox"/>	agus	judul	Kecelakaan	01 January 1970 01:00:00	Publish	

Showing 1 to 8 of 8 entries

Previous 1 Next

Gambar 5.13 Desain Halaman Laporan Kejadian

### 5.5.3.4 Halaman Data Polisi

Halaman data polisi berfungsi untuk menampilkan list dari user polisi dan mengelola data akun user polisi.

APUKAD Admin

Home > Tables > Data Polisi

Tables > Data Polisi

Results for "Data Polisi"

Display 10 records Search:

<input type="checkbox"/>	No KTP	Nama	Alamat	Telp	Username	
<input type="checkbox"/>	0673657623	ahmad	Surabaya	031	polisi	

Showing 1 to 1 of 1 entries

Previous 1 Next

No KTP

nomor identitas

NAMA

nama

Alamat

TELP

nomor telepon

Foto file input

Choose File No file chosen

User Name

username

Password

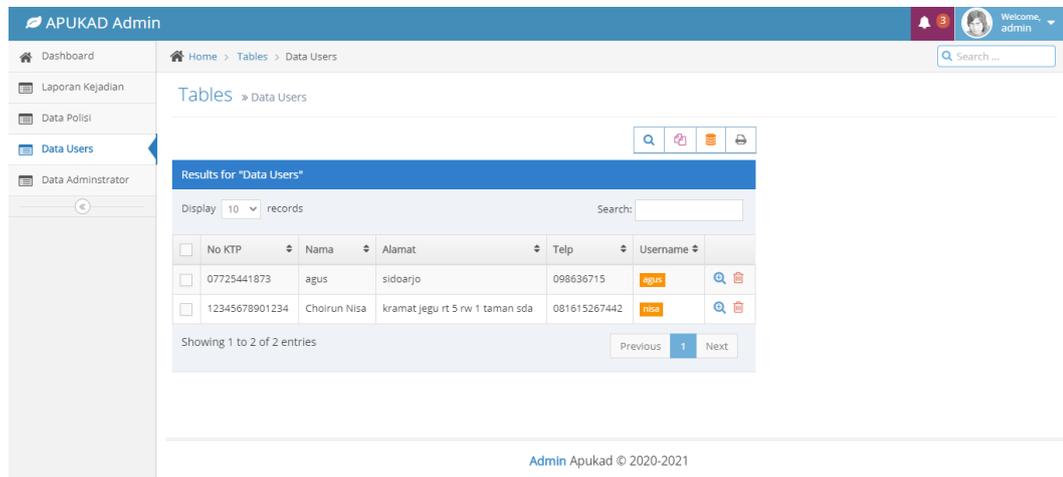
password

Simpan

Gambar 5.14 Desain Halaman Data Polisi

### 5.5.3.5 Halaman Data Users

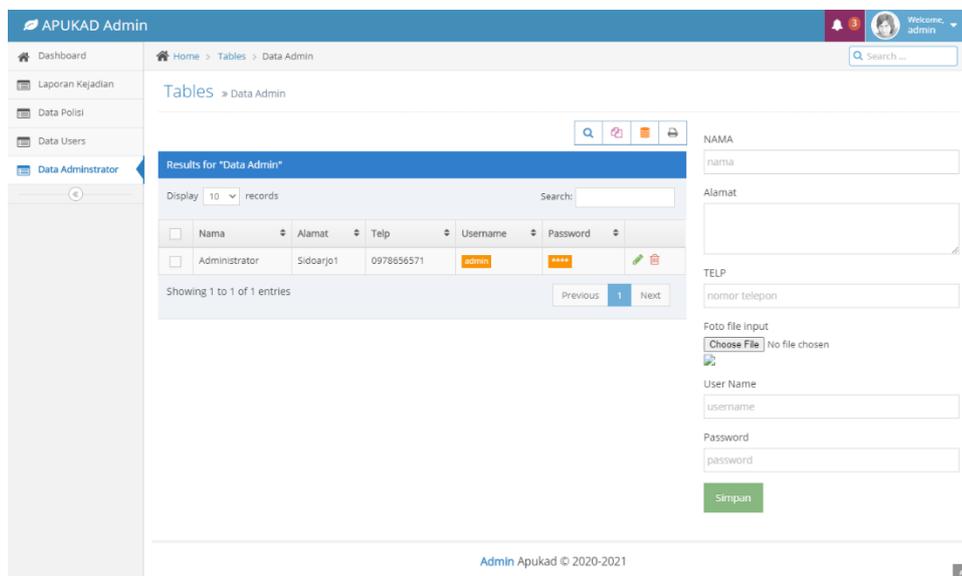
Halaman data user berfungsi untuk menampilkan list dari user public dan mengelola data akun user public.



