

DESAIN DAN IMPLEMENTASI APLIKASI ANDROID UNTUK PENENTUAN RUTE DAN PELACAKAN POSISI KENDARAAN MENGGUNAKAN PERANGKAT GPS SECARA INTERAKTIF

Design and Implementation of Android Application for Route Determination and Tracking Position on Vehicle using GPS Hardware Interactively

Mochamad Soebagja Budiana¹, Asep Mulyana, ST., MT.², Aris Hartaman, ST., MT.³

¹²³Prodi D3 Teknik Telekomunikasi, Fakultas Ilmu Terapan, Universitas Telkom

¹begiooo@gmail.com, ²asepm267@gmail.com, ³aris.hartaman@gmail.com

Abstrak

Aplikasi pemilihan rute trafik jalan raya pada *smart phone* dan kendaraan menggunakan algoritma rute terpendek tanpa mempertimbangkan kepadatan lalu lintas. Di sisi lain aplikasi yang ada saat ini mendapatkan data kepadatan lalu lintas dari fitur GPS pada *smart phone* pengendara. Padahal belum tentu semua pengendara mengaktifkan fitur GPS. Selain itu, aplikasi pemilihan rute pada kendaraan hanya memiliki fitur pemilihan rute saja. Walaupun kendaraan tersebut terpasang perangkat GPS, apabila terjadi kehilangan maka pemilik kendaraan tidak bisa melacak keberadaan kendaraan tersebut.

Untuk mengatasi permasalahan diatas, maka dibuatlah suatu aplikasi Android yang mampu mencari rute tercepat dengan mempertimbangkan kepadatan lalu lintas, mendapatkan data kepadatan lalu lintas dari kecepatan rata-rata kendaraan yang sudah terpasang perangkat GPS, dan melacak kendaraan yang hilang. Untuk merealisasikan monitoring kepadatan lalu lintas jalan raya, setiap kendaraan harus dipasang perangkat GPS agar data kepadatan lalu lintas yang didapat adalah data yang pasti dan *real-time*.

Proyek akhir ini menghasilkan sebuah aplikasi pemilihan rute trafik jalan raya dan pelacakan kendaraan (*tracking*) dan *database* yang sudah terintegrasi dengan *server* dan perangkat GPS yang dalam pengujian fungsionalitas dan implementasinya, tingkat keberhasilan fungsionalitas pada setiap *activity* dan *database* sebesar 100%. Dapat disimpulkan bahwa aplikasi dan *database* dapat berfungsi dengan baik.

Kata Kunci : Android, GIS, GPS, Rute, Kendaraan

Abstract

Road traffic selection applications on smart phones and vehicles use the shortest route algorithm without considering the traffic density. On the other hand the current applications get traffic density data from the GPS feature on the smart phone riders. Though not necessarily all riders activate the GPS feature. In addition, the route selection application on the vehicle only features route selection only. Although the vehicle is installed GPS device, in case of loss then the owner of the vehicle can not track the existence of the vehicle.

To overcome the above problems, then made an Android application that is able to find the fastest route by considering the density of traffic, get traffic density data from the average speed of vehicles that have been installed GPS devices, and track the missing vehicle. To realize the highway traffic density monitoring, each vehicle must be installed GPS device so that the data traffic density obtained is data that is sure and *real-time*.

This final project generates an application and database that is being built with GPS and servers that in testing the functionality and implementation, the success rate of functionality on each activity and database is 100%. Can be concluded application and database can function well.

Keywords : Android, GIS, GPS, Route, Vehicle

1. Pendahuluan

Untuk memenuhi kebutuhan pengendara diperlukan aplikasi yang mampu memilih rute trafik jalan raya. Maka dari itu dibuatlah aplikasi pada kendaraan untuk memenuhi kebutuhan tersebut dimana aplikasi tersebut mendapatkan data posisi awal dari perangkat GPS yang sudah terpasang pada *dashboard*. Akan tetapi, aplikasi ini masih menggunakan algoritma pemilihan rute trafik jalan raya terpendek. Artinya walaupun pada jalur yang akan dilalui terdapat kepadatan lalu lintas, aplikasi tersebut tetap akan melalui jalur tersebut. Selain aplikasi yang sudah terdapat pada kendaraan, aplikasi pada *smart phone* yang ada menggunakan data kepadatan lalu lintas yang didapat dari fitur GPS pada *smart phone* pengendara dimana perangkat GPS ini bersifat temporer yang artinya perangkat GPS hanya akan aktif selama pengendara menyalakan fitur GPS pada *smart phone*-nya. Di sisi lain, perangkat GPS yang sudah terpasang pada *dashboard* kendaraan hanya memiliki fitur untuk mencari rute saja. Walaupun

pada *dashboard* kendaraan sudah terpasang perangkat GPS, apabila terjadi kehilangan maka pemilik kendaraan tetap tidak melacak kendaraannya.

