Terbit *online* pada laman web jurnal : http://teknosi.fti.unand.ac.id/

Jurnal Nasional Teknologi dan Sistem Informasi

| ISSN (Print) 2460-3465 |ISSN (Online) 2476-8812|

Studi Kasus

Sistem *Monitoring* Kendaraan Secara *Real Time* Berbasis Android menggunakan Teknologi CouchDB di PT. Pura Barutama

Ramos Somya [a]

[a]*Program Studi S1 Teknik Informatika Universitas Kristen Satya Wacana, Jl. Diponegoro 52-60 Salatiga, Jawa Tengah, Indonesia*

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima Redaksi: 05 Februari 2018
Revisi Akhir: 28 Juli 2018
Diterbitkan *Online*: 31 Agustus 2018

KATA KUNCI

Sistem Monitoring Kendaraan,
CouchDB,
GPS

KORESPONDENSI

Telepon: +62 85640326685
E-mail: mailto:ramos.somya@staff.uksw.edu

A B S T R A C T

PT. Pura Barutama is a company with many production areas. Products that have been produced will be delivered to some destinations using company's vehicles. Those vehicles are managed by Vehicle Unit of PT. Pura Barutama. Based on the research, there is a problem in the term of vehicle tracking and monitoring. The delay of product delivery is often occur for no apparent reason from the drivers. Previously, Vehicle Unit already has a GPS Tracker installed on each vehicle, but this GPS Tracker can only be triggered using SMS to send the location of vehichles. This method causes the lack of tracking and monitoring information obtained by the company. This research developed a new model of tracking and monitoring system based on Android platform. CouchDB technology is also used to provide a real time tracking and monitoring system. The test result shows that this new system has advantages compared with the old tracking system and can be used to replace the old system.

1. PENDAHULUAN

PT. Pura Barutama Kudus memiliki banyak bidang produksi [1]. Barang-barang hasil produksi kemudian akan dikirimkan ke masing-masing customer. Unit Kendaraan adalah unit yang bertanggung jawab untuk semua pengiriman barang hasil produksi. Kendaraan yang dipakai terdiri dari kendaraan milik Unit Kendaraan sendiri dan kendaraan sewa dari pihak ketiga.

Selama ini proses pengiriman oleh Unit Kendaraan dimulai dari *request* dari masing-masing unit produksi. Saat satu unit produksi melakukan *request* pengiriman, maka unit kendaraan akan mengeluarkan sebuah Surat Pengiriman (SP) yang berisi informasi untuk pengiriman barang, di antaranya: rincian barang kiriman, waktu pengiriman, alamat tujuan pengiriman, dan nomor plat kendaraan yang dipakai.Surat pengiriman akan diberikan kepada *driver* kendaraan, sendangkan unit produksi akan diberikan nomor dari surat pengiriman yang merupakan nomor SP barang milik unit produksi yang bersangkutan.Kemudian setelah kapasitas dari satu kendaraan sudah terpenuhi (bisa terdiri dari beberapa SP) maka kendaraan akan berangkat dengan membawa satu nomor *Delivery Order* (DO). Saat barang sudah sampai ke *customer*, maka *customer* akan memberikan tanda tangan pada surat pengiriman (SP) sebagai bukti/verifikasi bahwa barang sudah sampai di tujuan. SP yang sudah diverifikasi kemudian akan dipakai oleh unit produksi untuk membuat faktur tagihan.

Selama proses pengiriman, unit produksi dapat memeriksa sampai mana proses pengiriman barang dengan bertanya kepada unit kendaraan menggunakan nomor SP. Banyaknya barang yang dikirimkan menyebabkan proses pengecekan barang kiriman dengan bertanya ke unit kendaraan menjadi lebih sulit. Keadaan ini bertambah sulit karena sistem GPS *tracker* yang dipakai adalah GPS pasif yang memerlukan *trigger* dengan cara mengirimkan SMS ke perangkat GPS untuk mendapatkan data lokasi kendaraan. Berdasarkan masalah yang telah dijelaskan, maka latar belakang dari penelitian ini adalah membuat sistem *monitoring* kendaraan secara *real time* yang dapat diakses langsung oleh masing-masing unit produksi tanpa meminta informasi kepada unit kendaraan. Teknologi *real time* memungkinkan untuk membuat suatu sistem *monitoring* dengan jeda waktu yang minimal. Informasi lokasi kendaraan akan langsung diketahui di saat yang bersamaan dengan bergeraknya kendaraan. Teknologi ini kemudian dapat digabungkan dengan perangkat Android sehingga dapat diakses secara *mobile* oleh siapa saja, di mana saja, dan kapan saja. Maka solusi yang diberikan untuk masalah yang ada adalah dengan membuat sistem

https://doi.org/10.25077/ TEKNOSI.v4i2.2018.053-060

monitoring kendaraan berbasis Android yang berfungsi untuk memantau lokasi kendaraan secara *real time*.

Berdasarkan latar belakang yang telah dijelaskan, didapatkan rumusan masalah yaitu bagaimana membuat sistem *monitoring* yang dapat memberikan laporan data lokasi kendaraan secara *real time* dan dapat diakses dengan cepat dan akurat saat dibutuhkan. Penelitian ini terfokus pada pembuatan *prototype* sistem *tracking*, sehingga penelitian ini memiliki batasan-batasan dalam pembahasannya. Batasan tersebut di antaranya, sistem *tracking* akan dibuat pada perangkat Android dengan GPS terintegrasi, sistem dibangun menggunakan Android *native programming* (Java) dengan *real time database* Apache CouchDB, sistem yang dibuat memerlukan versi minimal API Android 19, sistem *tracking* menggunakan proses bisnis pengiriman barang yang sudah ada di Unit Kendaraan PT. Pura Barutama.

2. TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Terdapat beberapa penelitian yang telah dilakukan sebelumnya mengenai sistem tracking menggunakan perangkat Android, salah satunya adalah penelitian yang dilakukan oleh Wahyu Kusuma dan Tity Septiani dengan judul "Aplikasi *Friend Tracker* berbasis Android *Smartphone* menggunakan GPS *Tracking*". Penelitian ini membahas tentang implementasi GPS *Tracking* yang digunakan dalam aplikasi pencarian lokasi teman. Hasil akhir dari penelitian ini yaitu sebuah sistem *tracking* menggunakan perangkat Android yang berfungsi untuk mengetahui posisi teman yang telah terdaftar dalam *friend list* [2].

Penelitian yang kedua dilakukan oleh Muhammad Dzani Alfikridengan judul "Aplikasi *Auto-Reporting Position Tracking* Berbasis Android Untuk Mengetahui Posisi *Device* Sebagai Sarana *Monitoring* Posisi Karyawan di PT Telkom Indonesia Kota Malang". Penelitian ini menghasilkan satu aplikasi berbasis Android yang digunakan untuk memantau lokasi setiap karyawan PT. Telkom Indonesia di kota Malang dengan tujuan meningkatkan produktifitas kerja para karyawan [3].

Perbandingan penelitian ini dengan dua penelitian sebelumnya adalah penggunaan teknologi *real time* pada sistem tracking. Aplikasi pada penelitian pertama hanya memberikan data lokasi terakhir saat pengguna melakukan *sign-up* ataupun melakukan *update* lokasi dalam profilnya. Sedangkan sistem *tracking* pada penelitian kedua menggunakan interval waktu dalam mengirimkan data lokasi setiap karyawan ke *database* SQL. Pelaporan data lokasi secara *real time* dalam penelitian ini dimungkinkan dengan menggunakan *real time database*.

2.1 *Real-Time Database System*

Real time database adalah *database* yang menggunakan *real time processing* untuk menangani data yang nilainya terus berubah, tidak seperti *database* tradisional yang menyimpan data tetap/*persistent* [4]. Real time *database* juga disebut dengan NOSqldatabase, di mana SQL *database* adalah sebutan untuk *relational database*. NOSql (Not Only Sql) memiliki kelebihan yang tidak dimiliki oleh SQL *database*. NOSqldatabase tidak terikat dengan skema (*schema-free*), mendukung replikasi dengan mudah, memiliki API yang lebih sederhana, dan mampu menangani data yang sangat besar (big data) [5]. CouchDB adalah salah satu contoh dari *real time database* system (RTDBS) yang dikembangkan oleh Apache. CouchDB tidak menggunakan tabel (baris dan kolom) untuk menyimpan data, melainkan menggunakan struktur *file* JSON (*key-value pair*). Sebuah *record* dalam CouchDB disebut juga satu dokumen (*document*). Satu dokumen merupakan satu *file* berformat JSON yang memiliki ID (*primary key*), REV (*revision* id), dan isi data yang disimpan. ID digunakan sebagai *unique identifier* yang membedakan dokumen satu dengan yang lainnya. Sedangkan *revision* id digunakan untuk menjaga konsistensi data [6].

2.2 CouchDB

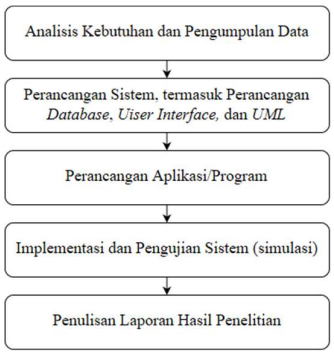
CouchDB memiliki API berbasis HTTP REST yang memudahkan *client* untuk mengakses *database*. Maka dalam mengakses CouchDB, *client* dapat menggunakan perintah seperti PUT, GET, dan DELETE yang ada pada HTTP [7]. LightCouch adalah *library* Java yang berfungsi sebagai antarmuka (*interface*) untuk berkomunikasi dengan CouchDB menggunakan HTTP. API dalam LightCouch diakses dengan membuat koneksi dari *client* ke *server* CouchDB. Client adalah objek utama yang melakukan setiap request ke *database*, dalam konteks ini adalah *request* dokument ataupun *view*. API LightCouch berfokus dalam CRUD dokumen, *view*, *attachment*, *design documents*, *changes notification*, dan operasi *database* CouchDB yang spesifik seperti *compaction* dan *replication* [8].

2.3 Google Maps

Google Maps adalah salah satu layanan yang disediakan oleh Google. Google Maps memiliki keunggulan dalam hal kelengkapan dan detail peta. Hal ini disebabkan karena Google Maps mengijinkan pengguna untuk berkontribusi dalam pengembangan peta [9]. Google Maps juga dapat diakses dari berbagai *platform*, salah satunya dari *platform* Android menggunakan Google Maps Android API [10]. API ini memungkinkan pengembang untuk mengintegrasikan Google Maps pada sistem yang dibuat.

3. METODE DAN PERANCANGAN SISTEM

Penelitian ini dilakukan dengan 5 tahapan penelitian yaitu: (1) analisis kebutuhan dan pengumpulan data; (2) perancangan sistem; (3) perancangan aplikasi; (4) implementasi dan pengujian sistem dengan cara simulasi; dan (5) penulisan laporan hasil penelitian [11]. Tahapan yang dilakukan dalam penelitian ini dapat dilihat pada Gambar 1.

