

PENGEMBANGAN SISTEM PEMANTAU DAN PENGENDALI KENDARAAN MENGGUNAKAN RASPBERRY PI DAN FIREBASE

Erma Susanti¹, Joko Triyono²

^{1,2}Jurusan Teknik Informatika, Fakultas Teknologi Industri, IST AKPRIND, Yogyakarta, Indonesia

¹erma@akprind.ac.id,²zainjack@gmail.com

Abstrak

Peningkatan kejahatan pencurian kendaraan bermotor sering terjadi karena lemahnya pengawasan terhadap kendaraan. Umumnya kendaraan seperti mobil, saat ini sudah dilengkapi dengan perangkat GPS (Global Positioning System), sedangkan pada sepeda motor belum. Penerapan teknologi IoT (Internet of Things) dapat diterapkan untuk mengatasi permasalahan pemantauan/pelacakan dan pengendalian terhadap kendaraan. Sistem pemantau dan pengendali kendaraan jarak jauh pada penelitian ini dikembangkan dengan teknologi web realtime menggunakan AngularJS dan Firebase. Sistem diintegrasikan dengan alat IoT yang diimplementasikan menggunakan Raspberry Pi 3 Model B, GPS Ublox Neo-6M, Relay DC dan Modem GSM. Cara kerja sistem pemantau adalah dengan melakukan akuisisi data posisi koordinat dari satelit GPS pada kendaraan, mengirimkan ke server Firebase dan selanjutnya menampilkannya secara real time melalui web dengan visualisasi Google Maps. Sistem pengendali bekerja untuk menghidupkan atau mematikan kendaraan dari jarak jauh. Hasil uji coba prototype sistem pemantau berhasil mendeteksi posisi kendaraan yang telah dipasang alat IoT dan menampilkan visualisasinya secara real time.

Kata Kunci : pemantau, pengendali, IoT, Raspberry Pi, Firebase.

1. Pendahuluan

Peningkatan jumlah kendaraan bermotor dewasa ini merupakan salah satu penyebab maraknya kejahatan pencurian kendaraan, baik yang berada di jalanan maupun yang sedang di parkir. Badan Pusat Statistik (BPS) Indonesia dalam laporan Statistik Kriminal 2013 mencatat kejahatan pencurian kendaraan bermotor dari tahun 2010 sampai dengan 2012 mengalami kenaikan dari 35.688 kejadian menjadi 41.816 kejadian atau naik sekitar 17,2% (BPS, 2013). Tindak pencurian kendaraan terbanyak yang tercatat adalah terjadi pada sepeda motor. Peningkatan tindak pencurian tersebut diduga terjadi karena lemahnya pengawasan keamanan terhadap kendaraan, khususnya sepeda motor, karena biasanya sepeda motor belum dilengkapi dengan perangkat GPS (Global Positioning System). Modul GPS untuk kendaraan yang ada di pasaran saat ini dinilai masih mahal, karena pemilik kendaraan tidak bisa mengontrol dan mengawasi sendiri kendaraannya,

selain itu biaya transfer data posisi masih relatif mahal.

Perangkat GPS yang banyak digunakan saat ini sebagian besar ditanamkan (*embedded*) pada perangkat seperti telepon genggam, komputer, atau modul GPS terpisah seperti pada mobil yang dikendalikan oleh mikrokomputer. Umumnya sistem yang ada selama ini, baru dapat melakukan pemantauan, tetapi belum dapat mengendalikan kendaraan, seperti menghidupkan atau mematikan mesin dari jarak jauh. Solusi untuk semua permasalahan tersebut adalah dengan membuat suatu sistem pemantau dan pengendali kendaraan dari jarak jauh menggunakan teknologi web *realtime*.

Sistem pemantau dan pengendali akan memanfaatkan teknologi GPS, *socket programming*, dan Google MAPS API. Sedangkan untuk perangkat keras untuk pemantau dan pengendali akan menggunakan *Raspberry Pi*, modul GPS, *relay DC*, dan modem GSM. *Raspberry Pi* dengan modul GPS ini akan diaplikasikan pada kendaraan bermotor

untuk mendapatkan posisi dari GPS secara *realtime*. Posisi kendaraan akan dilacak oleh satelit GPS dan data koordinat lokasi/posisi tersebut akan dikirimkan ke *web server* secara periodik, selanjutnya data akan ditampilkan dalam bentuk peta menggunakan Google MAPS API.

Framework untuk mengimplementasikan web *realtime* akan menggunakan teknologi *AngularJS*. Sedangkan, untuk *web server* akan menggunakan *Firebase* karena mendukung *socket programming* dan dapat diintegrasikan dengan berbagai platform *web*. *Firebase* digunakan untuk membantu membuat aplikasi *realtime*. Cara kerjanya, jika ada masukan (*input*) data maka pada saat itu juga keluaran (*output*) akan langsung tampil saat itu juga.

