

# Rancang Bangun Sistem Keamanan Kendaraan Berbasis Internet Of Things Menggunakan Raspberry Pi

Dani Rohpandi<sup>1)</sup>, Hilman Effendi<sup>2)</sup>, Susanto<sup>3)</sup>

Program Studi Teknik Informatika, STMIK Tasikmalaya  
Jl. RE Martadinata 272A Kota Tasikmalaya (0265) 310830  
danirtms@gmail.com<sup>1)</sup>, hilman@gmail.com<sup>2)</sup>, susantoprata@yahoo.co.id<sup>3)</sup>

## Abstrak

Modus pencurian kendaraan bermotor dari tahun ke tahun semakin beragam. Antisipasi yang dilakukan pemilik kendaraan dari tindak pencurian seperti memasang kunci stang, kunci unik, *Secure Key Sutter*, alarm motor dan gembok cakram bukan ditujukan untuk menghindari pencurian kendaraan, namun lebih kepada memperlambat sekaligus mempersulit kerja pencuri kendaraan. Tujuan dari penelitian ini adalah untuk membangun konsep IoT sebagai sarana keamanan kendaraan, dengan memanfaatkan mikrokontroler Raspberry Pi sebagai pengatur, diantaranya : sistem dapat memberikan notifikasi secara real-time kepada user apabila terjadi kasus pencurian, user mengontrol kendaraan menggunakan smartphone. Metode penelitian yang digunakan adalah metode deskriptif dengan pendekatan kualitatif dan model pengembangan perangkat lunaknya menggunakan model *Prototyping*. Garis besar sistem ini adalah penerapan sistem kontrol kendaraan melalui internet yang diimplementasikan dengan menggunakan GPS dan konsep Internet of Things (IoT), yang mana saat diaktifkan dapat menghasilkan respon sesuai dengan harapan.

**Kata kunci:** Internet Of Things, GPS, Keamanan Kendaraan, Raspberry Pi.

## 1. Pendahuluan

Modus pencurian kendaraan bermotor dari tahun ke tahun semakin beragam. Antisipasi yang dilakukan pemilik kendaraan dari tindak pencurian seperti memasang kunci stang, kunci unik, *Secure Key Sutter*, alarm motor dan gembok cakram bukan ditujukan untuk menghindari pencurian kendaraan, namun lebih kepada memperlambat sekaligus mempersulit kerja pencuri kendaraan. Terbatasnya pengawasan pada kendaraan ketika kendaraan disimpan di area parkir, mengakibatkan pemilik tidak dapat memastikan keamanan terhadap kendaraannya sehingga saat terjadi pencurian, pemilik kendaraan terlambat memberikan informasi kehilangan kepada polisi yang mengakibatkan kendaraan sulit ditemukan.

Dalam penelitian sebelumnya, terdapat tiga penelitian dengan permasalahan yang hampir sama yaitu penelitian yang pertama pencurian kendaraan bermotor menduduki

peringkat pertama dari 10 jenis kejahatan maka dari itu dibuatkan alat sistem pengamanan dan pengendalian sepeda motor berbasis mikrokontroler atmega8535[1]. Penelitian yang kedua yaitu semakin marak terjadi pencurian kendaraan bermotor, hal ini dikarenakan oleh minimnya tingkat keamanan kendaraan bermotor roda dua. Maka dari itu dibuatkan alat yang memanfaatkan teknologi GPS (*Global Positioning System*), Arduino dan *Smartphone* Android[2]. Penelitian yang ketiga yaitu sistem pengamanan mobil dengan menggunakan *id card/tag, engine* mobil dapat menyala apabila sesuai dengan *database*. Apabila mobil dimasuki oleh orang baik pemilik ataupun peminjam dan tidak memiliki *idcard/tag* maka mobil tidak dapat dihidupkan sama sekali karena sumber kelistrikkannya tidak terhubung ke bagian *starter*[3].