Untuk mengatasi permasalahan di atas, dibuat suatu aplikasi yang mampu memilih rute tercepat dengan mempertimbangkan kepadatan lalu lintas trafik jalan raya, mendapatkan data kepadatan lalu lintas yang akurat berdasarkan kecepatan rata-rata perangkat GPS, dan mampu melacak kendaraan yang hilang dengan dua mode, yaitu mode *on demand* dan mode periodik. Akan tetapi dalam implementasi untuk mendapatkan data kepadatan lalu lintas, setiap kendaraan harus terpasang perangkat GPS. Dalam pembuatan sistem ini terdapat empat bagian pengerjaan sistem, yaitu pengerjaan sistem pada perancangan GPS, pembuatan *server*, pembuatan database, dan pembuatan aplikasi pada Android. Pengerjaan sistem yang dikerjakan oleh penulis adalah bagian pembuatan *database* dan pembuatan aplikasi pada Android.

2. Dasar Teori

2.1. Android

Android merupakan sistem operasi *open source* berbasis Linux yang dirancang untuk perangkat bergerak (*mobile*). Android memungkinkan pengguna untuk memasang aplikasi pihak ketiga baik yang diperoleh dari Google Play, Amazon Appstore, atau dengan mengunduh dan memasang file berekstensi *.apk* dari situs pihak ketiga. Arsitektur Android terdiri dari beberapa lapisan dimana tiap lapisan terdiri dari program yang memiliki fungsi yang berbeda. Seperti gambar 2.1., berikut lapisan Android dari yang paling dalam hingga lapisan terluar.^[1]

- a. *Linux Kernel*
- b. *Libraries dan Android Runtime*
- c. *Application Framework*
- d. *Application*

2.2. Java

Bahasa pemrograman Java banyak mengadopsi perintah yang terdapat pada bahasa pemrograman C dan C++ namun dengan perintah dengan model yang lebih sederhana. Bahasa pemrograman Java ini berorientasi objek (*OOP-Object Oriented Programming*). Paradigma OOP menyelesaikan masalah dengan merepresentasikan masalah ke model objek sehingga mempunyai dua model parameter, yaitu *variable* dan *method*. *Variable* merupakan objek dan *method* merupakan perintah yang akan dijalankan oleh objek atau *variable*.

2.3. Java Development Kit (JDK)

JDK merupakan kepanjangan dari *Java Development Kit* yang mempunyai fungsi untuk mengkompilasi kode - kode *java* menjadi aplikasi *java*. Tanpa adanya *JDK* maka kode-kode *java* yang sudah di buat tidak dapat di jadikan aplikasi berbasis *Java*. Sedangkan *SDK* merupakan *tools* bagi para *programmer* yang ingin mengembangkan aplikasi berbasis *google android*.^[3]

2.4. Software Development Kit (SDK)

Android SDK mencakup seperangkat alat pengembangan yang komprehensif. *Android SDK* terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, contoh kode, dan *tutorial*. Saat ini *Android* sudah mendukung arsitektur x86 pada *Linux* (distribusi *Linux* apapun untuk *desktop* modern), *Mac OS X* 10.4.8 atau lebih, *Windows XP* atau *Vista*.^[3]

