

## Aplikasi Peringatan Rambu Lalu Lintas dengan Metode *Location Based Service* Berbasis *Mobile*

Risnu Arya Kusuma<sup>#1</sup>, Yus Sholva<sup>#2</sup>, Rudy Dwi Nyoto<sup>#3</sup><sup>#</sup>*Program Studi Informatika, Fakultas Teknik, Universitas Tanjungpura  
Jl. Prof. Hadari Nawawi, Kota Pontianak, 78115*<sup>1</sup>risnu.arya@student.untan.ac.id<sup>2</sup>sholvariza@untan.ac.id<sup>3</sup>rudydn@informatika.untan.ac.id

### Abstrak

Rambu lalu lintas adalah bagian dari perlengkapan jalan yang memuat lambang, huruf, angka, kalimat dan/atau perpaduan di antaranya, dan berfungsi untuk menginformasikan peraturan dan perintah yang ada dalam jalan tersebut. Dalam penerapannya, banyak sekali pengendara yang melanggar rambu yang dikarenakan ketidaktahuan terhadap rambu tersebut. Faktor yang menyebabkan ketidaktahuan pengendara terhadap rambu diantaranya adalah rambu tidak terlihat oleh pengendara yang dikarenakan posisi rambu yang tidak terjangkau pandangan dan kondisi plang rambu yang kurang baik serta ketidaktahuan pengendara terhadap arti dari rambu tersebut. Dengan permasalahan tersebut, diperlukan aplikasi yang dapat memberi tahu pengendara mengenai informasi rambu yang berada disekitarnya ketika sedang berkendara. Aplikasi ini memberikan peringatan rambu lalu lintas dengan metode location based service untuk menyesuaikan data rambu dengan lokasi pengendara, yang disampaikan dalam bentuk tampilan dan suara agar tidak mengganggu pengendara dalam mengemudikan kendaraan serta penilaian perilaku pengendara terhadap rambu yang dilanggar. Penilaian perilaku terhadap rambu yang berlaku dapat mendorong pengendara agar lebih mematuhi rambu. Hasil dari penelitian ini menghasilkan aplikasi yang dapat digunakan sebagai pemandu rambu ketika sedang berkendara. Pengujian sistem mendapatkan hasil cukup baik pada pengujian menggunakan metode pengujian BlackBox dengan memiliki beberapa batasan tertentu. Dengan adanya aplikasi ini, diharapkan pengendara dapat mematuhi rambu lalu lintas dan meminimalisir adanya pelanggaran serta kecelakaan lalu lintas.

**Kata kunci:** Rambu Lalu Lintas, Deteksi, Peringatan, Penilaian, Location Based Service, Aplikasi Mobile.

## Traffic Sign Warning Application with Location Based Service Method and Mobile Based

### Abstract

Traffic signs are part of road equipment that contains symbols, letters, numbers, sentences and/or a combination of them, and serves to inform the rules and instructions in the road. In its application, a lot of drivers violate the signs due to ignorance of these signs. Factors causing drivers' ignorance of signs include signs not seen by drivers due to the unreachable position of the sign and poor signposts and the drivers are not known of the meaning of the sign. With these problems, an application that can notify drivers about the sign information around them when driving is needed. This application provides a warning of traffic signs with location-based service methods to adjust the traffic sign data with the driver's location informed in the form of display and sound so as not to disturb the driver when driving the vehicle and assessing the drivers' behavior. Behavioral assessment of the prevailing signs can encourage drivers to more obey the signs. The results of this research are to produce applications that can be used as signs guide when driving. The system test result that can run quite well on testing using the BlackBox testing method by having certain restrictions. With this application, it is expected that drivers can obey traffic signs and minimize violations and traffic accidents.

**Keywords:** Traffic Sign, Detection, Warning, Scoring, Location-based service, Mobile app.

### I. PENDAHULUAN

Saat ini penggunaan kendaraan bermotor sebagai alat transportasi semakin meningkat setiap tahunnya.

Peningkatan penggunaan kendaraan juga mempengaruhi aktifitas jalan raya yang menyebabkan tingkat resiko kecelakaan meningkat. Kecelakaan lalu lintas sebagian besar diakibatkan oleh pengendara yang melanggar aturan

lalu lintas [1]. Pelanggaran lalu lintas adalah hal yang paling sering terjadi di jalan raya, yang dilakukan oleh sebagian besar masyarakat [2]. Kurang sadarnya masyarakat dalam hukum berlalu-lintas dapat dilihat dalam perilaku seperti semakin meningkatnya pelanggaran lalu lintas oleh pengendara [3]. Rambu lalu lintas adalah bagian dari perlengkapan jalan yang memuat lambang, huruf, angka, kalimat dan/atau perpaduan di antaranya. Rambu lalu lintas berfungsi untuk menginformasikan peraturan dan perintah yang ada dalam jalan tersebut. Penggunaan rambu lalu lintas bertujuan agar memberikan perhatian dan peringatan kepada pengendara agar mengurangi tingkat kecelakaan dan mentertibkan arus lalu lintas jalan [4]. Untuk mendukung kelancaran dan keamanan dalam berlalu lintas, maka disusun aturan-aturan serta perangkat pendukung dalam kegiatan berlalu lintas [5]. Dalam penerapannya, tidak sedikit pengendara yang masih melanggar rambu. Penyebab pelanggaran rambu adalah pengendara kurang mengetahui arti atau maksud dari rambu rambu yang ada di jalan, sosialisasi yang kurang terhadap masyarakat mengenai pentingnya mematuhi rambu rambu lalu lintas [6], dan posisi plang rambu yang terkadang tidak terlihat karena posisi pemasangan yang kurang terlihat dan kondisi plang rambu yang kurang baik.