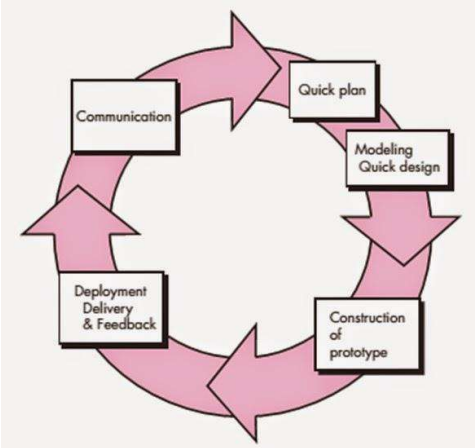


Gambar 1. Tahapan Penelitian

https://doi.org/10.25077/ TEKNOSI.v4i2.2018.053-060

Berdasarkan bagan pada Gambar 1 dapat dijelaskan bahwa tahapan penelitian yang dilakukan adalah: Tahap pertama analisis kebutuhan dan pengumpulan data, di mana pihak pengembang mencari informasi mengenai kebutuhan dari pengguna yang berhubungan dengan pembuatan sistem. Pencarian informasi dilakukan dengan cara wawancara secara langsung kepada pihak MIS (*Management and Information System*) PT. Pura Barutama. Berdasarkan hasil dari wawancara yang dilakukan didapatkan informasi bahwapihak unit kendaraan sudah memiliki sistem *monitoring* yang dibuat menggunakan GPS *tracker* berbasis SMS. Sistem monitoring kendaraan ini dianggap tidak praktis oleh unit produksi. Hal ini disebabkan karena unit produksi harus bertanya kepada unit kendaraan untuk meminta informasi lokasi kendaraan. Pihak MIS kemudian mencari cara untuk membantu menyelesaikan masalah tersebut. Pada tahap kedua, ketiga dan keempat dilakukan perancangan sistem monitoring kendaraanmenggunakan metode pengembangan sistem *Prototype*. Tahap kelima dilakukan penulisan laporan ilmiah dan artikel ilmiah.

Perancangan dan pembuatan aplikasi dalam penelitian ini dilakukan menggunakan metode *Prototype*. Metode ini menggunakan sebuah siklus dalam menghasilkan suatu produk. Produk yang dihasilkan dari metode prototype akan diuji, jika terdapat kekurangan maka akan dilakukan siklus *prototype* kembali untuk memperbaiki produk sebelumnya.

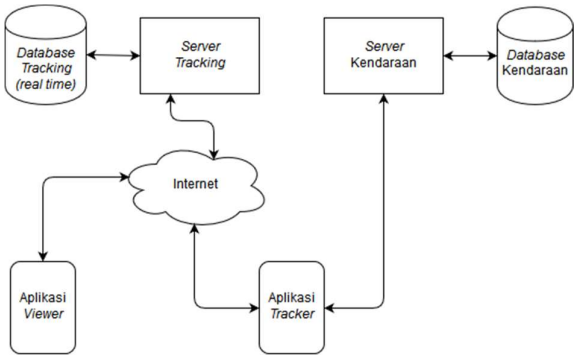


Gambar 2. Metode *Prototype* [12]

Seperti yang digambarkan pada Gambar 2, metode *Prototype* dimulai dengan communication yaitu analisis kebutuhan sistem yang akan dibuat. Analisa ini dapat diperoleh dari pengumpulan data, baik bersumber dari literatur maupun penelitian sebelumnya.

Langkah selanjutnya adalah *planning* dan *design*, di mana *software* akan dibuat rancangannya secara garis besar. Perancangan / *design software* ini meliputi bagaimana program akan berjalan, meliputi semua aspek *software* yang diketahui. Pembuatan desain program dapat dilakukan dengan atau tanpa *software* bantuan (seperti *software* UML). Pada tahap analisis sebelumnya telah diperoleh *system requirement* yang didapat dari pengumpulan data. Tentunya pada tahap perancangan/design, requirement pada tahap sebelumnya harus dipertimbangkan dengan matang untuk fitur apa saja yang memungkinkan untuk dibuat. Setelah dilakukan *planning* dan *design*, maka selanjutnya sistem akan masuk dalam tahap *construction* atau pembuatan. Desain akan diubah menjadi bahasa pemrograman yang dapat dijalankan oleh platform tujuan. *Programmer* akan menerjemahkan kebutuhan sistem yang didapat dari tahap-tahap sebelumnya. Pembuatan sistem tracking dalam penelitian ini menggunakan bahasa Java yang merupakan bahasa *native* dalam pemrograman aplikasi berbasis Android menggunakan Android Studio. Setelah sistem selesai diterjemahkan ke dalam bahasa pemrograman, semua fungsi sistem harus diuji coba agar dapat dinilai apakah sistem sudah siap untuk dipakai atau belum. Pengujian sistem yang dibuat dilakukan dengan *blackbox testing* [13]. Saat sistem sudah lulus tahap testing maka sistem sudah siap untuk masuk ke tahap *deployment*, namun jika masih terdapat kekurangan makaakan dikembangkan lebih lanjut dengan mengulangi siklus *prototyping*.

Tahap terakhir dari penelitian adalah analisis dan pengambilan kesimpulan. Analisis dilakukan untuk menilai apakah algoritma dan metode yang dipakai dalam pengenalan *real time tracking* sudah baik atau sebaliknya, dan faktor-faktor apa saja yang mempengaruhi performa *software*.