2.5. Hypertext Propocessor (PHP)

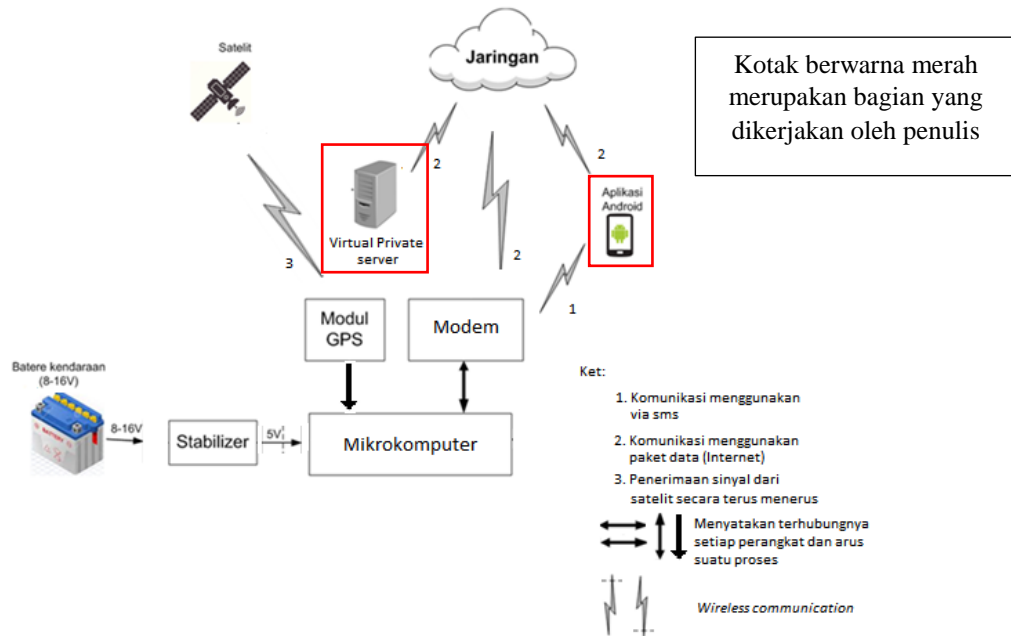
PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web*. Aplikasi *web* adalah aplikasi yang disimpan dan dieksekusi (oleh *PHP Engine*) di lingkungan *web server*. Setiap permintaan yang dilakukan oleh *user* melalui aplikasi klien (*web browser*) akan di respon oleh aplikasi web dan hasilnya akan di kembalikan lagi ke hadapan *user*. Dengan aplikasi *web*, halaman yang tampil di layar *web browser* dapat bersifat dinamis, tergantung dari nilai data atau parameter yang dikirimkan oleh *user* ke *web server*.^[2]

3. Perancangan Sistem

3.1. Blok Diagram Sistem Keseluruhan

Pada bab ini menguraikan langkah-langkah sistematis yang dilakukan dalam penelitian. Perancangan sistem merupakan kerangka dasar dari tahapan penyelesaian Proyek Akhir. Perancangan sistem penulisan pada Proyek

Akhir ini mencakup semua kegiatan yang dilaksanakan untuk memecahkan masalah atau melakukan proses analisa terhadap permasalahan. Berikut adalah blok sistem secara keseluruhan.



Gambar 1 Blok Diagram Sistem Keseluruhan

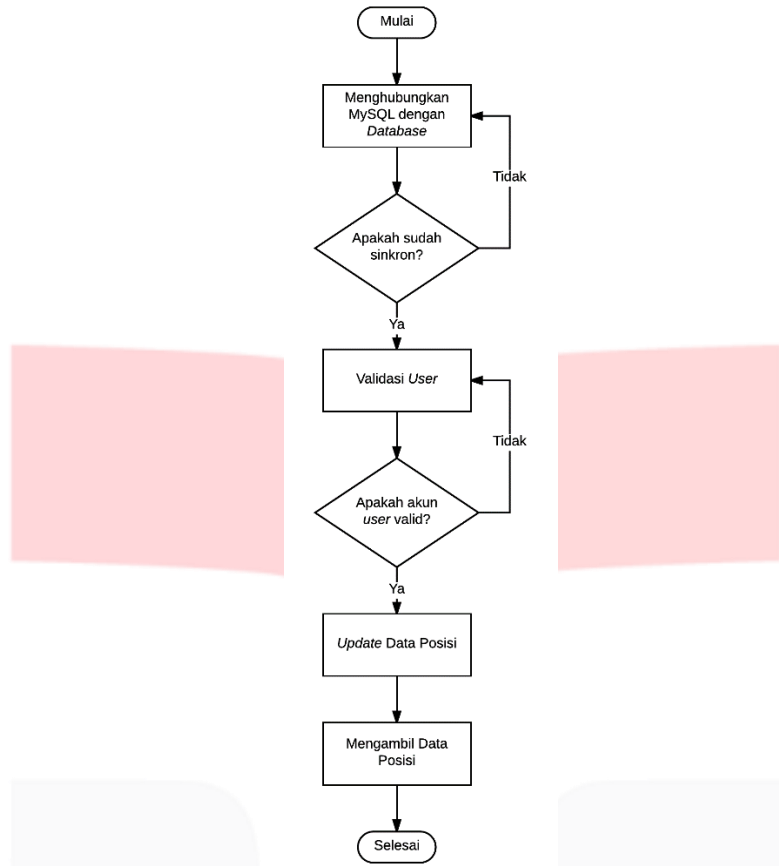
Keterangan Blok Diagram Sistem :