Dalam menyelesaikan permasalahan terkait rambu lalu lintas, beberapa peneliti sudah pernah membuat aplikasi pengenalan rambu berbasis objek. Kurniawan Teguh [5] dan Cecep Supriatna [6] membuat aplikasi media pembelajaran rambu dengan memanfaatkan metode *Augmented Reality*, sedangkan Kurnia Khafidhatun Rafiah [7] menggunakan algoritma *CAMPSHIFT* sebagai pengenalan objek gambar rambu. Aplikasi tersebut sebagai media edukasi mengenai rambu yang diharapkan meningkatkan pengetahuan pengguna terhadap arti dari rambu-rambu lalu lintas.

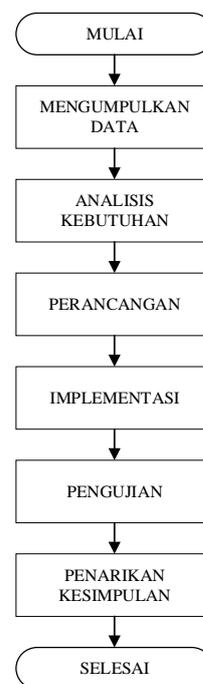
Penelitian terkait peringatan rambu lalu lintas sudah pernah dilakukan oleh peneliti lain, yaitu Marson James Budiman, dkk [8]. Pada penelitiannya, ia membuat sebuah aplikasi android yang dapat memberikan isyarat terhadap adanya rambu di sekitar lokasi pengguna. Cara kerja sistem yang dibuat adalah dengan menentukan lokasi pengguna dan menghitung jarak rambu dengan membandingkan jarak antara koordinat lokasi pengguna dan koordinat lokasi rambu. Aplikasi tersebut akan mendeteksi adanya rambu yang berada di sekitar jarak 25-30 meter dan membunyikan suara sesuai jenis rambu. Data rambu yang dimuat terdiri dari 4 jenis, yaitu rambu lampu lalu lintas, dilarang parkir, dilarang belok kanan dan dilarang belok kiri.

Kemudian penelitian yang penulis lakukan adalah membuat aplikasi peringatan rambu lalu lintas yang dapat menyesuaikan terhadap lokasi dan kondisi pengendara agar informasi rambu yang disampaikan tepat. Mengimplementasikan sistem dengan penggunaan metode *Location Based Service* pada perangkat *mobile* dapat digunakan untuk mendeteksi lokasi pengendara. Dalam penelitian ini menggunakan sistem operasi *Android* yang umum digunakan pada perangkat *mobile*. *Android* adalah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. *Android* merupakan sebuah kumpulan perangkat lunak untuk

perangkat *mobile* yang meliputi sistem operasi, *middleware* dan aplikasi utama *mobile* [9]. Dengan data lokasi pengendara, sistem akan memuat data rambu dan mencocokkan rambu yang sesuai dengan membandingkan koordinat rambu dengan lokasi pengendara dan kondisi pengendara untuk menginformasikan rambu yang berlaku ketika pengendara sedang berkendara, mencatat setiap pelanggaran rambu dan memberikan penilaian kepatuhan terhadap rambu, sehingga dapat dijadikan evaluasi untuk pengguna. Penggunaan media suara sebagai media penyampaian informasi rambu berguna agar tidak mengganggu pengendara ketika sedang berkendara. Sistem tersebut dapat menjadi solusi alternatif sebagai media informasi mengenai rambu lalu lintas kepada pengendara. Dengan penggunaan aplikasi ini, diharapkan dapat mengurangi pelanggaran rambu yang dapat membuat aktifitas lalu lintas lebih teratur dan tingkat resiko kecelakaan lebih rendah.

## II. METODOLOGI

Penelitian dilakukan dengan beberapa langkah-langkah sistematis, berikut langkah penelitian diperlihatkan pada Gambar 1.



Gambar 1. Metode penelitian

### A. Pengumpulan Data

Metode pengamatan merupakan pendekatan untuk melaksanakan pengamatan dengan pencatatan data secara langsung dan melakukan tanya jawab. Melakukan pengamatan langsung pada pihak Kepolisian dan Dinas Perhubungan mencakup seluruh aspek yang dibutuhkan dalam pembuatan aplikasi dalam menentukan masalah yang ada. Survei akan dilakukan untuk pengumpulan data agar data yang dimuat lebih akurat

