

PROTOTYPE ALAT IoT (INTERNET OF THINGS) UNTUK PENGENDALI DAN PEMANTAU KENDARAAN SECARA REALTIME

Erma Susanti¹, Joko Triyono²

¹Jurusan Teknik Informatika, Fakultas Teknologi Industri, IST AKPRIND Yogyakarta
Jl. Kalisahak No. 28 Kompleks Balapan Yogyakarta 55222 Telp 0274 563029

²Jurusan Teknik Informatika, Fakultas Teknologi Industri, IST AKPRIND Yogyakarta
Jl. Kalisahak No. 28 Kompleks Balapan Yogyakarta 55222 Telp 0274 563029

Email: erma@akprind.ac.id

Abstrak

Kendaraan bermotor merupakan salah satu objek yang sangat rawan untuk dicuri. Antisipasi dari tindak pencurian tidak cukup hanya dengan memasang kunci pengaman, tetapi juga perlu melengkapi kendaraan dengan sistem pemantau dan pengendali. Kendaraan seperti sepeda motor pada umumnya belum dilengkapi dengan GPS (Global Positioning System), sehingga perlu dilengkapi dengan perangkat tersebut dan juga sistem tertanam (*embedded system*) yang dapat dipantau dari jarak jauh. Penggunaan sistem tertanam dapat menjadi pilihan untuk diimplementasikan karena hemat daya dan harganya relatif murah. Penerapan sistem pengendali dan pemantau kendaraan jarak jauh pada penelitian ini diimplementasikan dengan membuat prototype alat IoT (Internet of Things) menggunakan Rapsberry Pi 3 Model B, modem GSM, GPS USB VK-172 Glonass, dan relay DC. Perangkat IoT akan dilengkapi aplikasi yang akan dikoneksikan dengan server Firebase dan selanjutnya dipasang pada kendaraan. Jika kendaraan dihidupkan maka alat IoT akan mengirimkan data posisi kendaraan dari satelit GPS ke server Firebase. Penggunaan server Firebase dipilih karena mendukung pemrograman socket, memiliki pustaka yang lengkap untuk berbagai platform web dan piranti bergerak, dapat digabungkan dengan berbagai framework, dan menyediakan layanan DbaaS (Database as a Service). Hasilnya, data posisi kendaraan yang diterima server, secara langsung (*realtime*) akan ditampilkan melalui aplikasi web sistem pengendali dan pemantau jarak jauh dengan tampilan visualisasi Google Maps dan tombol lock/unlock untuk menghidupkan dan mematikan/mengunci kendaraan.

Kata kunci: IoT, Internet of Things, Pemantau, Pengendali, Raspberry Pi, Firebase

Pendahuluan

IoT (*Internet of Things*) merupakan teknologi yang dapat mengkoneksikan suatu peralatan dengan Internet untuk menjalankan berbagai fungsi. Perangkat IoT dapat diimplementasikan menggunakan *embedded system* (sistem tertanam), karena cenderung hemat daya. Salah satu perangkat yang menggunakan teknologi pengontrol jarak jauh dengan sistem tertanam berbasis ARM (*Advanced RISC Machine*) adalah Raspberry Pi. Raspberry Pi (disingkat raspi) adalah suatu mini komputer yang dapat bekerja seperti halnya personal komputer. Raspberry Pi juga memiliki sambungan LAN yang dapat digabungkan dengan router menjadi jaringan nirkabel yang dapat diakses oleh berbagai perangkat dengan akses internet (Darmaliputra dan Hermawan, 2014). Raspberry Pi dapat dimanfaatkan untuk pembuatan berbagai perangkat IoT yang dapat membantu otomatisasi berbagai pekerjaan. Penerapan IoT pada penelitian ini adalah untuk pengendali dan pemantau kendaraan bermotor. Perangkat pemantau lokasi akan memanfaatkan GPS Glonass untuk mendapatkan data posisi *latitude*, *longitude* dan *waktu*. Selanjutnya data akan dikirim ke server Firebase dan ditampilkan kembali secara *realtime* ke dalam sistem pengendali dan pemantau berbasis web. Firebase dipilih karena tidak hanya menyediakan fasilitas penyimpanan data, tetapi juga menyediakan API untuk pengimplementasian web socket (Oriza, 2014). Sistem pemantau akan menampilkan visualisasi dalam bentuk peta yang menunjukkan lokasi kendaraan. Sistem tertanam ini menggunakan teknologi *realtime* karena memiliki waktu respon yang sangat singkat, sehingga pengguna aplikasi akan mendapatkan *update* informasi kendaraan secara langsung saat terjadi perubahan data, tanpa perlu melakukan *refresh* halaman. Pendekatan menggunakan teknologi monitoring terhadap pergerakan kendaraan secara *realtime* perlu dilakukan untuk mengamankan kendaraan (Muchlisin dan Istiyanto, 2011).