- Modul GPS berfungsi untuk menerima data lokasi dari satelit
- Mikrokomputer berfungsi untuk menterjemahkan data posisi, penyimpanan data posisi, dan tempat eksekusi program.
- Modem berfungsi untuk menerima sms dan mengkoneksikan mirkokomputer dengan internet.
- VPS digunakan sebagai tempat untuk penyimpanan data posisi di *database*.
- Aplikasi Android mengambil data posisi perangkat GPS dari *database*.

3.2. Perancangan *Database*

Database pada *server* berfungsi sebagai penyimpan data posisi (*latitude* dan *longitude*) dari perangkat GPS, dimana data posisi (*latitude* dan *longitude*) ini didapat dari perangkat GPS yang mengirimkan data posisi tersebut ke *server* secara *real-time*. Agar data posisi (*latitude* dan *longitude*) yang dikirim oleh perangkat GPS dapat disimpan didalam *database* pada *server* dan dapat diakses melalui halaman web, maka *database* harus dihubungkan dengan MySQL. Akan tetapi untuk mengambil data posisi (*latitude* dan *longitude*) yang tersimpan di dalam *database*, maka *user* harus melakukan *login* menggunakan akun yang telah dibuat. Apabila akun *user* valid, maka *user* dapat mengakses data posisi (*latitude* dan *longitude*) yang ada di dalam *database*. Validasi ini dilakukan di dalam file *login.php*.

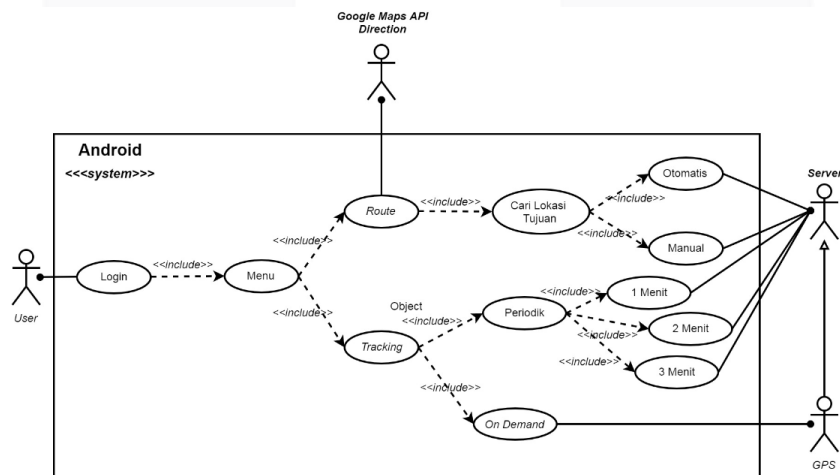
Agar data posisi (*latitude* dan *longitude*) perangkat GPS tidak mengalami kesalahan posisi, maka data posisi (*latitude* dan *longitude*) akan selalu di-*update*. Setelah data di-*update* maka data yang sudah di-*update* tersebut akan disimpan kembali di dalam *database* dan pada saat yang bersamaan data pada *database* yang sudah di-*update* tersebut akan diberikan kepada *user* dalam format *JSONArray* melalui halaman web. Data posisi (*latitude* dan *longitude*) ini akan di-*update* dan disimpan menggunakan file *update.php*, sedangkan untuk memberikan data posisi (*latitude* dan *longitude*) kepada *user* melalui halaman web menggunakan file *ambil_data.php*.



Gambar 2 Diagram Alir Perancangan Database

3.3. Use Case Diagram

Berikut merupakan diagram use case dari aplikasi pemilihan rute trafik jalan raya.



