

APLIKASI INTERNET OF THINGS UNTUK PENGENDALI DAN PEMANTAU KENDARAAN

INTERNET OF THINGS APPLICATIONS FOR CONTROLING AND MONITORING OF VEHICLES

Rastim¹, Kemas Muslim Lhaksana², Danang Triantoro Murdiansyah³

Prodi S1 Ilmu Komputasi, Fakultas Informatika, Universitas Telkom

rastim@telkomuniversity.ac.id, kemas.muslim@gmail.com, danang@telkomuniversity.ac.id,

Abstract

Internet of Things adalah suatu konsep dimana konektifitas Internet dapat dimanfaatkan sebagai pertukaran informasi antar perangkat, salah satunya yaitu untuk mengantisipasi dari tindak pencurian kendaraan. Dalam mengantisipasi pencurian kendaraan tersebut maka diperlukan suatu perangkat pengendali dan pemantau yang terintegrasi dengan sistem tertanam (embadded system) yang dapat dipantau dari jarak jauh. Aplikasi Internet of Things untuk pengendali dan pemantau kendaraan merupakan teknologi yang memanfaatkan aplikasi sebagai user interface dan perangkat sebagai machine control yang difungsikan untuk mematikan dan melakukan tracking kendaraan. Didalam perangkat pengendali dan pemantau terdapat GPS (global positioning system) yang digunakan untuk melakukan monitoring atau pengawasan posisi kendaraan tersebut. Dalam penelitian tugas akhir ini yaitu mencoba membuat aplikasi Internet of Things untuk pengendali dan pemantau kendaraan dengan menggunakan perangkat raspberry pi 3, modem GSM, GPS mudule, dan relay DC. Hasilnya data posisi yang diterima server secara langsung (realtime) akan ditampilkan melalui aplikasi mobile (android) dengan visualisasi yang terintegrasi google maps API.

Kata kunci : *Internet of Things, raspberry pi, modem gsm, gps module, relay DC, server*

Abstract

Internet of Things is a concept where Internet connectivity can be utilized as an exchange of information between devices, one of which is to anticipate from theft vehicle. In anticipation of no theft of the vehicle is required a device controller and monitor integrated with embedded system (embadded system) that can be monitored remotely. The Internet of Things application for vehicle controllers and monitors is a technology that utilizes the application as a user interface and device as a machine control that is enabled to turn off and track the vehicle. In the controller and monitoring device there is a GPS (global positioning system) used to perform monitoring or supervision of the position of the vehicle. In this thesis research is trying to make the application of Internet of Things to control and monitor the vehicle by using raspberry device pi 3, GSM modem, GPS mudule, and DC relay. The result of position data received by the server directly (realtime) will be displayed through the mobile application (android) with visualization integrated google maps API.

Keywords: Internet of Things, raspberry pi, gsm modem, gps module, DC relay, server.

Pendahuluan

Di dalam dunia teknologi yang semakin berkembang kita dituntut untuk berperan dalam mengikuti perkembangan tersebut. Banyak ilmu teknologi yang dapat kita terapkan dalam kehidupan sehari-hari seperti halnya dalam dunia kriminalitas pencurian kendaraan. Untuk mengatasi hal tersebut maka dibutuhkan suatu perangkat yang dapat mengendalikan dan memantau kendaraan tersebut. Oleh karena itu, peran Internet of Things sebagai pengendali dan pemantau kendaraan sangat dibutuhkan.

Internet of Things merupakan teknologi yang dapat mengkoneksikan suatu peralatan elektronik dengan Internet [1], salah satunya raspberry pi. Raspberry pi adalah perangkat yang menggunakan teknologi pengontrol jarak jauh dengan sistem ARM (advanced RISC machine). Penerapan Internet of Things pada penelitian ini adalah untuk pengendali dan pemantau kendaraan. Perangkat pengendali ini memanfaatkan relay DC, raspberry pi, kendaraan dapat dikendalikan menggunakan relay, dengan relay arus listrik dapat diputus maupun

dihubungkan. Sistem pengendali dapat bekerja untuk menghidupkan atau mematikan kendaraan [2]. Kendaraan yang telah terpasang perangkat Internet of Things dapat diakses secara remote melalui aplikasi mobile (android).

Perangkat pemantau lokasi memanfaatkan teknologi GPS module untuk mendapatkan data posisi latitude, longitude sehingga dapat menampilkan posisi di google maps. Modem GSM yang dikendalikan oleh raspberry pi akan mengirimkan koordinat lokasi ke web server yang kemudian akan ditampilkan melalui visualisasi mobile (android) yang terintegrasi google maps API. Google maps API digunakan untuk menemukan data koordinat berupa maps [3].

Tujuan dari penelitian tugas akhir ini adalah membuat Aplikasi Internet of Things berbasis android untuk mengamankan kendaraan dengan cara melakukan pengendalian dan pemantauan secara remote atau jarak jauh.

1. Tinjauan Pustaka

a. Related work

Sistem pemantau diimplementasikan dengan menampilkan pergerakan kendaraan berdasarkan lokasi di waktu tertentu, teknologi yang digunakan mengkombinasikan aplikasi mobile (android) dengan menggunakan microprocessor, desain perangkat tracking menggunakan GPS dan teknologi GSM/GPRS. Perangkat tersebut akan ditanamkan GPS module untuk mendapatkan koordinat geografi dengan interval waktu tertentu. Modem GSM selanjutnya mengirim data dan meng-update lokasi kendaraan secara realtime [3]. Penelitian secara detail mendeskripsikan desain dan aplikasi sistem tracking kendaraan. Perangkat in-vehicle, server dan aplikasi mobile (android) digunakan untuk sistem tracking kendaraan. Implementasi Internet of Things difokuskan pada akuisi lalu lintas dengan memanfaatkan kemampuan GPS sebagai sensor, komunikasi dengan GPRS/3G dan desain sistem monitoring berbasis mobile (android). Hasilnya menunjukkan interkoneksi antara GPS tracker [4], GPRS/3G dan jaringan Internet sebagai skema Internet of Things yang menerapkan metode desain iteraksi client-server.

b. Teknologi GPS (global positioning system)

GPS (global positioning system) adalah sistem untuk menentukan letak di permukaan bumi dengan bantuan penyalarsan (synchronization) sinyal satelit. System control didefinisikan sebagai pencegah tindakan pencurian menggunakan teknologi GPS. GPS digunakan untuk mendapatkan posisi lokasi kendaraan [7]. Data dikirimkan menggunakan modem GSM perangkat Internet of Things yang digunakan adalah GPS module UBLOX NEO 6MV2 ditampilkan di gambar 2.1, Dengan terintegrasinya relay DC yang digunakan sebagai pengontrol kendaraan jika terjadi tindak pencurian maka kendaraan dapat dimatikan melalui mobile aplikasi. Sistem tertanam Embadded ini menggunakan perangkat raspberry pi dengan antarmuka linux raspbian, menggunakan penyimpanan database secara realtime. Sistem bekerja dengan menggunakan GPS sebagai media pemberi informasi koordinat posisi dan Modem GSM sebagai pengirim informasi koordinat posisi ke database server.



Gambar 2.1 GPS module UBLOX NEO 6MV2[3]

c. Raspberry pi 3 model b

Pada Gambar 2.2 merupakan arsitektur port yang terdapat di perangkat raspberry pi 3 model B, port tersebut yang berfungsi sebagai pintu masuk komponen eksternal raspberry pi. Raspberry pi adalah adalah microprocessor yang mempunyai input/output digital port seperti pada board microcontroller yang terdiri dari port USB untuk mengkoneksikan berbagai perangkat USB seperti keyboard, mouse dan lain-lain. Mini USB port digunakan untuk menghubungkan ke power adaptop. Port ethernet/LAN untuk menghubungkan raspberry pi ke Internet. Port HDMI untuk mengkoneksikan monitor, raspberry pi juga mendukung audio/vidio. Beberapa pin GPIO (general purpose input/output) dapat digunakan untuk mengkoneksikan dengan perangkat elektronik lainnya. Sistem operasi yang direkomendasikan untuk perangkat raspberry pi adalah raspbian yang dijalankan dari SD card pada board raspberry pi. Raspbian merupakan distribusi resmi sistem operasi untuk perangkat raspberry pi berbasis debian [5]. Fungsi raspberry pi secara keseluruhan dapat dijalankan oleh sistem operasi, sistem operasi yang dapat digunakan pada raspberry pi antara lain raspbian, pedora, openElec, raspBMC, RISC OS, arch linux ARM, kali linux [6].

Selain itu raspberry pi sangat bagus untuk membangun web server murah yang dapat digunakan sebagai pengujian atau untuk penyimpanan file. Pada penelitian tugas akhir ini akses Internet yang digunakan adalah

menggunakan modem gsm dengan terkoneksi ke server eksternal atau menggunakan resource lain untuk mengirim dan menampilkan visualisasi koordinat gps.