Berdasarkan latar belakang yang sudah diuraikan sebelumnya, maka masalah yang timbul bagaimana membuat aplikasi sistem keamanan kendaraan berbasis IoT yang dapat mengontrol kendaraan. Batasan permasalahan yang diberlakukan yaitu :

- 1) Sistem kontrol kendaraan meliputi pengecekan lokasi kendaraan, *stop engine, start engine*, alarm kendaraan dan notifikasi kepada *user* apabila terjadi pencurian kendaraan.
- 2) Perangkat *monitoring* dan kontrol menggunakan *Smartphone* dengan OS Android versi *Lollipop*.
- 3) Sistem diterapkan pada sepeda motor.
- 4) Sistem harus terhubung ke jaringan internet.

## 2. Pembahasan

### 2.1. Perancangan

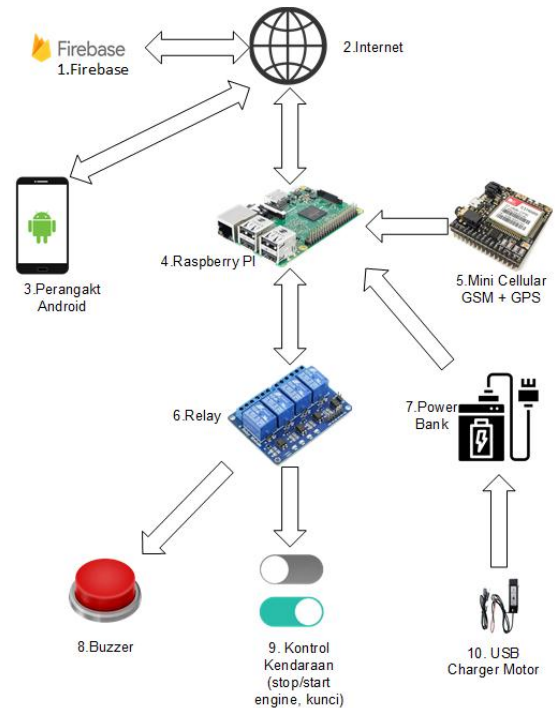
Model pengembangan perangkat lunak dalam pembuatan aplikasi ini menggunakan model *Prototyping*. *Prototyping* adalah proses pembuatan model sederhana *software* yang mengijinkan pengguna memiliki gambaran dasar tentang program serta melakukan pengujian awal. *Prototyping* memberikan fasilitas bagi pengembang dan pemakai untuk saling berinteraksi selama proses pembuatan, sehingga pengembang dapat dengan mudah memodelkan perangkat lunak yang akan dibuat.

Adapun tahapan yang harus dilakukan pada model prototipe adalah sebagai berikut:

1. Pengumpulan kebutuhan : developer dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya;
2. Perancangan : perancangan dilakukan cepat dan rancangan mewakili semua aspek *software* yang diketahui, dan rancangan ini menjadi dasar pembuatan *Prototype*;
3. Evaluasi *Prototype* : klien mengevaluasi *Prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan *software*.

Pembangunan sistem keamanan kendaraan berbasis IoT adalah untuk mengantisipasi dan menanggulangi pencurian kendaraan, sistem ini dibangun menggunakan mini PC *Raspberry Pi* dan android. Dengan membangun sistem ini pengguna dapat mengontrol kendaraan melalui *smartphone* android, sistem juga dapat memberikan notifikasi kepada pengguna apabila ada indikasi pencurian kendaraan. Sistem yang akan dibangun, aplikasi keamanan kendaraan yang berbasis IoT akan berkomunikasi dengan mini PC *Raspberry Pi* yang akan terhubung ke BaaS *Firestore*, adapun rancangan sistem yang akan dibangun dapat dilihat pada gambar 1. Berikut penjelasan alur sistem pada gambar :

1. Penggunaan *server Firestore* dari Google dipilih karena mendukung pemrograman socket. *Firestore* akan digunakan untuk mengirim/menerima data dari perangkat android ke *Raspberry* melalui jaringan internet begitu pula sebaliknya.
2. Android sebagai *User Interface (UI)* untuk mengontrol kendaraan dan menerima data dari kendaraan.
3. *Raspberry Pi* digunakan sebagai *server Mini Cellular GSM+GPS, Relay* dan *Buzzer* melalui *GPIO Raspberry*.
4. *Raspberry Pi* akan mengontrol kendaraan seperti mematikan/menyalakan mesin kendaraan dan menyalakan *Buzzer* melalui *Relay DC* sesuai dengan data yang ada pada *serverFirestore*.
5. *Mini Cellular GSM+GPS* digunakan sebagai *modem* agar *Raspberry Pi* terhubung ke jaringan Internet dan sebagai penerima sinyal GPS.
6. Data lokasi yang diterima oleh GPS akan di kirim *Raspberry Pi* ke *serverFirestore*.
7. *Power Bank* digunakan sebagai daya untuk *Raspberry Pi* agar selalu menyala.
8. *USB Charger* pada kendaraan akan digunakan untuk mengisi daya *Power Bank*.