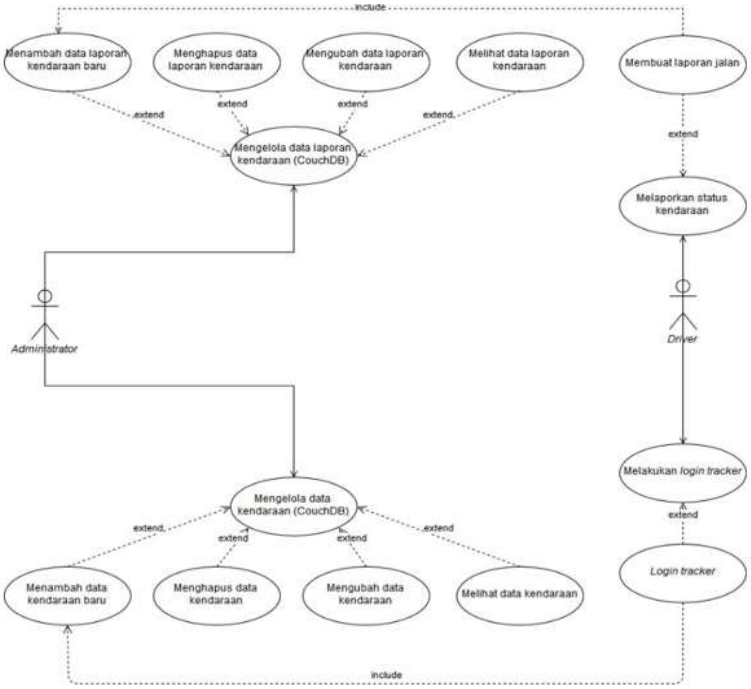
Gambar 3. Desain Arsitektur Sistem

Gambar 3 menjelaskan desain arsitektur sistem yang dibuat. Sistem *monitoring* dirancang untuk menggunakan 1 *server* dengan IP publik yang digunakan sebagai serverreal time database CouchDB, 1 *server offline* milik unit kendaraan, dan perangkat Android yang terpasang dalam setiap kendaraan. Perangkat Android dalam kendaraan digunakan untuk mengirimkan data lokasi kendaraan ke *server real time database* saat kendaraan bergerak. Kemudian data yang ada dalam server dapat diakses oleh pengguna aplikasi *monitoring* (dalam hal ini adalah masing-masing unit produksi). Perancangan sistem dalam penelitian ini dibuat dalam bentuk diagram UML. Diagram UML meliputi *usecase diagram* dan *class diagram*.

Gambar 4 merupakan *use case* yang menjelaskan bagaimana sistem *monitoring* kendaraan dibuat. Sistem memiliki 2 aktor, yaitu administrator yang memiliki hak akses penuh ke dalam sistem, termasuk mengelola data kendaraan, mengelola laporan kendaraan, dan mengelola data laporan akhir (*final report*) setiap kendaraan. Aktor yang kedua adalahdriveryang memiliki hak akses untuk melakukan *login* ke dalam sistem *tracking* dan melaporkan status kendaraan (termasuk melakukan verifikasi pengiriman barang).

Evaluasi *Prototype* sistem *monitoring* kendaraan dievaluasi dengan dua cara. Cara yang pertama adalah dengan melakukan uji lapangan secara langsung. Uji lapangan melibatkan staff MIS

dan pengembang sistem untuk menguji sistem tracking dengan cara membawa aplikasi *tracker* dan melakukan kegiatan sehari-hari. Evaluasi ini dilakukan dengan maksud menguji stabilitas algoritma tracking pada aplikasi *tracker*. Hal ini diperlukan mengingat aplikasi *tracker* harus berjalan selama berhari-hari tanpa mengalami *error*/*crash* dengan mempertimbangkan semua keadaan seperti tidak adanya koneksi internet di beberapa tempat. Evaluasi kedua dilakukan dengan cara simulasi proses bisnis yang dilakukan di unit MIS untuk mencocokkan proses bisnis yang diimplementasi dalam sistem aplikasi trackingdengan proses bisnis yang sudah ada sebelumnya.



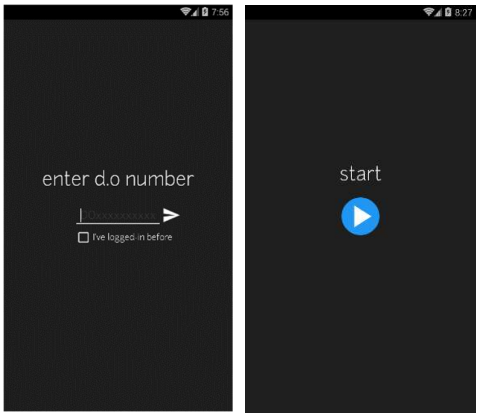
Gambar 4. *Use Case Diagram Prorotype* Sistem *Monitoring* Kendaraan

Evaluasi *Prototype* sistem *monitoring* kendaraan dievaluasi dengan dua cara. Cara yang pertama adalah dengan melakukan uji lapangan secara langsung. Uji lapangan melibatkan staff MIS dan pengembang sistem untuk menguji sistem tracking dengan cara membawa aplikasi *tracker* dan melakukan kegiatan sehari-hari. Evaluasi ini dilakukan dengan maksud menguji stabilitas algoritma tracking pada aplikasi *tracker*. Hal ini diperlukan mengingat aplikasi *tracker* harus berjalan selama berhari-hari tanpa mengalami *error*/*crash* dengan mempertimbangkan semua keadaan seperti tidak adanya koneksi internet di beberapa tempat. Evaluasi kedua dilakukan dengan cara simulasi proses bisnis yang dilakukan di unit MIS untuk mencocokkan proses bisnis yang diimplementasi dalam sistem aplikasi trackingdengan proses bisnis yang sudah ada sebelumnya.

4. HASIL DAN PEMBAHASAN

Sistem *monitoring* kendaraan dalam penelitian ini terdiri dari 2 aplikasi utama berbasis Android: (1) Aplikasi sisi *driver* (*tracker*) dan (2) Aplikasi sisi administrator (*viewer*). Aplikasi *tracker* digunakan oleh *driver* yang berfungsi sebagai pengganti GPS *tracker* berbasis SMS pada sistem yang terdahulu dengan tambahan fungsi pelaporan (termasuk pelaporan verifikasi). Sedangkan aplikasi *viewer* adalah aplikasi yang digunakan oleh administrator untuk melihat posisi setiap kendaraan yang sudah terpasang aplikasi *tracker* melalui API Google Maps, serta laporan *live* dari setiap *driver* kendaraan.

Garis besar proses bisnis dengan sistem *monitoring* yang baru adalah sebagai berikut: (1) *Driver* menggunakan aplikasi *tracker* untuk *login* dan memulai pengiriman barang. (2) Bila terjadi kendala di perjalanan, *driver* dapat melaporkan kendala melalui aplikasi *tracker*. (3) Ketika barang sudah sampai tujuan, *driver* melakukan verifikasi dengan *scan* QR *code* yang dimiliki *customer*. (4) Informasi letak kendaraan, laporan *driver*, dan status verifikasi barang dapat dipantau oleh masing-masing unit produksi menggunakan aplikasi *viewer*. Aplikasi *viewer*/sisi administrator juga dapat melakukan *override* pada proses verifikasi apabila terdapat masalah pada sistem verifikasi dengan QR code oleh aplikasi *tracker*/sisi *driver*.