Gambar 2.2 Raspberry pi 3 model B[3]

d. Relay DC

Raspberry pi mengendalikan perangkat listrik dengan bantuan relay. Relay berfungsi untuk mengontrol arus listrik dengan memberikan tegangan dan arus pada koil, dengan relay arus listrik dapat diputus atau dihubungkan ke perangkat Internet of Things. Perangkat Internet of Things yang digunakan adalah relay 4 channel ditampilkan pada Gambar 2.3. Relay adalah saklar yang dioperasikan secara elektronik perbedaannya dengan saklar biasa adalah jika saklar biasa dioperasikan dengan cara ditekan dengan tangan, relay dioperasikan dengan memberikan sebuah tegangan pemicu menggunakan perangkat komputer, IoT dan arduino [8]. Ketika pengguna menginginkan menyalakan kendaraan maka raspberry pi akan mengirim sinyal high ke pin kontrol relay dan ketika relay menerima sinyal high maka relay akan menutup sirkuit dari rangkaian terkontrol dalam penelitian tugas akhir ini berfungsi sebagai saklar kendaraan. Ketika raspberry pi menginginkan kendaraannya mati maka pinnya akan disetting ke sinyal low (0 volt) dan sebaliknya relay pun akan membuka rangkaian terkontrol yang artinya kendaraan akan mati.



Gambar 2.3 Relay 4 channel[15]

e. GSM Modem Huawei K4203

Memantau dan mengendalikan kendaraan secara remote memerlukan penggunaan data atau internet untuk mengirim data koordinat dan data relay dc ke database server sehingga dibutuhkan perangkat yang disebut dengan Modem GSM. Modem GSM Huawei K4203 merupakan salah satu type GSM (global system mobile communication) menggunakan teknologi berdasarkan waktu untuk mengirim paket data, yang sering disebut dengan timeslot (dimonopoli oleh pengguna selama waktu yang digunakan ketika sedang aktif dan juga ketika sedang diam). Modem GSM Huawei K4203 menggunakan teknologi 3G (Third Generation Technology) untuk mengirim dan menerima data. 3G (Third Generation Technology) merupakan teknologi yang memungkinkan pengiriman dan penerimaan data lebih cepat dibandingkan dengan teknologi 2G (Second Generation Technology). Sistem 3G dapat digunakan untuk transfer data (dalam bentuk paket data) yang berkaitan dengan email, gambar (MMS), text, wireless application protocol (WAP) dan world wide web (WEB).



Gambar 2.4 Modem GSM Huawei K4203

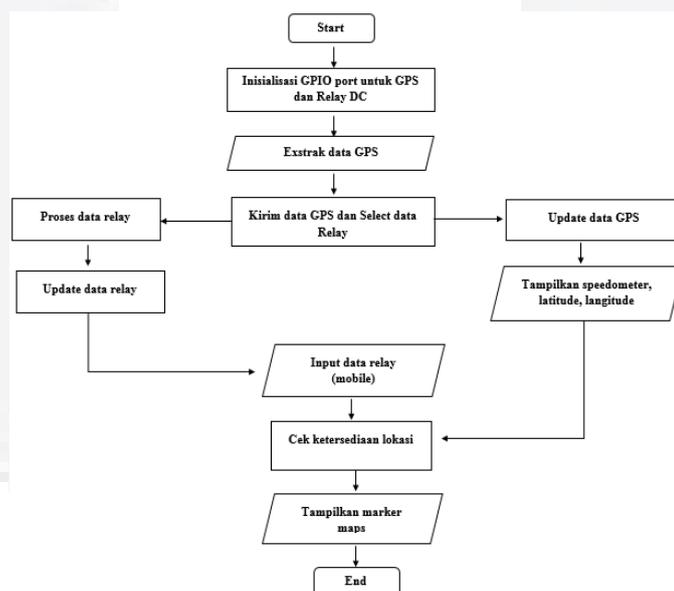
f. Android

Android adalah sistem operasi dengan sumber terbuka, dan google merilis kodenya di bawah lisensi apache. Kode dengan sumber terbuka dan lisensi perizinan pada android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi kustomisasi bahasa pemrograman Java [5]. Dengan memanfaatkan pemrograman java seorang developer dibekali API (Application Protocol Interface) geolocation untuk menampilkan informasi pemetaan koordinat yang digunakan dalam pengembangan aplikasi berbasis maps[11].

g. Database MySQL

Database MySQL adalah sistem manajemen database SQL (Structured Query Language) yang bersifat Open Source dan paling populer saat ini. Sistem database MySQL mendukung beberapa fitur seperti multithreaded, multi-user, dan SQL Database Management System (DBMS). Database ini dibuat untuk keperluan sistem database yang cepat, handal dan mudah digunakan, database mysql digunakan untuk menambahkan, menampilkan status dari perangkat Internet of Things [9]. Selama komunikasi dengan database server, data yang tersimpan di database akan dikonversi kedalam format JSON (JavaScript Object Nation)[10] yang akan diteruskan menjadi objek text yang dapat diakses menggunakan perangkat aplikasi berbasis android (mobile). Dalam penelitian ini database mysql akan digunakan sebagai tempat penyimpanan data users, data relay dc dan data koordinat lokasi.

3. Perancangan Sistem



Gambar 3.1 Diagram alir perancangan aplikasi Internet of Things untuk pengendali dan pemantau kendaraan

Berikut merupakan penjelasan dari perancangan sistem aplikasi pengendali dan pemantau kendaraan :

a. Inisialisasi GPIO port untuk GPS dan Relay DC

GPS module Ublox Neo 6MV2 mempunyai empat pin koneksi yaitu: Vcc, GND, TX dan RX. Pin RX dari raspberry pi harus dihubungkan ke pin TX dari GPS module. Sedangkan untuk relay dc memerlukan empat pin GPIO pada raspberry pi seperti, pin kontrol relay dc di GPIO 17, VCC(5Volt) dan GNC(Ground).

b. Raspberry pi melakukan extract data GPS

Salah satu fungsi GPS ialah mendeteksi adanya sinyal satellite dengan informasi berupa latitude, longitude dan speedometer yang dihasilkan dari GPS receiver informasi inilah yang selanjutnya akan diteruskan ke raspberry pi untuk di pecah menjadi data yang dapat di edit dan dikirim ke database server.

Untuk melakukan exstact data koordinat menggunakan pemrograman python. Dengan menggunakan library pynmea penulis dapat dengan mudah mengurai data NMEA menjadi objek yang dapat digunakan [13]. NMEA (National Marine Electronics Association) adalah sebuah standar komite Amerika Serikat yang mendefinisikan struktur data pesan, isi dan protokol untuk memungkinkan receiver GPS untuk berkomunikasi dengan peralatan elektronik lainnya. Protokol NMEA terdiri dari serangkaian pesan, pesan ini adalah set karakter ASCII . GPS menerima data dan menyajikannya dalam bentuk string pesan ASCII, tanda \$ (dolar) digunakan pada awal setiap pesan [14]. Protokol yang dipakai dalam penelitian ini terbagi menjadi 2 (dua), pertama menggunakan GGA(Global Positioning System Fixed) dan yang kedua RMC (Minimum Recomend Data for GPS). Gambar berikut merupakan hasil dari exstact data koordinat yang dihasilkan dari GPS receiver.

```

GNU nano 2.8.6 File: GPSdata.txt
1737.797850.42
1737.797850.42
1737.797850.42
1737.427850.41
1737.347850.44
1737.337850.80
1737.307850.87
1737.417849.97
1737.487849.83
1737.777849.16
1738.287849.11
1738.497848.86
^G Get Help ^O Write Out ^W Where Is ^X Cut Text ^J Justify
^X Exit ^R Read File ^N Replace ^U Uncut Text ^T To Spell

```

Gambar 3.2 Hasil data koordinat[3]

e. Update data GPS dan relay

Pengiriman data koordinat posisi dan relay ke database server menggunakan pemrograman python yang secara langsung dapat mengirimkan data hasil exstact (parsing) koordinat lokasi di raspberry pi. Dengan bantuan rest API inilah data akan di tampilkan di aplikasi mobile (android), rest API adalah web service yang menerapkan perpindahan antar state. Proses ini merupakan hasil dari output sistem aplikasi yang dibangun, aplikasi akan menampilkan koordinat posisi dalam visualisasi google maps dan sistem kendali switch on/off di aplikasi mobile (android).

Perancangan Aplikasi

Proses merancang aplikasi sebelum melakukan pengkodean (*coding*), yang diwujudkan dalam bentuk model. Model ini mampu menggambarkan persoalan dalam beberapa sudut pandang (jenis diagram) diantaranya yaitu, gambar 3.4 menjelaskan fungsionalitas dari sebuah sistem, gambar 3.5 menjelaskan struktur dan deskripsi kelas package dan objek, gambar 3.6 menjelaskan berbagai aktifitas dalam sistem yang sedang dirancang mulai dari titik awal, melalui kondisi yang mungkin terjadi, kemudian sampai pada titik akhir.