Tujuan dari penelitian ini adalah untuk mengembangkan *prototype* perangkat IoT untuk mengamankan kendaraan dengan cara melakukan pemantauan dan pengendalian terhadap kendaraan dari jarak jauh. Sistem

pengendali dapat bekerja untuk menghidupkan atau mematikan kendaraan. Kendaraan yang telah terpasang alat IoT tersebut dapat dikunci dari jarak jauh melalui aplikasi web yang dapat diakses melalui browser web dari komputer atau perangkat bergerak (*mobile*). Sistem pemantau akan menampilkan data posisi lokasi kendaraan (*latitude* dan *longitude*) yang diperoleh dari satelit GPS (*GPS Receiver*) dan ditampilkan dengan visualisasi dari Google Maps. Saat mesin kendaraan hidup maka aplikasi pada alat IoT akan terus berjalan, dan akan terkoneksi dengan server Firebase. Selanjutnya dari server Firebase informasi akan langsung ditampilkan pada aplikasi web sistem pengendali dan pemantau.

Tinjauan Pustaka

Sistem pemantau kendaraan diteliti oleh Lee, *et al.* (2014) dengan membuat desain dan mengimplementasikan suatu sistem untuk melihat perpindahan kendaraan berdasarkan lokasi di suatu waktu tertentu. Teknologi yang digunakan mengkombinasikan aplikasi telepon pintar (*smartphone*) dengan menggunakan mikrokontroler. Desain alat *tracking* menggunakan GPS dan teknologi GSM/GPRS. Alat tersebut akan ditanamkan ke dalam kendaraan untuk mendapatkan posisi kendaraan secara *realtime*. Sistem *tracking* kendaraan menggunakan modul GPS untuk mendapatkan koordinat geografi dengan interval waktu tertentu. Modul GSM/GPRS selanjutnya mengirim data dan meng-*update* lokasi kendaraan secara *realtime*. Aplikasi *smartphone* digunakan untuk secara berkelanjutan memonitoring lokasi keberadaan kendaraan. Google Maps API digunakan untuk menampilkan visualisasi kendaraan di peta melalui aplikasi *smartphone*. Hasil pengembangan dan pengujian sistem *tracking* untuk memantau lokasi pergerakan kendaraan secara *realtime*. Penelitian secara detail mendeskripsikan desain dan implementasi sistem *tracking* kendaraan. Perangkat *in-vehicle*, server, dan aplikasi *smartphone* digunakan untuk sistem *tracking* kendaraan. Pengujian sistem menunjukkan pemantauan lokasi kendaraan dari berbagai tempat dan diimplementasikan dengan biaya relatif murah.

Widyantara and Sastra (2015) melakukan penelitian tentang IoT untuk sistem monitoring lalu lintas di kota Denpasar. Tujuannya adalah untuk menampilkan visualisasi lalu lintas melalui web berbasis GPS/GPRS. Implementasi IoT difokuskan pada akuisisi lalu lintas memanfaatkan kemampuan GPS sebagai sensor, komunikasi data dengan GPRS dan desain sistem monitoring berbasis web/GIS. Hasilnya menunjukkan interkoneksi antara GPS Tracker, GPRS dan jaringan internet sebagai skema IoT untuk pengembangan ITMS (*Intelligent Traffic Monitoring System*) di Denpasar Bali. Target penelitian ITMS Denpasar adalah pengembangan pusat manajemen lalu lintas yang memiliki kemampuan memproses data lalu lintas dari alat GPS Tracker menjadi ITMS *realtime* berbasis Google MAP API dan menyediakan layanan informasi lalu lintas dalam bentuk GIS. IoT tersebut dibangun menggunakan model arsitektur jaringan dan menggunakan desain perangkat lunak untuk antarmuka aplikasi dengan menerapkan metode desain interaksi *client-server*.