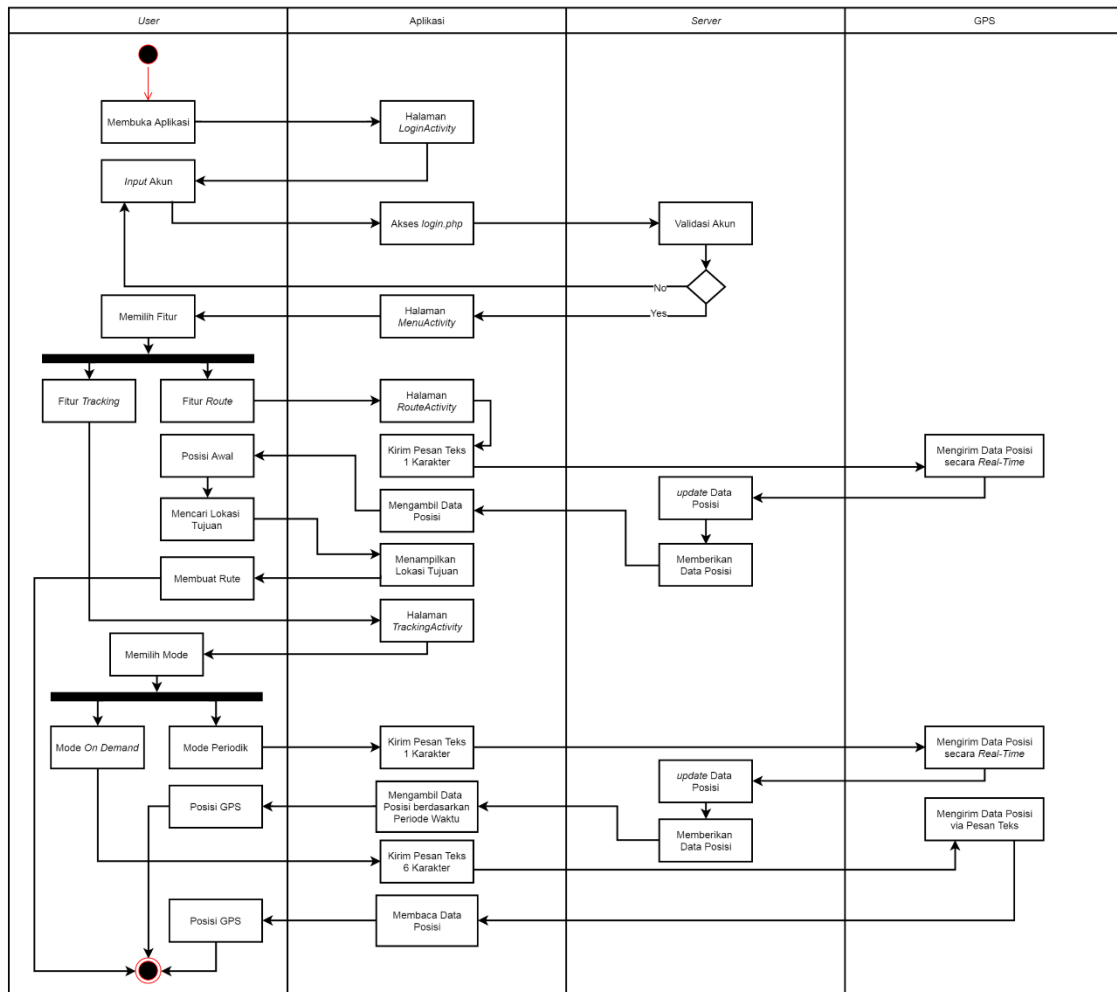
Gambar 3 Diagram Use Case

- User login sesuai dengan username dan password yang telah dibuat.
- Setelah user melakukan login, user akan masuk ke dalam halaman home activity dimana pada halaman ini user dapat memilih dua fitur yang disediakan oleh aplikasi, yaitu fitur pemilihan rute trafik jalan raya dan fitur tracking kendaraan.
- Jika user memilih fitur pemilihan rute trafik jalan raya, maka user akan masuk ke dalam halaman rute activity. Pada halaman ini user harus menentukan lokasi tujuan terlebih dahulu untuk membuat rute dari posisi sekarang sampai ke posisi tempat tujuan. Posisi sekarang didapat dari data posisi (latitude dan longitude) GPS

yang telah dikirim dan disimpan di *web server* dan data posisi GPS tersebut akan diambil oleh *smartphone* melalui jaringan internet.

