

Pembuatan Aplikasi Salesforce Automation System pada Toko IV Berbasis Android

Soen, Surya Soenaryo¹, Yulia², Henry Novianus Palit³

Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra

Jl. Siwalankerto 121-131, Surabaya 60236

Telp. (031)-2983455, Fax. (031)-8417658

E-mail: wei.surya@gmail.com¹, yulia@petra.ac.id², hnpalit@petra.ac.id³

ABSTRAK

Sistem informasi yang dimiliki Toko IV telah diimplementasikan di Toko IV sudah mendukung untuk keperluan jangka panjang pada sisi administratif. Namun, Toko IV masih belum memiliki aplikasi pendukung untuk sektor penjualan yang melakukan kontak langsung dengan pelanggan setiap hari.

Kondisi Toko IV yang memegang puluban *brand* untuk setiap kategori produk yang dijual membuat kewalahan para *salespeople*, karena mereka harus mempelajari dan memahami keunikan pada setiap produk. Selain itu, terkadang terdapat permasalahan komunikasi antara manager dan *salespeople*. Terdapat kemungkinan *sales people* tidak memahami promosi yang sedang berlangsung pada Toko IV. Selain itu, terdapat juga beberapa kejadian yang dimana mereka lupa akan tugas-tugas yang diserahkan olehnya.

Demi mengatasi kondisi sekarang, dibuat sebuah aplikasi Android. Aplikasi Android ini dibuat dengan menggunakan Ionic Framework, AngularJS, Google Firebase database, Amazon RDS database, dan SQLite database. Terdapat 7 fungsi utama, antara lain: *Account Management, Incentive Management, Company Timeline, Leads Management, Opportunity Management* dan *Company Statistics*. Juga terdapat fungsi tambahan lainnya untuk mendukung *sale people* dan *manager*, seperti halnya: *user tracking, private-chatting, group-chatting, dan real-time information* untuk *data-driven decision*.

Hasil akhir dari pengembangan aplikasi ini buat pengguna dapat memperoleh wawasan yang lebih akan produk dan mengetahui informasi terbaru yang penting dengan lebih cepat dan efektif. Aplikasi ini juga dapat bekerja dengan baik dengan sistem informasi yang sudah ada.

Kata Kunci: Salesforce, Ionic Framework, NoSQL Database, SQL Database dan JavaScript

ABSTRACT

Currently, the information system that was recently implemented in IV shop worked well in the long term for administration purpose. Meanwhile, IV shop still lacks any supporting application for their front-end sales department, which has direct contact with customer in daily basis.

The fact that IV shop sells various car-related products with various brands troubles their salespeople because they have to learn and understand each product's uniqueness. In addition, occasionally, there are communication problems between manager and salespeople. The salespeople may not fully understand the current promotion in IV shop. Furthermore, there were some occasions that they forgot their assigned tasks.

In order to solve the current condition, an android application were developed. This Android application was created by using Ionic Framework, AngularJS, Google Firebase database, Amazon RDS database, and SQLite database. This application has 7 major functions, which are: Account Management, Incentive Management, Company Timeline, Leads Management, Opportunity Management and Company Statistics. There are some additional functions to support the front-end salespeople and manager, such as: user tracking, private-chatting, group-chatting, and real-time information for data-driven decision.

The final result of this application development lets users get a better insight about products and know the recent important information faster and more effective. This application also works well with the existing system.

Keywords: Salesforce, Ionic Framework, NoSQL Database, SQL Database and JavaScript

1. PENDAHULUAN

Toko IV merupakan toko yang bergerak di bidang jasa dan penjualan aksesoris mobil di Semarang. Barang yang ditawarkan berupa pelek mobil, ban mobil, kaca film mobil, oli, audio mobil, dan berbagai jenis aksesoris lain. Toko IV juga menerima layanan berupa pemasangan aksesoris mobil, *spooring* dan *balancing* mobil, dan cuci mobil. Toko IV memegang puluhan *brand* untuk setiap kategori produk yang dijual. Untuk mendukung proses pencatatan di Toko IV, maka pada tahun 2001, Toko IV membentuk sebuah sistem informasi berbasis MS-DOS. Sistem informasi ini ditujukan untuk membantu pencatatan pembelian, pencatatan penjualan, dan melakukan konfirmasi kuota barang yang berada di gudang. Pada tahun 2015, toko IV telah membuat sebuah sistem informasi baru berbasis aplikasi *website*. Selain untuk membaharui sistem informasi toko IV menjadi teknologi terbaru, dimasukkan beberapa fitur tambahan untuk menunjang kebutuhan Toko IV, seperti halnya fungsi untuk memiliki banyak toko dan gudang. Terdapat dua alasan penunjang dalam pemilihan teknologi ini, yaitu: penggunaan teknologi aplikasi *website* dimaksudkan untuk rencana jangka panjang mereka. Rencana tersebut adalah kemampuan untuk membuka aplikasi toko IV dimanapun dan tidak diperlukan konfigurasi khusus pada perangkat keras. Fitur-fitur baru mereka juga termasuk salah satu rencana mereka untuk kemampuan melakukan integrasi antar toko dan gudang.