Penelitian Vigneshwaran, *et al.* (2015) membahas tentang pencegahan tindakan pencurian menggunakan teknologi GSM dan teknologi GPS. GPS digunakan untuk mendapatkan posisi lokasi kendaraan. Data dikirimkan menggunakan telepon genggam dengan GSM. Posisi yang didapatkan dari modul GPS dikirim ke mikrokontroler dengan mengirim ke pengguna *smartphone* melalui GSM dengan modem GSM dan modul GPS. Implementasi menggunakan mikrokontroler Atmel. Jika terjadi tindak pencurian, maka kendaraan dapat dimatikan melalui aplikasi Android. Solenoid dan komponen *relay* digunakan untuk mengontrol kendaraan. Hasil implementasi dan pengujian sistem secara *realtime* dilakukan dengan melakukan pengecekan sistem *tracking*. Sistem *tracking* diinstal ke dalam kendaraan dan dilakukan pengujian dengan melakukan perjalanan ke tiga kota. Sistem *tracking* menunjukkan status interval dan respon sukses.

Desain monitoring dan *tracking* kendaraan sekolah dari lokasi A ke lokasi B secara *realtime* dilakukan oleh Shinde and Mane (2015); Shinde, *et al.* (2015). Sistem yang diusulkan menggunakan teknologi berbasis *Embedded Linux Board* bernama Raspberry Pi dan menggunakan fitur penyimpanan basis data secara *realtime*. Sistem bekerja dengan GPS dan GSM yang digunakan untuk pelacakan dan monitoring kendaraan. Modul SIM908 digunakan untuk GPS, GPRS dan GSM. GPS memberikan informasi terkini kendaraan, GPRS mengirimkan hasil *tracking* informasi ke server dan GSM digunakan untuk mengirimkan pesan ke pemilik kendaraan. Sistem yang ada pada kendaraan akan dimonitor secara *realtime* melalui web. Sistem juga dilengkapi dengan sensor gas LPG MQ6 dan sensor suhu DS18B20 untuk deteksi gas dan temperatur. Sistem monitoring informasi kendaraan diimplementasikan menggunakan LAMP (Linux, Apache, MySQL, dan PHP). Hasil pengujian dilakukan untuk menguji jalannya sensor suhu dan sensor gas. Hasil sensor dari peralatan akan dikirimkan melalui pesan yang dikirim ke pemilik kendaraan.

Leelavathi, *et al.* (2016) meneliti tentang penggunaan IoT dengan sistem tertanam dengan membuat aplikasi mobil pintar (*smart car*). Metode *capturing image* dan pelacakan lokasi digunakan untuk mencari kendaraan yang hilang dan juga mendeteksi adanya indikasi aktivitas berbahaya pada kendaraan. Tujuan pengembangan alat IoT pada penelitian ini adalah untuk keamanan kendaraan dari pencurian. Desain sistem *tracking* untuk kendaraan menggunakan sensor pergerakan dan IoT. Sistem penguncian kendaraan menggunakan sensor seperti yang diterapkan pada pintu, deteksi pergerakan, dan GPS. Aplikasi *Smart car* digunakan untuk sistem penguncian pada kendaraan. Perangkat *mobile* dengan sistem operasi Android diinstal dengan aplikasi *smart car*. Untuk mengakses

aplikasi harus melakukan login. Aplikasi ini dapat digunakan untuk melakukan penguncian atau membuka pintu kendaraan. Aplikasi juga akan memberikan perintah suara jika ada indikasi pencurian atau aktivitas mencurigakan pada kendaraan. Perangkat *wifi dongle* juga digunakan untuk akses internet. Alat IoT dibuat menggunakan Raspberry Pi untuk aplikasi tertanam. Komponen antarmuka menggunakan Pi camera, modem GSM dan GPS, sensor pintu, sensor *proximity*, dan *wifi dongle*. *Default* sistem operasi yang digunakan adalah Linux, sistem operasi ini digunakan untuk menangani tugas dan komponen alat pada Chip. Skrip Python digunakan untuk pemrograman pada alat dan fungsionalitas. Sensor penguncian pintu akan terdeteksi jika ada pergerakan yang mencurigakan di dalam kendaraan.