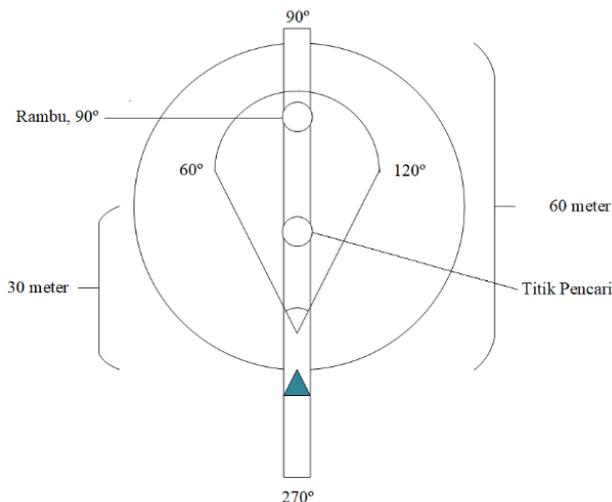
### B. Analisis Kebutuhan

Untuk menjalankan aplikasi dengan baik, aplikasi membutuhkan fungsi yang memuat rambu sesuai dengan

kondisi pengendara. Dalam menentukan rambu yang akan diinformasikan ke pengendara, aplikasi harus mengolah data-data yang diperlukan agar informasi yang diberikan sesuai dengan kondisi pengendara. Data yang diolah tidak hanya untuk diinformasikan ke pengendara ketika mengemudi, data yang diolah juga harus dapat digunakan untuk evaluasi pengendara terhadap tindakan pengendara mengenai rambu yang ditemui.

Data rambu lalu lintas memuat data titik koordinat lokasi rambu, nama rambu, jenis rambu untuk menentukan aksi peringatan, gambar dan suara untuk informasi peringatan. Untuk menampilkan data rambu yang sesuai dengan lokasi pengendara, maka diperlukan metode *Location Based Service* untuk mendeteksi lokasi. *Location Based Services* adalah layanan berbasis lokasi untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang pengguna gunakan. Layanan ini menggunakan teknologi global positioning system (GPS) dan cell-based location yang didapatkan dari operator penyedia jaringan pada perangkat. GPS adalah singkatan dari *Global Positioning System*, yang merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit [10]. GPS bekerja dengan menghubungkan sinyal satelit ke perangkat GPS itu sendiri yang selanjutnya Informasi dari GPS itu akan di transmisikan oleh beberapa satelit sehingga GPS *receiver* mampu untuk mengetahui dan menentukan seakurat mungkin dimana posisi pengguna fitur GPS [11]. Perangkat GPS akan menangkap sinyal NMEA dari satelit GPS yang menghasilkan koordinat latitude dan longitude terhadap lokasi perangkat GPS tersebut berada [12].

Setelah lokasi pengendara didapatkan, maka aplikasi akan mensortir rambu-rambu yang berada di sekitar pengendara dengan menghitung kesesuaian arah dan jarak antara kordinat lokasi dan kordinat rambu dengan metode *haversine*. Metode *haversine* merupakan metode yang digunakan dalam penghitungan jarak terdekat. Teorema *haversine* digunakan untuk menghitung jarak antara 2 titik dengan berdasarkan panjang garis lurus antara 2 titik pada garis bujur (*latitude*) dan garis lintang (*longitude*) [13].



Gambar. 2. Ilustrasi Pencari Rambu menunjukkan rambu akan terdeteksi ketika rambu berada di depan pengguna dan membandingkan kesesuaian antara arah pengendara dan arah ruas jalan.

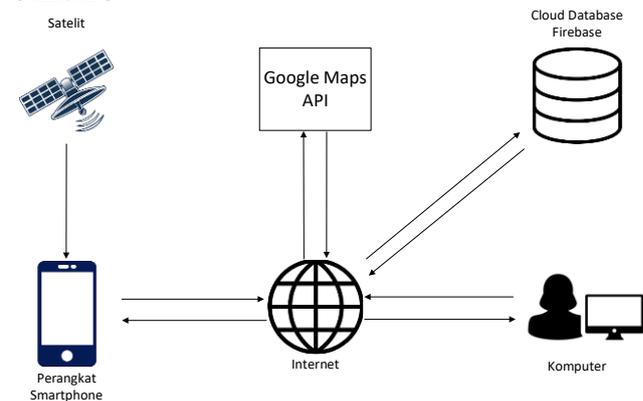
Dari ilustrasi pada Gambar 2, diasumsikan pengendara berjalan dengan arah 90° ke depan. Aplikasi akan membuat titik pencari rambu di depan sejauh 30 meter berdasarkan titik lokasi pengendara. Rambu akan dicari di sekitar radius 30 meter pada titik pencari rambu yang membuat aplikasi dapat mencari rambu hingga 60 meter ke depan. Rambu yang sesuai jarak akan disesuaikan kembali antara arah pengendara dengan arah rambu. Penyesuaian arah rambu akan dibuat rentang toleransi arah pengendara  $\pm 30^\circ$  kiri dan kanan berdasarkan arah pengendara. Pada contoh di atas, arah pengendara adalah 90°, maka akan menyesuaikan arah rambu yang berada di rentang 60° hingga 120°. Apabila jarak dan arah rambu sesuai (berada di rentang 60° hingga 120°), maka rambu tersebut akan masuk dalam data antrian rambu yang akan diinformasikan ke pengendara.

Data rambu yang dapat dimuat dalam aplikasi berupa rambu peringatan, rambu larangan, dan rambu perintah. Untuk menentukan pelanggaran, hanya rambu larangan dan

### C. Perancangan Sistem

#### 1) Arsitektur Aplikasi

Desain arsitektur aplikasi akan ditunjukkan pada Gambar 3.