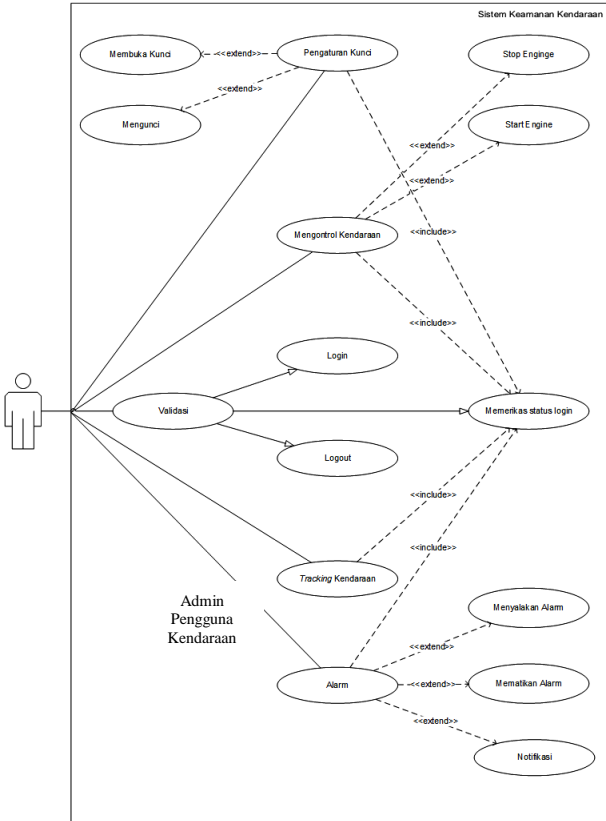


**Gambar 1.** Arsitektur Perancangan Sistem  
Kebutuhan fungsional yang ada pada sistem keamanan kendaraan ini:

1. Pemilik kendaraan dapat menyalakan dan mematikan mesin kendaraan dengan tombol ON/OFF menggunakan perangkat android.
2. Sistem akan menyalakan alarm, memberikan notifikasi kepada pemilik kendaraan dan mematikan mesin kendaraan apabila ada indikasi pencurian kendaraan (kendaraan berpindah dari lokasi parkir).
3. Pemilik kendaraan dapat melihat lokasi kendaraan dengan tombol *tracking* pada perangkat Android.
4. Sistem akan mengunci otomatis apabila pemilik kendaraan meninggalkan kendaraannya dengan radius yang telah diatur pada pengaturan *Auto Lock*.

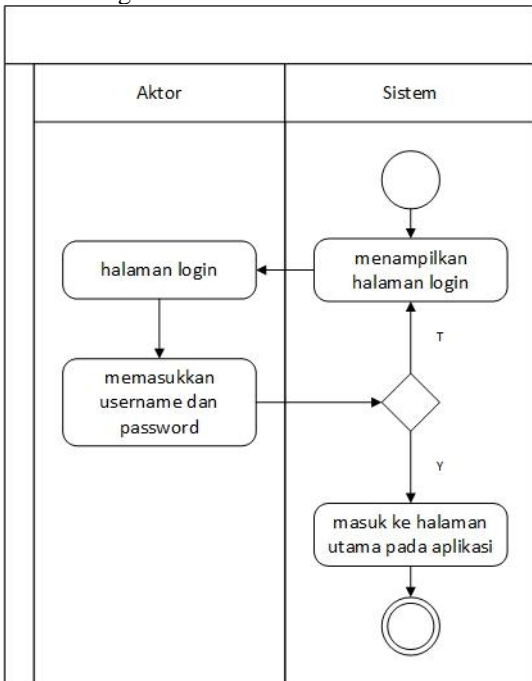
Pada perancangan sistem ini menggunakan beberapa UML diantaranya *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*.