Teori Raspberry Pi

Komponen *board* (papan) Raspberry Pi model B terdiri dari port USB untuk mengkoneksikan berbagai perangkat USB seperti *keyboard*, *mouse*, dan lain-lain. Mini USB port digunakan untuk menghubungkan ke *power adaptor*. Untuk terkoneksi ke jaringan bisa menggunakan port Ethernet/LAN atau pada Raspberry Pi 3 model B sudah dilengkapi dengan *wifi built-in*. Raspberry Pi juga sudah mendukung audio/video. Untuk mengkoneksikan ke monitor/tv dapat menggunakan HDMI atau RCA. Beberapa pin GPIO (*General Purpose Input/Output*) dapat digunakan untuk mengkoneksikan dengan perangkat elektronik lainnya. Gambar 1 merupakan komponen bagian depan Raspberry Pi model B (Schmidt, 2012). Sistem operasi yang direkomendasikan untuk perangkat Raspberry Pi adalah Raspbian. Raspbian merupakan distribusi resmi sistem operasi untuk perangkat Raspberry Pi berbasis Debian (Richardson and Wallace, 2013). Fungsi Raspberry secara keseluruhan dapat dijalankan oleh sistem operasi. Sistem operasi yang dapat digunakan pada Raspberry Pi antara lain Raspbian, Pidora, OpenElec, RaspBMC, RISC OS, Arch Linux ARM, dan lain-lain (Rakhman, dkk., 2014).



Gambar 1. Komponen Bagian Depan Raspberry Pi Model B (Schmidt, 2012)

Firestore

Firestore dapat bertindak sebagai basis data *realtime* sehingga aplikasi sistem pemantau dan pengendali akan dapat disinkronisasi dengan basis data untuk menyediakan data secara *realtime*. Keuntungan menggunakan basis data Firestore sebagai *backend* adalah memastikan bahwa *latency* pada proses penyimpanan data ke server sangat kecil. Semua operasi dengan firestore pada pengembangan aplikasi menggunakan AngularFire karena operasi API telah menyediakan objek dan *array* seperti *add*, *remove*, dan *get* (Phan, 2015). AngularFire memanfaatkan koneksi antara AngularJS dan Firestore. Kombinasi tersebut menggunakan tiga cara koneksi yaitu HTML, JavaScript, dan basis data Firestore. Firestore menjamin ‘*no server code*’ sehingga memudahkan pembuatan aplikasi. Firestore juga dapat melakukan sinkronisasi data dengan *local cache* pada perangkat yang ada sehingga aplikasi tetap responsif ketika koneksi jaringan lambat (Max, 2014).

Teknologi GPS

GPS (*Global Positioning System*) merupakan suatu sistem navigasi menggunakan satelit yang diletakkan pada orbit. GPS Receiver akan menerima sinyal posisi satelit (*latitude* dan *longitude*). *GPS Receiver* pada penelitian ini digunakan untuk mendeteksi lokasi kendaraan dan sekaligus menyediakan informasi yang dapat diakses dari jarak jauh melalui aplikasi sistem pemantau berbasis web. GPS menggunakan standar protokol NMEA untuk dapat mentransmisikan posisi data. Data posisi lokasi yang didapatkan dari GPS akan dikirimkan ke server secara *realtime* melalui Firestore.

Relay DC

Raspberry pi mengendalikan peralatan 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 alat IoT. Koil motor dapat digerakkan dengan menggunakan relay DC dengan tegangan koil sebesar 12V DC dan arus yang diberikan sekitar 20-30 mA.