Salah satu pilihan untuk membuat aplikasi *realtime* adalah menggunakan *web socket*. Beberapa implementasi menggunakan pustaka (*library*) dan *socket server* antara lain *socket.io* (*Node.js*), *Ratchet* (*PHP*), *Tornado* (*Python*), *Socky* (*Ruby*), dan lain-lain. Pilihan lain adalah dengan menggunakan layanan pihak ketiga dengan *Firebase*. Penggunaan *Firebase* dapat mengurangi waktu pembelajaran dan penelitian secara drastis. Selain itu, *Firebase* menggunakan konsep *realtime* dalam menyediakan layanan *DbaaS (Database as a Service)*. *Firebase* tidak hanya menyediakan fasilitas untuk menyimpan data, tetapi juga menyediakan API untuk implementasi *web socket* (Oriza, 2014). *Firebase* API dapat menyimpan dan melakukan sinkronisasi data secara *realtime*. *Firebase* juga dapat diintegrasikan dengan *server* yang menggunakan *Android*, *Node.js*, *Java Script*, *Java*, *AngularJS* dan lain-lain. Pengguna akan mendapatkan update secara langsung saat terjadi perubahan data, tanpa perlu *me-refresh* halaman.

Penelitian ini diharapkan dapat bermanfaat bagi orang/individu atau perusahaan yang bergerak dalam bisnis rental kendaraan untuk dapat mengamankan kendaraan dengan melakukan pemantauan dan pengendalian kendaraan, jika terjadi pencurian. Penggunaan teknologi web *realtime*, *Raspberry Pi* dan *Firebase* diharapkan dapat melakukan *monitoring* posisi dan pengendalian seperti mengirimkan perintah untuk mematikan dan mengunci kendaraan apabila terjadi pencurian.

2. Kajian Literatur dan Teori

2.1. Kajian Penelitian Terdahulu

Sistem pemantau posisi kendaraan berbasis aplikasi dekstop memanfaatkan *SMS Gateway* pernah dikembangkan oleh Hanifah, dkk (2010). Sistem serupa untuk pelacakan posisi kendaraan secara *realtime* berbasis *web* dengan teknologi GPS memanfaatkan *Google Maps* melalui komunikasi menggunakan *GPRS* juga pernah dilakukan oleh Purnama (2009), Muchlisin dan Istiyanto (2010), Junus (2012), Rifai (2013), Verma dan Bathia (2013). Teknologi pelacakan kendaraan tersebut memanfaatkan perangkat *GPS Receiver* yang dihubungkan dengan perangkat *GSM/GPRS* sebagai media pengirim data dari *server*.

Purnama (2009) melakukan penelitian menggunakan *GPS Tracker CT-58* dan sistem pemantau menggunakan *mTrack*. Junus (2012) melakukan *parsing data* dari *web server* untuk diimplementasikan ke dalam program *web*. Hasil *parsing* ditampilkan dalam bentuk peta menggunakan *Google Maps API* dan *Javascript*. Muchlisin dan Istiyanto (2010) menggunakan perangkat GPS dari *smartphone*. Hasil pengiriman data dari *smartphone* ke *server* berupa *latitude* dan *longitude* dalam format *NMEA-0183* yang diterima dari *GPS receiver*, data tersebut kemudian di-*parsing* ke *website* melalui koneksi *GPRS*. Verma dan Bathia (2013) membuat desain dan pengembangan sistem pelacakan kendaraan dengan *GPS*, *GSM*, mikrokontroler *MAX232* dan aplikasi web dengan *monitoring* menggunakan *Google Maps*.

Al-Khedher (2011) dan Maurya dkk. (2012) meneliti tentang sistem pelacakan kendaraan menggunakan aplikasi *Google Earth* yang diintegrasikan dengan *GPS-GSM*. Pada penelitian Al-Khedher (2011) Koordinat *GPS* difilter menggunakan *Kalman filter* untuk meningkatkan akurasi dari posisi yang diukur. Sistem terdiri dari dua komponen utama yaitu modul tertanam untuk transmisi di *GPS* kendaraan dan alat *GSM* untuk mengirimkan lokasi dan status informasi melalui *SMS*. Modul kedua adalah modul penerima untuk mengumpulkan informasi yang diterima dalam format yang kompatibel dengan *GoogleEarth* untuk memonitor lokasi dari jarak jauh. Lokasi kendaraan yang ditransmisikan kemudian difilter menggunakan *Kalman filter* untuk menerima pelacakan secara akurat.

