

Perancangan Aplikasi Inspeksi Kendaraan Niaga Menggunakan Framework PhoneGap

Yeremia Silvester Sutoyo¹, Hindriyanto D. Purnomo², Yos Richard Beeh³

¹ PT. Indomarco Prismaatama, ^{2,3} P Universitas Kristen Satya Wacana
¹Jln. Jl. Ancol I, No. 9-10, Ancol Barat Jakarta, ²Salatiga, INDONESIA

¹yeremia.s@indomaret.co.id, ²hindriyanto.purnomo@staff.uksw.edu,
³yos.fti.uksw@gmail.com

Intisari— Proses inspeksi kendaraan mobil *delivery* di seringkali masih dilakukan secara dengan mengisi *form* pemeriksaan kendaraan secara konvensional. Hasil inspeksi ini juga belum terhubung dengan layanan jasa perbaikan, pemeliharaan dan servis di bengkel. Dalam penelitian ini dirancang aplikasi *mobile* yang dapat membantu dalam proses inspeksi. Aplikasi dirancang menggunakan dasar *Framework PhoneGap* yang merupakan salah satu *framework* yang *open source* yang memungkinkan aplikasi berbasis *web (html)* menjadi aplikasi *native* untuk semua jenis *platform* sistem operasi *smartphone / tablet* yang ada (*Android, Ios, BalckBerry, Windows Phone, Symbian*, dan lain-lain). Hasil dari penelitian ini adalah rancangan aplikasi untuk membantu proses inspeksi kendaraan niaga yang dihubungkan dengan sistem aplikasi permintaan jasa perbaikan, pemeliharaan serta servis dalam bengkel

Kata kunci— Inspeksi Kendaraan, Framework PhoneGap.

I. PENDAHULUAN

PT. XYZ merupakan perusahaan yang dibidang ritel / waralaba dan menaungi banyak gerai waralaba. Laju pertumbuhan gerai yang cepat menyebabkan volume transaksi melebihi 45 juta struk perbulan. PT. XYZ memiliki diferensiasi unit serta memunculkan inovasi-inovasi ekonomi dan memprioritaskan pemenuhan kebutuhan dan kepuasan konsumen. Saat ini perusahaan tersebut memiliki 11.285 gerai, terdiri dari 40% gerai milik terwaralaba dan 60% gerai milik perusahaan. Sebagian besar pasokan barang dagangan untuk seluruh gerai berasal dari 22 pusat distribusi yang menyediakan lebih dari 4.800 jenis produk [1].

Pada proses didalam *distribution center* terdapat proses pengiriman barang dari *distribution center* ke toko outlet. Sebelum dan sesudah barang dikirim ke outlet menggunakan mobil *delivery*, inspektor harus melakukan pengecekan terhadap mobil *delivery*. Inspeksi kendaraan merupakan pemeriksaan fisik kendaraan niaga (*dry, perishable, bakery, LPG 3 kg dan LPG 12 kg*) yang dilakukan oleh pihak GA (*General Affairs*). Tujuan inspeksi kendaraan adalah untuk menentukan kelayakan kendaraan *Delivery Driver* dalam mendistribusikan barang dagangan [2].

Proses inspeksi kendaraan mobil *delivery* pada saat ini masih dilakukan secara manual dengan mengisi *form* pemeriksaan kendaraan, pada saat inspektor (petugas yang bertugas melakukan inspeksi) melakukan inspeksi dengan mengisi *form* inspeksi kendaraan, setelah selesai inspektor membawa hasil inspeksi ke bagian *distribution center* untuk di-input-kan kembali hasil inspeksi kendaraan yang kendaraannya membutuhkan perawatan pada aplikasi permintaan jasa

perbaikan, pemeliharaan untuk mengeluarkan surat PJPP (Permintaan Jasa Perbaikan, Pemeliharaan) yang akan dibawa ke bengkel untuk dijadikan dasar intruksi agar kendaraan yang bersangkutan untuk ditindak lanjuti. Data hasil inspeksi juga mejadi dasar aplikasi bengkel untuk mengeluarkan SPK (Surat Perintah Kerja) untuk mekanik. Oleh karena itu, data-data hasil inspeksi belum terhubung antara aplikasi permintaan jasa perbaikan, pemeliharaan dan aplikasi bengkel. Berdasarkan masalah tersebut, diperlukan aplikasi mobile yang dapat membantu dalam hal proses inspeksi serta data-data hasil inspeksi dapat terhubung secara paralel antara aplikasi permintaan jasa perbaikan, pemeliharaan dan aplikasi bengkel. Pemilihan perangkat mobile sebagai perangkat dalam proses inspeksi kendaraan dikarenakan perkembangan perangkat mobile sangatlah pesat terutama mobile smartpone. Ditinjau dari keefektivitas penggunaan, perangkat mobile yang dapat digunakan dan dibawa kemana saja untuk proses inspeksi kendaraan dan data hasil inspeksi dapat dikirim kedalam database menggunakan jaringan internet.

Pada saat ini, perkembangan smartpone semakin beragam, seperti Android, IOS, BlackBerry, Windows Phone, dan lain-lain. Framework PhoneGap adalah salah satu framework yang open source yang memungkinkan kita menjadikan aplikasi berbasis web (html) menjadi aplikasi native. Dengan Phonegap kita bisa membuat aplikasi berbasis web yang nantinya bisa dijalankan sebagai aplikasi native di semua jenis platform mobile smartpone / tablet (Android, Ios, BlackBerry, Windows Phone, Symbian, dan lain-lain)[3]. PhoneGap merupakan framework open source saat ini yang digunakan lebih dari 400.000 developers diseluruh dunia [4]. Dalam laporan Developer Economics Q1 2013 dari survei yang dilakukan, sebanyak 34% memilih menggunakan PhoneGap, 21% menggunakan Appcelerator, 19% menggunakan Adobe Air, 12% menggunakan Sencha, 11% menggunakan Qt, 9% menggunakan Unity, 7% menggunakan Corona, 7% menggunakan Mono dan 6% memilih menggunakan Marmalade [5].

Dalam penelitian ini dirancang aplikasi inspeksi kendaraan niaga menggunakan framework phonegap. Perancangan aplikasi ini bertujuan untuk mempermudah proses inspeksi kendaraan niaga, sehingga proses inspeksi kendaraan tidak lagi mengisi form pemeriksaan kendaraan melainkan data-data inspeksi dimasukkan kedalam aplikasi smartpone sehingga data-data hasil inspeksi kendaraan dapat terhubung secara paralel antara aplikasi permintaan jasa perbaikan, pemeliharaan dan aplikasi bengkel.