Sistem informasi yang dimiliki Toko IV sudah mendukung untuk mencapai tujuan jangka panjang mereka, namun hal tersebut hanya mendukung pada sisi administratif. Toko IV masih belum memiliki aplikasi pendukung untuk sektor penjualan yang berhadapan langsung dengan pelanggan, seperti halnya untuk *sales people* Toko IV maupun untuk

partner (cabang) Toko IV. Hingga saat ini, segala proses transaksi masih dilakukan secara manual. Antara pemilik dan *sales people* melakukan interaksi satu sama lain melalui bertemu secara langsung, ataupun melalui telepon. Baik dalam hal penyebaran informasi dari pemilik Toko IV, informasi harga barang, komparabilitas barang untuk suatu mobil, maupun detail seputar produk-produk yang dijual pada Toko IV. Proses penjualan dengan para pelanggan lama maupun calon pelanggan berupa interaksi secara langsung tanpa adanya media penunjang. Seperti halnya saat *sales people* membuat janji dengan pelanggan, menjelaskan sebuah produk kepada pelanggan, maupun pelanggan lama yang memiliki potensi untuk melakukan transaksi ulang. Segala proses penjualan masih dilakukan dengan pola-pola tersendiri dari masing-masing individu.

Kondisi Toko IV yang memegang puluhan *brand* untuk setiap kategori produk yang dijual membuat para *salespeople* Toko IV mengalami kesulitan. *Sales people* Toko IV tidak mampu untuk memahami informasi akan setiap produk tersebut, baik dari segi karakteristik produk, kelebihan, kekurangan, maupun komparabilitas produk terhadap suatu mobil tertentu. Hal ini menyebabkan *sales people* Toko IV kesulitan dalam menawarkan suatu produk kepada pelanggan, maupun memberikan masukan akan berbagai kebutuhan pelanggan. Saat ini, kondisi ini diatasi dengan cara pemilik yang langsung turun tangan untuk memberikan penjelasan tentang suatu produk, hingga proses negosiasi jika diperlukan. Terdapat pula permasalahan komunikasi antara pemilik Toko IV dan *sales people* Toko IV, seperti halnya *sales people* tidak memahami ataupun mengetahui akan promosi yang sedang diadakan pada Toko IV. Informasi lain yang telah disampaikan oleh pemilik Toko IV juga seringkali terlupakan oleh *sales people* Toko IV.

Kondisi sektor penjualan yang seperti ini bertolak belakang dengan teknologi informasi yang telah diimplementasikan pada sektor administrasi. Dengan adanya sebuah aplikasi yang dapat mencatat tahap penjualan dengan pelanggan, maka data tersebut dapat dipelajari untuk meningkatkan pelayanan Toko IV. Dengan adanya sebuah aplikasi yang dapat memberikan visualisasi dan informasi seputar produk yang berkaitan, maka pelanggan dapat memperoleh informasi yang tepat. Dengan adanya fitur untuk mengingatkan *sales people* akan sebuah catatan bisnis, maka tingkat penjualan dan kualitas pelayanan Toko IV dapat meningkat.

Untuk menjawab kondisi pada sektor penjualan, pada skripsi ini dibuatlah sebuah aplikasi pendukung *salesforce automation* (SFA) *system*. Aplikasi ini ditujukan untuk orang internal dan *partner* Toko IV, yang dapat diakses setiap saat melalui *smartphone*. Guna mendukung dan meningkatkan sistem informasi yang sudah tersedia, aplikasi ini dapat terintegrasi dengan sistem informasi Toko IV yang baru.