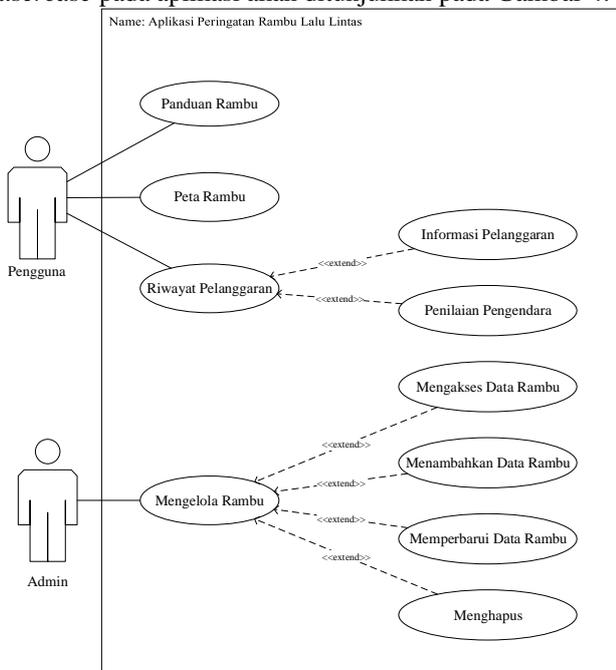
Gambar. 3. Arsitektur Aplikasi yang menjelaskan interaksi antar komponen pada sistem.

Dalam menjalankan aplikasi pengendara, aplikasi akan mengakses internet untuk mendapatkan data, layanan lokasi, dan layanan Google Maps API. Google Maps API adalah sebuah antarmuka salah satu bagian dari Google play services [14] yang berisikan tampilan, fungsi, dan metode dalam pengolahan peta digital untuk pengembangan aplikasi di berbagai platform. API ini menyediakan layanan berbasis peta digital, seperti tampilan jalan raya, lokasi pengguna, informasi jalan, informasi tempat, dan rute perjalanan. Google Maps API berupa suatu library yang berbentuk javascript [15] yang berguna untuk memodifikasi peta yang ada di Google Maps sesuai kebutuhan.

Aplikasi akan mengambil data rambu pada database, dilanjutkan dengan meminta data lokasi perangkat secara kontinu. Posisi pengendara akan ditampilkan pada tampilan peta yang dimuat melalui layanan Google Maps API. Segala proses pengolahan data rambu akan dilakukan di perangkat pengendara. Untuk aplikasi admin, aplikasi akan mengakses internet untuk mengambil, menambah, memperbarui, menghapus dan menampilkan data rambu.

2) Usecase Diagram

Usecase adalah abstraksi dari interaksi antara sistem dan aktor Dalam sistem ini terdapat 2 aktor dapat melakukan beberapa perilaku berbeda. Untuk aktor user dapat menjalankan fungsi pemandu rambu, melihat informasi rambu yang ada di jalan, dan melihat riwayat rambu yang ditemui. Sedangkan untuk aktor admin hanya dapat melakukan pengelolaan data rambu. Desain diagram usecase pada aplikasi akan ditunjukkan pada Gambar 4.



Gambar. 4. Usecase Diagram yang menjelaskan aksi yang dapat dilakukan pada aktor.

D. Implementasi

Sistem yang dibuat akan diimplementasikan menjadi sebuah aplikasi berbasis *mobile* yang terhubung dengan koneksi internet. Internet digunakan untuk mengambil data rambu, memuat tampilan peta, dan penentuan lokasi. Segala proses akan dilakukan di perangkat pengguna, seperti pengolahan data rambu dan penentuan informasi yang diberikan.

E. Pengujian

Pengujian sistem merupakan hal yang penting dari tahapan pembangunan sistem. Tahap pengujian bertujuan untuk menemukan masalah-masalah atau kesalahan yang mungkin terjadi pada aplikasi yang telah dibangun. Pengujian ini bermanfaat untuk mengetahui apakah aplikasi telah sesuai dengan kriteria kebutuhan sistem yang dibangun dan tujuan perancangannya.

F. Penarikan Kesimpulan

Penarikan kesimpulan dirumuskan berdasarkan analisis hasil pengujian dan mengacu pada tujuan dari penelitian yang dilakukan. Dari penelitian yang telah dilakukan didapatkan sebuah pengetahuan baru yang dapat dijadikan sebagai bahan penelitian selanjutnya.

III. HASIL DAN PEMBAHASAN

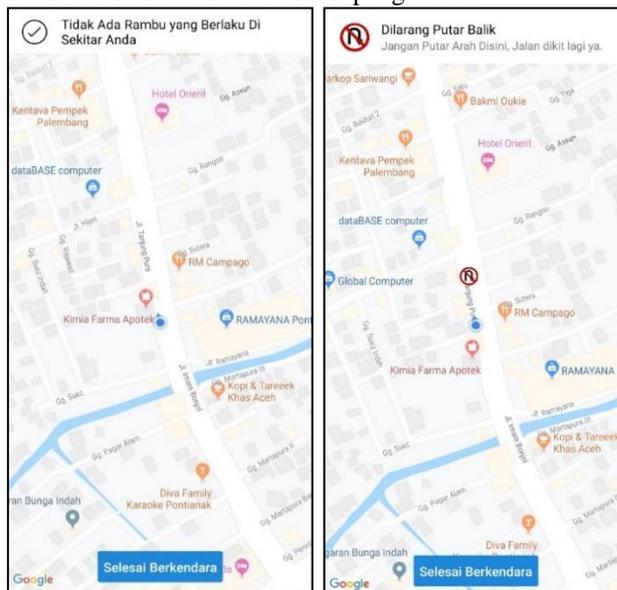
A. Implementasi

Antarmuka halaman utama seperti gambar 4 merupakan halaman pertama yang diakses oleh pengguna. Pada halaman ini terdapat menu pemandu rambu, peta rambu, riwayat rambu dan hapus data perjalanan yang bisa digunakan oleh pengguna ketika ingin menggunakan aplikasi.