Huang dkk. (2011) pernah mendesain arsitektur sistem pemantauan posisi GPS kendaraan yang diintegrasikan dengan CORS dan *mobile GIS*. Teknologi GPS digunakan untuk melakukan pelacakan objek bergerak dan CORS merupakan sistem layanan posisi jaringan secara *realtime* yang menghasilkan informasi lokasi *geospatial* untuk pergerakan target secara *realtime*. Khan dan Mishra (2012) membuat desain sistem *monitoring* kendaraan menggunakan GPS dan modem GSM dengan prosesor ARM diinstall pada kendaraan. Desain dan integrasi sistem dengan GPS-GSM digunakan untuk menyediakan informasi lokasi dan pelacakan secara *realtime* menggunakan SMS. Desain dan implementasi sistem pelacakan kendaraan lainnya yang menggunakan teknologi GPS/GSM/GPRS dan aplikasi *smartphone* juga dibuat oleh Lee dkk. (2014). Implementasi menggunakan mikrokontroler Arduino Atmega328, modul GPS, modul GSM/GPRS, komunikasi HTTP, Web Server dan database MySQL serta Google Maps API.

Sistem pelacakan dan pengunci kendaraan berbasis GSM dan GPS juga pernah diimplementasikan oleh Ramani dan Valarmathy (2013). Sistem menggunakan mikrokontroler Atmel AT89C52, modul GPS dan modul GSM. Sistem menempatkan kendaraan dalam mode *sleep*, tetapi jika ada indikasi pencurian maka mode penguncian jarak jauh dapat diaktifkan dengan mengirim sms ke mikrokontroler untuk mengeluarkan sinyal kontrol untuk menghentikan mesin motor.

Berdasarkan kajian literatur-literatur tersebut, umumnya penelitian-penelitian yang ada lebih banyak membahas tentang sistem *monitoring* atau pelacakan atau pemantau kendaraan, sangat sedikit penelitian yang sekaligus membahas tentang sistem pengendalian hidup/mati kendaraan dari jarak jauh. Oleh karena itu, penelitian ini akan mengimplementasikan *prototype* sistem pengendali dan juga pemantau kendaraan bermotor jarak jauh menggunakan teknologi web *realtime* dengan nodeJS untuk aplikasi *client* dan *angularFire* (*AngularJS* dan *Firebase*) untuk aplikasi *server*. Aplikasi *client* akan ditanamkan pada alat IoT yang diimplementasikan dengan menggunakan perangkat *Raspberry Pi*, modul GPS Ublox, *Relay DC*, dan *modem GSM*. Alat IoT yang dipasang pada kendaraan akan mengirimkan data posisi kendaraan secara *realtime* yang diperoleh dari modul GPS ke *serverFirebase*. Selanjutnya dari *serverFirebase*

data akan diproses untuk ditampilkan kembali secara *realtime* ke sistem pemantau dalam bentuk visualisasi peta menggunakan *Google Maps*. Sedangkan pada sistem pengendali, jika diberikan perintah melalui sistem pemantau, maka prosesnya dari *Server Firebase*, perintah akan dikirimkan ke alat IoT untuk menghidupkan atau mematikan kendaraan.

2.2. Raspberry Pi

Raspberry Pi merupakan sebuah komputer mini hasil karya sebuah yayasan nirlaba Inggris “*Raspberry Pi Foundation*”. *Raspberry Pi* berlisensi *Open-Source* sehingga rancangan perangkat keras yang dirilis ke publik dapat bebas dipelajari, dimodifikasi, dirakit, didistribusikan/disebarkan dan dijual sesuai dengan rancangan aslinya. Awalnya *Raspberry Pi* dikembangkan untuk membantu proses pengajaran dasar ilmu komputer di sekolah-sekolah dengan biaya yang minim. *Raspberry Pi* juga dapat digunakan untuk keperluan perangkat keras seperti aplikasi *web server*. Aplikasi *web server* merupakan perangkat lunak yang dapat berjalan pada sisi server dan bertugas menerima permintaan dari *browser web* serta menerjemahkan permintaan (*request*) dan mengembalikan hasil (*response*) ke *browser* (Upton dan Halfacree, 2012).

Sistem pemantau atau pelacakan posisi kendaraan jarak jauh dilakukan melalui *web* yang dikendalikan oleh *web server*. Komputer yang digunakan untuk *web server* harus selalu dalam keadaan hidup, sehingga hal ini menyebabkan pemborosan energi. *Embedded system* digunakan untuk mengatasi masalah pemborosan energi. Kelebihan dari *embedded system* adalah pada harga yang murah dan hemat daya. Teknologi pengontrol jarak jauh dimana pengontrolannya dilakukan oleh *embedded system* berbasis ARM (*Advanced RISC Machine*) yaitu *Raspberry Pi*. Penggunaan *Raspberry Pi* memberikan keuntungan karena dapat berfungsi sebagai komputer, namun memiliki daya yang kecil dan harganya relatif murah. *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 dihubungkan dengan internet dan *web server* dapat dikendalikan secara *remote* dari jarak jauh. *Raspberry Pi* mendapat koneksi internet melalui modem dan *wirelessrouter*. Awalnya *Raspberry Pi* dibuat pada tahun 2006 dan

dirilis pada 19 Februari 2012 dengan dua model: Model A dan Model B. Setelah terjual 3 juta unit pada Mei 2014, maka pada Juli 2014 dikeluarkan model B+ (Sachdeva dan Katchii, 2014). *Raspberry* merupakan mini komputer yang dapat digunakan untuk berbagai macam keperluan dan bekerja seperti halnya komputer. Gambar adalah perangkat *Raspberry Pi 3 Model B*.