3.4.1 Use case diagram

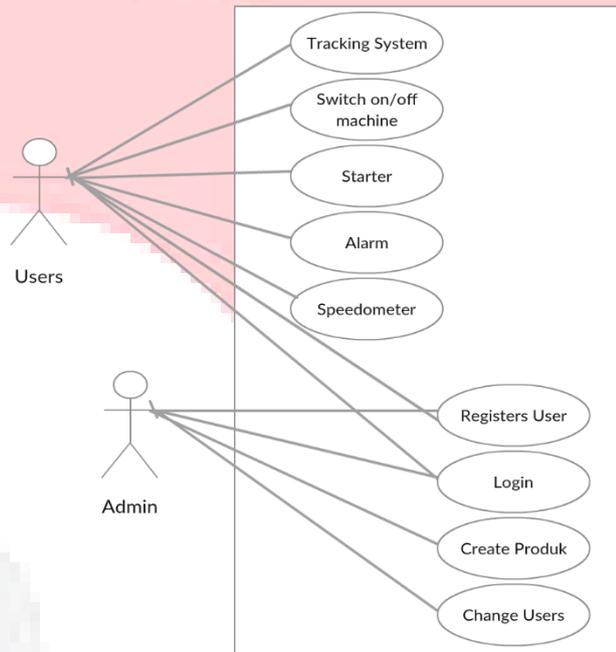
Dimana dalam aplikasi ini terdapat beberapa aktor yang aktif diantaranya:

a). Users

Tugas utama *users* ialah memeriksa perangkat supaya tetap berjalan dengan baik, *users* bisa mengontrol kendaraan melalui aplikasi *mobile (android)*, *users* bisa mengelola *tracking system*, *switch on/off machine* dan melakukan pengontrolan kecepatan (*speedometer*).

b). Admin

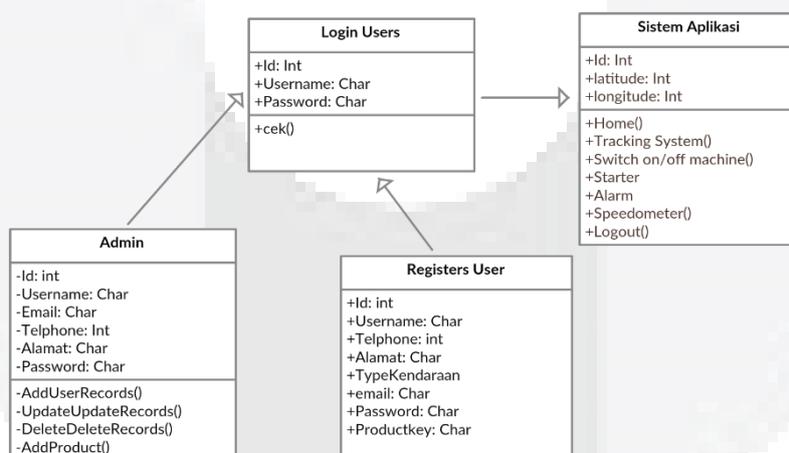
Tugas admin ialah memeriksa users supaya dapat terhubung dengan *server*, mengelola data *users* dapat melakukan *update*, *delete*, *create user* langsung menggunakan perangkat aplikasi untuk administrator dan mengelola data produk admin bertugas memperbaiki dan mengembangkan lebih lanjut mengenai perangkat dan aplikasi *Internet of Things* supaya jauh lebih baik dari pengembangan sebelumnya.



Gambar 3.3 Use case diagram

3.4.2 Class diagram

Class diagram berisi kelas-kelas untuk membuat (*generate*) *dataset* (data terkumpul) mulai dari membuat *registers user* sampai dengan sistem aplikasi berikut merupakan class diagram aplikasi *Internet of Things* untuk pengendali dan pemantau kendaraan.



Gambar 3.4 Class diagram

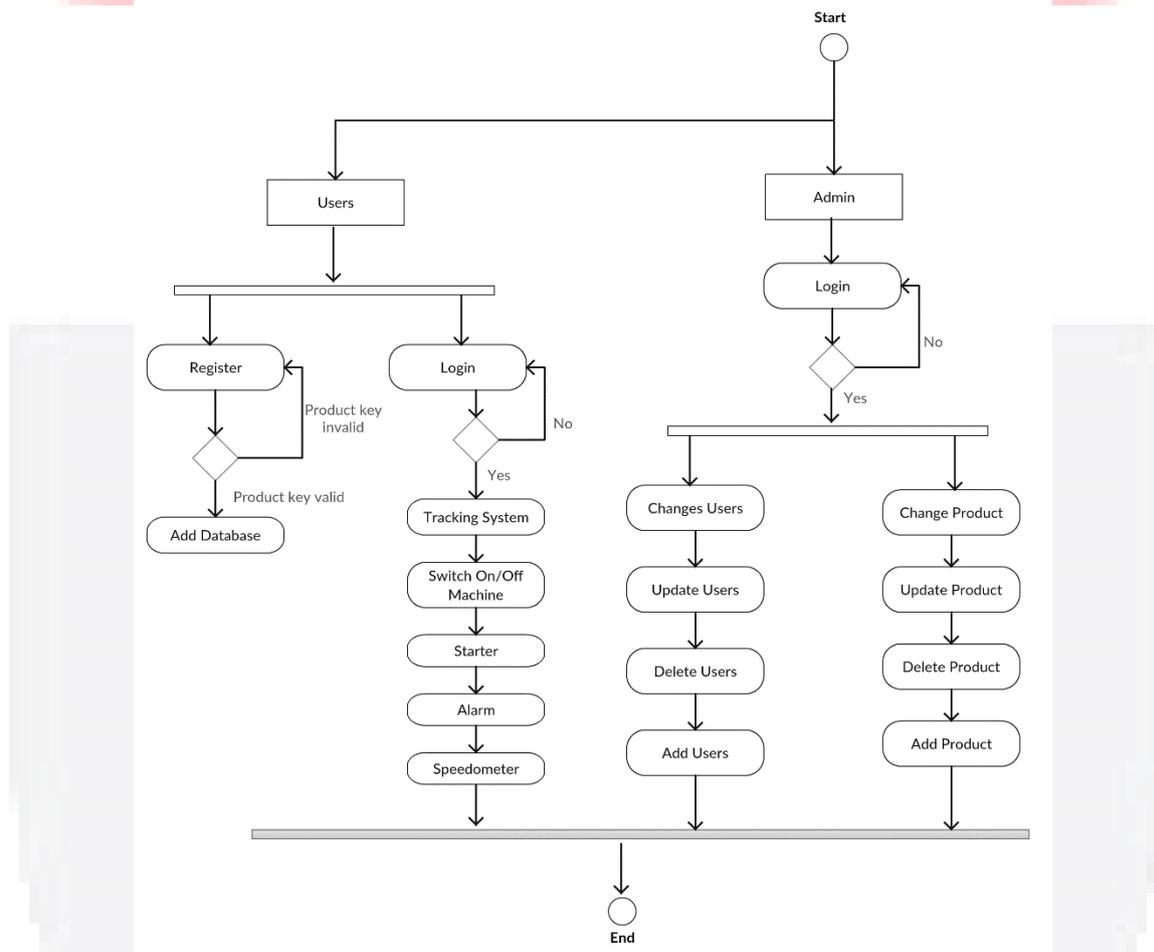
Berikut merupakan penjelasan dari class diagram :

- *Class / table login* ber-agregasi dengan class admin dan class registers user karena login user tidak dapat berdisi sendiri tanpa adanya user dan admin yang melakukan akses ke fitur aplikasi.

- *Class / table* sistem aplikasi tidak dapat berdiri sendiri tanpa adanya *login user* yang melakukan akses ke sistem aplikasi.
- *Class / table* sistem aplikasi dapat menyimpan berbagai macam fitur diantaranya *tracking sistem, switch on/off machine, speedometer dan logout*.

3.4.3 Activity diagram

Berikut merupakan diagram aplikasi *Internet of Things* untuk pengendali dan pemantau kendaraan yang menggambarkan proses-proses dan jalur-jalur aktivitas secara umum (global).



Gambar 3.5 Activity diagram

Berikut merupakan penjelasan dari activity diagram :

- *User* melakukan registrasi sesuai dengan ketentuan sistem, dengan melakukan validasi perangkat (*product key*) melalui *form register*, jika *product key* sesuai maka sistem aplikasi akan menambahkan *user* tersebut.
- *User* mengidentifikasi dirinya (dengan cara memasukkan nama pengguna (*username*) serta kata sandi (*password*) ke sistem. *User* melakukan *tracking* kendaraan sesuai dengan *id* perangkat *Internet of Things*.
- *User* melakukan validasi kendaraan, jika kendaraannya dibawa oleh orang yang tidak diketahui *user* (bukan pemilik) maka *user* (pemilik) bisa mengambil tindakan dengan cara mematikan kendaraan, melakukan *tracking* mengaktifkan sistem alarm di sistem aplikasi *mobile* yang sudah di sediakan.

- Tugas admin ialah melakukan *update* sistem, menerima segala macam masukan (permasalahan terkait sistem) dan admin mempunyai hak akses ke seluruh *users* pada sistem, seperti *change users*, *update users*, *delete users* dan *add users*.