II. KAJIAN PUSTAKA

Penelitian tentang system informasi operasional kendaraan belum banyak dilakukan. Diantara sedikit literature terkait hal itu, Lee dkk [6] merancang aplikasi sistem informasi operasional kendaraan berbasis Web. Sistem yang dibangun Lee dkk meliputi proses administrasi operasional dan perawatan. Sistem tersebut dipakai untuk mengetahui operasional kendaraan dalam sistem pemakaian kendaraan seperti : truk, mobil, motor, dan pemeliharaan kendaraan, data kendaraan yang dimiliki serta penyusunan laporan setiap periodenya secara cepat dibandingkan menggunakan sistem manual. Aplikasi tersebut secara otomatis dapat mengidentifikasi kendaraan yang sudah harus di-service

berdasarkan dari total jam pemakaian kendaraan tersebut serta dapat mencetak slip perjalanan berangkat, slip perjalanan kembali, slip perbaikan kendaraan, laporan perjalanan kendaraan, laporan perawatan kendaraan pada periode tertentu dan mengetahui status kendaraan ada [6].

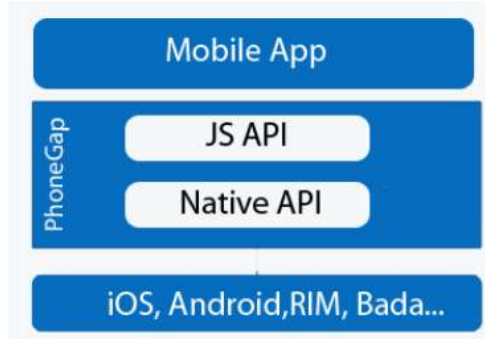
Shrivass dkk [7] menuliskan adanya peningkatan luar biasa dalam penggunaan iPhone, Android berbasis Ponsel, Smartphone, dan komputer tablet. Penggunaan ponsel lebih tinggi dari pada penggunaan komputer. Ada kesempatan untuk transplantasi di bidang akademik dan informasi dari komputer ke ponsel untuk meningkatkan layanan dan efisiensi manajemen. Maka dari itu dalam penelitian ini, Shrivass & Pardeshi mengembangkan aplikasi mobile cross-platform untuk Sistem Informasi Mahasiswa menggunakan framework PhoneGap. Dengan menggunakan framework PhoneGap pengembang dapat membuat aplikasi untuk beberapa jenis sistem operasi perangkat mobile dengan source code yang sama. Tulisan ini bertujuan untuk berbagi informasi antara mahasiswa, anggota staf, departemen dan administrasi perguruan tinggi. Aplikasi Sistem Informasi Mahasiswa ini, membantu mahasiswa untuk memeriksa hasil kuliah mereka, kehadiran, melihat data diri, memeriksa pengumuman, dan lain-lain serta Anggota Staff dan mahasiswa dapat melihat jadwal dari kelas mereka dan juga menyediakan layanan yang mudah digunakan langsung dari perangkat mobile [7].

Dalam penelitian ini dirancang Aplikasi Insepsi Kendaraan Niaga menggunakan Framework PhoneGap untuk memberikan kemudahan dalam proses inspeksi kendaraan niaga. Pemeriksaan mesin kendaraan dilakukan dalam kondisi 1-2 jam setelah kendaraan tidak digunakan. Tujuan dari inspeksi kendaraan adalah untuk menentukan kelayakan kendaraan Delivery Driver dalam mendistribusikan barang dagangan ke outlet [2].

PhoneGap adalah framework open source dari Nitobi dibuat pada awal tahun 2008. PhoneGap cukup populer dikalangan pengguna terutama karena fleksibilitasnya sederhana dan kemudahan penggunaan [8]. PhoneGap salah satu framework yang open source yang memungkinkan kita menjadikan aplikasi berbasis web (html) menjadi aplikasi native. Dengan PhoneGap kita bisa membuat aplikasi berbasis web yang nantinya bisa dijalankan sebagai aplikasi native di semua jenis platform mobile smartphone / tablet (Android, Ios, BalckBerry, Windows Phone, Symbian, dan lain-lain) [3]. PhoneGap merupakan framework open source saat ini yang digunakan lebih dari 400.000 developers diseluruh dunia [4]. Dalam laporan Developer Economics Q1 2013 dari survei yang dilakukan, sebanyak 34% memilih menggunakan PhoneGap, 21% menggunakan Appcelerator, 19% menggunakan Adobe Air, 12% menggunakan Sencha, 11% menggunakan Qt, 9% menggunakan Unity, 7% menggunakan Corona, 7% menggunakan Mono dan 6% memilih menggunakan Marmalade [5].

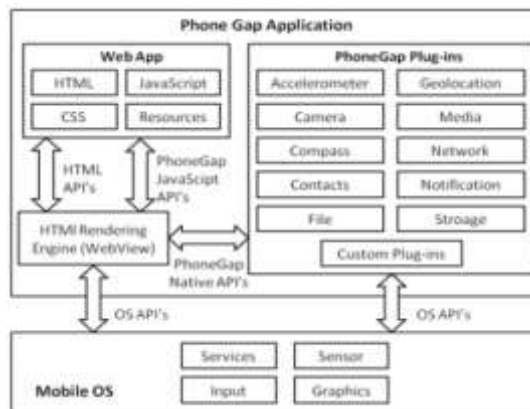
Arsitektur PhoneGap terutama terdiri dari tiga lapisan yaitu Aplikasi Web, PhoneGap, dan (OS dan API Native). Pada Gambar 1, lapisan atas merupakan sumber kode aplikasi. Lapisan tengah terdiri dari JavaScript dan API Native. Lapisan ini bertanggung jawab untuk antarmuka antara aplikasi web dan Lapisan PhoneGap. Selain itu, lapisan ini juga menangani antarmuka antara JavaScript yang digunakan oleh aplikasi dengan API native yang digunakan oleh sistem

operasi mobile. Fungsi Lapisan ini adalah untuk menjaga hubungan antara JavaScript API dan native API masing-masing dari sistem operasi mobile [7].



Gambar 1. Arsitektur PhoneGap

Pada Gambar 2 ditunjukkan skema yang lebih rinci yang disediakan oleh IBM. Gambar 2 mewakili semua komponen tentang aplikasi web, HTML mesin rendering, API PhoneGap dan lapisan OS. Selain itu, beberapa antarmuka yang berbeda ditunjukkan pada detail, seperti antarmuka antara lapisan API PhoneGap dan API native [7].



Gambar 2. Skema Lengkap dari Arsitektur PhoneGap

PhoneGap menggunakan jQuery library JavaScript dalam pengembangan framework dan lebih mudah membangun aplikasi berbasis jQuery untuk mengakses fitur native aplikasi. PhoneGap mendukung accelerometer, kamera, kompas, kontak, file, lokasi, media, jaringan, notification (peringatan, suara, getaran) dan penyimpanan [9].