- d. Jika *user* memilih fitur *tracking*, maka *user* akan masuk ke dalam halaman *tracking activity*. Pada halaman ini *user* dapat menggunakan dua mode pelacakan, yaitu mode *on demand* dan mode *periodic*. Mode *on demand* merupakan mode pelacakan yang memberikan posisi kendaraan hanya pada saat *user* meminta posisi kendaraan tersebut. Sedangkan mode *periodic* merupakan mode pelacakan yang memberikan posisi kendaraan pada periode waktu yang telah ditentukan oleh *user*. data posisi (*latitude* dan *longitude*) kendaraan ini didapat dari SMS yang dikirim oleh GPS ke *smartphone* yang nantinya akan dibaca oleh *smartphone* sebagai data posisi kendaraan.

3.4. Activity Diagram



Gambar 4. Activity Diagram

Saat *user* membuka aplikasi, sistem aplikasi akan menampilkan halaman *LoginActivity*. Setelah itu *user* harus meng-input-kan *field username* dan *password*. Saat *user login*, maka sistem aplikasi akan mengakses *login.php* untuk melakukan validasi. Jika akun valid, maka *user* akan masuk ke dalam halaman *MenuActivity*. Di dalam halaman ini, *user* dapat memilih dua fitur yang disediakan oleh aplikasi, yaitu fitur pemilihan rute dan fitur pelacakan kendaraan (*tracking*).

Jika *user* memilih fitur pemilihan rute, maka *user* akan masuk ke dalam halaman *RouteActivity*. Halaman ini merupakan tempat *user* untuk membuat rute dari posisi awal sampai ke lokasi tujuan. Saat *user* masuk ke dalam *RouteActivity*, maka sistem aplikasi akan mengirim pesan teks ke perangkat GPS sebanyak satu karakter. Setelah itu perangkat GPS akan mengirim data posisi (*latitude* dan *longitude*) secara *real-time* ke *server*. *Server* akan meng-update data posisi tersebut dan akan memberikan data posisi yang sudah ter-update melalui halaman web dalam bentuk *JSONArray*. Setelah itu aplikasi akan mengambil data posisi untuk dijadikan sebagai posisi awal.

Saat *user* sudah mencari lokasi tujuan, maka sistem aplikasi akan menampilkan lokasi tujuan tersebut. Jika lokasi tujuan sudah ditentukan, maka *user* dapat membuat rute dari posisi awal sampai ke lokasi tujuan.

Jika *user* memilih fitur pelacakan kendaraan (*tracking*), maka *user* akan masuk ke dalam halaman *TrackingActivity*. Di halaman ini *user* dapat memantau posisi kendaraan (dengan asumsi kendaraan sudah terpasang perangkat GPS) menggunakan dua mode, yaitu mode pelacakan *on demand* dan mode pelacakan periodik. Saat *user* menggunakan mode *on demand*, maka aplikasi akan mengirim pesan teks ke perangkat GPS sebanyak enam karakter. Setelah itu perangkat GPS akan mengirim data posisi (*latitude* dan *longitude*) melalui pesan teks. Lalu sistem aplikasi akan membaca data posisi yang sudah dikirim menggunakan pesan teks tersebut yang nantinya data posisi ini akan ditampilkan di halaman *TrackingActivity*.

Saat *user* memilih mode pelacakan periodik, maka sistem aplikasi akan mengirim pesan teks ke perangkat GPS sebanyak satu karakter agar perangkat GPS mengirim data posisi (*latitude* dan *longitude*) secara *real-time* ke *server*. Setelah itu *server* akan meng-*update* data posisi tersebut dan menyimpannya di dalam *database*. Lalu *server* akan memberikan data posisi tersebut melalui halaman web dalam bentuk *JSONArray*. Halaman web ini akan diakses oleh sistem aplikasi yang nantinya data *JSONArray* ini akan diambil oleh sistem aplikasi berdasarkan periode waktu yang telah ditentukan. Data *JSONArray* ini akan dijadikan sebagai posisi perangkat GPS.

4. Keluaran dan Hasil Analisa

4.1. Pengujian Fungsionalitas Sistem

Tujuan dari pengujian fungsionalitas sistem adalah untuk mengetahui apakah sistem sudah berfungsi dengan baik.