Gambar 5.15 Desain Halaman Data Users

### 5.5.3.6 Halaman Data Administrator

Halaman data administrator berfungsi untuk menampilkan list dari user administrator dan mengelola data akun user administrator.



Gambar 5.16 Desain Halaman Data Administrator

## BAB VI

### HASIL DAN PENGUJIAN

#### 6.1 Pengujian Sistem

Aplikasi pusat komando pemanggilan darurat kejadian kriminal atau kendala lalu lintas berbasis Android menggunakan metode pengujian sistem *black-box testing* dengan teknik komparabilitas aplikasi dan *functional testing*.

*Black box testing* adalah suatu pengujian yang tidak melihat dari struktur coding suatu program. Pengujian ini biasanya meliputi seputar kinerja program. Dan pengujian komparabilitas aplikasi adalah suatu pengujian yang bertujuan untuk melihat performa aplikasi yang terinstal pada berbagai smartphone dan berbagai versi android sesuai spesifikasi minimal dari versi android yang didukung oleh aplikasi android tersebut. Dan *functional testing* adalah pengujian berdasarkan studi kasus yang akan diberikan pada suatu komponen, modul atau fitur yang akan di *testing*. *Functional testing* dilakukan dengan cara memberikan inputan pada komponen, modul atau fitur kemudian memeriksa hasil *output* nya. Apabila *output* yang dihasilkan sesuai dengan harapan atau benar, apabila tidak sesuai maka bagian tersebut terdapat *error*. Pengujian dengan teknik *functional testing* dilakukan dengan cara menguji apakah alur jalannya program sudah sesuai dengan kebutuhan user atau belum.

#### 6.2 Hasil Pengujian Sistem

Berikut ini adalah hasil pengujian Aplikasi pusat komando pemanggilan darurat kejadian kriminal atau kendala lalu lintas berbasis Android, baik secara teknik komparabilitas aplikasi dan *functional testing*.

##### 1. Komparabilitas Aplikasi

Pada pengujian ini, kinerja aplikasi *mobile* untuk user masyarakat dan polisi akan dilihat pada berbagai *smartphone* dan berbagai versi android yang berjalan pada *smartphone* saat pengujian. Sedangkan, kinerja web aplikasi untuk user admin akan dilihat pada berbagai browser di Windows. Pada pengujian tersebut device

yang digunakan menggunakan koneksi internet yang sama yaitu internet indihome paket 10mbps.

Tabel 6.1. Tabel kinerja tampilan *mobile* aplikasi pada *smartphone* uji.

No	Perangkat Android	Versi Android	Tampilan Halaman				
			Home	My Location	Peta Kejadian	List Kejadian	Akun
1.	Infinix Mobility Limited Infinix X627	9.0	1.28 sec	2.21 sec	0.90 sec	1.03 sec	0.70 sec
2.	Samsung SM- G955F	9.0	1.25 sec	2.00 sec	0.73 sec	0.91 sec	0.51 sec
3.	Android Virtual Devices	8.1	1.30 sec	3.10 sec	0.70 sec	1.77 sec	0.90 sec
4.	Android Virtual Devices	7.1.1	1.33 sec	3.19 sec	0.76 sec	1.39 sec	0.99 sec
5.	MEmu Android Virtual Devices	5.1.1	1.19 sec	2.66 sec	0.67 sec	0.57 sec	0.50 sec
6.	ASUS T00F Zenfone 5 A501CG	4.4	force close	force close	force close	force close	force close
Kecepatan Rata-rata			1,05833 3 sec	2,19333 3 sec	0,62666 7 sec	0,945 sec	0,6 sec

Penjelasan dari tabel 6.1 diatas adalah *mobile* aplikasi dapat menampilkan tampilannya dengan baik pada berbagai versi android (minimal versi 5.1.1) sampai pada versi android yang terbaru saat ini, yaitu android 9.0. Sedangkan pada android versi 4.4, aplikasi mengalami force close.