Dalam perancangan aplikasi ini, untuk user interface dan action menggunakan jQuery Mobile dan jQuery. jQuery Mobile adalah framework yang berbasis HTML 5 / CSS 3 dan javascript untuk aplikasi web yang dijalankan di perangkat mobile. Apabila ingin membuat aplikasi web untuk perangkat mobile (smartphone / tablet PC) kita bisa menggunakan jQuery Mobile [3] dan jQuery merupakan salah satu dari sekian banyak JavaScript library yaitu kumpulan fungsi JavaScript yang siap pakai, sehingga mempermudah dan mempercepat dalam membuat kode JavaScript. Dengan menggunakan jQuery, skrip JavaScript yang panjang dapat disingkat menjadi beberapa baris kode saja [10].

Untuk pertukaran data antara client dan service diaplikasi ini menggunakan format JSON. JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari bahasa pemrograman JavaScript, standar ECMA-262 edisi ke-3 – Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dan lain-lain. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data [11].

III. RANCANGAN SISTEM INSPEKSI KENDARAAN

Dalam penelitian ini digunakan model proses prototyping. Model prototyping adalah salah satu metode pengembangan perangkat lunak yang dibuat dengan pendekatan aspek desain, fungsi, dan user-interface. Tahap-tahap dalam model prototyping adalah sebagai berikut: (1) Mendengarkan Pelanggan, (2) Membangun/Memperbaiki Mock-Up, (3) Pelanggan Melihat/Menguji Mock-Up. Hasil dari model proses prototyping yang dilakukan adalah sebagai berikut:

Tabel 1 menunjukkan hasil dari prototyping yang telah dilakukan. Pada tahapan pertama yaitu mendengarkan pelanggan, langkah yang dilakukan adalah wawancara terhadap user untuk mengumpulkan data-data dan kebutuhan yang diperlukan untuk pembuatan sistem. Berdasarkan data-data dan kebutuhan sistem yang telah dikumpulkan maka dilakukan tahapan kedua yaitu membangun/memperbaiki mock-up. Pada tahapan kedua, langkah yang dilakukan adalah melakukan perancangan atau memperbaiki perancangan sesuai dengan kebutuhan yang telah dikumpulkan. Tahapan selanjutnya adalah tahapan ketiga yaitu pelanggan melihat/menguji mock-up, pada tahapan ini user melakukan evaluasi terhadap sistem yang telah dibuat.

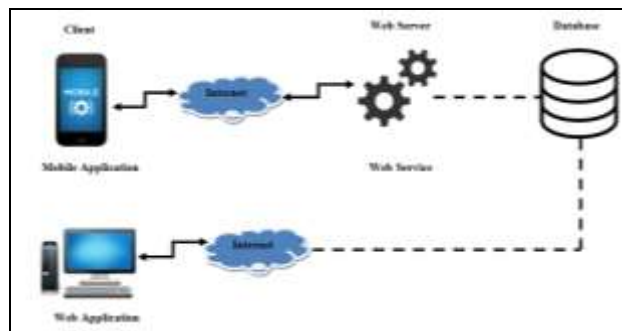
Proses bisnis inspeksi kendaraan yaitu petugas yang bertugas untuk melakukan inspeksi disebut inspektor. Inspektor itu sendiri merupakan bagian dari departemen General Affairs (GA) yang terdapat pada Distribution Center (DC). Sebelum dan sesudah barang dikirim ke outlet menggunakan mobil delivery, mobil delivery harus melakukan pengecekan kendaraan atau melakukan inspeksi kendaraan yang dilakukan oleh inspektor terhadap mobil delivery, yang bertujuan untuk menentukan kelayakan kendaraan delivery dalam mendistribusikan barang dagang. Pada saat kendaraan delivery akan pergi dan masuk ke dalam distribution center, kendaraan akan berhenti pada titik-titik tertentu untuk dilakukan inspeksi. Proses inspeksi dibutuhkan waktu 3 menit untuk inspektor melakukan pemeriksaan terhadap kendaraan delivery. Pada proses inspeksi kendaraan delivery, inspektor mengisi form pemeriksaan kendaraan dimana pada form pemeriksaan kendaraan sudah terdapat daftar komponen kendaraan yang akan diinspeksi dan status dari hasil inspeksi yaitu OK dan NOK. Jika terdapat komponen kendaraan yang bersatus NOK maka dari hasil inspeksi tersebut dari pihak distribution center akan memberikan persetujuan bahwa kendaraan delivery tersebut memang benar dalam kondisi tidak baik dan dikeluarkan PJPP

(Permintaan Jasa Perawatan/Perbaikan) yang akan diteruskan kepada pihak bengkel sebagai dasar instruksi kepada begkel bahwa kendaraan delivery tersebut untuk ditindak lanjuti.

TABEL I
HASIL PROTOTYPING

<i>Prototyping</i>	Tahap 1 (Mendengarkan Pelanggan)	Tahap 2 (Membangun/ Memperbaiki <i>Mock-Up</i>)	Tahap 3 (Pelanggan Melihat/ Menguji <i>Mock-Up</i>)
<i>Prototyping</i> I	Data yang diperoleh : a) <i>Form</i> Pemeriksaan Kendaraan b) Data Kendaraan c) Data unit perusahaan d) Data cabang perusahaan Kebutuhan yang diperlukan : a) Aplikasi menampilkan <i>form login</i> . b) Inspektur dapat melakukan pengisian data kendaraan. c) Aplikasi menampilkan daftar komponen kendaraan. d) <i>Checklist</i> OK dan NOK pada setiap daftar komponen kendaraan. e) <i>Text-box</i> untuk mengisi keterangan kerusakan. f) Aplikasi menampilkan <i>form</i> konfirmasi <i>driver</i> yang terdiri dari NIK dan <i>Password driver</i> .	a) Membangun aplikasi sesuai data dan kebutuhan yang telah dikumpulkan	Hasil Evaluasi : a) Perubahan pada pilihan OK dan NOK pada daftar komponen kendaraan. b) Perubahan menggunakan <i>radio button</i> sebelumnya menggunakan <i>check box</i> .
<i>Prototyping</i> II	Kebutuhan yang diperlukan : a) Hak akses masing-masing <i>user</i> yaitu inspektur, admin dan <i>driver</i> . b) Inspektur : melakukan inspeksi, melihat <i>history</i> inspeksi, dan ubah <i>password</i> . c) Admin : melihat daftar pegawai, melihat detail data pegawai, menambah data pegawai, mengubah data pegawai, menghapus data pegawai, melihat daftar item/komponen kendaraan, menambah data <i>item</i> /komponen kendaraan, mengubah data <i>item</i> /komponen kendaraan, menghapus data <i>item</i> /komponen	a) Mengubah pilihan OK dan NOK pada daftar komponen kendaraan menggunakan <i>radio button</i> , b) membedakan hak akses pada saat <i>login</i> terhadap inspektur, admin dan <i>driver</i> . c) Di dalam menu masing-masing <i>user</i> sudah terdapat fungsi-fungsi sesuai kebutuhan yang telah didiskusikan bersama <i>user</i> .	Hasil Evaluasi : a) Perubahan pada daftar item kendaraan yang akan diinspeksi. b) Perubahan dengan menampilkan daftar komponen kendaraan yang akan diinspeksi sesuai kategori yaitu bagian mesin, bagian luar, bagian dalam dan lain-lain dari kendaraan. sebelumnya terlalu panjang dan banyak