Pengguna Kendaraan merupakan aktor yang menggunakan dan memiliki hak akses untuk menggunakan kendaraan, kunci kendaraan, mengontrol kendaraan, *Tracking* kendaraan dan mendapatkan notifikasi dari kendaraan.



Gambar 2. Use Case Diagram

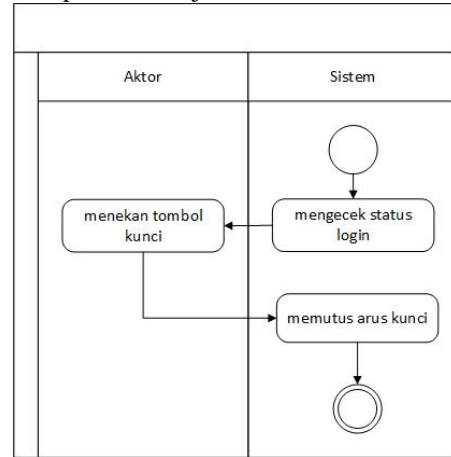
Proses login admin setelah mengakses website program. Pada tampilan utama website, admin akan diminta memasukkan username dan password, dijelaskan pada gambar 3 sebagai berikut :



Gambar 3. Activity Diagram Admin Pengguna Kendaraan

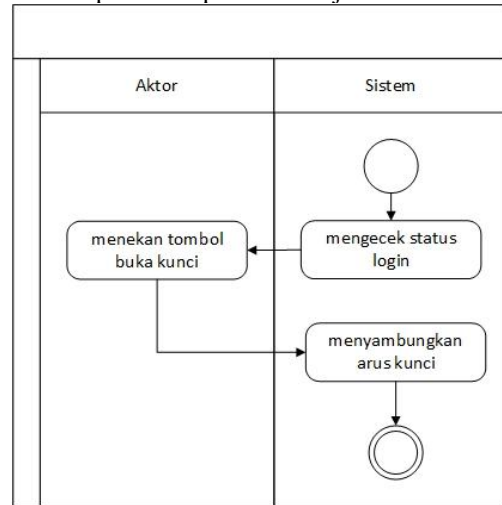
Selanjutnya setelah proses login berhasil, maka pengguna dapat mengakses fasilitas-fasilitas yang disediakan.

Pengguna dapat mengaktifkan fasilitas pengamanan untuk memutus arus kunci, sehingga kendaraan tidak dapat dihidupkan atau dijalankan.



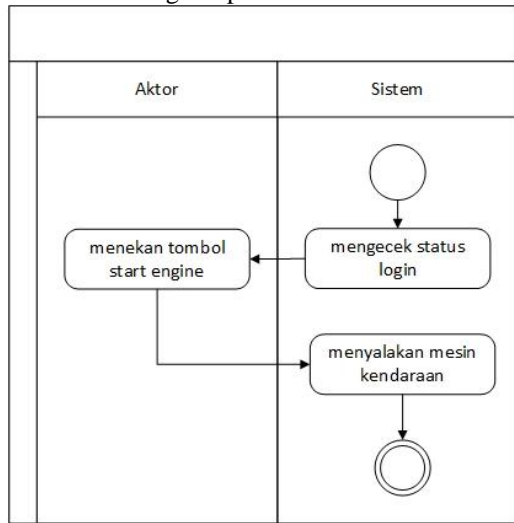
Gambar 4. Activity Diagram : Proses Mengunci Kendaraan

Pengguna dapat mengaktifkan fasilitas untuk menyambungkan arus kunci pada relay, sehingga kendaraan dapat dihidupkan atau dijalankan.



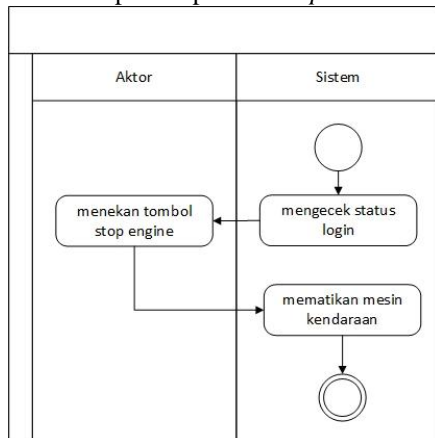
Gambar 5. Activity Diagram : Proses Membuka Kunci Kendaraan

Pengguna dapat mengaktifkan fasilitas *start engine* secara otomatis melalui aplikasi pada *smartphone* Android untuk menghidupkan mesin kendaraan.