2. TINJAUAN PUSTAKA

2.1 JavaScript

JavaScript, juga dikenal sebagai ECMAScript, adalah bahasa *script client-side* yang dikembangkan oleh Brendan Eich (Netscape Communications Corporation). Kode JavaScript dapat dimasukkan ke dalam *file* HTML atau disimpan ke dalam *file* eksternal berekstensi “.js”. Implementasi JavaScript memungkinkan sisi klien untuk berinteraksi dengan pengguna (*user*), berkomunikasi secara *asynchronous*, mengubah isi dokumen HTML yang ditampilkan, atribut HTML, dan *style* HTML (CSS), serta memvalidasi data sebelum di-*submit* [14]

2.2 AngularJS 1

AngularJS merupakan sebuah ekstensi dari HTML dengan beberapa atribut baru. [15] AngularJS merupakan sebuah Javascript *framework*, yang dimana ditambahkan pada halaman HTML dengan tanda `<script>`. AngularJS memperlebar atribut HTML dengan perintah, dan mengikat data pada HTML dengan berbagai ekspresi.

Beberapa ekstensi AngularJS pada HTML, antara lain:

- `ng-app`: untuk memberikan ekstensi AngularJS pada HTML.
- `ng-model`: untuk mengikat nilai pada HTML (`input`, `select`, `textarea`) kepada data aplikasi.
- `ng-bind`: untuk mengikat data aplikasi kepada HTML `view`.

Beberapa fitur dan benefit dari penggunaan AngularJS, antara lain [2]:

- *Cross platform*
 - a) Aplikasi *web* yang progresif. Menggunakan kapabilitas *platform web modern* untuk mengirim pengalaman seperti aplikasi. Performa tinggi, *offline*, dan tanpa instalasi.
 - b) *Native*. Membangun aplikasi *mobile native* dengan strategi dari Ionic Framework, NativeScript, dan React Native.
 - c) *Desktop*. Membuat aplikasi yang telah terpasang pada Mac, Windows, dan Linux dengan menggunakan metode Angular yang sama yang telah anda pelajari sebagai nilai tambah *web*; kemampuan untuk mengakses Native OS APIs
- Kecepatan dan performa
 - a) Pembuatan kode. Angular mengubah *template* anda menjadi kode yang telah dioptimasi secara tinggi untuk JavaScript Virtual Machine masa kini, memberikan anda seluruh keuntungan akan sebuah *framework*.
 - b) *Universal*. Menyajikan tampilan utama aplikasi anda pada `node.js`, `.NET`, dan server lainnya untuk *near-instant rendering* hanya pada HTML dan CSS.
 - c) Pembagian kode. Aplikasi Angular mengambil data secara cepat dengan komponen *router* baru, yang mengirim *automatic code-splitting*, sehingga pengguna hanya mengambil data kode yang dibutuhkan untuk render tampilan berdasarkan permintaan.
- Produktivitas
 - a) *Templates*. Dengan cepat membuat tampilan pengguna dengan mudah dan sintaks *template* yang kuat.
 - b) Angular CLI. *Command line tools*: mulai membangun dengan cepat, menambahkan komponen dan testing lalu secara instan diterapkan.
 - c) IDEs. Memperoleh *error* secara instan dan berbagai *feedback* dari *editor* dan IDE.
- *Full development story*
 - a) *Testing*. Dengan menggunakan Karma sebagai *unit tests*, anda dapat mengetahui jika anda memiliki hal yang rusak setiap kali anda simpan. Dan Protractor membuat skenario percobaan anda berjalan lebih cepat dan stabil.
 - b) *Animation*. Membuat Animasi dengan performa yang tinggi, kompleks, berkoreografi dengan kode yang sedikit melalui *Intuitive API* milik Angular.

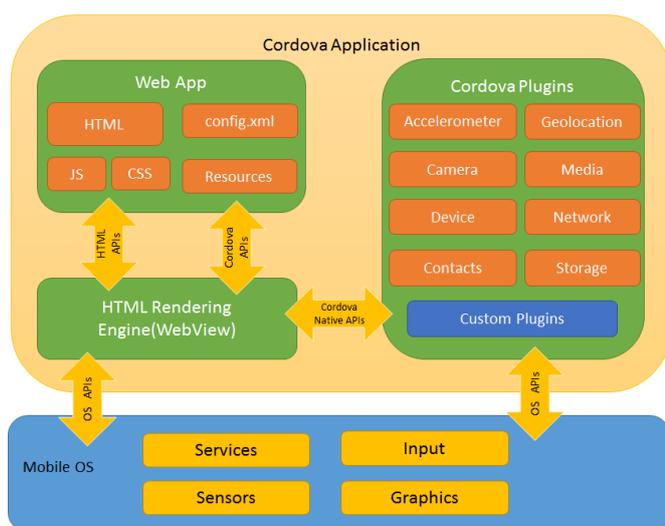
- c) Aksesibilitas. Membuat aplikasi yang dapat diakses dengan komponen *ARIA-enabled*, buku petunjuk, dan *all-y test infrastructure* yang telah terpasang. [2]