Gambar. 5. Halaman Utama Aplikasi menampilkan tiap-tiap menu pada aplikasi.

Antarmuka halaman pemandu rambu pada gambar 6 merupakan halaman yang tampil ketika pengguna memilih menu pemandu rambu. Halaman pemandu rambu sendiri berfungsi untuk memberikan informasi rambu yang ditemui dalam bentuk tampilan dan suara. Tampilan yang dimuat berupa daftar rambu yang ditemui dan peta yang berisi rambu berdasarkan lokasi pengendara.



Gambar. 6. Halaman Pemandu Rambu menampilkan informasi rambu yang berlaku disekitar lokasi pengendara ketika sedang berjalan.

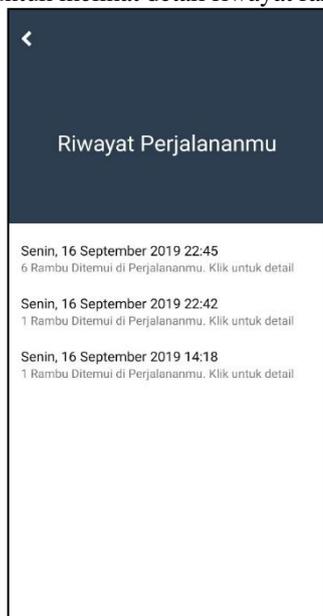
Antarmuka halaman peta rambu merupakan halaman yang dapat diakses pengguna untuk menampilkan peta

rambu. Halaman peta rambu pada gambar 7 berisikan peta Google dengan rambu-rambu yang berada di jalan. Ikon rambu dapat diklik untuk mendapatkan arti mengenai rambu tersebut.



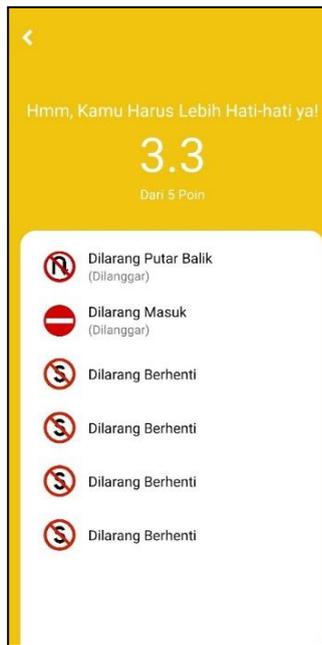
Gambar 7. Halaman Peta Rambu menampilkan informasi lokasi rambu-rambu yang berlaku di jalan.

Halaman riwayat perjalanan pada gambar 8 adalah daftar perjalanan pengguna saat menjalankan aplikasi ketika berkendara. Tampilan yang dimuat berupa daftar riwayat perjalanan pengguna berdasarkan waktu pencatatan yang dapat di klik untuk melihat detail riwayat rambu.



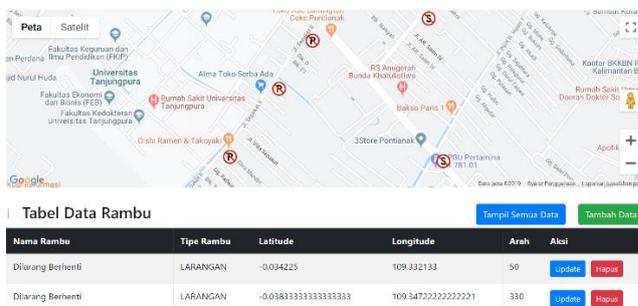
Gambar 8. Halaman Riwayat Perjalanan menampilkan riwayat perjalanan yang dilakukan oleh pengendara.

Antarmuka halaman riwayat detail rambu pada gambar 9 merupakan halaman yang berfungsi untuk menampilkan data rambu yang ditemui berdasarkan riwayat perjalanan yang dipilih sebelumnya. Pengguna dapat melihat lokasi rambu yang ditemui dengan mengklik pada daftar rambu.



Gambar 9. Halaman Detail Riwayat Rambu menampilkan rambu-rambu yang ditemui ketika pengendara sedang berkendara dan menampilkan penilaian terhadap kepatuhan rambu lalu lintas

Antarmuka halaman pengelolaan data rambu pada gambar 10 merupakan halaman yang berfungsi untuk menampilkan data rambu yang telah didaftarkan. Pengguna dapat melakukan pengelolaan data rambu, yakni, melihat lokasi rambu, menambahkan data rambu, memperbarui data rambu dan menghapus data rambu.



Gambar 10. Halaman Pengelolaan Data Rambu yang digunakan pengelola untuk mengelola data rambu, seperti penambahan, perubahan, dan menghapus data rambu.