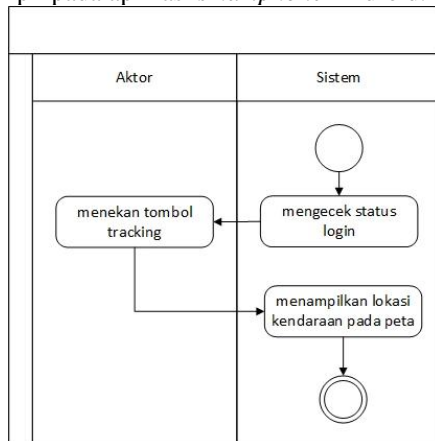
Gambar 6. Activity Diagram : Proses Start Engine

Pengguna dapat mematikan mesin kendaraan secara otomatis melalui aplikasi pada *smartphone* Android



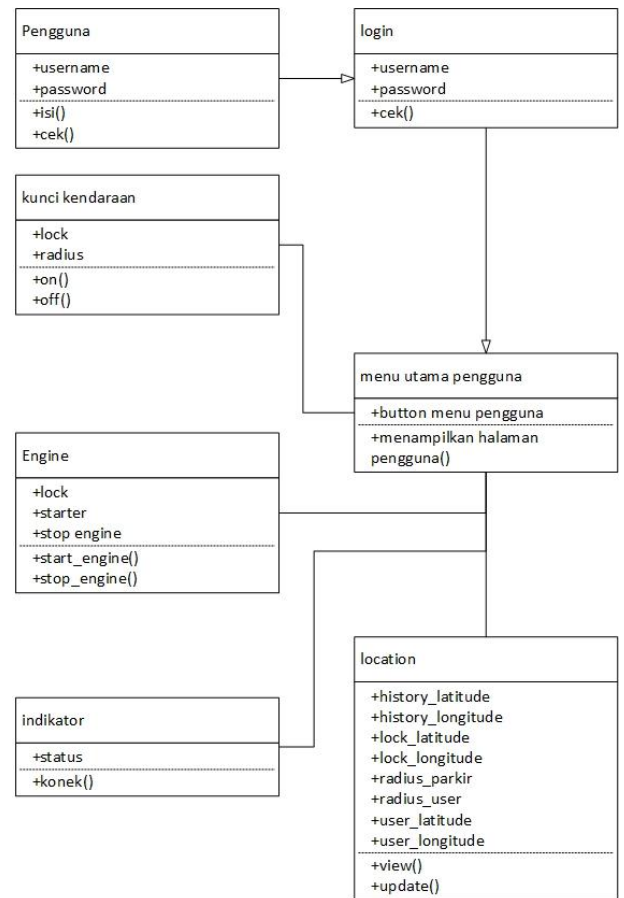
Gambar 7. Activity Diagram : Proses StopEngine

Pengguna dapat melacak posisi kendaraan melalui peta yang tampil pada aplikasi *smartphone* Android.



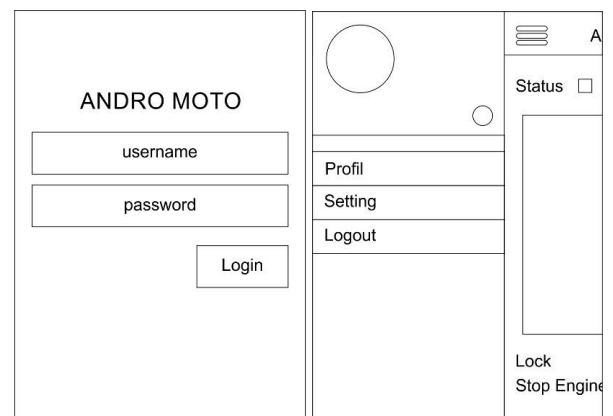
Gambar 8. Activity Diagram : Proses Tracking Kendaraan