2.3 Cordova

Apache Cordova atau yang biasa disingkat dengan cordova merupakan sebuah *mobile-development framework* berbasis *open source*, yang dimana membantu untuk menggunakan teknologi standar pada website, HTML5, CSS3, dan Javascript untuk *cross-platform development*. Aplikasi dieksekusi dalam beberapa *wrappers* yang ditargetkan untuk setiap *platform*, dan bergantung pada *standard-compliant API Bindings* untuk mengakses kemampuan pada setiap perangkat seperti diantaranya sensor, data, status jaringan, dan lain sebagainya. Pada Gambar 1. ditunjukkan beberapa komponen untuk sebuah aplikasi Cordova. [3]

2.4 Ionic Framework 1

Ionic framework merupakan sebuah *framework* yang digunakan untuk membuat aplikasi *mobile* dengan teknologi *website*. *Framework* ini gratis berbayar dan *open source*, yang dimana menawarkan sebuah *library* yang berisi komponen-komponen HTML, CSS, dan JavaScript yang telah dioptimalisasikan untuk *mobile*, berbagai gerakan, dan berbagai alat untuk membangun aplikasi dengan tingkat interaksi yang tinggi. Ionic framework dibuat dengan menggunakan Sass dan dioptimalisasikan untuk AngularJS. [7]



Gambar 1. Arsitektur cordova

2.5 NoSQL

NoSQL adalah sebuah metode penyimpanan data generasi terbaru, yang dimana poin-poin utama yang ditekankan adalah sebagai sebuah *database* tanpa relasi, terdistribusi, *open-source*, dan dapat menjadi tambah luas secara horizontal.

Maksud awal dari pengembangan NoSQL adalah sebagai *modern web-scale database*, yang dimana pergerakan dimulai pada awal 2009 dan mulai berkembang dengan cepat. Seringnya karakteristik yang diaplikasikan seperti hal nya: *schema-free*, *easy replication support*, *simple API*, konsisten, data yang berskala besar, dan masih banyak lagi. [9]

Beberapa kelebihan dari database dengan teknologi NoSQL, sebagai berikut:

- Selalu tersedia: sebuah *database* yang dapat terus tersedia walau infrastruktur dari database tersebut ada yang mengalami kerusakan.
- Terdistribusi secara geografis: data yang aktif secara penuh, dapat diakses kapanpun saat dibutuhkan.
- *Low latency*: waktu respon yang dibutuhkan cukup cepat untuk *cloud applications*.
- Skalabilitas secara horizontal: dapat diprediksi skalanya untuk memenuhi kebutuhan data untuk keadaan sekarang maupun kedepan.
- Kemudahan pemasangan: tidak membutuhkan *object-relation mapping* atau *schema*.

3. ANALISIS DAN DESAIN

3.1 Data Flow Diagram (DFD)

DFD menggambarkan aliran data (*data flow*) di dalam sistem informasi, yaitu dari sumber data (input) ke penerima data (output). DFD juga menggambarkan semua proses dalam sistem informasi dari sudut pandang data (*viewpoint of data*) dan setiap jenis pengguna pada sistem informasi toko IV. Pembuatan DFD dimulai dengan pembuatan context diagram yang merupakan gambaran sistem informasi secara keseluruhan. Setelah pembuatan *context diagram*, dapat dibuat diagram level 0 dan level selanjutnya untuk menggambarkan subproses dan aliran data yang lebih detail. Aliran data yang ditentukan di dalam context diagram harus digambarkan di dalam semua diagram berikutnya [7]. *Data flow diagram aplikasi Salesforce Automation System mobile application Toko IV* dapat dilihat pada Gambar 2.