Gambar 5. Halaman *Login* & *StartAppTracker*

Sistem *monitoring*/*tracking* kendaraan dimulai saat *driver* melakukan *login* melalui aplikasi *tracker* menggunakan nomor *delivery order* yang telah diberikan sebelumnya. Untuk melakukan *login*, *smartphone driver* harus terkoneksi

https://doi.org/10.25077/ TEKNOSI.v4i2.2018.053-060

ke jaringan lokal PT. Pura Barutama untuk mencocokkan nomor *delivery order* yang dimiliki *driver*, dengan nomor *delivery order* yang ada di *database* lokal. Pada halaman login juga disediakan *checkbox* "*I've logged in before*", yang berfungsi untuk melakukan login di luar jaringan lokal Pura dengan syarat sudah pernah melakukan *login* di jaringan lokal. Jika masukan *driver* valid, maka *driver* akan dibawa pada halaman start untuk memulai menjalankan sistem *tracking*. Antarmuka halaman *login* dan *start tracking* dapat dilihat pada Gambar 5.

Service sistem *monitoring* kendaraan akan dijalankan dalam perangkat Android setelah *driver* menekan tombol *start* pada layar aplikasi. Mekanisme start *service* ini melalui beberapa proses: (1) melakukan *knocking* ke *server* untuk mendaftarkan IP *address* perangkat Android, (2) jika IP *address* perangkat sudah terdaftar maka perangkat diijinkan untuk mengakses *port database* CouchDB, (3) aplikasi *tracker* akan membuat koneksi ke *database* CouchDB, (4) dan yang terakhir *service tracking* akan dijalankan untuk melaporkan setiap perubahan data lokasi kendaraan yang didapat dari GPS perangkat Android ke *database* CouchDB.

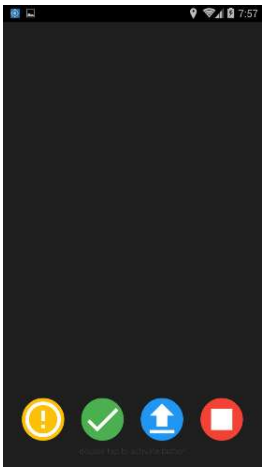
Mekanisme *knocking* perlu dilakukan untuk mencegah akses dari pihak yang tidak bertanggung jawab ke *database* CouchDB. *Knocking* dilakukan dengan mengakses *port* tertentu pada *host server* (yang bersifat rahasia) untuk sekadar mendaftarkan IP *address* agar mendapat akses untuk mengakses *port* yang lainnya. *Request knocking* dilakukan sebelum aplikasi dapat mengakses *port* CouchDB.

Jika *knocking* berhasil, maka aplikasi akan mencoba untuk membuat koneksi ke *database* CouchDB. Koneksi ke dalam CouchDB tidak sama dengan koneksi ke *database* SQL. Koneksi ke *database* CouchDB, atau sebagian besar *real time database*, bersifat *persistent* atau tetap. Koneksi akan terus terhubung bukan hanya saat melakukan *query* saja seperti pada *database* SQL.

Setelah koneksi dibuat, maka aplikasi akan mendaftarkan data kendaraan yang melakukan *login* ke *database* CouchDB. Data yang dibuat merupakan data kendaraan dan data laporan kendaran. Data dalam CouchDB akan disimpan dalam bentuk JSON.

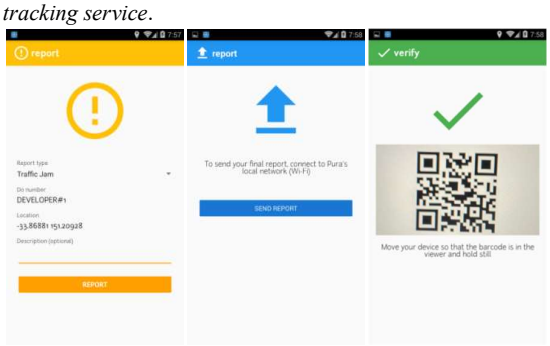
Selanjutnya aplikasi *tracker* akan mengirimkan data *latitude* dan *longitude* untuk meng*update record* yang ada pada *database* CouchDB untuk melakukan *tracking*. *Update* data lokasi kendaraan dilakukan dengan cara: (1) aplikasi mengambil nilai *latitude* dan *longitude* dengan modul GPS yang terintegrasi pada perangkat Android, (2) meng*update record* pada *database* dengan menyertakan *revision* id "_rev" dari *record* terakhir, (3) jika *update* maka lakukan proses *reconnect* ke *database* CouchDB. Setiap kali suatu *record* pada CouchDB dilakukan proses *update* maka atribut _rev akan berubah, sedangkan untuk melakukan *update* dibutuhkan nilai _rev yang terakhir. Hal ini disebabkan karena adanya proteksi dari CouchDB yang berfungsi untuk menjaga validitas nilai (hanya *record* terakhir/yang paling baru yang dapat di*update*).

Pengambilan data lokasi kendaran menggunakan classLocationListener yang tersedia dalam Android SDK. Untuk mencapai performa aplikasi yang baik sekaligus penggunaan baterai yang hemat, maka perubahan lokasi hanya akan dicatat ketika perangkat telah bergerak sejauh minimal 200 meter dan 15 detik. Ini menyebabkan perangkat tidak akan mengirimkan posisinya terus-menerus jika hanya berdiam dalam satu tempat yang sama.



Gambar 6. Halaman Utama Aplikasi *Tracker*

Fitur tambahan yang ada pada aplikasi *tracker* adalah adanya fitur pelaporan untuk *driver*. Fitur ini dibuat untuk menanggulangi kurangnya komunikasi *driver* kepada unit yang menyebabkan ketidakjelasan jika terjadi *delay* dalam pengiriman barang. Fitur ini dapat diakses melalui halaman utama aplikasi tracker yang akanmuncul setelah *service tracking* sudah berjalan. Terdapat 4 menu dalam halaman utama aplikasi *tracker* yang dapat dilihat pada Gambar 6, menu dari kiri ke kanan adalah: (1) menu *report*, (2) menu *verification*, (3) menu *final report*, dan (4) menu *stop tracking service*.