4. Hasil Dan Analisis

4.1 Pengujian Sistem

Pengujian dan analisis sistem dalam penelitian ini secara umum terbagi kedalam dua hal, yaitu:

1. Pengujian dan analisis perangkat keras yang meliputi pengujian dan analisis rangkaian *microprocessor raspberry pi3 model b* yang tersambung dengan *relay dc 4 channel*, *GPS module UBLOX NEO 6MV2* dan *modem GSM Huawei K4203*.
2. Pengujian dan analisis perangkat lunak yang diimplementasikan menggunakan pemrograman *mobile (android)*, sebagai pengendali dan pemantau kendaraan Dalam pengujian perangkat lunak ini menggunakan metode pengujian *black box*. Pengujian *black box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Dengan demikian pengujian dengan metode *black box* mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional program aplikasi. Kebenaran perangkat lunak hanya dilihat berdasarkan keluaran yang dihasilkan dari kondisi masukan yang diberikan fungsi tersebut tanpa melihat bagaimana proses untuk mendapatkan hasil keluaran suatu aplikasi.

4.2 Pengujian dan analisis perangkat keras

Pengujian dan analisis perangkat keras ini bertujuan untuk mengetahui proses yang dilakukan setiap rangkaian, analisis dilakukan dengan menilai hasil pengujian dari rangkaian *microprocessor raspberry pi3 model b* yang tersambung dengan *relay dc 4 channel* dan *GPS module UBLOX NEO 6MV2*. untuk dibandingkan dengan konsep perancangan dan informasi masing-masing perangkat pada masing-masing dataset. Pada penelitian ini setiap *module* akan bekerja pada level tegangan *5volt relay dc 4 channel* dan tegangan *3.3volt* untuk *GPS module UBLOX NEO 6MV2*. Adapun hal-hal yang diujikan adalah sebagai berikut:

Tabel 4.1 Perangkat Pengendali dan Pemantau

Requirement yang diuji	Butir Uji
Relay DC 4 Channel	Melakukan Exstact / parsing data relay
GPS	Melakukan Exstact / parsing data gps

4.2.1 Pengujian Relay DC 4 Channel

Pada tahap ini adalah pengujian dari sisi perangkat *Internet of Things* dimana data hasil keputusan dari *relay* langsung dikirim ke *database server* menggunakan perangkat *microprocessor* yang sudah tersambung ke jaringan *Internet*. Ada beberapa hal mengenai konfigurasi sambungan *port* dari *relay dc* ke *microprocessor*, adapun *relay dc* memerlukan empat pin *GPIO* pada *raspberry pi* seperti, pin kontrol *relay dc* di *GPIO 17*, *VCC(5Volt)* dan *GNC(Ground)*. Berikut merupakan sambungan *port GPIO raspberry pi* ke *Relay Dc*.

	switch on/off, starter dan alarm tidak menyala (Off) .	starter dan alarm tidak menyala (Off)	<input type="checkbox"/> ditolak
--	--	---------------------------------------	----------------------------------

4.2.2 Pengujian GPS

Pada tahap kedua ini adalah pengujian dari sisi perangkat *Internet of Things* dimana data hasil *extract (parsing)* dikirim ke *database server (server hosting)* menggunakan perangkat *microprocessor* yang sudah tersambung ke jaringan *Internet*. Ada beberapa hal mengenai konfigurasi sambungan *port* dari *gps module* ke *microprocessor*, adapun *GPS module Ublox Neo 6MV2* mempunyai empat pin koneksi yaitu: *Vcc, GND, TX dan RX*. Pin *RX*. Berikut merupakan sambungan *port GPIO raspberry pi* ke *GPS module*

Tabel 4.4 Inialisasi GPIO ke GPS Ublox Neo

Pin GPIO Raspberry pi 3	Pin GPS module Ublox Neo 6MV2
Pin 1 (3.3 v)	VCC
Pin 10 (GPIO 15)	TX
Pin 8 (GPIO 14)	RX
Pin6 (Gnd)	GND

Konfigurasi selanjutnya ialah melakukan pengiriman hasil *extract (parsing)* data, adapun pemrograman yang digunakan ialah pemrograman *python* dengan menggunakan *library pynmea2* dan untuk tranfrer data ke *database server* menggunakan *library MySQLdb*, program aplikasi tersebut menghasilkan keluaran berupa *longitude, latitude dan speedometer (monitoring* kecepatan). Berikut gambar 4.2 merupakan hasil keluaran dari pengujian *GPS*.

```

pi@raspberrypi: ~/Execute/starup/maps
pi@raspberrypi: ~/Execute/starup/maps 110x24
Time: Thu Jan 18 16:58:30 2018, Latitude: -6.97814666667N, Longitude: 107.634396667W, Speed: 0.850068km/h
Time: Thu Jan 18 16:58:31 2018, Latitude: -6.97814933333N, Longitude: 107.634396W, Speed: 0.457444km/h
Time: Thu Jan 18 16:58:32 2018, Latitude: -6.9781495N, Longitude: 107.634395167W, Speed: 0.516708km/h
Time: Thu Jan 18 16:58:33 2018, Latitude: -6.9781525N, Longitude: 107.634393833W, Speed: 0.501892km/h
Time: Thu Jan 18 16:58:34 2018, Latitude: -6.97815533333N, Longitude: 107.6343905W, Speed: 0.503744km/h
Time: Thu Jan 18 16:58:35 2018, Latitude: -6.978158N, Longitude: 107.6343875W, Speed: 0.559304km/h
Time: Thu Jan 18 16:58:36 2018, Latitude: -6.97815866667N, Longitude: 107.63439W, Speed: 0.209276km/h
Time: Thu Jan 18 16:58:37 2018, Latitude: -6.97815966667N, Longitude: 107.634391667W, Speed: 0.503744km/h
Time: Thu Jan 18 16:58:39 2018, Latitude: -6.97816316667N, Longitude: 107.634393167W,ed: 0.216684km/h
Time: Thu Jan 18 16:58:40 2018, Latitude: -6.978165N, Longitude: 107.634393667W, Spe Speed: 0.2315km/h
Time: Thu Jan 18 16:58:41 2018, Latitude: -6.97816733333N, Longitude: 107.634394833W,ed: 0.451888km/h
Time: Thu Jan 18 16:58:42 2018, Latitude: -6.978168N, Longitude: 107.634396333W, Spe Speed: 0.646348km/h
Time: Thu Jan 18 16:58:44 2018, Latitude: -6.97817066667N, Longitude: 107.634396167W, Speed: 0.11112km/h
Time: Thu Jan 18 16:58:45 2018, Latitude: -6.978172N, Longitude: 107.634396667W, Speed: 0.079636km/h
Time: Thu Jan 18 16:58:46 2018, Latitude: -6.97817316667N, Longitude: 107.634399667W, Speed: 0.622272km/h
Time: Thu Jan 18 16:58:47 2018, Latitude: -6.97817466667N, Longitude: 107.634399167W, Speed: 0.322248km/h
Time: Thu Jan 18 16:58:48 2018, Latitude: -6.97817733333N, Longitude: 107.634399167W, Speed: 0.294468km/h
Time: Thu Jan 18 16:58:49 2018, Latitude: -6.97817916667N, Longitude: 107.634398167W, Speed: 0.285208km/h
Time: Thu Jan 18 16:58:50 2018, Latitude: -6.978181N, Longitude: 107.634396833W, Speed: 0.266688km/h
Time: Thu Jan 18 16:58:51 2018, Latitude: -6.978181N, Longitude: 107.634398333W, Speed: 0.29632km/h
Time: Thu Jan 18 16:58:52 2018, Latitude: -6.978182N, Longitude: 107.6344005W, Speed: 0.38892km/h
Time: Thu Jan 18 16:58:53 2018, Latitude: -6.97818483333N, Longitude: 107.634402W, Speed: 0.461148km/h
Time: Thu Jan 18 16:58:54 2018, Latitude: -6.97818733333N, Longitude: 107.634402667W, Speed: 0.34262km/h
    
```

Gambar 4.2 Output ekstrakt data gps

Berikut ini adalah hasil pengujian sistem berdasarkan hasil implementasi aplikasi *Internet of Things* pada *gps* menggunakan metode *black box* sesuai dengan *requirement* pengujian aplikasi.

Tabel 4.5 Pengujian GPS

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Parsing / Exstract data gps	Akan menghasilkan koordinat gps latitude, langitude dan speedometer	Menghasilkan data koordinat, latitude, langitude dan speedometer	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak
Send data GPS	Data GPS dikirim ke database server dan langsung di panggil oleh rest API	Data GPS dikirim ke database server dan langsung di panggil oleh rest API	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak

4.3 Pengujian dan analisis perangkat lunak

Dari hasil implementasi dan pengujian terhadap perangkat lunak, maka dapat dilakukan analisis bahwa secara umum perangkat lunak dapat berjalan dengan baik sehingga tidak menutup kemungkinan untuk dapat diterapkan pada kondisi yang sebenarnya berikut adalah analisis hasil pengujian sistem. Dalam implementasi terdapat beberapa antar muka aplikasi yang menjelaskan tentang masukan, proses dan keluaran dari sistem aplikasi *Internet of Things* untuk pengendali dan pemantau kendaraan, adapun aplikasi terbagi kedalam 2 (dua) katagori yang pertama aplikasi *Internet of Things* untuk *Administrator* dan yang kedua aplikasi *Internet of Things* untuk *Users*, dimana antara aplikasi tersebut memiliki fungsi dan cara kerja masing-masing.