3.2 Desain Alur Aplikasi

3.2.1 Proses Pembuatan Objektiv

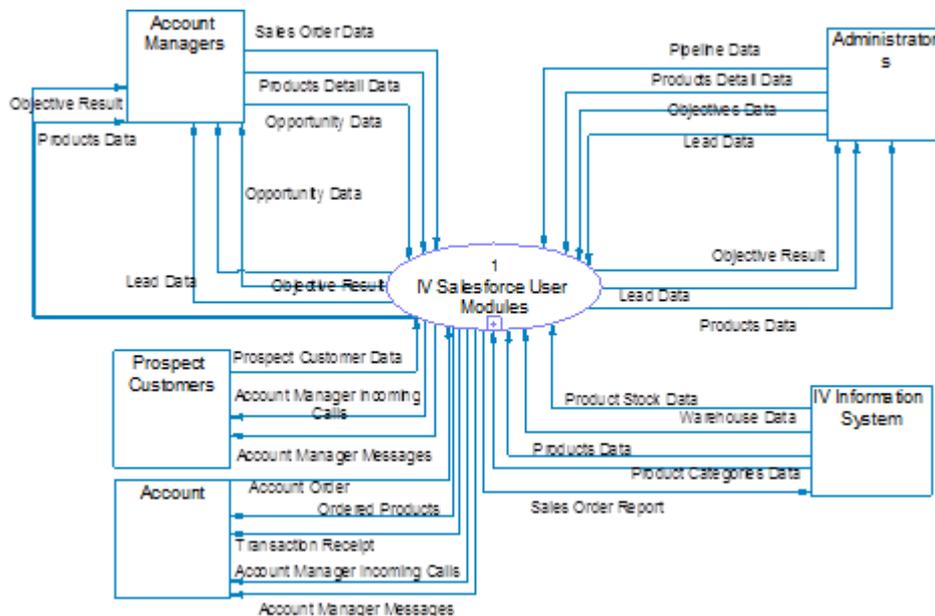
Proses pemberian tugas dimulai dari *administrator* membuat sebuah dorongan (*incentive*) yang dapat ditujukan untuk *partner* atau *account manager*. Tugas dibagi menjadi 4 jenis, antara lain: berdasarkan catatan manual (*note*), berdasarkan keuntungan bersih (*net profit*), berdasarkan keuntungan kotor (*gross profit*), dan berdasarkan produk (*products*). Setelah itu *administrator* menentukan *account manager* atau *partner* yang akan ditunjuk beserta informasi revelan lainnya. Data tersebut akan disimpan kedalam koleksi (*collection*) rewarding dan muncul di halaman utama *partner / account manager* yang ditunjuk.

3.2.2 Proses Pembuatan Lead

Proses lead dimulai dari *administrator / account manager* membuat sebuah *lead* baru. Hal yang membedakan antara kedua jenis akun ini adalah, jika akun tersebut adalah *administrator*, maka akun tersebut mampu membuat sebuah lead yang ditujukan untuk akun lainnya. Namun, jika akun tersebut adalah *account manager*, maka akun tersebut akan secara otomatis menunjuk diri sendiri sebagai penanggung jawab *lead* yang akan dibuat. Data *lead* yang telah dibuat akan disimpan di *database* dan muncul di halaman utama menu *lead management* milik akun yang ditugaskan.

Akun yang ditugaskan dapat membuka modul tersebut dan melakukan berbagai aktivitas dengan pelanggan yang dituju, antara lain: mencatat sejarah telepon dengan pelanggan, membuat sebuah tugas (*task*), membuat sebuah acara (*event*), berbagi informasi umum dengan anggota lainnya melalui *chat*, dan melihat detail informasi seputar pelanggan yang dituju.

lanjut ke fase berikutnya, mengubah status lead menjadi berhasil, dan mengubah status *lead* menjadi gagal. Akun



Gambar 2. Data Flow Diagram Salesforce Automation System mobile application Toko IV

dalam melakukan perubahan status dengan lanjut ke fase selanjutnya hingga fase terakhir. Jika sudah di fase terakhir, akun hanya memiliki pilihan untuk mengubah status lead tersebut menjadi berhasil atau gagal. Jika diubah menjadi berhasil, maka *lead* tersebut dapat dikonversi menjadi sebuah oportunitas.

3.2.3 Proses Pembuatan Oportunitas

Proses oportunitas dimulai dari *administrator* / *account manager* membuat sebuah oportunitas baru. Hal yang membedakan antara kedua jenis akun ini adalah, jika akun tersebut adalah *administrator*, maka akun tersebut mampu membuat sebuah oportunitas yang ditujukan untuk akun lainnya. Namun, jika akun tersebut adalah *account manager*, maka akun tersebut akan secara otomatis menunjuk diri sendiri sebagai penanggung jawab oportunitas yang akan dibuat. Data oportunitas yang telah dibuat akan disimpan di *database* dan muncul di halaman utama menu *opportunity management* milik akun yang ditugaskan.