Tabel 1 Pengujian Fungsionalitas Sistem

Fungsi	Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan
Login dengan meng-input-kan <i>username</i> dan <i>password</i>	<i>User</i> berhasil masuk ke <i>Activity Menu</i>	<i>User</i> berhasil masuk ke <i>Activity Menu</i>	Sesuai
Menekan tombol <i>Route</i>	<i>User</i> masuk ke <i>Activity Route</i>	<i>User</i> masuk ke <i>Activity Route</i>	Sesuai
Menekan tombol <i>Tracking</i>	<i>User</i> masuk ke <i>Activity Tracking</i>	<i>User</i> masuk ke <i>Activity Tracking</i>	Sesuai
Menekan tombol <i>Logout</i>	<i>User</i> kembali ke <i>Login Activity</i>	<i>User</i> kembali ke <i>Login Activity</i>	Sesuai
Menekan Tombol Cari	<i>User</i> mendapatkan lokasi yang dicari	<i>User</i> mendapatkan lokasi yang dicari	Sesuai
Menekan Tombol Rute	<i>User</i> mendapatkan rute beserta estimasi jarak dan waktu	<i>User</i> mendapatkan rute beserta estimasi jarak dan waktu	Sesuai
	Aplikasi mengirim satu karakter via SMS	Aplikasi mengirim satu karakter via SMS	Sesuai
Men-Tap Layar <i>Smartphone</i>	Aplikasi membuat <i>waypoints</i>	Aplikasi membuat <i>waypoints</i>	Sesuai
Men-Tap Layar <i>Smartphone</i> > 3 kali	Aplikasi menghapus semua <i>waypoints</i>	Aplikasi menghapus semua <i>waypoints</i>	Sesuai
Menekan Tombol Lacak untuk Memulai Pelacakan	Aplikasi mengirim enam karakter via SMS	Aplikasi mengirim enam karakter via SMS	Sesuai
	Aplikasi mulai melacak posisi kendaraan	Aplikasi mulai melacak posisi kendaraan	Sesuai
Menekan Tombol Berhenti Lacak untuk Menghentikan Pelacakan	Aplikasi mengirim dua karakter via SMS	Aplikasi mengirim dua karakter via SMS	Sesuai
	Aplikasi berhenti melacak posisi kendaraan	Aplikasi berhenti melacak posisi kendaraan	Sesuai

Menekan <i>Option 1</i> menit	Aplikasi mengirim tiga karakter via SMS	Aplikasi mengirim tiga karakter via SMS	Sesuai
	Aplikasi menerima data posisi setiap 1 menit sekali	Aplikasi menerima data posisi setiap 1 menit sekali	Sesuai
Menekan <i>Option 2</i> menit	Aplikasi mengirim empat karakter via SMS	Aplikasi mengirim empat karakter via SMS	Sesuai
	Aplikasi menerima data posisi setiap 2 menit sekali	Aplikasi menerima data posisi setiap 2 menit sekali	Sesuai
Menekan <i>Option 3</i> menit	Aplikasi mengirim lima karakter via SMS	Aplikasi mengirim lima karakter via SMS	Sesuai
	Aplikasi menerima data posisi setiap 3 menit sekali	Aplikasi menerima data posisi setiap 3 menit sekali	Sesuai

4.2. Pengujian *Delay* dalam Kondisi Diam

Tabel 2 Pengujian *Delay* dalam Kondisi Diam

No	Halaman	Aksi	Pengujian ke-	<i>Delay</i>
1	<i>Login Activity</i>	Menyinkronisasikan data <i>user</i> dengan data pada <i>database</i> lalu masuk ke dalam halaman <i>Home Activity</i>	1	01.551 <i>second</i>
			2	01.251 <i>second</i>
			3	01.451 <i>second</i>
			4	02.469 <i>second</i>
			5	01.001 <i>second</i>
2	<i>Route Activity</i>	Membuat rute trafik jalan raya	1	01.151 <i>second</i>
			2	01.701 <i>second</i>
			3	01.701 <i>second</i>
			4	01.650 <i>second</i>
			5	01.451 <i>second</i>
3	<i>Tracking Activity</i>	Mengirim SMS dan menerima SMS data posisi perangkat GPS	1	3.760 <i>second</i>
			2	4.570 <i>second</i>
			3	4.450 <i>second</i>
			4	4.500 <i>second</i>
			5	4.460 <i>second</i>