Tabel 6.2. Tabel kinerja tampilan *web* aplikasi pada *browser* uji.

No	Browser	Versi Browser	Tampilan Halaman				
			Dashbo ard	Lapora n Kejadi an	Data Polisi	Data Users	Data Administra tor
1.	Chrome	83.0.4103.16	0.75 sec	0.89 sec	0.77 sec	0.61 sec	0.64 sec
2.	Firefox	78.0.1	1.37 sec	1.25 sec	1.20 sec	0.92 sec	0.85 sec
3.	Micros oft Edge	83.0.478.58	1.58 sec	1.10 sec	0.83 sec	0.82 sec	0.80 sec
Kecepatan Rata-rata			1,23333 3 sec	1,08 sec	0,9333 33 sec	0,7833 33 sec	0,763333 sec

Penjelasan dari 6.2 diatas adalah web aplikasi dapat menampilkan tampilannya dengan baik pada berbagai browser diantaranya yaitu: Chrome, Firefox, dan Microsoft Edge.

Dan berikut ini hasil pengujian pengambilan data pada server.

Tabel 6.3. Tabel pengambilan data dari aplikasi *mobile* ke server.

No.	Perangkat Android	Versi Android	Pengambilan Data		
			Peta Kejadian	List Kejadian	Akun
1.	Infinix Mobility Limited Infinix X627	9.0	1.98 sec	0.87 sec	0.67 sec
2.	Samsung SM-G955F	9.0	1.75 sec	0.98 sec	0.73 sec

3.	Android Devices	Virtual	8.1	2.07 sec	1.73 sec	0.70 sec
4.	Android Devices	Virtual	7.1.1	2.00 sec	1.60 sec	0.76 sec
5.	MEmu Virtual Devices	Android	5.1.1	0.88 sec	1.09 sec	0.47 sec
6.	ASUS T00F Zenfone 5 A501CG		4.4	Force close	Force close	Force close
Kecepatan Rata-rata				1,446667 sec	1,045 sec	0,555 sec

Penjelasan dari tabel 6.3 diatas adalah *mobile* aplikasi dapat mengambil data dari database server dengan baik pada berbagai versi android (minimal versi 5.1.1) sampai pada versi android yang terbaru saat ini, yaitu android 9.0. Sedangkan pada android versi 4.4 aplikasi mengalami force close.

Tabel 6.4. Tabel pengambilan data dari web aplikasi ke server.

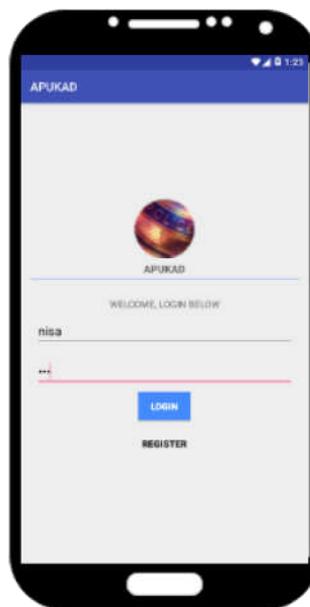
No	Browse r	Versi Browser	Tampilan Halaman				
			Dashboa rd	Lapora n Kejadi an	Data Polisi	Data Users	Data Administra tor
1.	Chrome	83.0.4103.1 16	0.75 sec	0.89 sec	0.77 sec	0.61 sec	0.64 sec
2.	Firefox	78.0.1	1.37 sec	1.25 sec	1.20 sec	0.92 sec	0.85 sec
3.	Micros oft Edge	83.0.478.58	1.58 sec	1.10 sec	0.83 sec	0.82 sec	0.80 sec
Kecepatan Rata-rata			1,23333 3 sec	1,08 sec	0,9333 33 sec	0,7833 33 sec	0,763333 sec

Penjelasan dari 6.4 diatas adalah web aplikasi dapat menampilkan tampilannya dengan baik pada berbagai browser diantaranya yaitu: Chrome, Firefox, dan Microsoft Edge.