Akun yang ditugaskan dapat membuka modul tersebut dan melakukan berbagai aktivitas dengan pelanggan yang dituju, antara lain: mencatat sejarah telepon dengan pelanggan, membuat sebuah tugas (*task*), membuat sebuah acara (*event*), memilih produk yang ditawarkan, berbagi informasi umum dengan anggota lainnya melalui *chat*, dan melihat detail informasi seputar pelanggan yang dituju. Perubahan fase pada oportunitas dibagi menjadi 3, yaitu: lanjut ke fase berikutnya, mengubah status oportunitas menjadi berhasil, dan mengubah status oportunitas menjadi gagal. Akun dalam melakukan perubahan status dengan lanjut ke fase selanjutnya hingga fase terakhir. Jika sudah di fase terakhir, akun hanya memiliki pilihan untuk mengubah status oportunitas tersebut menjadi berhasil atau gagal. Jika diubah menjadi berhasil, maka oportunitas tersebut dapat dikonversi menjadi sales order dan sales order detail.

3.2.4 Proses Konversi Oportunitas Menjadi Sales Order

Proses konversi oportunitas dimulai dari penugasan oportunitas yang telah selesai dan siap untuk di konversi. Jika setuju untuk dikonversi, maka data *sales order* dan *sales order detail* akan disimpan pada 2 jenis *database*, yang pertama

adalah *SQL database* sebagai *database* utama dan dilanjutkan dengan penyimpanan pada *NoSQL database* sebagai *database* untuk membantu mendukung aplikasi *mobile* dan proses *rewarding*. Setelah selesai disimpan ke dalam *database*, akan melakukan pengecekan dengan *rewarding* yang ditujukan ke akun tertentu. Jika kuota yang ditargetkan telah memenuhi, maka akan secara otomatis dilakukan perubahan pada *reward* yang diberikan sebelumnya.

4. HASIL DAN PEMBAHASAN

4.1 Menu Rewarding

Menu *rewarding* menunjukkan *task* / *incentive* yang telah

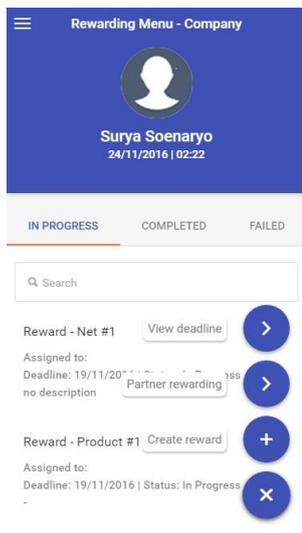
dibuat oleh akun dengan tipe *administrator* dan ditujukan ke akun tertentu. Terdapat fitur untuk melihat tanggal *deadline* yang dapat diakses melalui tombol lingkaran di sisi kanan bawah. Untuk akun dengan tipe *administrator* akan ditampilkan menu pembuatan oportunitas pada tombol lingkaran di kanan bawah untuk membuat *task* / *incentive* baru. Desain tampilan menu *rewarding* pada halaman perusahaan dan *partner* dapat dilihat pada Gambar 3 dan Gambar 4.

4.2 Menu Informasi Utama

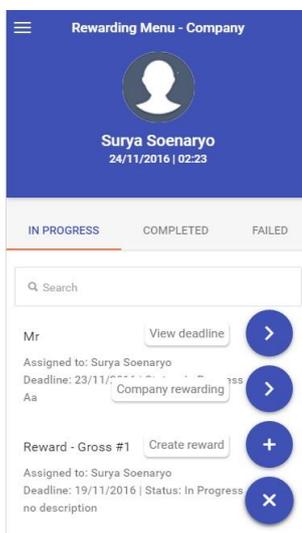
Menu informasi utama menunjukkan informasi yang tersebar pada toko tertentu. Pada setiap informasi, terdapat informasi jumlah pengguna yang telah melihat dan jumlah komentar pada informasi tersebut. Pengguna dapat membuat sebuah informasi baru maupun melakukan *group chat*. Desain tampilan menu informasi utama dapat dilihat pada Gambar 5.

4.3 Menu Lead

Menu *lead* menunjukkan penugasan dari *administrator* ataupun



Gambar 3. Desain Tampilan Menu *Rewarding* Perusahaan



Gambar 4. Desain Tampilan Menu *Rewarding Partner*



Gambar 5. Desain Tampilan Menu *Informasi Utama*

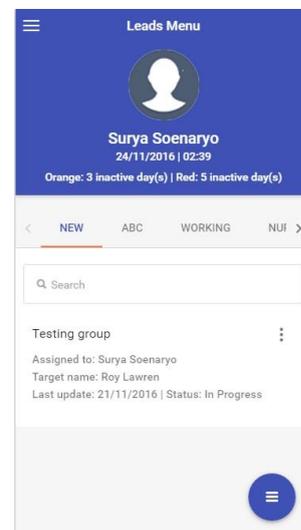
account manager untuk melakukan pendekatan ke *account* yang dituju hingga muncul *oportunitas* untuk melakukan transaksi. Desain tampilan menu *lead* dapat dilihat pada Gambar 6.