Untuk membuat aplikasi pada *smartphone* Android, membutuhkan database untuk merekam data pengguna serta status-status lainnya yang dibutuhkan, seperti yang dapat dilihat pada gambar 9 :

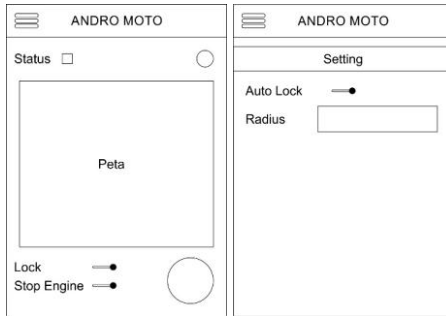


Gambar 9. Class Diagram pada sistem yang dibangun.

Untuk perancangan tampilan pada aplikasi *smartphone* Android, dapat dilihat pada gambar 10 dan gambar 11.



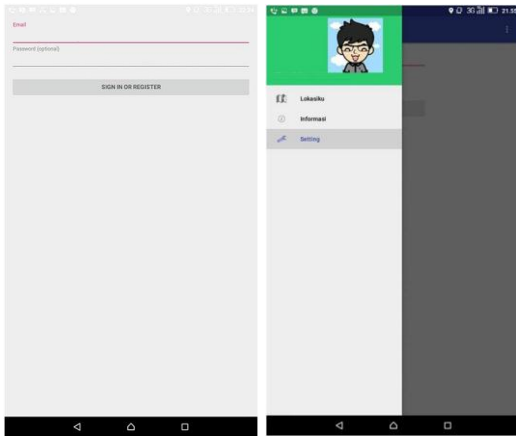
Gambar 10. Rancangan Layout Login dan Halaman Menu



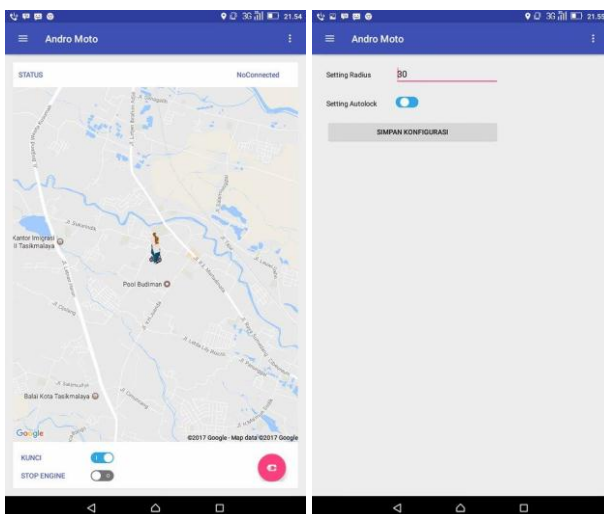
Gambar 11. Rancangan Layout Halaman Utama dan Setting

## 2.2. Implementasi

Implementasi dari perancangan tampilan pada aplikasi *smartphone* android dapat dilihat pada gambar 12 dan gambar 13



Gambar 12. Halaman Login dan Menu Program



Gambar 13. Halaman Utama Program dan Pengaturan Auto Lock

Adapun implementasi perangkat keras (*hardware*) yang digunakan penulis untuk menjalankan program diatas sebagai berikut :

1. Sepeda Motor Yamaha Vixion
2. Relay DC 5V 4 channel
3. Raspberry Pi 3B
4. Kartu SD Card 16 GB Class 10
5. GPS Unblock 6 M
6. Mini Cellular Fona 808
7. Antenna GPS Reciever
8. Step down 5V 3A
9. Powerbank
10. Buzzer 12 V 110 DB
11. Seperangkat Laptop/Notebook
12. Seperangkat gadget Android Lollipop

## 2.3. Pengujian Sistem

Dalam tahap terakhir setelah implementasi sistem yaitu masuk pada tahap pengujian sistem. Pada tahap ini penulis menguji program menggunakan *BlackBox Testing* yang mana pengujiannya dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Berikut pengujian yang dilakukan oleh penulis :