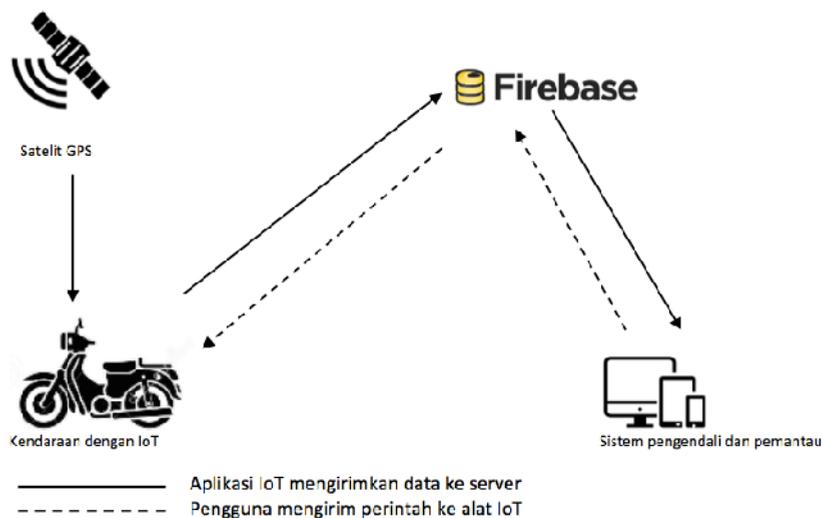
Metode Penelitian

Implementasi penelitian ini terdiri dari perangkat keras dan perangkat lunak. Perangkat keras untuk pembuatan alat IoT antara lain sistem tertanam menggunakan Raspberry Pi 3 Model B, modul GPS menggunakan USB GPS VK-172 Glonass, dan Relay DC. Gambar 2 adalah perangkat keras utama yang digunakan untuk penelitian ini yaitu Relay DC, USB GPS Glonass, dan Raspberry Pi 3 Model B. Perangkat lunak untuk aplikasi pada alat IoT terdiri dari sistem operasi Raspbian dan NodeJS. Sedangkan perangkat lunak untuk server dan aplikasi sistem pemantau dan pengendali jarak jauh dikembangkan menggunakan AngularFire (AngularJS dan Firebase) dan Google Maps API. Aplikasi tersebut berbentuk web yang dapat diakses dari berbagai perangkat baik itu komputer maupun perangkat *mobile* melalui browser.



Gambar 2. Perangkat Keras untuk membuat alat IoT

Arsitektur dari aplikasi IoT dan koneksi Raspberry Pi ke sistem pemantau dan pengendali dapat dilihat pada Gambar 3. Langkah pertama lakukan instalasi perangkat Raspberry Pi dengan sistem operasi Raspbian dan buat kode program untuk aplikasi *client* menggunakan NodeJS. Selanjutnya koneksikan perangkat Raspberry Pi dengan internet menggunakan *wifi dongle* atau 3G/4G modem GSM dan pasang USB GPS. Untuk Raspberry Pi 3 Model B sudah tersedia perangkat *wifi* sehingga tidak perlu perangkat *wifi dongle* tambahan. Setelah itu hidupkan perangkat dan nyalakan *service* aplikasi untuk bisa mendapatkan data posisi *latitude* dan *longitude* dari satelit GPS. Data yang diperoleh dari *GPS receiver* secara *realtime* akan dikirim ke server, dan dari server Firebase data lokasi akan diproses untuk ditampilkan kembali ke dalam bentuk peta Google (Google Maps). Gambar panah menunjukkan alat IoT akan mengirimkan data posisi ke server, sedangkan gambar panah putus-putus menunjukkan alur sistem jika pengguna mengirimkan perintah kepada alat IoT.