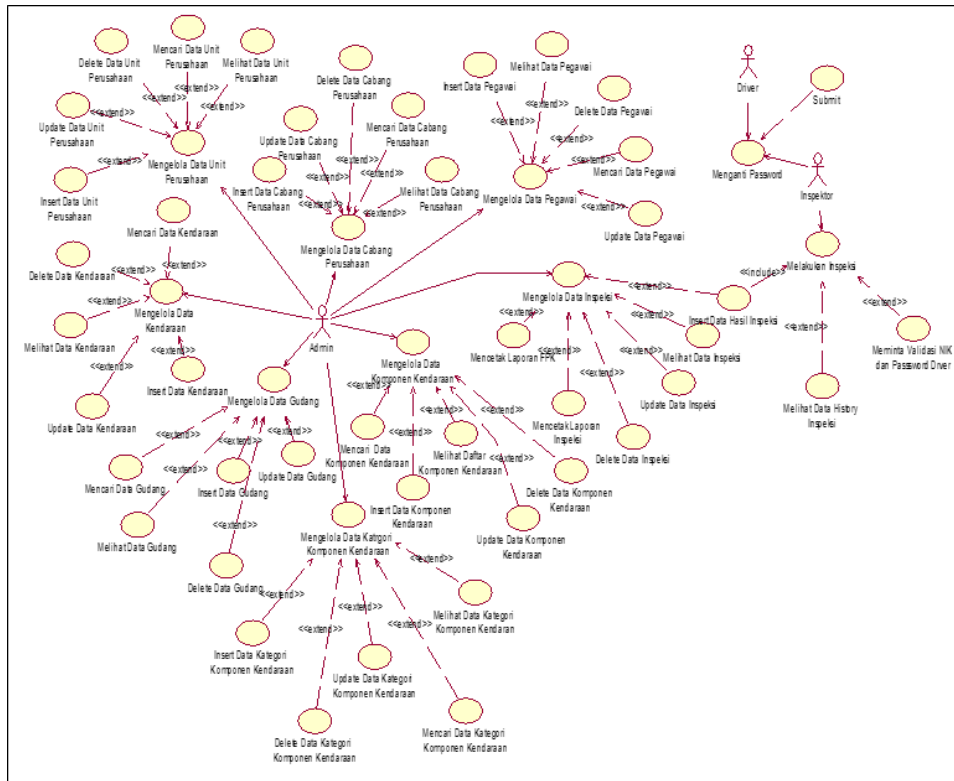
	kendaraan, dan ubah password.	
	d)Driver : melakukan ubah password.	
Prototyping III	<p>Kebutuhan yang diperlukan :</p> <p>a)Memberi <i>alert service</i> kendaraan sesuai dengan posisi KM kendaraan pada saat inspektor menginputkan posisi KM kendaraan sesuai dengan ketentuan yang ada</p>	<p>a)Menampilkan daftar komponen kendaraan yang akan diinspeksi sesuai kategori yaitu bagian mesin, bagian luar, bagian dalam dan lain-lain.</p> <p>b)Fungsi <i>alert service</i> kendaraan sesuai posisi KM kendaraan pd saat inspektor meng-inputkan posisi KM kendaraan sesuai dengan ketentuan yang ada.</p>
		<p>Hasil Evaluasi :</p> <p>a)Tidak ada perubahan</p>



Gambar 3. Rancangan Arsitektur Sistem Inspeksi Kendaraan

Gambar 3 merupakan rancangan arsitektur sistem inspeksi kendaraan, aplikasi mobile dijalankan pada smartphone. Web service menghubungkan antara aplikasi mobile dan database dengan pertukaran data menggunakan format JSON. Pada client terlebih dahulu diinstal aplikasi inspeksi kendaraan, kemudian dengan dukungan internet dapat menjalankan aplikasi untuk proses inspeksi kendaraan. Data yang tersimpan diambil oleh sistem dari server web service yang menggunakan format JSON untuk pertukaran data. Aplikasi web dapat diakses melalui komputer yang berfungsi sebagai aplikasi pengelolaan data yang diperlukan pada aplikasi mobile serta keluaran yang dihasilkan oleh aplikasi mobile berupa laporan inspeksi kendaraan.

Perancangan aplikasi inspeksi kendaraan niaga dilakukan menggunakan Unified Modeling Language (UML). Adapun diagram yang dibuat adalah Use Case Diagram dan Class Diagram. Use Case Diagram menggambarkan interaksi yang terjadi antara aktor dengan sistem [13]. Use Case Diagram dari sistem yang akan dibangun dapat dilihat pada Gambar 5.