4.4 Menu *Oportunitas*

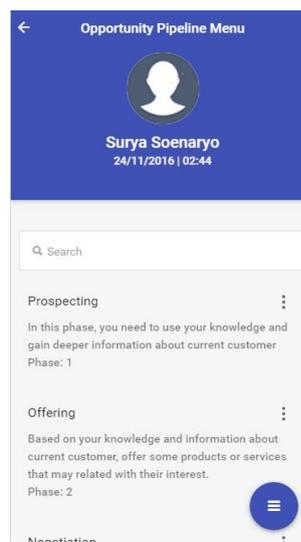
Menu *oportunitas* menampilkan data *oportunitas* yang sedang berlangsung, selesai, gagal, maupun yang telah dikonversi. Di dalam menu lingkaran di sudut kanan bawah, terdapat tombol untuk mengakses pengaturan alur (*pipeline*) *oportunitas* dan pembuatan *oportunitas* baru. Desain tampilan menu alur (*pipeline*) *oportunitas* dapat dilihat pada Gambar 7. Desain tampilan detail *oportunitas* dapat dilihat pada Gambar 8.

4.5 Menu *Statistik*

Menu *statistik* menampilkan performa penjualan. Pada halaman menu *statistik*, ditampilkan 5 jenis diagram utama, yaitu: *leads performance*, *opportunities performance*, *rewards performance - general*, *rewards performance - partner*, dan *rewards performance - shop*. Terdapat *menu* untuk melihat *statistik* per toko, *salespeople*, dan pelanggan. Menu tersebut dapat diakses pada menu di bagian akhir halaman / melalui tombol lingkaran di sisi kanan bawah. Desain tampilan menu *statistik* dapat dilihat pada Gambar 9.



Gambar 6. Desain Tampilan Menu *Lead*



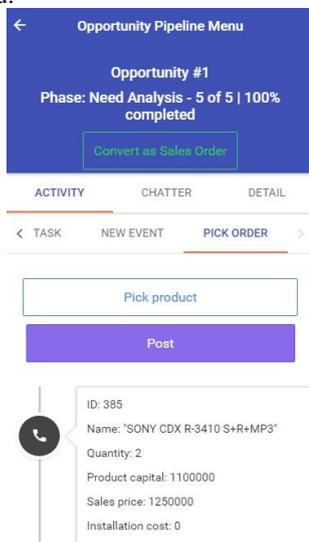
Gambar 7. Desain Tampilan Menu Alur (*Pipeline*) *Oportunitas*

5. KESIMPULAN DAN SARAN

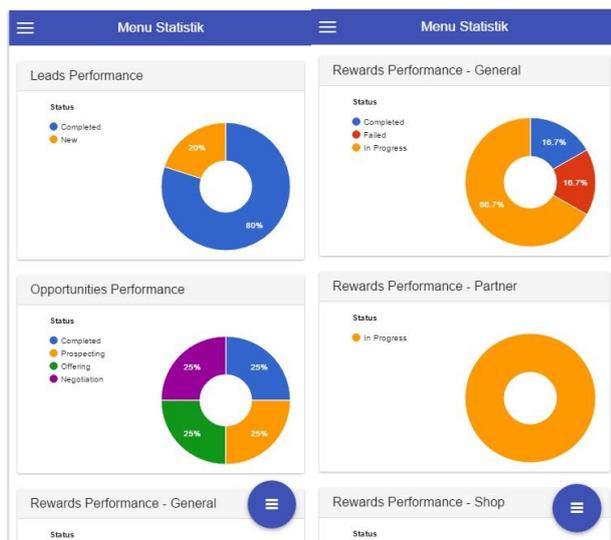
5.1 Kesimpulan

Dari hasil perancangan dan pembuatan aplikasi *Salesforce Automation System* untuk Toko IV berbasis Android dapat diambil kesimpulan antara lain:

- Proses pembelajaran teknologi dan *debugging* memakan waktu cukup lama karena proses perpindahan jenis *data* dari *database* satu ke *database* lain berbeda, sehingga perlu dilakukan modifikasi terlebih dahulu.
- Aplikasi masih awam oleh kondisi *salespeople* maupun *administrator* sekarang ini, sehingga perlu diadakan pelatihan dan panduan penggunaan terlebih dahulu.
- Fitur *newsfeed* dan *live chatting* membantu pengguna untuk berbagi informasi secara cepat dan fitur *mark* pada *newsfeed* sangat berguna untuk mengetahui apakah *account manager* sudah mengetahui informasi tersebut atau tidak.
- Fitur *product management* cukup membantu *user* untuk mencari maupun menambahkan catatan seputar produk yang dituju.