Gambar 3. Arsitektur Alat IoT Untuk Pengendali dan Pemantau Kendaraan

Perangkat IoT dirancang dengan sistem tertanam menggunakan *board* (papan) tunggal Raspberry Pi untuk menanamkan aplikasi yang akan terhubung dengan server Firebase. Server Firebase pada penelitian ini dipilih karena memiliki banyak pustaka (*library*) untuk pengembangan aplikasi web maupun *mobile* serta mendukung penerapan DbaaS (*Database as a Service*) untuk *cloud computing*. Perangkat Raspberry Pi 3 Model B sudah dilengkapi dengan perangkat wifi untuk akses Internet dan juga terdapat 4 port USB yang dapat digunakan untuk menghubungkan beberapa komponen tambahan. Sebagai bahan percobaan, perangkat USB GPS VK-172 Glonass akan digunakan sebagai *GPS receiver*, kemudian komponen Relay DC juga ditambahkan sebagai pengendali on/off kendaraan. Lokasi GPS yang didapat akan dikirim menggunakan Internet ke server Firebase dan selanjutnya akan ditampilkan melalui aplikasi sistem pemantau dengan visualisasi menggunakan Google Maps.

Implementasi alat IoT terdiri dari beberapa tahapan. Pertama tahap persiapan alat IoT dengan melakukan instalasi sistem operasi Raspbian untuk perangkat Raspberry Pi. Kemudian instal NodeJS dan membuat kode untuk akuisisi perangkat GPS. Selanjutnya membuat kode untuk Google Maps, kode perintah ke GPIO (*General Purpose Input Output*), dan kode untuk koneksi ke server. Setelah itu aplikasi ditanamkan pada alat IoT dan akan dijalankan *service* aplikasi untuk selalu terkoneksi dengan server. Aplikasi harus terus berjalan (*running*) jika kendaraan dihidupkan.

Implementasi untuk aplikasi server adalah berbentuk aplikasi web yang ditampilkan dengan visualisasi peta yang menunjukkan lokasi kendaraan dan juga tombol on/off untuk kendali kendaraan. Aplikasi server dibuat dengan menggabungkan antara AngularJS dengan Firebase (AngularFire).

Cara kerja sistem, pertama kendaraan dengan IoT akan melakukan akuisisi lokasi melalui perangkat GPS secara langsung. Data *latitude*, *longitude* dan waktu yang didapatkan terlebih dahulu akan dibandingkan dengan data yang sudah disimpan terakhir, apabila kendaraan tidak berubah posisi maka data tersebut akan disimpan ke dalam penyimpanan lokal dan selanjutnya akan di *broadcast* ke server.

Sebelum perangkat dipasang, terlebih dahulu dipersiapkan sistem operasi Raspbian Jessie yang akan menangani perangkat Raspberry Pi. Kemudian dilakukan instalasi aplikasi NodeJS dan pengaturan pin GPIO. Kemudian aplikasi dipasang pada alat IoT untuk mendapatkan data lokasi dari satelit GPS. Aplikasi pada IoT tersebut akan langsung terkoneksi ke server saat kendaraan dihidupkan. Aplikasi akan terus dijalankan selama kendaraan dalam keadaan hidup. Jika data didapat maka data akan dikirimkan ke dengan server. Sedangkan pada sistem pengendali, jika pengguna memberikan perintah melalui aplikasi web, maka perintah akan dikirim ke server, dan server akan mengirimkan perintah ke alat IoT. Terakhir lakukan wiring pada kendaraan dan lakukan pengujian fungsional aplikasi pada kendaraan.