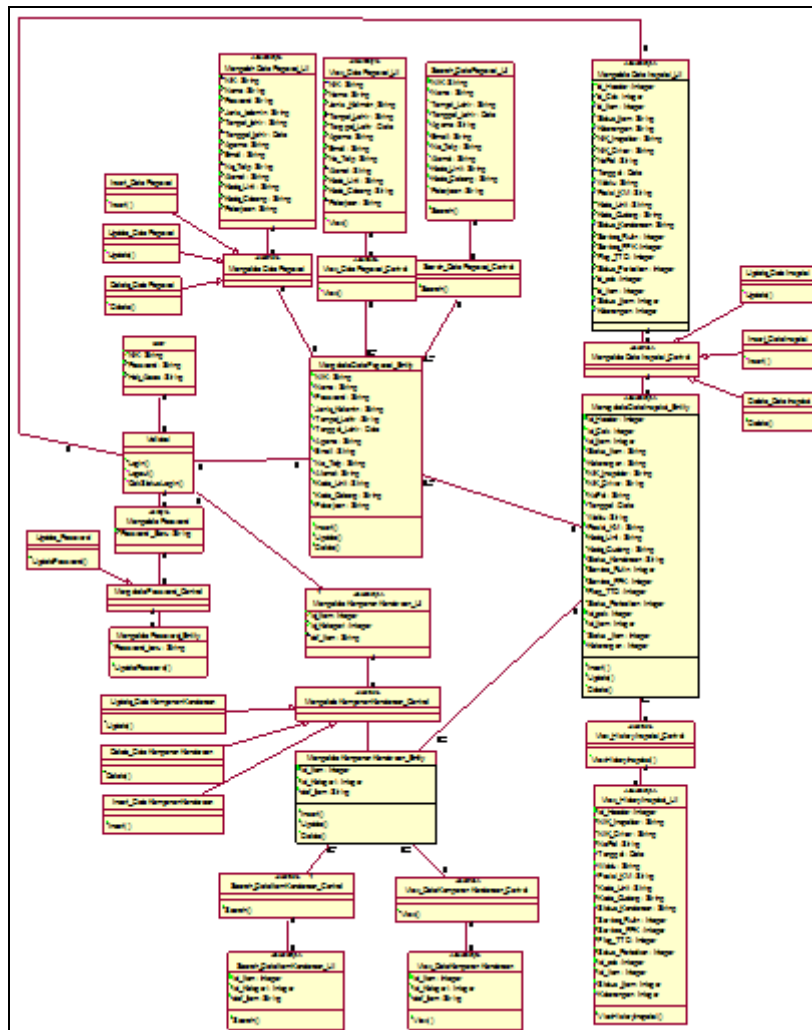
Gambar 5. Use Case Diagram

Pada Gambar 5 Use Case Diagram terdiri dari tiga aktor yaitu admin, inspektor dan driver. Tiap aktor memiliki peran masing-masing, aktor admin mempunyai peran dalam mengelola data pegawai, mengelola data kendaraan, mengelola data kategori komponen kendaraan, mengelola data komponen kendaraan yang harus diinspeksi, mengelola data gudang perusahaan, mengelola data unit perusahaan, mengelola data cabang perusahaan dan mengubah password admin. Inspektor mempunyai peran untuk melakukan inspeksi, melihat history inspeksi dan mengubah password inspektor. Sementara driver dalam aplikasi hanya dapat mengubah password driver yang digunakan untuk verifikasi tanda tangan setelah inspektor melakukan inspeksi.

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem [13]. Pada perancangan ini Class Diagram dapat dilihat pada Gambar 6.

Dalam membuat Class Diagram, yang dipecah dari Use Case sebelumnya terdapat tiga class yang saling berhubungan yaitu boundary, controller dan entity. Boundary merupakan apa saja yang akan ditampilkan dalam user interface pada masing-masing aktor dalam perancangan aplikasi inspeksi kendaraan niaga. Controller merupakan operasi-operasi yang terdapat dalam aplikasi ini yaitu insert, update, delete, view dan search. Sedangkan entity merupakan gambaran hubungan antar entitas yang ada pada aplikasi ini. Pada aplikasi ini terdapat tiga entity yang saling terhubung yaitu Mengelola Data Pegawai_Entity, Mengelola Komponen Kendaraan_Entity, Mengelola Data Inspeksi_Entity yang saling terhubung. Beberapa controller yaitu Mengelola Data Pegawai_Control, Mengelola

Komponen Kendaraan_Control, Mengelola Data Inspeksi_Control, View_Data Pegawai_Control, Search_Data Pegawai_Control, View_History Inspeksi-Control,View_Data Komponen Kendaraan_Control, Search_Data Komponen Kendaraan-Control. Tiap controller masing-masing memiliki hubungan dengan entity yang ada. Pada controller terdapat fungsi-fungsi yang digunakan untuk mengakses entity. Tiap masing-masing boundary yaitu Mengelola Data Pegawai_UI, Mengelola Data Inspeksi_UI, Mengelola Komponen Kendaraan_UI terhubung dengan controller yang berfungsi untuk mengakses ke dalam database.



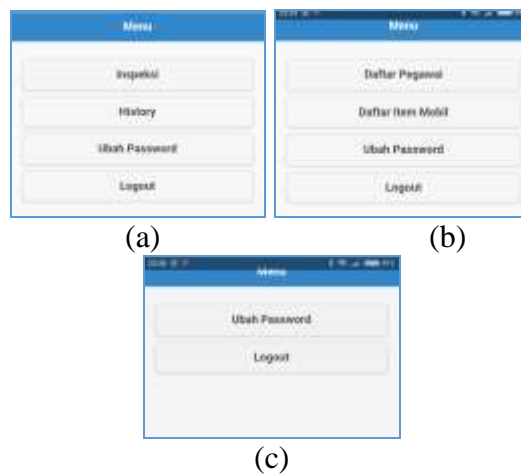
Gambar 6. Use Case Diagram

Dalam membuat *Class Diagram*, yang dipecah dari *Use Case* sebelumnya terdapat tiga *class* yang saling berhubungan yaitu *boundary*, *controller* dan *entity*. *Boundary* merupakan apa saja yang akan ditampilkan dalam *user interface* pada masing-masing aktor dalam perancangan aplikasi inspeksi kendaraan niaga. *Controller* merupakan operasi-operasi yang terdapat dalam aplikasi ini yaitu *insert*, *update*, *delete*, *view* dan *search*. Sedangkan *entity* merupakan gambaran hubungan antar entitas yang ada pada aplikasi ini. Pada aplikasi ini terdapat tiga

entity yang saling terhubung yaitu Mengelola Data Pegawai_Entity, Mengelola Komponen Kendaraan_Entity, Mengelola Data Inspeksi_Entity yang saling terhubung. Beberapa *controller* yaitu Mengelola Data Pegawai_Control, Mengelola Komponen Kendaraan_Control, Mengelola Data Inspeksi_Control, View_Data Pegawai_Control, Search_Data Pegawai_Control, View_History Inspeksi-Control, View_Data Komponen Kendaraan_Control, Search_Data Komponen Kendaraan-Control. Tiap *controller* masing-masing memiliki hubungan dengan *entity* yang ada. Pada *controller* terdapat fungsi-fungsi yang digunakan untuk mengakses *entity*. Tiap masing-masing *boundary* yaitu Mengelola Data Pegawai_UI, Mengelola Data Inspeksi_UI, Mengelola Komponen Kendaraan_UI terhubung dengan *controller* yang berfungsi untuk mengakses ke dalam *database*.