No	Daftar Masukan	Proses yang diharapkan	Pengamatan	Kesimpulan
1	Mengisi halaman login dengan memasukan <i>username</i> dan <i>password</i> .	Jika <i>username</i> dan <i>password</i> tersedia maka program akan menampilkan halaman awal program dan jika <i>username</i> dan <i>password</i> tidak tersedia di <i>database</i> maka akan kembali ke halaman login.	Masuk ke halaman awal jika sukses dan menampilkan kembali halaman login jika gagal	Diterima
2	Switch Kunci	Pengguna dapat mengunci semua arus listrik pada kendaraan.	Sistem mampu mematikan semua arus listrik pada kendaraan.	Diterima
4	SwitchStopEngine	Admin dapat mengaktifkan atau menonaktifkan fitur Engine yang ada pada kendaraan melalui <i>smartphone</i> .	Sistem mampu mengaktifkan dan menonaktifkan Engine kendaraan melalui <i>smartphone</i> .	Diterima
5	Button Starter	Pengguna dapat menghidupkan mesin kendaraan melalui <i>smartphone</i> .	Sistem mampu menghidupkan mesin kendaraan.	Diterima

## Seminar Nasional Sistem Informasi dan Teknologi Informasi 2018

SENSITEK 2018

STMIK Pontianak, 12 Juli 2018

6	Peta <i>Tracking</i>	Sistem dapat menampilkan peta lokasi kendaraan pada aplikasi <i>smartphone</i> .	Sistem mampu menampilkan lokasi kendaraan secara <i>realtime</i> .	Diterima
7	Menu Profil	Pengguna dapat mengubah nama, <i>username</i> dan <i>password</i> .	Sistem mampu mengubah nama, <i>username</i> dan <i>password</i> pada profil.	Diterima
8	Menu <i>Setting</i>	Sistem menampilkan halaman <i>Setting</i> .	Menampilkan halaman <i>Setting</i> .	Diterima
9	Menu <i>Auto Lock</i>	Sistem dapat mengunci dengan mematikan semua arus listrik dengan radius yang di atur pada halaman <i>Setting</i> .	Sistem mampu mengunci kendaraan dengan radius yang telah ditentukan.	Diterima
10	Menu <i>Logout</i>	Pengguna dapat mengeluarkan akun pada aplikasi yang dibangun.	Sistem berjalan sesuai yang diharapkan.	Diterima

### 3. Kesimpulan dan Saran

Dari hasil penelitian ini dapat diambil kesimpulan bahwa sistem yang dibangun dengan konsep IoT dan mini PC

Raspberry Pi sebagai sarana keamanan kendaraan sudah dapat memberikan notifikasi secara *real-time* kepada *user* apabila terjadi kasus pencurian, dan begitupun *user* sudah mampu mengontrol kendaraan jarak jauh menggunakan *smartphone* Android.

Agar sistem ini dapat digunakan sesuai dengan yang diharapkan, maka ada beberapa saran yang dapat dijadikan bahan pertimbangan yaitu sistem yang dibangun untuk kedepannya bisa diterapkan di berbagai jenis kendaraan. Kemudian perangkat *monitoring* dan kontrol bisa digunakan di berbagai macam OS selain Android. Dan penambahan sensor serta fasilitas pada sistem kontrol yang dibuat bisa dijadikan bahan pengembangan untuk menambahkan notifikasi yang lainnya.

### Daftar Pustaka

- [1]. D.N. Bagenda, I. Prasetya, Prototipe Sistem Keamanan Dan Pengendalian Sepeda Motor Berbasis Mikrokontroler ATmega 8535, Jurnal LPKIA, Vol. 1, No. 1, pp. 1-6, 2014.
- [2]. F. Rizkidiniah, M. Yamin, Perancangan Dan Implementasi *Prototype* Sistem GPS (Global Positioning System) Dan SMS Gateway Pada Pencarian Kendaraan Bermotor Berbasis Arduino Uno, Jurnal smanTIK Vol. 2, No. 2, pp. 87-92, 2016.
- [3]. R.H. Muhammad, R.S. Adi, Rancang Bangun Sistem Pengamanan Mobil Menggunakan *ID Card* Dengan Metode *Radio Frequency Identification*, KOPERTIP : Jurnal Ilmiah Manajemen Informatika dan Komputer, Vol. 01, No. 01, pp. 39-44, 2017.