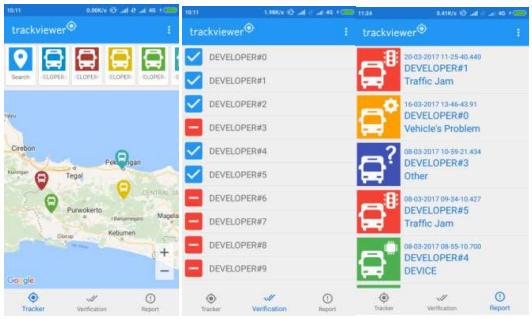
Gambar 7. Halaman *Report, Verification*, dan *Final Report*

Menu *report* (Gambar 7) digunakan untuk melaporkan setiap kejadian yang ada dalam perjalanan. Fitur ini akan menyimpan setiap laporan *driver* ke dalam *database* SQLiteinternal Android. Data laporan yang disimpan adalah: jenis laporan, nomor *delivery order*, waktu kejadian, lokasi, dan keterangan. Data yang disimpan dari fitur *report* ini akan dilaporkan saat *driver* sudah sampai di PT. Pura dan digunakan untuk melakukan audit perjalanan yang dilakukan.

Saat *driver* sudah sampai di tujuan, *driver* wajib untuk melakukan verifikasi sebagai bukti bahwa *customer* sudah menerima barang yang dikirim oleh PT. Pura menggunakan menu verification.

Sebelumnya, *customer* terlebih dahulu dikirimkan QR Code yang berisi kode verifikasi oleh unit di PT. Pura (fitur ini terdapat dalam aplikasi administrator). Untuk melakukan verifikasi, *driver* hanya perlu untuk menscan QR Code yang dimiliki customer seperti pada Gambar 7. Data verifikasi ini akan disimpan di SQLite sekaligus ke *real time database* agar unit dapat mengetahui segera saat *driver* melakukan verifikasi.

Selanjutnya, menu *final report* digunakan saat *driver* sudah kembali ke PT. Pura. Driver wajib untuk "melaporkan" perjalanannya ke sistem. Untuk melaporkan, perangkat Android harus terkoneksi ke jaringan lokal PT. Pura. Semua data perjalanan (*history* lokasi perjalanan dan laporan/*report* pada menu pertama) akan dimasukkan ke sistem (Gambar 7).



Gambar 8. Tampilan Setiap Menu Aplikasi *Viewer*

Menu terakhir, sesuai dengan namanya, menu ini akan menghentikan *service tracking*. Menu ini digunakan saat *driver* sudah tidak melakukan perjalanan atau saat aplikasi terdapat masalah sehingga membutuhkan *restart*.

Gambar 8 adalah tampilan dari aplikasi *viewer* yang digunakan administrator untuk melihat status semua kendaraan yang sedang dalam pengiriman barang. Terdapat 3 menu utama dalam aplikasi *viewer*: (1) Menu *Tracker* digunakan untuk melihat lokasi setiap kendaraan dalam peta, (2) *Verification* digunakan untuk melihat status verifikasi setiap barang yang dikirim, dan (3) Menu *report* digunakan untuk melihat data laporan yang dibuat *driver*. Tampilan menu *viewer* ini memanfaatkan Google Maps, di mana akan menampilkan lokasi aktif dari kendaraan yang dipantau.

Konsep *real time* yang ditawarkan oleh *real time database* dapat direalisasikan dengan adanya *changes listener* pada CouchDB (atau *real time database* lainnya). *Change listener* akan memberitahu kepada sistem saat terdapat *record* yang berubah pada *database*. Maka saat aplikasi *tracker* melakukan *update* terhadap salah satu *record* kendaraan, aplikasi *viewer* akan langsung mendapatkan notifikasi dan menampilkannya pada Google Maps. Proses ini lebih sederhana dan tidak memberatkan *server* dibanding menggunakan *database* SQL yang harus melakukan *long polling* untuk *select* guna mengetahui apakah terdapat *record* yang berubah pada *database*.

Tabel 1. Hasil *Blackbox Testing*

No	Deskripsi	Hasil yang Diharapkan	Hasil yang Diberikan Sistem	Status Pengujian
1	Penguji melakukan *login* dengan aplikasi *tracker* sebagai *driver*	Aplikasi menerima informasi *login* dan berganti ke halaman *start service*	Sesuai yang diharapkan	*Valid*
2	Penguji memilih tombol START pada halaman *start service*	Aplikasi akan melakukan serangkaian proses untuk memulai *tracking service*, sekaligus membuat *record* pada CouchDB	Sesuai yang diharapkan	*Valid*
3	Penguji membuat laporan dari aplikasi *tracker*	Laporan terkirim ke *database* CouchDB dan tersimpan dalam SQLite	Sesuai yang diharapkan	*Valid*
4	Penguji melakukan *scanQRCode* dalam menu *verification* pada aplikasi *tracker*	*QRCode* terbaca dan laporan terkirim ke *database* CouchDB dan tersimpan dalam SQLite	Sesuai yang diharapkan	*Valid*
5	Penguji menggunakan menu *final report* untuk melaporkan *trip log*	*Log* tersimpan dalam *database* lokal PT. Pura	Sesuai yang diharapkan	*Valid*
6	Penguji melakukan *monitoring* kendaraan secara *real time* melalu aplikasi *viewer*	Posisi dari setiap kendaraan dapat dilihat secara *live* pada tampilan peta Google Maps	Sesuai yang diharapkan	*Valid*
7	Penguji melakukan verifikasi melalui aplikasi *viewer* tanpa *QRCode*	Pengiriman terverifikasi tanpa *QRCode*	Sesuai yang diharapkan	*Valid*
8	Penguji mencoba menggunakan fitur *auto-reconnect* saat koneksi *database* terputus karena sinyal *internet* yang tidak memadai	Sistem melakukan *auto-reconnect* dan dapat berjalan normal kembali	Sesuai yang diharapkan	*Valid*