**B. Pengujian aplikasi dengan Metode BlackBox**

Pengujian dengan metode *black box* pada aplikasi dilakukan untuk menguji kesesuaian logika yang dibuat dengan hasil yang ditampilkan. Pengujian dilakukan dengan mengakses tiap-tiap menu pada pada aplikasi, lalu melakukan pengamatan terhadap respon yang diberikan oleh aplikasi.

Berdasarkan Tabel I terlihat bahwa hasil pengujian memberikan hasil yang sesuai dengan yang diharapkan. Semua menu yang dipilih dapat berjalan dengan baik.

TABEL I  
TABEL PENGUJIAN SETIAP MENU APLIKASI

No.	Hal Yang Dilakukan	Hasil Yang Diharapkan	Hasil
1.	Menjalankan fungsi pemandu rambu	Aplikasi menginformasikan rambu di sekitar pengendara	Berhasil
2.	Menjalankan menu peta rambu	Aplikasi menampilkan peta dengan rambu-rambu sesuai lokasi	Berhasil
3.	Menjalankan fungsi riwayat perjalanan	Aplikasi menampilkan data riwayat perjalanan	Berhasil
4.	Melihat detail rambu pada riwayat perjalanan	Aplikasi menampilkan detail rambu dan lokasinya	Berhasil

TABEL II  
HASIL PENGUJIAN

No.	Hal Yang Dilakukan	Hasil Yang Diharapkan	Hasil	Keterangan
1.	Mendeteksi lokasi pengendara	Aplikasi memuat peta digital dan menampilkan lokasi pengendara secara realtime	Berhasil	
2.	Menjalankan Pelacakan pada Latar Belakang Aplikasi	Aplikasi mencatat dan menyembunyikan informasi rambu yang ditemui ketika aplikasi sedang di <i>minimize</i>	Tidak Berhasil	Modul <i>watchPosition</i> tidak dapat berjalan pada mode latar belakang. Penggunaan modul <i>startLocationUpdatesAsync</i> tidak dapat berjalan semestinya.
3.	Menemui Rambu	Aplikasi mengeluarkan informasi rambu berupa tampilan dan suara	Berhasil	
4.	Meninggalkan Rambu	Aplikasi menghapus informasi rambu pada tampilan	Berhasil	
5.	Melanggar Rambu	Aplikasi mendeteksi pelanggaran rambu yang dilakukan pengendara	Berhasil	

6.	Menginformasikan pelanggaran rambu	Aplikasi mengeluarkan informasi pelanggaran rambu berupa tampilan dan suara serta mencatat pelanggaran	Berhasil	
7.	Mencatat rambu yang ditemui	Aplikasi menyimpan data rambu yang ditemui	Berhasil	
8.	Memberikan penilaian perilaku pengendara	Aplikasi menghitung nilai perilaku berdasarkan rambu yang dilanggar	Berhasil	

Berdasarkan Tabel II terlihat bahwa hasil pengujian memberikan hasil yang sesuai dengan yang diharapkan. Aksi pada setiap fungsi dapat berjalan dengan baik, hanya fungsi dapat berjalan pada latar belakang yang memiliki kendala.

TABEL III  
TABEL PENGUJIAN SETIAP AKSI PENENTUAN PELANGGARAN

No.	Hal Yang Dilakukan	Hasil Yang Diharapkan	Hasil	Keterangan
1.	Menemui rambu larangan berhenti dan tidak berhenti.	Aplikasi tidak mendeteksi adanya pelanggaran rambu.	Berhasil	
2.	Menemui rambu larangan berhenti dan berhenti di sekitar rambu berlaku.	Aplikasi menentukan bahwa pengguna melanggar rambu dan memberitahukan bahwa telah melanggar rambu.	Berhasil	
3.	Menemui rambu larangan putar balik dan terus berjalan kedepan.	Aplikasi tidak mendeteksi adanya pelanggaran rambu.	Berhasil	
4.	Menemui rambu larangan putar balik dan memutar arah.	Aplikasi menentukan bahwa pengguna melanggar rambu dan memberitahukan bahwa telah	Berhasil	

No.	Hal Yang Dilakukan	Hasil Yang Diharapkan	Hasil	Keterangan
5.	Menemui rambu pembatas kecepatan dan tidak melebihi kecepatan yang ditentukan	Aplikasi tidak mendeteksi adanya pelanggaran rambu.	Berhasil	
6.	Menemui rambu pembatas kecepatan dan berjalan melebihi kecepatan yang ditentukan	Aplikasi menentukan bahwa pengguna melanggar rambu dan memberitahukan bahwa telah melanggar rambu.	Berhasil	
7.	Menemui rambu larangan belok dan mengikuti arah lain.	Aplikasi tidak mendeteksi adanya pelanggaran rambu.	Berhasil	
8.	Menemui rambu larangan belok dan berbelok ke arah yang dilarang.	Aplikasi menentukan bahwa pengguna melanggar rambu dan memberitahukan bahwa telah melanggar rambu.	Berhasil	
9.	Menemui rambu perintah belok dan mengikuti arah yang ditentukan	Aplikasi tidak mendeteksi adanya pelanggaran rambu.	Berhasil	
10.	Menemui rambu perintah belok dan mengikuti arah lain.	Aplikasi menentukan bahwa pengguna melanggar rambu dan memberitahukan bahwa telah melanggar rambu.	Berhasil	
11.	Menemui rambu dilarang masuk dan berbelok atau memutar arah.	Aplikasi tidak mendeteksi adanya pelanggaran rambu.	Berhasil	
12.	Menemui rambu dilarang masuk dan	Aplikasi menentukan bahwa pengguna	Berhasil	