Gambar 8. Desain Tampilan Detail Oportunitas



Gambar 9 Desain Tampilan Menu Statistik

- Berdasarkan hasil kuesioner, 84% pengguna menilai tampilan aplikasi baik, 100% pengguna menilai fitur di *account management* cukup baik, 100% pengguna menilai

fitur di *rewarding management* cukup baik, 72% pengguna menilai fitur di *newsfeed* baik, 100% pengguna menilai fitur di *lead management* cukup baik, 100% pengguna menilai fitur di *opportunity management* cukup baik, 64% pengguna menilai fitur di *products management* baik. 100% pengguna menilai fitur di *statistics management* cukup baik, 64% pengguna menilai kemudahan menggunakan aplikasi baik, 72% pengguna menilai kesesuaian dengan kebutuhan baik, 68% pengguna menilai keseluruhan aplikasi baik.

- Hasil kuesioner menyimpulkan bahwa pengguna cukup senang dengan tampilan aplikasi, namun masih awam dengan fitur-fitur yang tersedia didalamnya.

5.2 Saran

Saran yang dapat diberikan untuk menyempurnakan dan mengembangkan aplikasi ini lebih lanjut antara lain:

- Pembuatan panduan penggunaan aplikasi dan penjelasan akan kosakata yang tidak umum, sehingga pengguna dapat lebih memahami fitur-fitur yang ada.
- Menstabilkan aplikasi agar lebih nyaman digunakan.
- Memperbaiki tipe dan *field database* utama, karena masih terdapat banyak *field* kosong dan tidak digunakan, sehingga fungsi tidak dapat berjalan dengan maksimal.
- Membuat modul pemesanan instan, sehingga aplikasi dapat digunakan untuk pemesanan produk tanpa butuh alur yang panjang.

6. DAFTAR REFERENSI

- [1] Arndt, D. A. & Harkins, J. 2013. *A framework for configuring sales support structure*, Journal of Business & Industrial Marketing, Vol. 28 Iss: 5, pp.432 – 443.
- [2] Angular. n.d. Features and Benefits – Angular 2. URI=<https://angular.io/features.html>
- [3] Apache Cordova. n.d. Architecture overview of Cordova platform. Retrieved March 14, 2016, URI=<http://cordova.apache.org/docs/en/latest/guide/overview/>
- [4] Buttle, F. & Maklan, S. 2015. Customer Relationship Management. Concepts and technologies (3rd ed.). Third Avenue, New York, NY: Routledge.
- [5] Encyclopædia Britannica. n.d. *Information System*. Retrieved August 8, 2015, URI=<http://www.britannica.com/topic/information-system>
- [6] Golden, B. n.d. Boris Golden – Academic Homepage. Retrieved October 9, 2016, URI=http://www.lix.polytechnique.fr/~golden/systems_architecture.html
- [7] Ionic Framework. n.d. *Advanced HTML5 Hybrid Mobile App Framework*. Retrieved March 14, 2016, URI=<http://ionicframework.com/>
- [8] Kendall, K. E. & Kendall, J. E. 2011. *Systems Analysis and Design* (8th ed.). One Lake Street, Upper Saddle River, New Jersey: Prentice Hall.
- [9] NOSQL Databases. n.d. *NOSQL Databases*. Retrieved March 14, 2016, URI=<http://nosql-database.org/>
- [10] Odoo CRM. n.d. The #1 Open Source CRM Software | Odoo. Retrieved September 25, 2016, URI=<https://www.odoo.com/page/crm>
- [11] Román, S. & Rodríguez, R. 2015. *The influence of sales force technology use on outcome performance*, Journal of

- [12] Business & Industrial Marketing, Vol. 30 Iss: 6, pp.771 – 783.
- [13] Salesforce. n.d. *CRM Cloud Software & Cloud Computing*. Retrieved May 16, 2016, URI= <https://www.salesforce.com/>
- [14] Sinisalo, J. et al. 2015. *Barriers to the use of mobile sales force automation systems: a salesperson's perspective*, Journal of Systems and Information Technology, Vol. 17 Iss: 2, pp.121 - 140.
- [15] W3Schools. n.d. *JavaScript Introduction*. Retrieved August 9, 2015, URI= http://www.w3schools.com/js/js_intro.asp
- [16] W3Schools. n.d. *AngularJS Introduction*. Retrieved March 14, 2016, URI= http://www.w3schools.com/angular/angular_intro.asp