Gambar 1. Raspberry Pi 3 Model B

2.3. Relay DC

Raspberry Pi dapat mengendalikan peralatan listrik dengan bantuan *Relay*. *Relay* merupakan salah satu komponen *output*/keluaran yang dapat digunakan pada peralatan elektronik. *Relay* berfungsi untuk menghubungkan ataupun memutuskan arus listrik dengan cara memberikan tegangan dan arus tertentu pada bagian koil. Untuk menggerakkan koil motor dapat menggunakan alat relay DC dengan menggunakan tegangan koil 12V DC, serta memerlukan arus sekitar 20-30mA. Salah satu contoh *relay* yang akan digunakan untuk penelitian ini adalah menggunakan *Relay 2 Channel* seperti terlihat pada Gambar 2. Relay 5V dengan 2 *channel* keluaran dapat digunakan sebagai saklar elektronik yang berfungsi untuk mengendalikan perangkat listrik dengan tegangan arus yang besar. *Relay* ini akan dihubungkan dengan *Raspberry Pi*.



Gambar 2. Modul Relay DC

2.4. GPS

GPS (*Global Positioning System*) merupakan sistem penentuan posisi lokasi dan navigasi satelit yang dikelola oleh Amerika Serikat. Sistem GPS didesain untuk memperoleh data posisi dan kecepatan dalam bentuk tiga dimensi serta informasi waktu, secara terus menerus di seluruh dunia, dan tidak bergantung pada waktu dan cuaca. Posisi suatu titik dinyatakan dengan koordinat. Selain GPS, juga ada sistem navigasi satelit GLONASS milik Rusia. Pada dasarnya GPS terdiri dari tiga segmen utama, yaitu segmen angkasa, segmen sistem control dan segmen pemakai. Alat penerima sinyal GPS (*GPS Receiver*) diperlukan untuk menangkap sinyal data dan memproses sinyal yang diperoleh satelit GPS untuk digunakan pada penentuan posisi lokasi, kecepatan, waktu maupun parameter turunan lainnya (Abidin, 2007).

Modul GPS dengan *Raspberry Pi* adalah sirkuit elektronik kecil yang memungkinkan papan *Raspberry Pi* untuk mendapatkan posisi, ketinggian, kecepatan, tanggal dan jam pada UTC (*Universal Time Coordinated*). Modul GPS ini menggunakan standar protokol NMEA untuk mentransmisikan posisi data melalui *port* serial. Modul GPS untuk penelitian ini dapat dilihat pada Gambar 3.



Gambar 3. Modul GPS

2.5. Firebase

Firebase merupakan *platform* untuk aplikasi *realtime*. Ketika data berubah, maka aplikasi dengan *firebase* akan meng-*update* secara langsung melalui setiap *device* (perangkat) baik *web* atau *mobile*. *Firebase* mempunyai *library* (pustaka) yang lengkap untuk sebagian besar platform *web* dan *mobile* dan dapat digabungkan dengan berbagai *framework* seperti *node*, *java*, *Java Script*, *AngularJS*, dan lain-lain. *Application Programming Interface* (API) untuk menyimpan dan sinkronisasi data akan disimpan sebagai bit-bit dalam bentuk *JSON* pada *cloud* dan akan disinkronisasi secara

realtime. Layanan pada *Firebase* meliputi autentikasi pengguna, pengaturan keamanan, dan *hosting*. Perubahan data pada satu *client* akan disinkronisasi pada semua *client* yang terdaftar ke data tersebut dalam hitungan mili detik. Kelebihan dari *Firebase* adalah dapat menerima data dari 1 juta perangkat secara bersamaan.

Penelitian ini akan menggabungkan *Firebase* dengan *AngularJS* untuk menyediakan *webrealtime*. *Firebase* dapat melakukan sinkronisasi data dengan data *local cache* pada perangkat yang ada sehingga aplikasi tetap responsif bahkan saat koneksi jaringan lambat atau tidak tersedia. *AngularJS* dengan *Firebase* diklaim menjamin 'no server code' sehingga memudahkan dalam pembuatan aplikasi (max, 2014).