Pengujian sistem dilakukan untuk menguji fungsi yang telah dibuat dalam *prototype* sistem *monitoring* kendaraan. Pengujian sistem yang dilakukan menggunakan metode *blackbox testing*. Pengujian dilakukan menggunakan perangkat Android Xiaomi Redmi 3 Pro dengan Sistem Operasi Android 5.1.1. *Blackbox Testing* dilakukan untuk mengetahui bahwa semua fungsi dan fitur yang ada di dalam sistem berjalan sesuai dengan harapan. Pengujian dilakukan dengan cara mencoba semua fungsi yang telah dibuat, kemudian membandingkan hasil pengujian dengan hasil yang diharapkan. Hasil *blackbox testing* dapat dilihat pada Tabel 1.

Hasil *blackbox testing* pada Tabel 1 menunjukkan bahwa status dari setiap fungsi adalah valid. Berdasarkan hasil yang didapat,

 https://doi.org/10.25077/ TEKNOSI.v4i2.2018.053-060

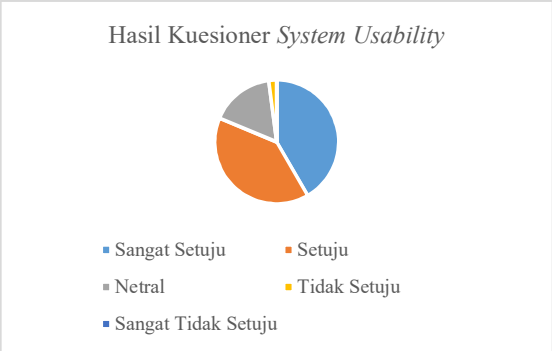
maka dapat disimpulkan bahwa sistem yang dibuat berjalan dengan baik dan sesuai yang diharapkan.

Usability Testing dilakukan untuk mengetahui apakah sistem telah memenuhi proses bisnis dan kebutuhan yang ada pada proses *tracking* pengiriman barang (*System Usability*). Hasil *usability testing* pada sistem ini diketahui dengan menggunakan kuesioner sejumlah 8 pertanyaan positif [14] yang terdapat pada Tabel 2.

Tabel 2. Daftar Pertanyaan *Usability Testing*

No	Daftar Pertanyaan Skala	Penilaian				
1	Secara keseluruhan, sistem mudah digunakan	1	2	3	4	5
2	Fungsi-fungsi sistem sudah sesuai dengan yang diharapkan	1	2	3	4	5
3	Sistem bersifat *user friendly*, *user*tidak kesulitan dalam penggunaan dan mudah untuk belajar menggunakan sistem	1	2	3	4	5
4	Sistem *tracking* dapat digunakan untuk mengakomodasi kebutuhan *tracking* kendaraan	1	2	3	4	5
5	Sistem yang dibuat dapat menggantikan sistem *tracking* terdahulu	1	2	3	4	5
6	Fungsi tambahan seperti *live reporting* dan *verifikasi* sangat membantu dalam proses pengiriman barang produksi	1	2	3	4	5
7	Sistem dapat dikembangkan lebih jauh untuk mengakomodasi kebutuhan lain dalam proses pengiriman barang	1	2	3	4	5
8	Aplikasi yang dibuat tidak terasa berat saat dijalankan dalam perangkat Android	1	2	3	4	5

Kuesioner dilakukan saat mempresentasikan hasil akhir sistem. Responden dari kuesioner *usability testing* terdiri dari 6 orang staff MIS yang merupakan pengembang sistem dan aplikasi di PT. Pura. Hasil kuesioner diperlihatkan pada Gambar 9.



Gambar 9. Hasil Kuesioner *System Usability*

Berdasarkan hasil dari kuesioner, kemudian dilakukan perhitungan hasil dengan Skala Likert. Skala Likert adalah skala yang digunakan untuk mengukur persepsi, pendapat seseorang atau kelompok, dalam hal ini konteksnya adalah sistem *monitoring* kendaraan yang telah dibuat [15]. Perhitungan menggunakan bobot nilai Skala Likert dapat dilihat pada Tabel 3 dan hasilnya pada Tabel 4.

Hasil akhir dari Skala Likert dihitung dengan cara mengalikan jumlah setiap kategori jawaban dengan bobot nilainya [15]. Total nilai didapatkan dengan menjumlahkan setiap hasil perkalian sebelumnya. Pada Tabel 5 didapatkan hasil akhir nilai dari kuesioner yang dilakukan adalah 202.

Tabel 3. Bobot Nilai Skala *Likert*

Jawaban	Bobot Nilai
SS (Sangat Setuju)	5
S (Setuju)	4
N (Netral)	3
TS (Tidak Setuju)	2
STS (Sangat Tidak Setuju)	1

Kemudian untuk mengetahui persentase kepuasan mengenai sistem yang dibuat digunakan rumus % Kepuasan. % Kepuasan dihitung dengan cara membagi total nilai dengan Y * 100, dengan Y = 5 * total responden * total pertanyaan yang diajukan. Hasil yang didapat dengan menggunakan rumus % Kepuasan dengan total responden = 6 dan total pertanyaan = 8 adalah:

Y = 5 x 6 x 8 ⇒
Y = 240 ∴
% Kepuasan = total nilai /Y x 100
% Kepuasan = 202 / 240 x 100
% Kepuasan = 84.16%

Tabel 5. Perhitungan Hasil Kuesioner menggunakan Bobot Nilai Skala *Likert*

Jawaban	Jumlah	Total Nilai
SS (Sangat Setuju)	20	100
S (Setuju)	19	76
N (Netral)	8	24
TS (Tidak Setuju)	1	2
STS (Sangat Tidak Setuju)	0	0
	Total	**202**

Perhitungan kepuasan kemudian dilakukan dengan mencocokkan hasil yang didapat dari rumus % Kepuasan dengan tabel persentase pada Skala Likert. Tabel 5 menunjukkan tabel persentase yang digunakan untuk mengkategorikan skala kepuasan berdasarkan nilai yang didapat.