4.3.1 Aplikasi Internet of Things untuk Administrator

Adapun aplikasi *Internet of Things* untuk pengendali dan pemantau kendaraan khusus untuk penggunaan *private (akses administrator)*. Adapun hal-hal yang diujikan menggunakan metode *black box* ini adalah sebagai berikut:

Tabel 4.6 Pengujian Aplikasi Internet of Things untuk Administrator

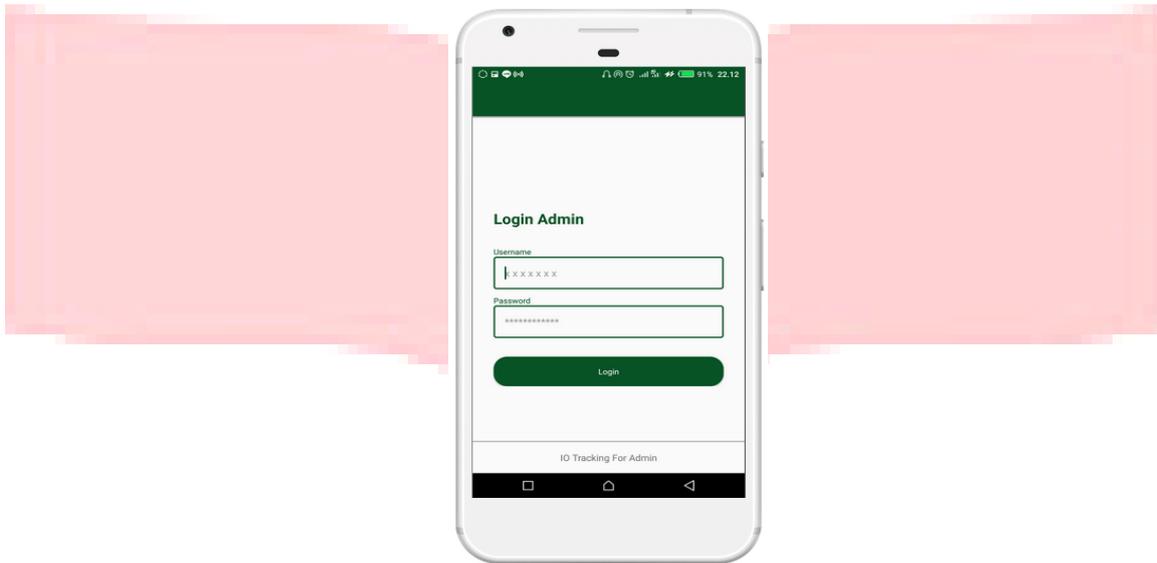
Requirement yang diuji	Butir Uji
Login	Melakukan Login
Data Produk	Menginputkan Data Produk
Data Users	Update Data Users

Berikut merupakan penjelasan dan rancangan implementasi aplikasi *Internet of Things* untuk administrator.

a. Pengujian Form Login Administrator

Form login berfungsi sebagai penentu apakah pengguna memiliki izin akses sistem, pada form ini hanya disediakan satu tombol "*Login*" dimana setelah melakukan eksekusi login, seorang *administrator* memasuki *form* info produk yang akan menampilkan beberapa informasi tentang produk dan info *users*, apabila pengguna

(*administrator*) tidak memiliki izin akses maka dengan sendirinya sistem akan menolak bahwa pengguna tidak memiliki hak untuk login dengan menampilkan pesan “*invalid username or password*”.



Gambar 4.3 Login Administrator

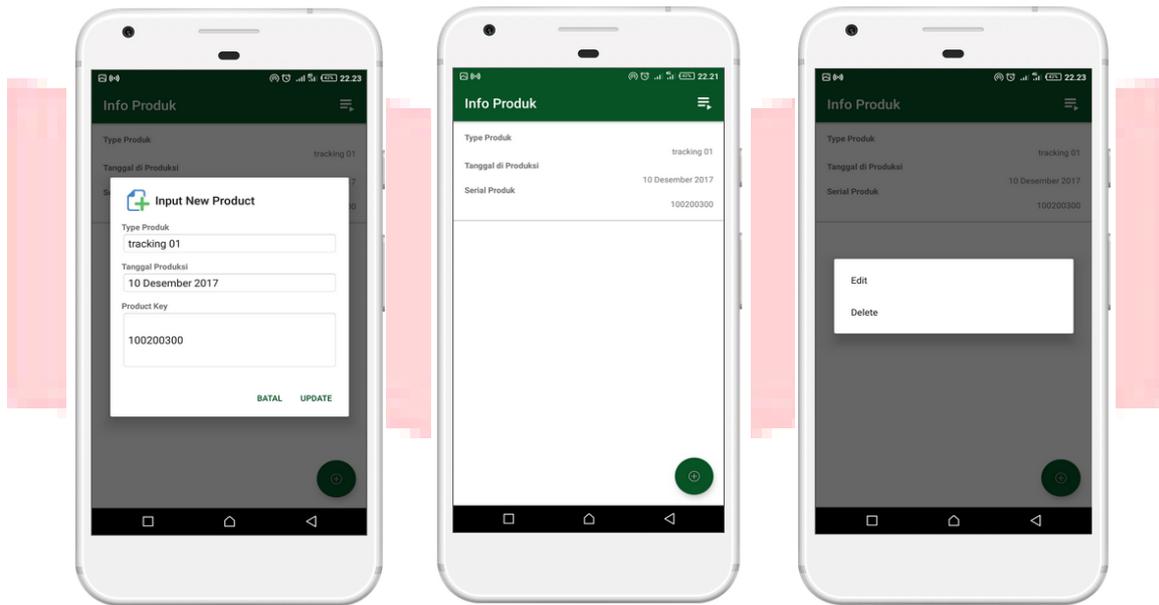
Berikut ini adalah hasil pengujian sistem berdasarkan hasil implementasi aplikasi *Internet of Things* pada *form login* menggunakan metode *black box* sesuai dengan *requirement* pengujian aplikasi.

Tabel 4.7 Pengujian Form Login Administrator

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Username dan password terisi dengan benar	Akan menampilkan dashboard monitoring system	Menampilkan dashboard monitoring system	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak
Username dan password kosong atau username dan password salah	Akan menampilkan invalid username or password	Menampilkan invalid username or password	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak

b. Pengujian Data Produk

Form data produk berfungsi sebagai pendataan suatu produk baru, *form* data produk ini digunakan setelah produk selesai di produksi dan sebelum *prototype* produk di edarkan maka harus di catat di *form* data produk, selain itu *form* ini juga berfungsi sebagai hak akses seorang pengguna bisa atau tidak melakukan pendaftaran sesuai dengan *product key* perangkat *Internet of Things* untuk pengendali dan pemantau kendaraan yang tercatat di dalam produk tersebut.



Gambar 4.4 Form data produk

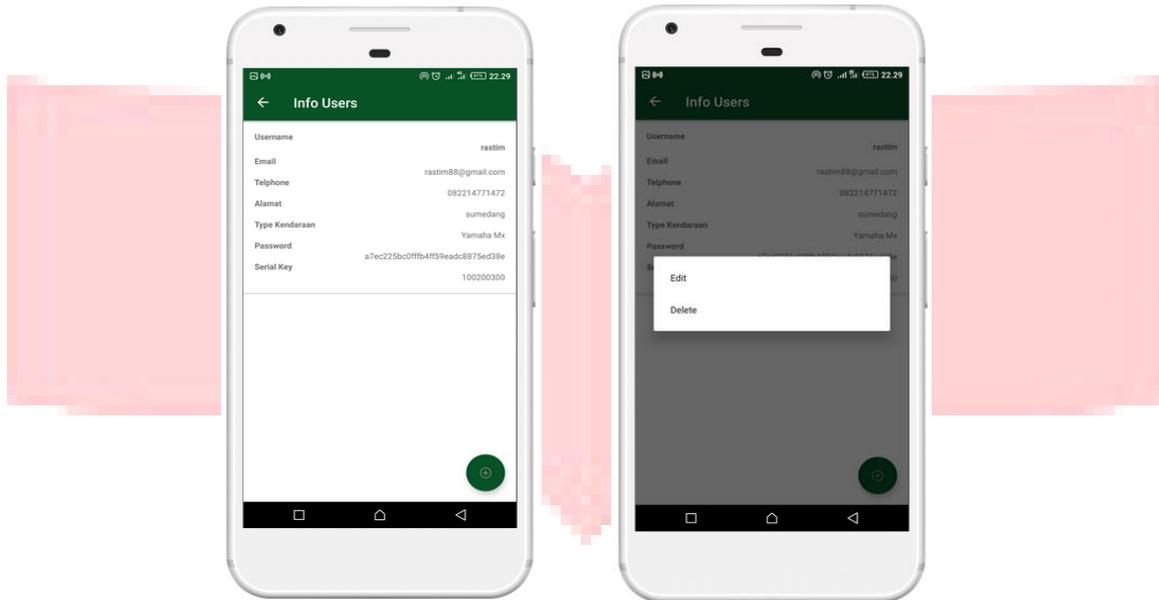
Berikut ini adalah hasil pengujian sistem berdasarkan hasil implementasi aplikasi *Internet of Things pada data produk* menggunakan metode *black box* sesuai dengan *requirement* pengujian aplikasi.