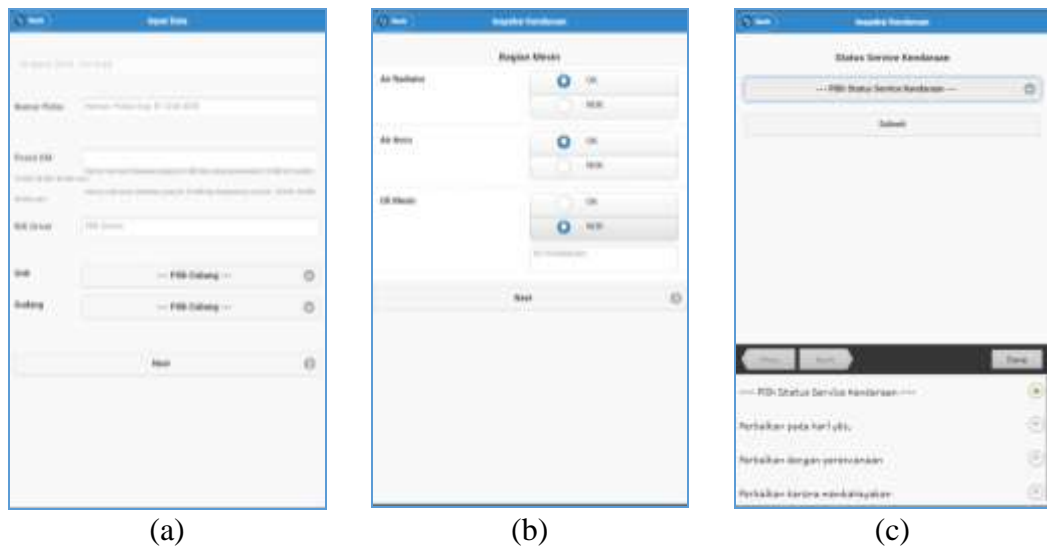
IV. HASIL DAN PEMBAHASAN

Pada saat aplikasi dijalankan akan muncul form login. Pada proses login user dibedakan menjadi tiga yaitu Admin, Inspektor, dan Driver. Untuk melakukan login, user harus menginputkan NIK dan Password yang sudah terdaftar dalam database. Jika login berhasil maka akan ditampilkan halaman menu utama sesuai dengan hak akses masing-masing user, seperti pada Gambar 7.



Gambar 7. Menu pada Masing-Masing User (a) Menu Inspektor, (b) Menu Admin, (c) Menu Driver

Pada halaman menu inspektor terdapat menu inspeksi yang berfungsi untuk proses inspeksi kendaraan yang akan dilakukan. Ketika memilih menu inspeksi maka akan tampil *form* Inspeksi Seperti Gambar 8.



Gambar 8. (a) Form Inspeksi Input Data Kendaraan, (b) Form Checklist Item Kendaraan, (c) Form Status Service Kendaraan

Pada form Inspeksi halaman input data kendaraan yang akan diinspeksi oleh inspektur. Inspektur diminta untuk meng-input-kan Nomor Polisi, Posisi KM, NIK Driver, Unit dan Gudang Perusahaan. Pada pilihan combo box unit dan gudang, hasil yang ditampilkan sesuai data yang ada didalam database. Setelah inspektur mengisi form input-an, kemudian klik tombol Next. Pada saat inspektur meng-klik tombol Next maka sistem didalamnya akan memeriksa Nomor Polisi dan NIK Driver sudah terdaftar dalam database atau tidak. Jika sudah terdaftar maka akan menampilkan form checklist item kendaraan. Jika tidak maka akan muncul pemberitahuan bahwa Nomor Polisi atau NIK Driver tidak ada. Pada halaman form Checklist Item Kendaraan terdapat beberapa kategori bagian kendaraan yaitu bagian mesin, bagian luar, bagian dalam dan lain-lain dari kendaraan yang akan diinspeksi. Pada bagian terakhir inspektur diminta untuk memilih status service kendaraan yang telah diinspeksi. Pada saat inspektur menekan tombol submit maka data hasil inspeksi akan tersimpan di dalam database.



Gambar 9. Form Konfirmasi Driver

Setelah semua proses inspeksi selesai maka akan ada halaman konfirmasi driver kendaraan seperti Gambar 9 untuk memastikan bahwa kendaraan tersebut telah diinspeksi.

Adapun pseudocode untuk proses inspeksi kendaraan dapat dilihat pada Kode Program 1.

```

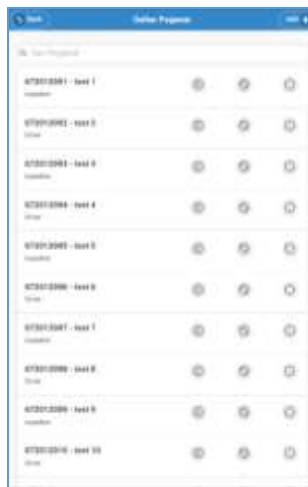
1. Start
2. Input Nomor_Polisi;
3. Input Posisi_KM;
4. Input NIK_Driver;
5. Input Unit;
6. Input Gudang;
7. get Tanggal;
8. get Waktu;
9. Cek Item_Checklist_Mobil;
10. get amountCorrect= 0;
11. getJSON (URL + 'DaftarItem',
    function(data){
12. get item = data.items;
13. get id = [];
14. get value = [];
15. get ket = [];
16. each (item, function (index, cek){
17. get radios = getElementsByName ('radio-
    mini + cek.id_item + ');
18. for (j = 0; j< radios.lenght; j++){
19. amountCorrect++;
20. id.push(cek.id_item);
21. value.push(radio.value);
22. ket.push('textarea'+ cek.id_item +
    '').val();
23. } }) });
24. if amountCorrect <= 0 then
25. get Status = OK;
26. get Keterangan = "-";
27. get Status_Perbaikan = 0;
28. get Id_Item = 0;
29. get flag_ttd = 0;
30. get Nomor_Polisi;
31. get Posisi_KM;
32. get NIK_Driver;
33. get Unit;
34. get Gudang;
35. get status_item = OK;
36. get Status_Perbaikan;
37. get url+InsertOK;
38. Endif
39. Else if amountCorrect > 0 then
40. get Status = NOK;
41. get Keterangan = ket[];
42. get Status_Perbaikan = 1;
43. get Id_Item = id[];
44. get flag_ttd = 0;
45. get Nomor_Polisi;
46. get Posisi_KM;
47. get NIK_Driver;
48. get Unit;
49. get Gudang;
50. get status_item = value[];
51. get Status_Perbaikan;
52. get url+InsertNOK;
53. Endif
54. End.