4.3. Pengujian *Delay* dalam Kondisi Bergerak

Tabel 3 Pengujian *Delay* dalam Kondisi Bergerak

No	Halaman	Aksi	Pengujian ke-	<i>Delay</i>
1	<i>Login Activity</i>	<i>User</i> masuk ke dalam aplikasi menggunakan <i>username</i> dan <i>password</i> yang sudah dibuat	1	1.320 <i>second</i>
			2	1.130 <i>second</i>
			3	2.140 <i>second</i>
			4	1.060 <i>second</i>
			5	0.870 <i>second</i>
2	<i>Route Activity</i>	Membuat rute trafik jalan raya	1	1.310 <i>second</i>
			2	1.210 <i>second</i>
			3	1.210 <i>second</i>
			4	1.420 <i>second</i>
			5	1.360 <i>second</i>
3	<i>Tracking Activity</i>	Mengirim SMS dan menerima SMS data posisi perangkat GPS	1	4.260 <i>second</i>
			2	4.770 <i>second</i>

			3	4.450 <i>second</i>
			4	4.650 <i>second</i>
			5	4.500 <i>second</i>

5. Kesimpulan dan Saran

5.1. Kesimpulan

Kesimpulan yang dapat diambil dari proyek akhir Aplikasi Pemilihan Rute Trafik Jalan Raya Berbasis GIS pada Android adalah:

- Aplikasi dapat mengambil data posisi (*latitude* dan *longitude*) dalam kondisi bergerak ataupun diam.
- Berjalannya fungsi interaktif, yaitu *user* dapat menerima dan mengirim pesan ke arah *user* dan sebaliknya.
- Aplikasi dapat mengambil data posisi (*latitude* dan *longitude*) dari halaman web dan pesan teks.
- Terdapat dua fitur pada aplikasi, yaitu fitur pemilihan rute trafik jalan raya dan fitur *tracking* kendaraan.
- Aplikasi dapat membuat rute dari posisi awal sampai ke lokasi tujuan.
- Aplikasi dapat menunjukkan estimasi waktu tempuh perjalanan.
- Terdapat dua mode pelacakan, yaitu mode *on demand* dan mode periodik. Mode *on demand* merupakan pelacakan kendaraan dimana perangkat GPS hanya akan mengirimkan data posisi (*latitude* dan *longitude*) hanya pada saat ada permintaan dari *user*. Sedangkan mode periodik merupakan mode pelacakan dimana perangkat GPS akan mengirimkan data posisi (*latitude* dan *longitude*) dalam rentan waktu tertentu (periodik).
- Berdasarkan hasil pengujian fungsionalitas pada setiap halaman, semua fungsi mempunyai persentase tingkat keberhasilan sebesar 100%.
- Berdasarkan hasil pengujian *quality of service* pada kondisi diam, *delay* rata-rata untuk validasi akun adalah 1,544 *second*, sedangkan *delay* rata-rata untuk membuat rute dari lokasi awal ke lokasi tujuan adalah 1,530 *second*, dan untuk *delay* rata-rata pengiriman pesan teks sampai penerimaan pesan teks adalah 4,438 *second*.
- Berdasarkan hasil pengujian *quality of service* pada kondisi diam, *delay* rata-rata untuk validasi akun adalah 1,304 *second*, sedangkan *delay* rata-rata untuk membuat rute dari lokasi awal ke lokasi tujuan adalah 1,530 *second*, dan untuk *delay* rata-rata pengiriman pesan teks sampai penerimaan pesan teks adalah 4,526 *second*.

5.2. Saran

Adapun saran dalam merancang ataupun mengimplementasikan aplikasi pemilihan rute trafik jalan raya untuk pengembangan yang lebih baik lagi, maka berikut saran yang dapat dipertimbangkan:

- Membuat aplikasi agar dapat berfungsi di *platform smartphone* manapun, seperti iOS dan Windows.
- Pengembangan *interface* agar lebih menarik.
- Dapat menggunakan data *map* milik sendiri.
- Dapat membuat rute trafik jalan raya menggunakan algoritma milik sendiri.

Daftar Pustaka :

- [1] Gramlich, Nicolas. 2009. “*Android Programming*”. Germany: Anddev. 8
- [2] Raharjo, Budi. 2011. “*Belajar Otodidak Pemrograman Web dengan PHP + Oracle*”. Bandung: Penerbit Informatika.
- [3] Sfaaat, Nazruddin. 2011. “*Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*”. Bandung : Penerbit Informatika.