Tabel 4.8 Pengujian Data Produk

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Penambahan data produk	Data masuk kedalam database	Data masuk kedalam database	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak
Penyimpanan data	Data baru disimpan kedalam database	Data masuk kedalam database	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak
Perubahan data produk	Data dapat dirubah data lama dapat dirubah menjadi data yang baru	Data pada database berubah	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak
Penghapusan data produk	Data dapat dihapus pada database	Data terhapus pada database	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak

c. Pengujian Data Users

Form data users berfungsi untuk melihat users yang aktif dan sebagai pendataan suatu *users* baru setelah *users* melakukan registrasi, dengan *form user* seorang administrator bisa melakukan *update*, *delete* dan *create* pengguna.



Gambar 4.5 Form data users

Berikut ini adalah hasil pengujian sistem berdasarkan hasil implementasi aplikasi *Internet of Things pada data users* menggunakan metode *black box* sesuai dengan *requirement* pengujian aplikasi.

Tabel 4.9 Pengujian Data Users

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Penambahan data Users	Data masuk kedalam database	Data masuk kedalam database	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak
Penyimpanan data	Data baru disimpan kedalam database	Data masuk kedalam database	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak
Perubahan data Users	Data dapat dirubah data lama dapat dirubah menjadi data yang baru	Data pada database berubah	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak
Penghapusan data produk	Data dapat dihapus pada database	Data terhapus pada database	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak

4.3.2 Aplikasi Internet of Things untuk Users

Adapun aplikasi *Internet of Things* untuk pengendali dan pemantau kendaraan khusus untuk penggunaan *public* (akses pengguna). Adapun hal-hal yang diujikan menggunakan metode *black box* ini adalah sebagai berikut:

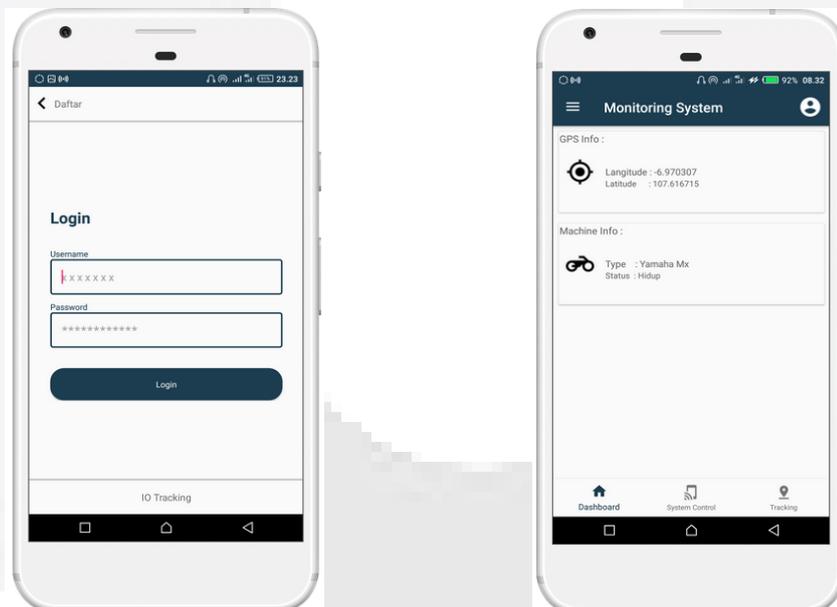
Tabel 4.10 Pengujian Aplikasi Internet of Things Users

Requirement yang diuji	Butir Uji
Login	Melakukan Login
Data Pendaftaran	Menginputkan Data Pendaftaran
Switch On/Off Kendaraan	Menekan Button On/Off
Speedometer	Ditampilkannya speedometer
Starter Kendaraan	Menekan Button Starter
Alarm Kendaraan	Menekan Button Alarm
Tracking Kendaraan	Menekan Button Tracking

Berikut merupakan penjelasan dan rancangan implementasi aplikasi *Internet of Things* untuk administrator.

a. Pengujian Form Login

Form login berfungsi sebagai penentu apakah pengguna memiliki izin akses sistem, pada form ini hanya disediakan satu tombol “*Login*” dimana setelah melakukan eksekusi login pengguna memasuki *form dashboard* yang akan menampilkan beberapa informasi tentang koordinat *gps* dan status dari kendaraan dimaksud, apabila pengguna tidak memiliki izin akses maka dengan sendirinya sistem akan menolak bahwa pengguna tidak memiliki hak untuk login dengan menampilkan pesan “*invalid username or password*”.



Gambar 4.6 Form Login Users

Berikut ini adalah hasil pengujian sistem berdasarkan hasil implementasi aplikasi *Internet of Things* pada *form login* menggunakan metode *black box* sesuai dengan *requirement* pengujian aplikasi.

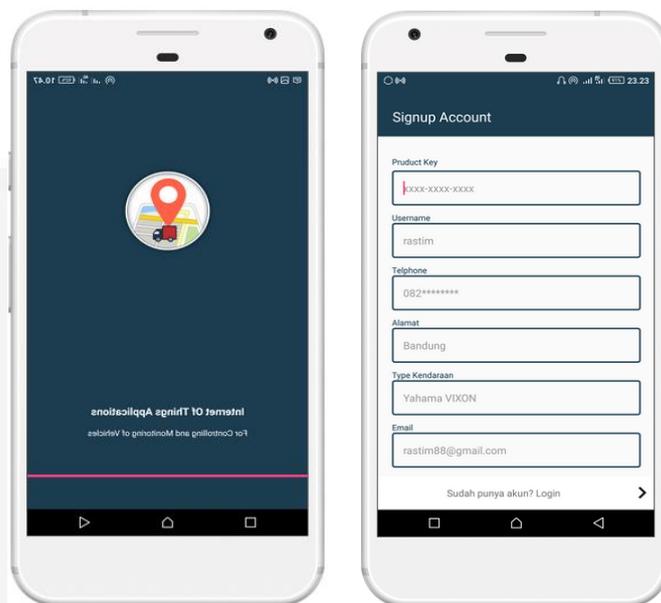
Tabel 4.11 Pengujian form login users

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
--------------	-----------------	------------	------------

Username dan password terisi dengan benar	Akan menampilkan dashboard monitoring system	Menampilkan dashboard monitoring system	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak
Username dan password kosong atau username dan password salah	Akan menampilkan invalid username or password	Menampilkan invalid username or password	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak

b. Pengujian Data Pendaftaran

Form registrasi ini berfungsi sebagai penentu apakah seorang layang untuk registrasi dan mengakses sistem aplikasi, pada form ini hanya disediakan satu tombol “Daftar” yang berfungsi sebagai pemberian identitas pengguna dan apabila pengguna berhasil memasukan *product key* dan pilihan yang lain terisi dengan benar maka pengguna di izinkan untuk *login* dan mengakses sistem pengendali dan pemantau kendaraan.



Gambar 4.7 Form Registrasi

Berikut ini adalah hasil pengujian sistem berdasarkan hasil implementasi aplikasi *Internet of Things* pada form registrasi menggunakan metode *black box* sesuai dengan *requirement* pengujian aplikasi.

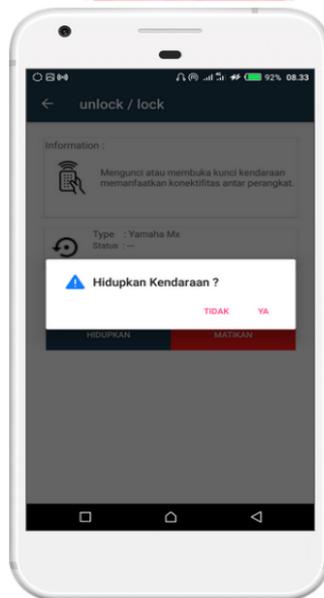
Tabel 4.12 Pengujian Data Pendaftaran

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Klik daftar terisi dengan benar	Akan menampilkan data berhasil ditambahkan	Menampilkan data berhasil ditambahkan	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak
Klik daftar ada kolom tidak terisi atau kosong	Akan menampilkan kolom tidak boleh kosong	Menampilkan kolom tidak boleh kosong	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak

Klik daftar dengan product key salah atau tidak sesuai dengan product key perangkat	Akan menampilkan product key tidak cocok silahkan dicoba kembali	Menampilkan product key tidak cocok silahkan dicoba kembali	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak
---	--	---	--

c. Pengujian Switch On/Off Kendaraan

Unlock/lock kendaraan ini memanfaatkan data *relay* sebagai acuan apabila nilai *relay high* dinyatakan dengan angka 1 (satu) yang artinya mesin berhasil di nyalakan sebaliknya apabila nilai *relay low* maka dinyatakan dengan angka 0 (nol) yang artinya mesin berhasil di matikan.