Tabel 5. Tabel Persentase Kategori Nilai Skala *Likert*

% Kepuasan	Keterangan
0% - 19.99%	Sangat Buruk
20% - 39.99%	Kurang Baik
40% - 59.99%	Cukup
60% - 79.99%	Baik
80% - 100%	Sangat Baik

Berdasarkan semua perhitungan dengan Skala Likert didapatkan kesimpulan bahwa sistem yang dibuat sudah memenuhi keinginan dari responden. Kesimpulan ini diambil dari nilai % Kepuasan yang mencapai 84.16% dan dicocokkan menggunakan nilai persentase pada Tabel 4 yang mengkategorikan nilai yang didapat dalam kategori sangat baik [15]. Sistem yang dibuat juga mendapatkan beberapa masukan seperti peningkatan informasi yang diberikan oleh aplikasi *viewer*, seperti estimasi waktu dan jarak kendaraan akan mencapai tujuan. Selain itu aplikasi *tracker* juga diberikan masukan untuk dapat memantau kegiatan kendaraan selain posisinya. Hal ini meliputi berapa lama kendaraan berhenti di satu tempat dan memberikan laporan jika kendaraan melenceng jauh dari rute yang seharusnya.

5. KESIMPULAN

Berdasarkan penelitian yang dilakukan, didapatkan kesimpulan yaitu untuk membuat sebuah sistem *monitoring* kendaraan yang dapat diakses oleh unit produksi PT. Pura secara langsung adalah dengan membuat sistem *monitoring* kendaraan yang diimplementasikan ke dalam 2 aplikasi *mobile* berbasis Android. Aplikasi pertama adalah aplikasi yang dipakai *driver* kendaraan untuk keperluan *tracking* dan *reporting*, sedangkan aplikasi kedua adalah aplikasi monitoring dan verifikasi yang dipakai oleh administrator (dalam hal ini adalah perwakilan unit yang sedang menggunakan jasa transportasi PT. Pura.

Sistem *monitoring* kendaraan yang dibuat pada penelitian ini dapat memberikan laporan data lokasi kendaraan secara *real time* dan dapat diakses dengan cepat dan akurat saat dibutuhkan. Hal ini dapat dicapai dari penggunaan *real time database* yang diintegrasikan dengan sistem monitoring. *Prototype* sistem *monitoring* kendaraan dalam penelitian ini juga dapat digunakan untuk menggantikan sistem GPS tracker yang lama. Beberapa pertimbangan untuk menggunakan sistem yang baru adalah adanya fitur *real time* yang tidak mengharuskan mengirim SMS hanya untuk mendapat data lokasi setiap kendaraan. Selain itu sistem *monitoring* yang baru juga memiliki sistem pelaporan dan verifikasi barang yang terintegrasi, jadi tidak sebatas untuk keperluan *tracking* saja. Biaya operasional yang dikeluarkan juga akan lebih sedikit karena tidak memerlukan pembelian pulsa SMS dan menyewa alat GPS *tracker* dengan biaya bulanan, namun cukup membeli kartu SIM penyedia akses internet.

DAFTAR PUSTAKA

[1] Pura Group. "Creating Value Through Innovation." Internet: http://id.puragroup.com, May 25, 2017 [Feb. 05, 2018].

[2] R. Radifan. "Perancangan dan Pembuatan Sistem Informasi Lokasi Friend Finder Berbasis GPS pada Sistem Operasi Android". Skripsi, Teknik Informatika Institut Teknologi Sepuluh November Surabaya, 2010.

[3] D. Alfikri. "Aplikasi Auto-Reporting Position Tracking Berbasis Android Untuk Mengetahui Posisi Device Sebagai Sarana Monitoring Posisi Karyawan di PT Telkom Indonesia Kota Malang". Skripsi, Teknik Informatika Universitas Muhammadiyah Malang, 2012.

[4] R. Abbott, H. Garcia-Molina. "What is a Real-Time Database System?". Abstracts of the Fourth Workshop on Real-Time Operating systems, IEEE, 1987, pp. 134–138.

[5] Firdausillah, Fahri, E.Y. Hidayat, I.N. Dewi. "NoSQL: Latar Belakang, Konsep, dan Kritik." Semantik (2012), Semarang.

[6] Apache Foundation. "Apache CouchDB." Internet: https://wiki.apache.org/couchdb/, Jan. 17, 2017 [Feb. 05, 2018].

[7] C. Anderson, J. Lehnardt, dan N. Slater. (2013). CouchDB: The Definitive Guide. O'Reilly Media.

[8] Lightcouch Guide. "Couch Guide." Internet: http://www.lightcouch.org/docs.html, May 25, 2017 [Feb. 05, 2018]

[9] Google. "Contribute to Google Maps and Earn Points." Internet: https://support.google.com/maps/answer/6304221, May 10, 2017 [Feb. 05, 2018].

[10] A. Kindarto. (2008). Asyik Berinternet dengan Beragam Layanan Google. Yogyakarta: Andi.

[11] Z. Hasibuan. (2007). Metodologi Penelitian Pada Bidang Ilmu Komputer Dan Teknologi Informasi. Depok.

[12] R.S. Pressman. (1997). Rekayasa Perangkat Lunak Edisi ke-2. LN Harnaningrum, penerjemah: Yogyakarta: Andi.

[13] J. S. Dumas. (1999). A Practical Guide to Usability Testing. Intellect Books.

[14] J. R. Lewis. "IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use," in International Journal of Human-Computer Interaction, 1995, pp. 57-78.

[15] H. Sarjono. "Sekilas Tentang Skala Likert." Internet: https://sbm.binus.ac.id/2015/06/26/sekilas-tentang-skala-likert/, June 26, 2015 [Jul. 28, 2018].

 https://doi.org/10.25077/ TEKNOSI.v4i2.2018.053-060