Hasil dan Pembahasan

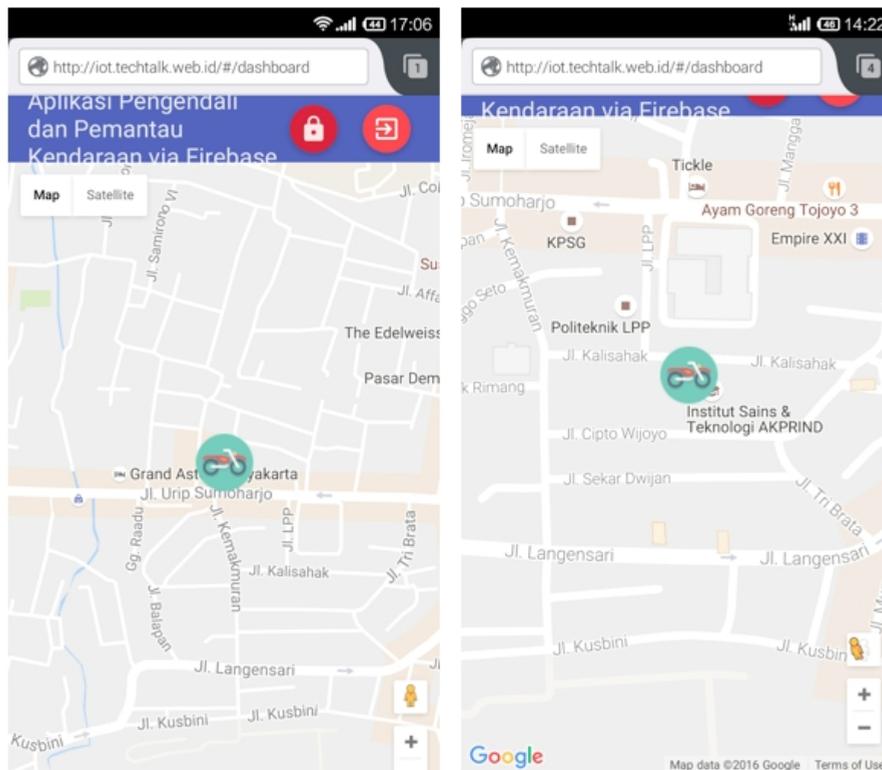
Pengujian sistem pemantau dan pengendali dilakukan dengan menguji fungsionalitas aplikasi web apakah sudah dapat mengirim perintah dan direspon oleh alat IoT. Proses pengujian pertama dengan melakukan *wiring* pada kendaraan dengan memasang Relay DC dan menghubungkannya dengan alat IoT. Selanjutnya dilakukan pengujian secara langsung pada kendaraan.

Hasil *prototype* awal alat IoT untuk pemantau dan pengendali dapat dilihat pada Gambar 4. Pengujian pertama dengan melakukan pengujian terhadap kinerja perangkat GPS. Alat IoT tersebut telah dilengkapi dengan sistem operasi Raspbian dan aplikasi *client* yang dibuat dengan NodeJS. Aplikasi *client* ditanamkan di alat untuk menjalankan *service* untuk mendapatkan data posisi dengan mengirimkan sinyal ke satelit GPS, selanjutnya data akan dikirim ke server pemantau secara *realtime*. Sedangkan alat pengendali dibuat dengan menghubungkan alat IoT dengan Relay DC.



Gambar 4. Prototype alat IoT untuk pemantau dan pengendali

Alat IoT pada saat dihidupkan akan tersinkronisasi dengan Server Firebase. Pergerakan kendaraan selanjutnya dapat dipantau melalui aplikasi web sistem pemantau dan pengendali. Aplikasi dilengkapi dengan login pengguna untuk keamanan akses aplikasi oleh pengguna yang tidak berhak. Hasil sistem pemantauan ditunjukkan pada Gambar 5. Data *latitude* dan *longitude* posisi GPS kendaraan dikirimkan ke server Firebase. Dari server Firebase selanjutnya ditampilkan menggunakan Google Maps. Aplikasi ditunjukkan dengan Google Maps dan icon posisi kendaraan dengan IoT. Aplikasi pengendali dapat digunakan untuk mematikan dan mengunci kendaraan jika terjadi pencurian.