Gambar 4.8 Form unlock/lock

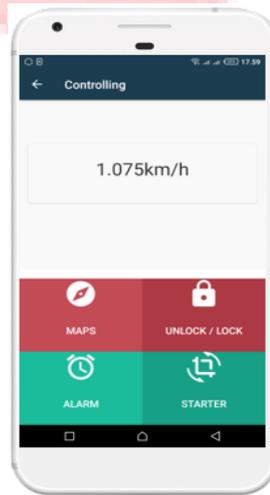
Berikut ini adalah hasil pengujian sistem berdasarkan hasil implementasi aplikasi *Internet of Things* pada *form Switch On/Off* kendaraan, menggunakan metode *black box* sesuai dengan *requirement* pengujian aplikasi.

Tabel 4.13 Pengujian Switch On/Off

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Klik Button Matikan	Akan menampilkan keterangan berhasil dimatikan	Menampilkan keterangan berhasil dimatikan	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak
Klik Buttom Hidupkan	Akan menampilkan keterangan berhasil dihidupkan	Menampilkan keterangan berhasil dihidupkan	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak

d. Pengujian Speedometer

Speedometer ini memanfaatkan *sensor gps* untuk melakukan pemetaan kecepatan dimana apabila sinyal *gps* menunjukkan *hight* maka akan terlihat seberapa cepat kendaraan tersebut melaju dan sebaliknya apabila *sensor gps* dalam keadaan *los connection* maka hanya akan ditampilkan 0 km/h. Pada *speedometer* ini memanfaatkan data *gps* untuk melakukan *extract* data *\$GPRMC*. *Function \$GPRMC (Recomended Minimum Specific GPS / Transit data)* merupakan *function* untuk mengetahui data kecepatan dalam km (kilometer), posisi dan waktu. Hanya data dengan header *\$GPRMC* yang akan diproses kemudian data tersebut dipisahkan menjadi data kecepatan (*speed_over_grnd*) dan disusun berdasarkan logika pemrograman, jika data tidak diketahui maka masukan nilai 0 sebaliknya jika data diketahui atau ditemukan maka jalankan *function \$GPRMC*. Selanjutnya data akan dikonversi ke angka yang dapat dikenali dan dimasukkan kedalam *database* untuk dikonversi ke data *json* yang dapat dipanggil menggunakan aplikasi *mobile (android)*.



Gambar 4.9 Form Speedometer

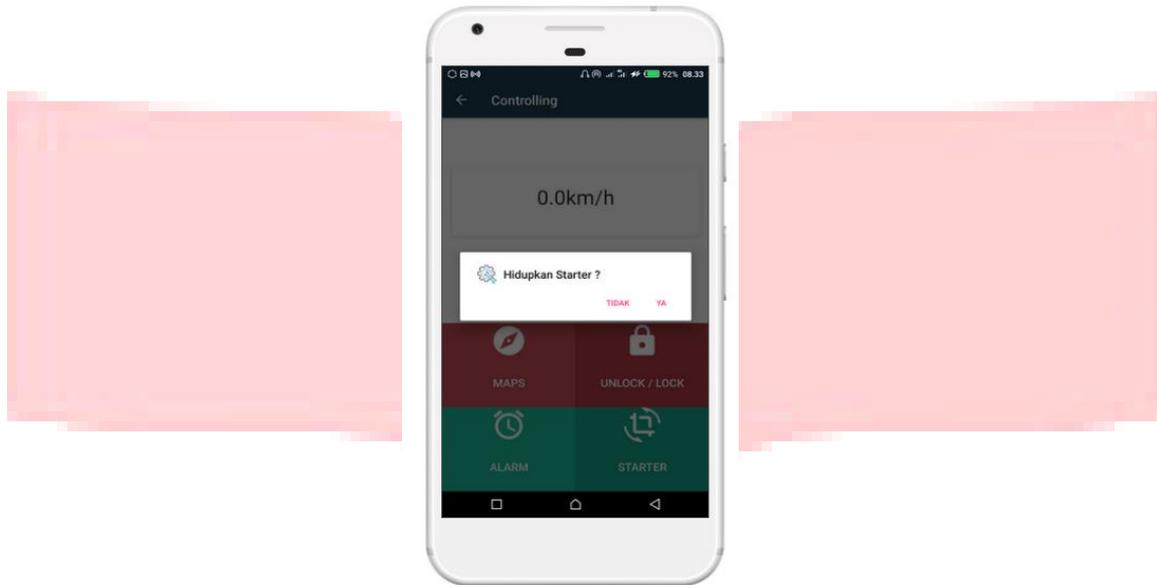
Berikut ini adalah hasil pengujian sistem berdasarkan hasil implementasi aplikasi *Internet of Things* pada *form speedometer* menggunakan metode *black box* sesuai dengan requirement pengujian aplikasi.

Tabel 4.14 Pengujian Speedometer.

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
GPS hasil extract (parsing)	Akan menampilkan kecepatan laju berdasarkan koordinat gps	Menampilkan kecepatan laju kendaraan	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak

e. Pengujian Starter Kendaraan

Pada dasarnya sistem *starter* kendaraan memanfaatkan data *relay* sebagai sistem acuan yang menyatakan aliran listrik mengalir atau tidak, pada sistem *starter* penulis berasumsi bahwa pada dasarnya *starter* hanya sebagai pemantik, begitu mesin berhasil dinyatakan menyala (kondisi mesin menyala) maka aliran listrik *relay* secara otomatis di matikan, penulis membatasi bahwa *delay* untuk aliran listrik pada *starter* adalah 2 (dua) detik, selama 2 (dua) detik aliran listrik akan dinyalakan.



Gambar 4.10 Form Starter

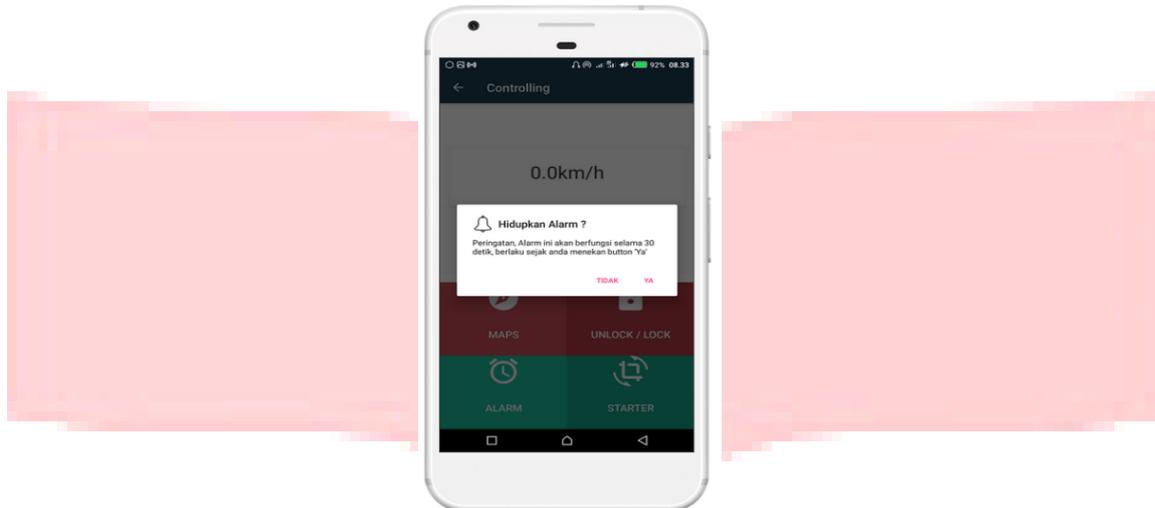
Berikut ini adalah hasil pengujian sistem berdasarkan hasil implementasi aplikasi *Internet of Things* pada pada *form starter* mesin menggunakan metode black box sesuai dengan requirement pengujian aplikasi.

Tabel 4.15 Pengujian Starter Mesin

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Klik Button Starter	Akan menampilkan keterangan starter berhasil dihidupkan	Menampilkan keterangan starter berhasil dihidupkan	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak

f. Pengujian Alarm Kendaraan

Alarm berfungsi sebagai isyarat peringatan bahwa kendaraan tersebut sedang di awasi, sistem alarm pada dasarnya memanfaatkan data relay sama seperti sistem pada *unlock/lock* dan *starter* sebagai acuan yang menyatakan aliran listrik mengalir atau tidak, pada sistem *alarm* penulis berasumsi bahwa pada dasarnya *relay* yang diterapkan di *alarm* hanya sebagai pemberi *delay*, begitu *alarm* berhasil dinyatakan menyala maka aliran listrik *relay* secara otomatis dialirkan, penulis membatasi bahwa *delay* untuk aliran listrik pada *alarm* adalah 30 (tiga puluh) detik, selama 30 (tiga puluh) detik aliran listrik akan dinyalakan.