Firebase digunakan untuk menyimpan data yang diterima. *Firebase* diperlukan untuk menyediakan layanan penyimpanan *cloud* yang memungkinkan untuk *middleware* dapat digabungkan dengan *interface* aplikasi dari pihak ketiga. *Firebase* memungkinkan pengembang untuk menciptakan aplikasi *mobile* dan *web* yang dapat digunakan untuk *generated* data secara *realtime* (Boman dkk., 2014).

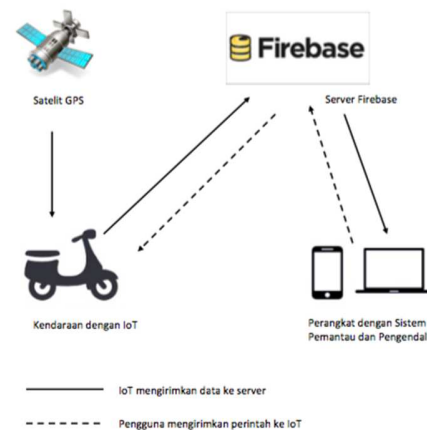
3. Metode Penelitian

3.1. Arsitektur Sistem

IoT (*Internet Of Things*) pada *Raspberry Pi* merupakan modul GPS untuk *Raspberry Pi* yang merupakan sebuah sirkuit elektronik kecil yang memungkinkan *board* (papan) *Raspberry Pi* untuk mendapatkan posisi lokasi. Modul ini menggunakan standar protokol NMEA (<http://www.nmea.org>) untuk mentransmisikan posisi data melalui *port* serial. Modul GPS menggunakan sinyal dari Satelit GPS mendeteksi lokasi/posisi kendaraan dan mengirimkan data posisi tersebut ke *web server* secara *realtime* melalui *Firebase*. Perangkat pemantau dapat melakukan pemantauan dengan mengambil data posisi melalui *web server*. Perangkat pengendali dapat melakukan pengendalian terhadap kendaraan dengan mengirimkan perintah ke kendaraan yang sudah dipasang dengan IoT.

Kendaraan akan melakukan akuisisi lokasi melalui perangkat GPS secara kontinyu dan *realtime*. Data *latitude* dan *longitude* yang didapatkan terlebih dahulu dibandingkan dengan data yang disimpan terakhir, apabila datanya sama

(artinya kendaraan tidak berubah posisi), maka data yang baru tersebut akan disimpan ke dalam database lokal dan di "broadcast" ke *server*. Model arsitektur sistem pemantau dan pengendali kendaraan dapat dilihat pada Gambar 4.

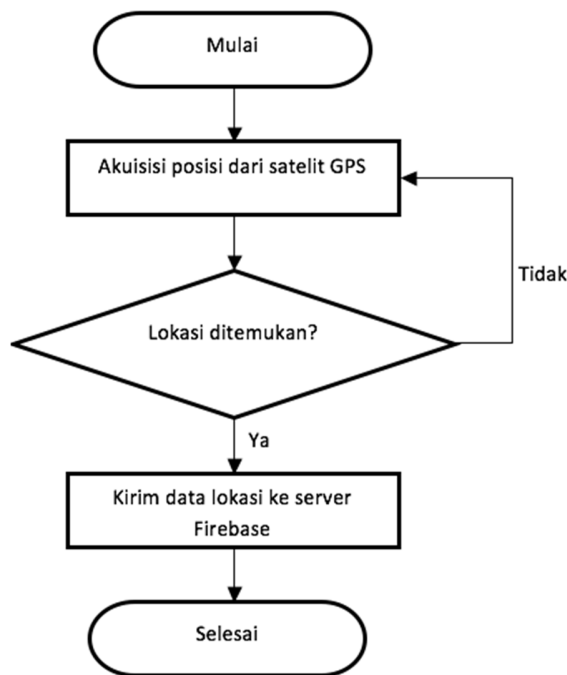


Gambar 4. Model Arsitektur Sistem

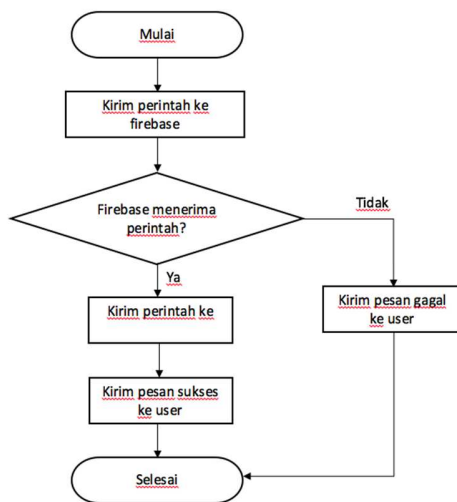
3.2. Rancangan Sistem

Cara kerja proses akuisisi posisi dari GPS ke kendaraan yaitu pertama perangkat IoT akan melakukan akuisisi posisi dari satelit GPS, jika lokasi ditemukan maka perangkat akan mengirimkan data lokasi tersebut ke *serverFirebase*. Sedangkan jika lokasi tidak ditemukan maka GPS akan mengulang kembali proses akuisisi posisi dari satelit GPS. Diagram alir proses akuisisi posisi GPS kendaraan dapat dilihat pada Gambar 5.