```

Kode Program 1 dapat dijelaskan sebagai berikut pada langkah 1 sampai 8 merupakan input-an data inspeksi kendaraan yang akan diinspeksi yang terdiri dari Nomor_Polisi,Posisi_KM, NIK_Driver, Unit, Gudang, Tanggal dan waktu. Langkah 9 merupakan pemeriksaan komponen kendaraan yang akan diinspeksi. Langkah 10 merupakan pendeklarasian variabel amountCorrect yang berfungsi

untuk variabel penyimpanan jumlah radio button yang dipilih berstatus NOK. Langkah 11 sampai 23 menjelaskan, dari daftar komponen kendaraan yang telah diinspeksi dengan memanggil fungsi DaftarItem, adakah radio button yang terpilih merupakan NOK, jika ada maka variabel amountCorrect akan bertambah sesuai dengan banyak radio button yang terpilih merupakan NOK kemudian nilai id, value dan ket dimasukkan kedalam variabel array yang telah dideklarasikan yaitu id, value dan ket menggunakan fungsi push yang telah disediakan oleh JQuery. Langkah 24 sampai 53 menjelaskan jika variabel amountCorrect kurang dari sama dengan 0 maka data-data yang ditampung dalam variabel Status, Keterangan, Status_Perbaikan, Id_Item, falg_ttd, Nomor_Polisi, Posisi_KM, NIK_Driver, Unit, Gudang, status_item, Status_Perbaikan akan disimpan kedalam database dengan fungsi InsertOK. Jika variabel amountCorrect lebih dari 0 maka data-data yang ditampung dalam variabel Status, Keterangan = ket[] (array), Status_Perbaikan, Id_Item = id[] (array), falg_ttd, Nomor_Polisi, Posisi_KM, NIK_Driver, Unit, Gudang, status_item = value[] (array), Status_Perbaikan akan disimpan kedalam database dengan fungsi InsertNOK.

Pada menu admin terdapat menu daftar pegawai yang dapat dilihat pada Gambar 10. Dalam halaman daftar pegawai terdapat tiga tombol fungsi yaitu tombol fungsi Add, Detail, Edit, dan Delete.



Gambar 10. Halaman Daftar Pegawai

Pada menu admin juga terdapat menu daftar item yang dapat dilihat pada Gambar 11. Dalam halaman daftar item terdapat tiga tombol fungsi yaitu tombol fungsi Add, Edit, dan Delete.



Gambar 11. Halaman Daftar Item Checklist Kendaraan Mobil

Framework PhoneGap memberikan kemudahan pada proses pembuatan aplikasi. Dengan *PhoneGap* kita bisa membuat aplikasi berbasis *web* yang nantinya bisa dijalankan sebagai aplikasi *native*. Dengan menggunakan *framework PhoneGap* pengembang dapat membuat aplikasi untuk beberapa jenis sistem operasi perangkat *mobile* dengan *source code* yang sama. Adapun uji coba yang telah dilakukan terhadap beberapa device, aplikasi dapat berjalan pada Android versi 4.2.1 (*Jelly Bean*), Android Versi 4.4.2 (*KitKat*), Android versi 5.0 (*Lollipop*), Android versi 6.0 (*Marshmallow*) dan iOS versi 8.

A. Pengujian Aplikasi

Pengujian yang dilakukan adalah pengujian terhadap responden/pengguna sistem. Pengujian dilakukan kepada 5 responden user testing pegawai Software Developer (SD) 5 PT. XYZ. Perhitungan hasil pengujian menggunakan skala Likert. Pilihan jawaban pada kuisioiner yang diberikan terdiri atas lima pilihan jawaban dalam skala Likert, antara lain: Sangat Setuju (SS) dengan 5 point, Setuju (S) dengan 4 point, Netral (N) dengan 3 point, Tidak Setuju dengan 2 point, dan Sangat Tidak Setuju dengan 1 point. Untuk mendapatkan skor masing-masing pilihan jawaban dihitung dengan mengalikan point dari pilihan jawaban dengan jumlah responden [14]. Hasil dalam persen diperoleh dari Jumlah Skor per Jumlah skor ideal dikali 100, dengan jumlah skor ideal = Skor tertinggi dikalikan dengan jumlah responden. Diperoleh jumlah skor ideal = 25. Hasil perhitungan kuesioner menggunakan skala Likert untuk fungsi-fungsi pada menu inspektor dapat dilihat pada Tabel 2.

TABEL II
HASIL PERHITUNGAN MENGGUNAKAN SKALA LIKERT UNTUK FUNGSI-FUNGSI
MENU INSPEKTOR DAN KEBUTUHAN INSPEKSI KENDARAAN

No	Pertanyaan	SS	S	N	TS	STS	Jumlah Skor	Hasil
1	Apakah fungsi Login pada aplikasi Inspeksi Kendaraan dapat berjalan dengan baik sesuai fungsinya ?	4	1				24	96%
2	Apakah fungsi inspeksi kendaraan pada aplikasi Inspeksi Kendaraan dapat berjalan dengan baik sesuai fungsinya?	2	3				22	88%
3	Apakah fungsi alert untuk pemberitahuan service rutin kendaraan berjalan dengan baik dan sesuai dengan ketentuan ?	3	1	1			22	88%
4	Apakah fungsi untuk melihat data history inspeksi kendaraan dapat berjalan dengan baik sesuai fungsinya ?	1	4				21	84%
5	Apakah fungsi ubah password pada menu inspektor di aplikasi Inspeksi Kendaraan dapat berjalan dengan baik sesuai fungsinya ?	4	1				24	96%
6	Apakah aplikasi Inspeksi Kendaraan sudah menyediakan form input data sesuai dengan kebutuhan ?	4	1				24	96%
7	Apakah fungsi pada aplikasi Inspeksi Kendaraan sudah cukup untuk melakukan proses inspeksi kendaraan?	4	1				24	96%
8	Apakah aplikasi Inspeksi Kendaraan ini sudah cukup membantu user dalam melakukan inspeksi kendaraan ?		4	1			19	76%
9	Apakah tampilan user interface pada aplikasi Inspeksi Kendaraan sudah baik?		3	2			18	72%
10	Apakah perlu aplikasi Inspeksi Kendaraan dikembangkan lagi ?	4		1			23	92%
Total							221	88,4 %

Berdasarkan hasil perhitungan kuisioner didapat total skor 221. Hasil dalam persen diperoleh dari total skor dibagi nilai maksimal = $(221 / 5 \text{ (point terbesar)} \times 10 \text{ (banyak pertanyaan)} \times 5 \text{ (responden)}) \times 100 = (221 / 250) \times 100 = 88,4 \%$. Dari hasil perhitungan kuisioner berdasarkan skala Likert, diperoleh kesimpulan bahwa aplikasi sudah memenuhi kebutuhan pada proses inspeksi kendaraan serta fungsi-fungsi pada menu inspektor dapat berjalan dengan baik yang dibuktikan dengan angka 88,4%.

Hasil perhitungan kuesioner menggunakan skala Likert untuk fungsi-fungsi pada menu admin dapat dilihat pada Tabel 3.