Gambar 4.11 Form Alarm

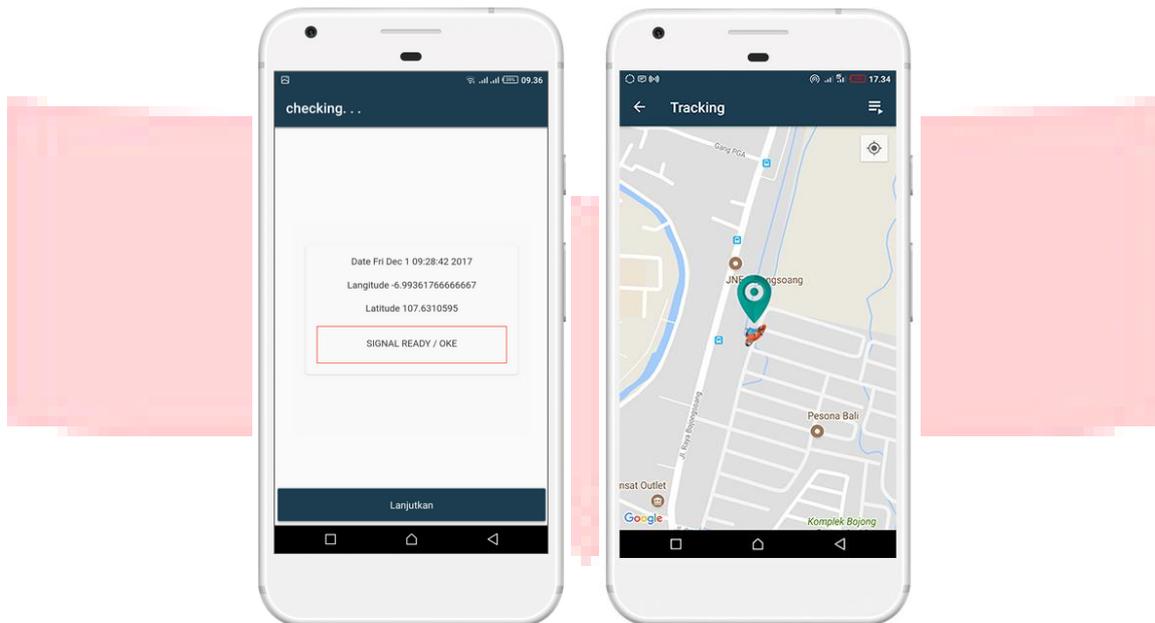
Berikut ini adalah hasil pengujian sistem berdasarkan hasil implementasi aplikasi *Internet of Things* pada *alarm* menggunakan metode *black box* sesuai dengan *requriment* pengujian aplikasi.

Tabel 4.16 Pengujian Alarm

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Klik Button Alarm	Akan menampilkan keterangan alarm berhasil dihidupkan	Menampilkan keterangan alarm berhasil dihidupkan	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak

g. Pengujian Tracking Kendaraan

Tracking system berfungsi untuk mengetahui pemetaan lokasi berdasarkan *sensor gps*, *sensor gps* akan melakukan *update* koordinat (mengubah lokasi) setiap 3 detik sekali, *tracking system* ini memanfaatkan *library pynmea2* untuk melakukan *exstarack* atau parsing data *gps* dan *google maps API* sebagai visualisasi *maps*. Hal pertama yang dilakukan adalah sistem akan mengecek ketersediaan sinyal *gps*, apabila sinyal *gps* tersedia maka akan menampilkan data *gps* dan keterangan “GPS READY / OK”, sebaliknya apabila sinyal *gps* tidak tersedia maka akan ditampilkannya data *gps* dan keterangan “GPS LOSS CONNECTION”. Langkah pengujiannya adalah dengan melakukan pemantauan *tracking gps* ke setiap wilayah salah satunya dikota Bandung, dengan melakukan perjalanan dan di pantau pergerakan objek *maps*, untuk menghindari *maps* tidak berjalan ataupun tidak memenuhi spesifikasi perancangan perangkat aplikasi. Berdasarkan hasil uji untuk pemantau kendaraan yang divisualisasikan kedalam *maps* dengan terintegrasinya *google maps API* ditampilkan dengan baik.



Gambar 4.12 Form Tracking system

Berikut ini adalah hasil pengujian sistem berdasarkan hasil implementasi aplikasi *Internet of Things* pada *form tracking system* menggunakan metode *black box* sesuai dengan requirement pengujian aplikasi.

Tabel 4.17 Pengujian Tracking Kendaraan

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Klik Button Tracking	Akan menampilkan informasi posisi koordinat dan kualitas sinyal gps.	Menampilkan informasi koordinat dan kualitas sinyal gps	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak
Klik Button Lanjutkan	Akan menampilkan posisi koordinat kendaraan dalam bentuk marker maps	Menampilkan keterangan koordinat kendaraan dalam bentuk marker maps	<input checked="" type="checkbox"/> diterima <input type="checkbox"/> ditolak

Kesimpulan

Berikut merupakan kesimpulan yang diambil dari hasil analisis dan pengujian sistem yang telah dibuat di tugas akhir ini :

1. Berdasarkan hasil pengujian dengan kasus black box dapat ditarik kesimpulan bahwa perangkat aplikasi dapat mengetahui fungsi-fungsi yang tidak benar atau hilang, kesalahan interface, kesalahan dalam akses database, kesalahan kinerja, inisialisasi, kesalahan terminasi dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan pada perancangan aplikasi.
2. Pada tingkat kesalahan yang didapatkan terjadinya suatu error diakibatkan kesalahan pada saat pengiriman data relay GPIO 17, GPIO 18, GPIO 27 dan data GPS, ke database server, dengan demikian harus diketahui bahwa akses Internet harus sudah siap sebelum pengiriman data relay dan gps berlangsung maka dari itu penulis melakukan penambahan value pada saat perangkat mulai diaktifkan.

3. Tracking system sudah dapat memenuhi kebutuhan untuk melakukan proses update posisi secara realtime menggunakan aplikasi mobile (android), sistem yang dibuat juga telah memenuhi kebutuhan pengguna untuk dapat mengakses peta dan posisi kendaraan melalui aplikasi mobile (android). Uji coba ini menunjukkan bahwa pada saat cuaca cerah dan tidak terhalang oleh gedung akses terhadap satelite GPS oleh perangkat Internet of Things bekerja sangat baik untuk menemukan data koordinat. Maka dari itu penulis mengistilahkan perangkat Internet of Things ini hanya bersifat outdoor (hanya berfungsi diluar ruangan).

Daftar Pustaka

- [1] Mohammed.Z.K.A, & Ahmed.E.S.A., 2014, *Internet of Things Applications, Challenges and Related Future Technologies*, vol. 67 (2).
- [2] Vigneshwaran.K, Sumithra.S, & Janani.R., 2015, *An Intelligent Tracking System Based on GSM and GPS Using Smartphones*, vol. 4 (5).
- [3] Sunehra.D., 2017, *Real Time Vehicle Tracking on Google Maps Using Raspberry Pi Web Server*, vol. 4 (6).
- [4] Widyantara.I.M.O, & Sastra.N.P., 2015, *Internet of Things for Intelligent Traffic Monitoring System: A Case Study in Denpasar*, vol. 30 (3).
- [5] Mogal.A.K., 2011, *Wireless Mobile Communication – A Study of 3D Technology*, Vol. 03 (05).
- Sachdeva.P, & Katchii., 2014, *A Review Paper on Raspberry Pi*, Vol.4 (6)
- [6] Tarapiah.S, Atalla.S, & Abuhania.R., 2013, *Smart On-Board Transportation Management System Using GPS/GSM/GPRS Technologies to Reduce Traffic Violation In Developing Countries*, vol. 3 (4).
- Iyuditya, & Dayanti.E., 2013, *Sistem Pengendali Lampu Ruang Secara Otomatis Menggunakan PC Berbasis Mikrokontroler Arduino Uno*, Vol. 10.
- [8] Bhaumik.V, Patel.N.B, & Modi.V., 2015, *Home Automation and Monitoring System Using Raspberry Pi and Android*.
- [9] Pandurove.M, Milakovic.S, Lukic.N, Savic.G, Segedinac.M, & Konjovic.Z., 2015, *Software Framework for REST Client Android: Canvas LSM Case Study*
- [10] Samuel.M.J., 2015, *Implementation of GPS Base Object Location and Route Tracking on Android Device*, Vol. 3 (2).
- [11] Lee.S, Tewolde.G, & Kwon.J., 2014, *Design and Implementation of Vehicle Tracking System Using GPS/GSM/GPRS Technology and Smartphone Application*.
- [12] Parthasarathy.J., 2006, *Positioning and Navigation System Using GPS*, Vol. 36 (6).
- Verma.P, & Bhatia.J.S., 2013, *Design and Development of GPS-GSM Based Tracking System with Google Map Based Monitoring*, vol.3 (3).
- [13] Perumal.T, Sulaiman.M.N, Syarif.K.Y, Ramli.A.R, & Leong.C.Y., 2013, *Development of an Embedded Smart Home Management Scheme*, Vol. 7 (2).
- [14]
- [15]