## 2. Pengujian *Functional Testing*

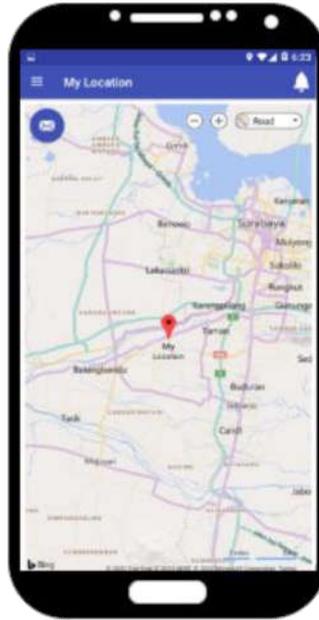
Berikut ini adalah pengujian Aplikasi pusat komando pemanggilan darurat kejadian kriminal atau kendala lalu lintas berbasis Android pada *functional testing*. Berdasarkan gambar 4.1. dan 4.2, alur dari aplikasi ini adalah sebagai berikut:

Diawali dengan user public (masyarakat) melaporkan kejadian atau peristiwa dengan melalui aplikasi *mobile*. Dengan melakukan login terdahulu.



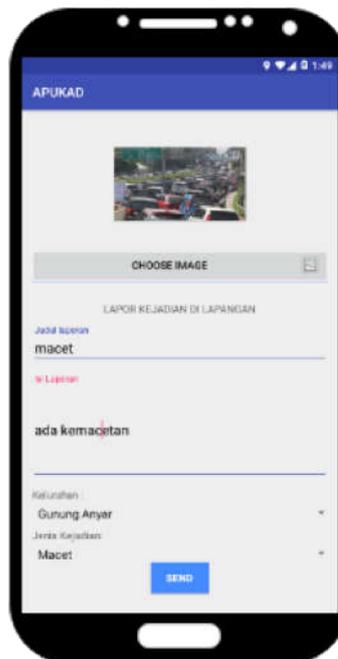
6.1 Halaman Login

Setelah sukses melakukan login, user public akan membuka menu My Location pada Navigation Drawer. Pada halaman menu My Location, user public menekan tombol ikon pesan untuk melakukan pelaporan. Seperti pada gambar dibawah ini:



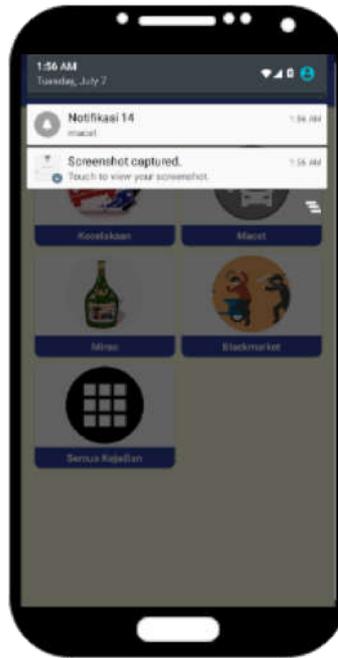
6.2 Halaman My Location

Setelah menekan tombol lapor, maka user public akan mengisi data data laporan pada form pelaporan.



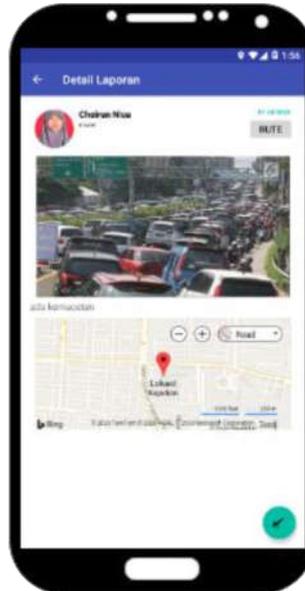
6.3 Halaman Form Laporan

Setelah user public sukses mengirimkan laporan, maka user polisi akan menerima notifikasi laporan untuk segera ditindak lanjuti.