Diagram alir berikutnya adalah rancangan proses pengiriman perintah dari *user* ke kendaraan. Pertama *user* mengirim perintah ke *Firebase*. Selanjutnya *Firebase* akan menerima perintah, jika berhasil maka perintah akan dikirimkan ke alat IoT pada kendaraan dan akan mengirim pesan sukses ke *user*. Sedangkan jika *Firebase* gagal menerima perintah, maka akan dikirim pesan gagal ke *user*. Gambar 6 merupakan rancangan proses pengiriman perintah dari *user* ke kendaraan.



Gambar 5. Diagram alir proses akuisisi posisi dari GPS pada kendaraan



Gambar 6. Diagram alir proses pengiriman perintah dari user ke kendaraan

3.3. Implementasi Sistem dengan Angular Fire dan Google Maps API

Prototipe alat IoT dikembangkan menggunakan perangkat keras berupa mikrokontroler papan tunggal (*single board*) *Raspberry Pi 3 Model B* yang dilengkapi dengan

modul GPS Ublox Neo-6M untuk mendapatkan posisi koordinat kendaraan dan *Relay DC* untuk mematikan dan menghidupkan kendaraan. Alat IoT diintegrasikan sistem pemantau dan pengendali kendaraan jarak jauh berbasis *web* yang dikembangkan menggunakan perangkat lunak seperti *AngularFire*, *Firestore* dan *Google Maps API*. Untuk dapat mengirimkan data posisi GPS ke *server* maka dibuat kode koneksi untuk menghubungkan perangkat *Raspberry* ke *server Firestore*. Selain itu juga perlu dibuat kode program untuk koneksi ke *server*. Koneksi ke *server* dilakukan dengan menjalankan aplikasi *web* yang ada pada alat IoT secara otomatis (*auto-running*) dengan membuat penjadwalan melalui *cron*. Data posisi yang didapat dari alat IoT selanjutnya dikirimkan ke *server Firestore* secara *realtime*, dan akan ditampilkan dengan visualisasi peta dari *Google Maps* dengan icon posisi kendaraan.

```

var Firebase = require('firebase'),
    FirebaseHandler = require('./app/firebase-handler'),
    GPSParser = require('./app/gps-parser'),
    gps = new GPSParser('/dev/ttyUSB0', 9600),
    gpio = require('./app/rpi-gpio-dummy'),
    _ = require('lodash'),
    argv = process.argv,
    GPIOPins =
[7,11,12,13,15,16,18,22,29,31,32,33,35,36,37,38,40];
    
```

```

Firebase.initializeApp({
  serviceAccount: 'app/service_account.json', databaseURL:
'https://vehicle-iot.firebaseio.com'});
    
```

```

var dbRef = Firebase.database().ref();
var powerPin = (argv[2] === '-pin' || argv[2] === '-p')
&& !isNaN(parseInt(argv[3])) &&
_.includes(GPIOPins, parseInt(argv[3])) ? argv[3] : 11;
var Handler = new FirebaseHandler(dbRef, gpio,
powerPin);
Handler.watchCommand();
Handler.ready() => {
  gps.on('gps-data', data => {
    Handler.handleGPS(data);
  });
});
    
```

4. Hasil dan Pembahasan

Hasil percobaan penelitian berupa *prototype* alat IoT untuk pemantau dan pengendali kendaraan dan sistem pengendali dan pemantau kendaraan berbasis *web realtime* dengan *AngularFire* (*AngularJS* dan *Firestore*).

4.1. Prototype Alat IoT

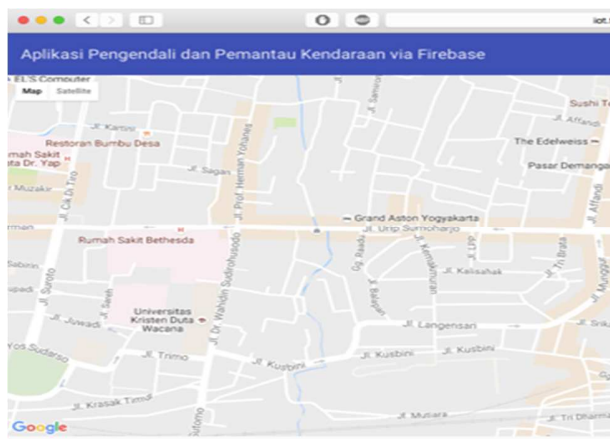
Prototype alat IoT untuk pemantau kendaraan dibuat dengan *Raspberry Pi* model B dan modul GPS. Alat IoT tersebut juga telah dilengkapi dengan sistem operasi *Raspbian* dan aplikasi *web* yang dibangun dengan *nodejs*. Aplikasi *web* akan dijalankan setiap kali alat IoT dihidupkan. Aplikasi *web* yang telah ditanamkan pada alat tersebut berfungsi untuk mendapatkan data posisi kendaraan dan selanjutnya dikirimkan secara *realtime* ke *server Firebase*.