Gambar 5. Sistem Pemantau dan Pengendali berbasis Web

Kesimpulan

Perangkat IoT untuk pengendali dan pemantau kendaraan bermotor diimplementasikan menggunakan perangkat keras antara lain Raspberry Pi 3 Model B, USB GPS VK-172 Glonass, relay DC dan dikoneksikan dengan Internet menggunakan modem 3G/4G GSM. Sedangkan untuk perangkat lunak terdiri dari perangkat lunak untuk *client* dan *server*. Aplikasi *client* dikembangkan dengan menggunakan NodeJS, sedangkan untuk server dikembangkan menggunakan AngularJs dan Firebase (AngularFire). Aplikasi *client* akan diletakkan pada Raspberry Pi dan akan secara otomatis berjalan setiap kendaraan hidup. Selanjutnya perangkat akan melakukan akuisisi GPS dengan mencari posisi koordinat lokasi *latitude* dan *longitude*. Data posisi tersebut kemudian dikirimkan secara langsung (*realtime*) ke server Firebase. Selanjutnya dari server Firebase akan merespon lokasi yang didapatkan tersebut dengan menampilkannya ke dalam Google Maps. Sedangkan untuk sistem pengendali, maka pengujian dilakukan dengan mencoba menghidupkan dan mematikan kendaraan dari jarak jauh melalui aplikasi web yang dapat diakses dari komputer atau perangkat *mobile*. Hasil uji coba alat IoT dan sistem pemantau jarak jauh secara fungsional dapat merespon posisi kendaraan dan menampilkannya secara langsung. Uji coba alat IoT untuk sistem pengendali juga dapat merespon perintah yang diberikan melalui sistem.

Hasil penelitian masih memerlukan banyak pengembangan, karena masih mengalami kendala pada keamanan peletakan alat dan kerentanan alat dari air hujan. Kendala lainnya adalah pada koneksi internet yang sering tidak stabil dan koneksi ke satelit GPS yang tidak berjalan semestinya.

Daftar Pustaka

- Darmaliputra, A. dan Hermawan, H., (2014), "Pembuatan Web Server Berbasis Raspberry Pi Untuk Kontrol Lampu dan AC" *Calyptra*, Vol. 3 (1).
- Lee, S., et al., (2014), "Design and Implementation of Vehicle Tracking System Using GPS/GSM/GPRS Technology and Smartphone Application" *IEEE World Forum on Internet of Things (WF-IoT)*, pp. 353-358.

- Leelavathi, T. C., et al., (2016), "IoT for Smart Car using Raspberry Pi" *International Research Journal of Engineering and Technology (IRJET)*, Vol. 03 (06) pp. 1376-1379
- Max, D. (2014), "Building Real Time Apps With Firebase and AngularJS", <http://davemax.com>, diakses 19 Maret 2015.
- Muchlisin, Y. D. dan Istiyanto, J. E., (2011), "Implementasi Pelacakan Kendaraan Bermotor Menggunakan Gps dan Gprs dengan Integrasi Googlemap" *IJCCS – Indonesian Journal of Computing and Cybernetics Systems*, Vol. 5 (2) pp. 76-84.
- Oriza, A. (2014), "Firebase Membantu Kita Membuat Aplikasi Realtime", <http://www.codepolitan.com/firebase-membantu-kita-membuat-aplikasi-realtime/>, diakses 21 Maret 2015
- Phan, H., (2015), "Ionic Cookbook", Packt Publishing, pp. 209-210
- Rakhman, E., et al., (2014), "Raspberry Pi – Mikrokontroler Mungil yang Serba Bisa", Andi.
- Richardson, M. and Wallace, S., (2013), "Getting Started with Raspberry Pi", O'Reilly Media, Inc., pp. 10
- Schmidt, M., (2012), "Raspberry Pi A Quick - Start Guide", The Pragmatic Programmers, LLC., pp. 2
- Shinde, P. and Mane, (2015), "Advanced Vehicle Monitoring and Tracking System based on Raspberry Pi" *IEEE Sponsored 9th International Conference on Intelligent Systems and Control (ISCO)*.
- Shinde, P., et al., (2015), "Real Time Vehicle Monitoring and Tracking System based on Embedded Linux Board and Android Application" *International Conference on Circuit, Power and Computing Technologies (ICCPCT)*.
- Vigneshwaran, K., et al., (2015), "An Intelligent Tracking System Based on GSM and GPS Using Smartphones" *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE)*, Vol. 4 (5) pp. 3897-3903.
- Widyantara, I. M. O. and Sastra, N. P., (2015), "Internet of Things for Intelligent Traffic Monitoring System: A Case Study in Denpasar" *International Journal of Computer Trends and Technology (IJCTT)*, Vol. 30 (3), pp. 169-173.