TABEL III
HASIL PERHITUNGAN MENGGUNAKAN SKALA LIKERT UNTUK FUNGSI-
FUNGSI MENU ADMIN

No	Pertanyaan	SS	S	N	TS	STS	Total	Hasil
1	Apakah fungsi daftar pegawai pada menu admin di aplikasi Inspeksi Kendaraan dapat menampilkan daftar pegawai yang terdaftar ?	1	4				21	84%
2	Apakah fungsi save pada daftar pegawai untuk menyimpan data pegawai dapat berjalan dengan baik sesuai dengan fungsinya ?	3	2				23	92%
3	Apakah fungsi detail pada daftar pegawai dapat menampilkan detail setiap pegawai dapat berjalan dengan baik sesuai fungsinya?	5					25	100%
4	Apakah fungsi edit data pegawai pada daftar pegawai dapat berjalan dengan baik sesuai dengan fungsinya ?	3	2				23	92%
5	Apakah fungsi delete data pegawai dapat berjalan dengan baik sesuai dengan fungsinya ?	3	2				23	92%
6	Apakah fungsi daftar komponen mobil pada menu admin di aplikasi Inspeksi Kendaraan dapat menampilkan daftar pegawai yang terdaftar?	3	2				23	92%
7	Apakah fungsi save pada daftar komponen kendaraan untuk menyimpan data komponen kendaraan dapat berjalan dengan baik sesuai dengan fungsinya ?	2	3				22	88%
8	Apakah fungsi edit data komponen kendaraan pada daftar komponen kendaraan dapat berjalan dengan baik sesuai dengan fungsinya ?	2	3				22	88%
9	Apakah fungsi delete data komponen kendaraan dapat berjalan dengan baik sesuai dengan fungsinya ?	2	3				22	88%
10	Apakah fungsi ubah password pada menu admin di aplikasi Inspeksi Kendaraan dapat berjalan dengan baik sesuai fungsinya ?	5					25	100%
Total							229	91,6%

Berdasarkan hasil perhitungan kuisisioner didapat total skor 229. Hasil dalam persen diperoleh dari total skor dibagi nilai maksimal = $(229 / 5 \text{ (point terbesar)} \times 10 \text{ (banyak pertanyaan)} \times 5 \text{ (responden)}) \times 100 = (229 / 250) \times 100 = 91,6 \%$. Dari angka 91,6% hasil perhitungan kuisisioner tersebut, diperoleh kesimpulan bahwa fungsi-fungsi pada menu admin dapat berjalan dengan baik.

V. KESIMPULAN

Berdasarkan user acceptance testing dapat disimpulkan bahwa aplikasi sudah memenuhi kebutuhan pada proses inspeksi kendaraan serta fungsi-fungsi pada menu inspektor dapat berjalan dengan baik yang dibuktikan dengan hasil persentase yang didapat yaitu 88,4% dan fungsi-fungsi pada menu admin dapat berjalan dengan baik yang dinyatakan dari hasil perhitungan kuisioner untuk fungsi-fungsi menu admin diperoleh presentase 91,6% serta data-data hasil inspeksi dapat berhubungan secara paralel antara aplikasi permintaan jasa perbaikan, pemeliharaan dan aplikasi bengkel yang dibuktikan dengan hasil simulasi aplikasi. Berdasarkan hasil pengujian aplikasi pada beberapa device dapat disimpulkan bahwa dengan source code yang sama, aplikasi dapat dibangun dan berjalan dengan baik pada Android versi 4.2.1 (Jelly Bean), Android Versi 4.4.2 (KitKat), Android versi 5.0 (Lollipop), Android versi 6.0 (Marshmallow) dan iOS versi 8. Dari penelitian yang telah dilakukan, adapun saran yang perlu diperhatikan yaitu lebih memanfaatkan API PhoneGap yang tersedia seperti File dan Storage untuk pengembangan offline mode pada aplikasi inspeksi kendaraan niaga, agar dapat digunakan di luar jangkauan jaringan lokal perusahaan yang menggunakan wifi.

REFERENSI

- [1] Profil PT. Indomarco Prismatam. <http://www.indomaret.co.id>. Diakses tanggal 14 Oktober 2015.
- [2] Anonim.2015. Memorandum Sosialisasi perubahan alur proses inspeksi kendaraan niaga di Cabang Indomaret Nomor : 365 /CPS/15. Jakarta : PT. Indomarco Prismatama.
- [3] Harahap, Nazruddin Safaat. 2014. ANDROID Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Bandung : Informatika.
- [4] Georgiev, Martin, Suman Jana & Vitaly Shmatikov, 2014, Breaking and Fixing Origin-Based Access Control in Hybrid Web/Mobile Application Frameworks, <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4254737/>. Diakses tanggal 12 Desember 2015.
- [5] Pros and Cons of the Top 5 Cross-Platform Tools, <http://www.developereconomics.com/pros-cons-top-5-cross-platform-tools/>. Diakses tanggal 20 Januari 2016.
- [6] Lee, Alberthus Tommy. 2009. “Aplikasi Sistem Informasi Operasional Kendaraan Berbasis Web Pada CV MADHYA Semarang”. Salatiga : Jurusan Sistem Informasi Universitas Kristen Satya Wacana.
- [7] Shrivas, Avinash dan Pardeshi, Anandkumar, 2013, To Study and Design a Cross-Platform Mobile Application for Student Information System using PhoneGap Framework, International Journal of Emerging Technology and Advanced Engineering, 3(9) : 390 – 395.
- [8] Hartmann, Gustavo, Geoff Stead & Asi Degani. 2011. Cross-Platform Mobile Development. [e-book]. <https://wss.apan.org/jko/mole/Shared%20Documents/Cross->

- Platform%20Mobile%20Development.pdf. Diakses tanggal 12 Desember 2015.
- [9] Redda, Yonathan Aklilu. 2012. “Cross Platform Mobile Application Development”. Trondheim : Jurusan Master in Information System Norwegian University of Science and Technology.
- [10] Rohingun. 2015. Smart Trik JQuery Without Plugin. Yogyakarta : Andi.
- [11] JSON. Pengenalan JSON. <http://www.json.org/json-id.html>, diakses tanggal 18 November 2015.
- [12] Chandra.K, Fendhika, 2013, Perancangan dan Implementasi Sistem Informasi Manajemen Praaktek Kerja Lapangan Berbasisi Web Menggunakan YUI Library (Studi Kasus : PT.PLN (Persero) P3B Jawa Bali APP Salatiga), Jurnal Teknologi Informasi AITI, 10(2) : 101 – 200.
- [13] Sukamto, Rossa A. & M. Shalahuddin. 2013. Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Bandung : Informatika.
- [14] Widoyoko, Eko Putro. 2012. Teknik Penyusunan Instrument Penelitian. Yogyakarta : Pustaka Pelajar.