Prototype alat IoT untuk pengendali kendaraan dibuat dengan *Raspberry Pi* model B dan dihubungkan dengan *Relay DC* untuk menghidupkan dan mematikan kendaraan. *prototype* alat IoT untuk kendali kendaraan dapat dilihat pada Gambar 7.



Gambar 7. Prototype Alat IoT

4.2. Aplikasi *WebRealtime* sebagai Sistem Pemantau dan Pengendali Kendaraan



Gambar 8. Aplikasi Pengendali dan Pemantau Kendaraan Berbasis Web

Alat IoT saat dihidupkan akan tersinkronisasi dengan *Firebase Server*. Berdasarkan data yang terkirim ke *server Firebase* tersebut, pergerakan kendaraan dapat dipantau secara *realtime* dari jarak jauh melalui web. Aplikasi berbasis *web* untuk pemantauan dan pengendalian yang dibuat dilengkapi dengan *login* karena memerlukan keamanan data dan tidak sembarang untuk dapat dilihat dan digunakan orang lain.

Data posisi pemantauan yang disimpan di *server* adalah data posisi terakhir yang terkirim ke *server*. Aplikasi web untuk pemantauan dilengkapi dengan *Google Maps* dan *icon* posisi pada peta, sedangkan aplikasi pengendali jarak jauh adalah berupa tombol *on/off* untuk menghidupkan dan mematikan mesin kendaraan bermotor dari jarak jauh. Aplikasi pengendali dapat digunakan untuk mematikan dan mengunci kendaraan, saat terjadi pencurian. Setelah login maka tampilan sistem pemantau berbasis *web* berupa data visualisasi *Google Maps* dan sistem pengendali untuk *on/off* kendaraan dari web seperti terlihat pada Gambar

4. Penutup

Sistem pemantau dan pengendali kendaraan bermotor jarak jauh menggunakan teknologi *web realtime* telah berhasil dikembangkan dengan menggunakan teknologi *AngularFire (AngularJS dan Firebase)*. Selanjutnya sistem dioperasikan untuk melakukan pemantauan terhadap kendaraan yang telah dipasang dengan perangkat IoT yang terdiri dari sebuah *Raspberry Pi 3 Model B*, modul GPS *Ublox Neo-6M, Relay DC 2 Channel*, modem GSM dan didalamnya telah dilengkapi dengan program koneksi ke *server*.

Selanjutnya dilakukan pengujian fungsionalitas pemantauan kendaraan dengan menjalankan kendaraan yang telah terpasang alat IoT dan melihat hasil pemantauan data pergerakan posisi kendaraan melalui aplikasi web sistem pemantau yang telah dilengkapi dengan *Google Maps*. Data yang tersimpan pada *serverFirebase* adalah data posisi terakhir kendaraan yang dikirimkan ke *serverFirebase*. Hasil pengamatan menunjukkan bahwaterjadi perubahan posisi kendaraan yang terlihat dari pergerakan icon posisi pada peta.