#### 6.4 Notifikasi

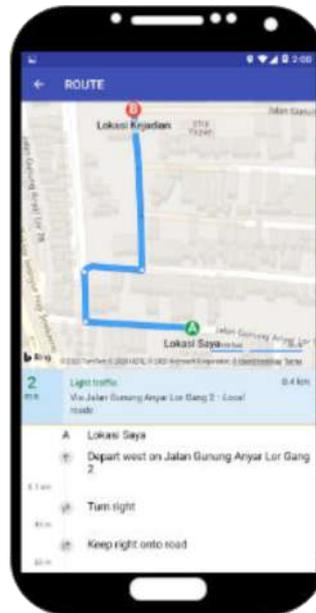
User polisi yang menerima notifikasi akan membuka notifikasi dan menelusuri lokasi kejadian.



#### 6.5 Detail Laporan

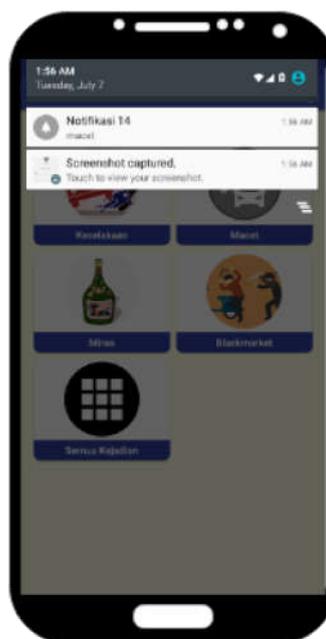
Pada detail laporan user polisi terdapat tombol verifikasi berwarna hijau yang digunakan untuk menyatakan bahwa memang benar laporan tersebut benar

adanya dan segera ditindaklanjuti. Sehingga user polisi perlu mensurvei lokasi kejadian tersebut dahulu sebelum mengverifikasi laporan.



6.6 Rute Jalan

Setelah dipastikan kebenaran laporan yang dilaporkan oleh user public, maka user polisi akan memverifikasi laporan sehingga laporan akan terpublish pada user public yang berada di dalam radius 1000m.



6.7 Laporan terpublish

### 6.3 Analisis Hasil Pengujian

Kesimpulan dari pengujian komparabilitas *mobile* aplikasi diatas adalah aplikasi berjalan dengan cukup baik pada Android versi 5.1.1-9.0.0 dan diketahui bahwa spesifikasi minimum android yang dibutuhkan untuk menjalankan aplikasi tersebut adalah android versi 5.1.1, kemudian dari pengujian komparabilitas *web* aplikasi diatas bisa menampilkan semua halaman-halamannya dengan baik pada berbagai browser diantaranya yaitu: Chrome, Firefox, dan Microsoft Edge. Berikut adalah hasil dari pengujian kecepatan aplikasi *mobile* dan web administrator:

Tabel 6.5. Tabel kriteria penilaian.

No.	Kategori	Rentang Waktu
1.	Sangat Cepat	0.00 sec s/d 0.99 sec
2.	Cepat	1.00 sec s/d 1.99sec
3.	Normal	2.00 sec s/d 2.99 sec
4.	Lama	3.00 sec s/d 5.99sec

Tabel 6.6. Tabel hasil kesimpulan kecepatan.

No.	Pengujian	Kecepatan Rata-rata	Kecepatan
A. Kinerja Tampilan <i>Mobile</i> Aplikasi			
1.	Menampilkan menu Home	1,058333333 sec	Cepat
2.	Menampilkan menu My Location	2,193333333 sec	Normal
3.	Menampilkan menu Peta Kejadian	0,626666667 sec	Sangat Cepat
4.	Menampilkan menu List Kejadian	0,945 sec	Sangat Cepat
5.	Menampilkan menu Akun	0,6 sec	Sangat Cepat
B. Kinerja Tampilan Web Administrator			
1.	Menampilkan menu Dashboard	1,233333333 sec	Cepat
2.	Menampilkan menu Laporan Kejadian	1,08 sec	Cepat