	tetap terus berjalan kedepan.	melanggar rambu dan memberitahukan bahwa telah melanggar rambu.		
No.	Hal Yang Dilakukan	Hasil Yang Diharapkan	Hasil	Keterangan
13.	Menemui rambu wajib mengikuti arah bundaran dan mengikuti arah bundaran.	Aplikasi tidak mendeteksi adanya pelanggaran rambu.	Berhasil	
14.	Menemui rambu wajib mengikuti arah bundaran dan memutar arah berlawanan.	Aplikasi menentukan bahwa pengguna melanggar rambu dan memberitahukan bahwa telah melanggar rambu.	Tidak Berhasil	Posisi pemasangan titik rambu yang tidak tentu dan diameter dari bundaran berpengaruh pada penentuan pelanggaran. Hal ini menyebabkan pelanggaran tidak terdeteksi

Berdasarkan Tabel III terlihat bahwa hasil pengujian hampir semua memberikan hasil yang sesuai dengan yang diharapkan. Aplikasi dapat menentukan pelanggaran dengan baik. Akan tetapi letak pemasangan rambu wajib mengikuti arah bundaran yang tidak pasti dan diameter bundaran yang berbeda-beda menyebabkan inkonsistensi aplikasi dalam menentukan pelanggaran rambu jenis ini.

TABEL IV  
TABEL PENGUJIAN MENU PENGELOLAAN DATA RAMBU

No.	Hal Yang Dilakukan	Hasil Yang Diharapkan	Hasil
1.	Mengakses menu kelola rambu admin	Aplikasi menampilkan peta rambu dengan menu yang dapat dilakukan	Berhasil
2.	Menambah dan memperbarui data rambu	Aplikasi menyimpan data penambahan dan pembaruan rambu	Berhasil
3.	Menghapus data rambu	Aplikasi menghapus data yang dipilih	Berhasil

Berdasarkan hasil pengujian yang tertera pada Tabel IV terlihat bahwa pengujian yang dilakukan menunjukkan aplikasi dapat mengelola data rambu. Admin dapat menambahkan data rambu, pembaruan data rambu dan menghapus data rambu pada menu tersebut.

Berdasarkan Tabel V pada pengujian kompatibilitas sensor dapat ditarik kesimpulan bahwa aplikasi dapat berjalan minimal pada sistem operasi android versi 5.0. Aplikasi tidak kompatibel dengan android dibawah versi 5.0. Hal ini dikarenakan versi tersebut sudah terlalu lama dan tidak mendukung paket dari react-native terbaru.

TABEL V  
TABEL PENGUJIAN KOMPATIBILITAS VERSI ANDROID

No.	Hal Yang Dilakukan	Hasil Yang Diharapkan	Hasil
1.	Android 4.4 KitKat	Tidak Berhasil	Tidak kompatibel dengan versi SDK
2.	Android 5.0 Lollipop	Berhasil	
3.	Android 6.0 Marshmallow	Berhasil	
4.	Android 7.1 Nougat	Berhasil	
5.	Android 8.1 Oreo	Berhasil	
6.	Android 9.0 Pie	Berhasil	

Dari pengujian yang telah dilakukan, adapun hasil analisis yang didapat dari aplikasi peringatan rambu lalu lintas adalah sebagai berikut:

1. Aplikasi dapat memandu rambu yang berlaku di sekitar pengendara ketika aplikasi sedang dijalankan. Pengguna hanya menjalankan aplikasi dan aplikasi akan menemukan rambu yang berlaku di sekitar perjalanan.
2. Aplikasi hanya dapat memandu rambu saat aplikasi dijalankan, aplikasi tidak dapat memandu rambu ketika aplikasi berjalan di proses latar belakang. Hal ini dikarenakan penggunaan modul *watchPosition* yang hanya dapat berjalan ketika aplikasi aktif. Modul yang didukung latar belakang adalah *startLocationUpdatesAsync*, akan tetapi hasil keluaran dari modul tersebut sering kali gagal dan terlambat dalam mengeluarkan data lokasi. Hal ini menyebabkan fungsi-fungsi pada aplikasi tidak berjalan dengan baik.
3. Jumlah rambu yang didukung sebanyak 33 jenis berupa rambu peringatan, larangan dan perintah.
4. Aplikasi dapat mendeteksi 7 macam pelanggaran rambu umum, yakni mendeteksi berhenti, pembatas kecepatan, putar balik, larangan belok, wajib belok, larangan masuk, dan wajib mengikuti arah bundaran.
5. Aplikasi yang dibangun tidak memerlukan perangkat tambahan seperti sensor, mikrokontroler, dll. Cukup menggunakan jaringan internet yang umumnya sudah tersedia di perangkat smartphone.
6. Aplikasi dapat memberikan penilaian perilaku pengendara terhadap rambu yang ditemui, hal ini dapat membuat aplikasi lebih menarik dan dapat mendorong pengguna agar lebih patuh terhadap rambu.
7. Sistem menyediakan fitur khusus yang digunakan oleh admin untuk pengelolaan data rambu.