Daftar Pustaka

- Abidin, H.Z.(2007)*Penentuan Posisi dengan GPS dan Aplikasinya*.Jakarta: PT Pradnya Paramita.
- Al-Khedher, M.A. (2011). Hybrid GPS-GSM Localization of Automobile Tracking System, *International Journal of Computer Science & Information Technology (IJCSIT), arXiv*, Vol. 3, No. 6.
- Boman, J., Taylor, J., Ngu, A.H.(2014). Flexible IoT Middleware for Integration of Things And Application, *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 Internatinal Conference on IEEE*, 481-488.
- BPS, (2013).*Statistik Kriminal 2013*.Diakses pada 20 Maret 2015dariWorld Wide Web:<http://www.bps.go.id>
- Darmaliputra, A., Hermawan, H.(2014). Pembuatan Web Server Berbasis Raspberry Pi Untuk Kontrol Lampu dan AC, *Calyptra*, Vol. 3, No. 1.
- Hanifah, R., Isnanto, R. R., Christyono, Y.(2010). Simulasi Sistem Informasi Geografis (SIG) Pemantau Posisi Kendaraan Via SMS Gateway, *TRANSMISI*, Vol. 12, No. 2, 45-59.
- Huang, Z., Huang, D., Xu, Z., Xu., Z.(2011). GPS Vehicle Positioning Monitoring System Integrated with CORS and Mobile GIS, *Procedia Environmental Sciences, SciVerse ScienceDirect*, Vo. 10, 2498-2504.
- Junus, M.(2012). Sistem Pelacakan Posisi Kendaraan dengan Teknologi GPS & GPRS Berbasis Web, *Jurnal ELTEK*, Vol. 10, No. 2, 58-67.
- Khan, A., Mishra, R.(2012). GPS-GSM Based Tracking System, *International Journal of Engineering Trends and Technology*, Vol. 3, No. 2, 161-164.
- Lee, S., Tewolde, G.,Kwon, J.(2014). Design and Implementation of Vehicle Tracking System using GPS/GSM/GPRS Technology and Smartphone Application, *Internet of Things (WF-IoT), IEEE*, 353-358.
- Maurya, K., Singh, M., Jain, N.(2012), Real Time Vehicle Tracking System using GSM and GPS Technology- An Anti-theft Tracking System, *International Journal of Electronics and Computer Science Engineering*, 1103-1107.
- Max, D., (2014).*Building Real Time Apps With Firebase and AngularJS*.Diakses pada 19 Maret 2015 dariWorld Wide Web:<http://davemax.com>.
- Muchlisin, Y. D., 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, No. 2, 76-84.
- Oriza, A., (2014).*Firebase Membantu Kita Membuat Aplikasi Realtime*.Diakses pada 21 Maret 2015dariWorld Wide Web:<http://www.codepolitan.com/firebase-membantu-kita-membuat-aplikasi-realtime/>.
- Purnama, B. E., (2009). Pemanfaatan Global Positioning System Untuk Pelacakan Objek Bergerak, *Speed-Sentra Penelitian Engineering dan Edukasi*, Vol. 2, No. 2, 6-12.
- Ramani, R., Valarmathy, S., SuthanthiraVanitha, N., Selvaraju, S., Thirupathi, M., Thangam, R., (2013). Vehicle Tracking and Locking System

Based on GSM and GPS, *International Journal of Intelligent Systems and Applications*, Vol. 5, No. 9, 86-93.

Rifai, A.(2013). Sistem Informasi Pemantauan Posisi Kendaraan Dinas Unsri Menggunakan Teknologi GPS, *Jurnal Sistem Informasi (JSI)*, Vol. 5, No. 2, 603-610.

Sachdeva, P.,Katchii, S. (2014). A Review Paper on Raspberry Pi, *Inpressco*, Vol. 4, No. 6, 3818-3819.

Upton, E., Halfacree, G., (2012).*Raspberry Pi User Guide*, Willey, West Sussex, England.

Verma, P., Bhatia, J. S., (2013). Design and Development of GPS – GSM Based Tracking

System with Google Map Based Monitoring, *International Journal of Computer Science, Engineering and Applications (IJCSEA)*, Vol. 3, No. 3,33-40.

Biodata Penulis

Erma Susanti, memperoleh gelar S1 di IST AKPRIND Yogyakarta. Memperoleh gelar S2 di Universitas Gadjah Mada. Saat ini menjadi pengajar di IST AKPRIND Yogyakarta.

Joko Triyono, memperoleh gelar S1 di IST AKPRIND Yogyakarta. Memperoleh gelar S2 di Universitas Gadjah Mada. Saat ini menjadi pengajar di IST AKPRIND Yogyakarta.

BERITA ACARA PELAKSANAAN HASIL SEMINAR SESI PARALEL KNASTIK 2016

Judul : Pengembangan Sistem Pemantau Dan Pengendali Kendaraan Menggunakan Raspberry Pi dan Firebase

Pemakalah : Erma Susanti, Joko Triyono

Moderator : Laurentius Kuncoro Probo Saputra, S.T., M.Eng.

Notulis : Rama

Peserta : 12 orang di ruang : B.3.3

Tanya Jawab :

- Kelemahan dari IoT adalah database servernya, karena masih menumpang di server orang lain. Untuk sistem ini, database server milik sendiri atau orang lain?
- Untuk pemantauan dan pengendalian, saat dikendalikan dari perangkat (smartphone) dikirimkan ke server, mengirimnya melalui apa?
- Pertimbangan mengapa memilih raspberry dalam sistem ini?

Masukan Seminar :

Pengembangan IoT untuk pemantauan dan pengendalian kendaraan ini masih memiliki kelemahan, yaitu masih menggunakan server orang lain sebagai database server.

Yogyakarta, 19 November 2016

Moderator Kelas

Laurentius Kuncoro P.S., S.T., M.Eng.

Penyaji Makalah

Erma Susanti