3.	Menampilkan menu Data Polisi	0,933333333 sec	Sangat Cepat
4.	Menampilkan menu Data Users	0,783333333 sec	Sangat Cepat
5.	Menampilkan menu Data Administrator	0,763333333 sec	Sangat Cepat
C. Proses Pengambilan Data Dari Aplikasi <i>Mobile</i> Ke Server			
1.	Pengambilan data pada menu Peta Kejadian	1,446666667 sec	Cepat
2.	Pengambilan data pada menu List Kejadian	1,045 sec	Cepat
3.	Pengambilan data pada menu Akun	0,555 sec	Sangat Cepat
D. Proses Pengambilan Data Dari Web Aplikasi Ke Server			
1.	Pengambilan data pada menu Dashboard	1,233333333 sec	Cepat
2.	Pengambilan data pada menu Laporan Kejadian	1,08 sec	Cepat
3.	Pengambilan data pada menu Data Polisi	0,933333333 sec	Sangat Cepat
4.	Pengambilan data pada menu Data Users	0,783333333 sec	Sangat Cepat
5.	Pengambilan data pada menu Data Administrator	0,763333333 sec	Sangat Cepat

Kesimpulan dari pengujian kecepatan komparabilitas *mobile* aplikasi diatas adalah aplikasi berjalan dengan rentang kecepatan antara 0.555 sec s/d 2,193333333 sec. Sedangkan, kesimpulan dari pengujian kecepatan komparabilitas web aplikasi administrator diatas adalah aplikasi berjalan dengan rentang kecepatan antara 0,763333333 sec s/d 1,233333333 sec. Koneksi internet yang lambat dapat mempengaruhi kecepatan dalam pengaksesan sebuah halaman.

Pada pengujian *functional testing*, *mobile* aplikasi dan *web* aplikasi dapat bekerja dengan sangat baik. Dan kesimpulan dari pengujian *functional testing* diatas adalah aplikasi dapat berjalan dengan semestinya dan mengeluarkan pemberitahuan kepada pengguna aplikasi.

Dari hasil pengujian diatas, maka aplikasi dapat dinyatakan bahwa sistem telah layak uji secara komabilitas aplikasi dan *functional testing*.

## **BAB VII**

### **PENUTUP**

#### **7.1 Kesimpulan**

Berdasarkan hasil perancangan dan pengujian yang telah dilakukan, dapat disimpulkan bahwa:

1. Aplikasi berjalan dengan cukup baik pada Android versi 5.1.1-9.0.0 dan diketahui bahwa spesifikasi minimum android yang dibutuhkan untuk menjalankan aplikasi tersebut adalah android versi 5.1.1.
2. Aplikasi web untuk admin dapat berjalan dengan baik pada berbagai browser diantaranya yaitu: Chrome, Firefox, dan Microsoft Edge.
3. Untuk menghindari adanya laporan fiktif, maka terdapat fitur verifikasi polisi atas laporan yang terkirim berfungsi sebagai benar terjadi atau tidaknya suatu laporan tersebut. Dengan mengverifikasi, maka laporan akan otomatis terpublish sehingga user public (masyarakat) akan mendapatkan pemberitahuan berupa notifikasi
4. Pada saat dilakukannya pelaporan, sistem akan mengambil koordinat lokasi pelapor dan pelapor wajib menyertakan foto dari kejadian tersebut.
5. Aplikasi mengirimkan notifikasi kepada user public (masyarakat) yang berada dalam radius 1000m dari lokasi kejadian.

#### **7.2 Saran**

Berdasarkan hasil perancangan dan pengujian yang telah dilakukan, terdapat beberapa saran untuk pengembangan selanjutnya, yaitu:

1. Diharapkan pada penelitian selanjutnya dapat dikembangkan dalam versi iOS operasi sistem agar pengguna iPhone juga dapat menggunakan aplikasi tersebut.
2. Pada penelitian selanjutnya, bisa ditambahkan fitur-fitur yang dikhususkan untuk menjadi wadah dari respon masyarakat terhadap laporan dari suatu kejadian yang terpublish.

3. Diharapkan pada penelitian selanjutnya jumlah laporan bisa diteliti lebih lanjut sebagai pendeteksi laporan fiktif atau laporan benar adanya.
4. Diharapkan pada penelitian selanjutnya, sistem bisa dikoneksikan dengan sistem informasi kependudukan.