#### IV. KESIMPULAN

##### A. Kesimpulan

Setelah dilakukan pengujian dan analisa aplikasi yang telah dibuat, dapat disimpulkan bahwa aplikasi yang dibuat dapat digunakan sebagai pemandu rambu ketika sedang berkendara. Dalam penggunaannya, aplikasi akan mengambil lokasi terkini pengendara dan mencari rambu yang berlaku di sekitar dengan algoritma pencocokan terhadap kondisi pengendara. Aplikasi akan menampilkan

informasi rambu yang berlaku di sekitar pengendara ketika rambu berada 40 meter di depan posisi pengendara dan kesesuaian antara arah ruas jalan dan arah pengendara. Informasi rambu akan menghilang ketika pengendara melewati jarak berlaku rambu yang telah ditentukan atau pengendara dideteksi telah tidak berada pada jalur dimana rambu tersebut berlaku.

Berdasarkan pengujian aplikasi pada beberapa perangkat, Aplikasi dapat dipasang dan menjalankan fitur-fitur dengan baik pada perangkat *mobile* pada sistem operasi android dengan minimum versi *Android 5.0*. Fitur-fitur tersebut meliputi pemandu rambu, melihat peta rambu, mengakses riwayat perjalanan beserta rambu yang ditemui dan memberikan penilaian perilaku pengendara berdasarkan rambu yang ditemui dan dilanggar.

#### DAFTAR PUSTAKA

- [1] D. Handayani, R. O. Ophelia dan W. Hartono, "Pengaruh Pelanggaran Lalu Lintas Terhadap Potensi Kecelakaan Pada Remaja Pengendara Sepeda Motor," *e-Jurnal Matriks Teknik SIPIL*, pp. 838-843, 2017.
- [2] R. O. Sasambe, "Kajian Terhadap Penyelesaian Pelanggaran Peraturan Lalu Lintas Oleh Kepolisian," *Lex Crimen*, vol. V, pp. 82-90, 2016.
- [3] S. Sadono, "Budaya Tertib Berlalu-Lintas "Kajian Fenomenologis Atas Masyarakat Pengendara Sepeda Motor Di Kota Bandung"," *Channel*, vol. 4, pp. 61-79, 2016.
- [4] Republik Indonesia, "Peraturan Menteri Perhubungan Republik Indonesia Nomor PM 13 Tentang Rambu Lalu Lintas," pp. 1-161, 2014.
- [5] K. T. Martono, "Rancang Bangun Media Pengenalan Rambu-Rambu Lalu Lintas Dengan Memanfaatkan Teknologi Augmented Reality," *Jurnal Teknologi Informasi-Aiti*, vol. 14, pp. 49-58, 2016.
- [6] C. Supriatna dan Sutono, "Media Sosialisasi Rambu-Rambu Lalulintas Dengan Metode Augmented Reality Berbasis Android," *Media Jurnal Informatika*, vol. 8, pp. 9-17, 2016.
- [7] K. K. Rafiah, R. Magdalena dan N. Andini, "Deteksi Rambu-Rambu Lalu Lintas Untuk Membantu Pengguna Jalan Raya Dengan Algoritma Camshift," *e-Proceeding of Engineering*, vol. 2, pp. 429-435, 2015.
- [8] M. J. Budiman, S. Walukow dan I. Patolenganeng, "Pengembangan Sistem Pemberi Isyarat Jenis Rambu Lalu Lintas Bagi Pengguna kendaraan Bermotor Berbasis GPS," *Industrial Research Workshop and National Seminar Politeknik Negeri Bandung*, pp. 396-400, 2017.
- [9] S. H. Nazruddin, Rancang Bangun Aplikasi Multiplatform, Bandung: Informatika, 2015.
- [10] S. Alfeno dan R. E. C. Devi, "Implementasi Global Positioning System (GPS) dan Location Based Service (LBS) pada Sistem Informasi Kereta Api untuk Wilayah Jabodetabek," *JURNAL SISFOTEK GLOBAL*, vol. 7, pp. 27-33, 2017.
- [11] N. Rachmat, A. Muhajirin dan Mukhsin, "Tracking Kendaraan Mobil Dengan Pemanfaatan GPS Berbasis Android," *Jurnal Kajian Ilmiah UBJ*, vol. 15, pp. 103-120, 2015.
- [12] M. R. Fahlivi dan Athariq, "Sistem Tracking Position Berdasarkan Titik Koordinat GPS Menggunakan Smartphone," *Jurnal Infomedia*, vol. 2, pp. 25-29, 2017.
- [13] M. A. F. C. Firdous, "Sistem Pencarian Jarak Terdekat Menggunakan Metode Haversine Berbasis Android," *Artikel Skripsi Universitas Nusantara PGRI Kediri*, pp. 1-8, 2017.
- [14] Kholil, "Pemanfaatan Sistem Informasi Geografis (SIG) Dalam Aplikasi Pelaporan dan Pelacakan Kejahatan Berbasis Android," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 6, pp. 51-58, 2017.

- [15] C. A. Pamungkas, "Aplikasi Penghitung Jarak Koordinat Berdasarkan Latitude dan Longitude Dengan Metode Euclidean Distance dan Metode Haversine," *Jurnal INFORMA Politeknik Indonusa Surakarta*, vol. 5, pp. 8-13, 2019.