

Integrasi dan Sinkronisasi Program Sikasir Sebagai Proses Monitoring Menggunakan Platform Web

Gagas Julio¹, Eliyani²

Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana
Jl. Raya Meruya Selatan, Kembangan, Jakarta 11650

juliogagas95@gmail.com¹,
eliyani@mercubuana.ac.id²

Abstract – Central Bureau of Statistics Indonesia stated in 2012 the number Small and Medium Enterprises in Indonesia reached 56,534,592. The biggest problem for SMEs in Indonesia is lack of system that can be used to monitor business easily without having to come to the outlet. This research is intended to develop a Web Apps application for SMEs management based on the concept of Single Page Application named SIKASIR. This application can be accessed by three actors namely owner as admin, manager, and cashier. Token is needed to run the application, the token will expire every two hours. Some menus in this application are management of stock, staff, and partners and reports. The Waterfall method was used to develop the application using JSON programming language and Angular JS as a framework. Application was tested using Black Box method. The result of the testing showed that the url of the application could be accessed successfully, and all menus run well. The system of Web Apps can be used as data providers and data monitoring wherever and whenever with high accuracy and updated.

Keywords: Web Apps, JSON, Single Page Application, Angular JS

Abstrak - Badan Pusat Statistik Indonesia menyatakan pada tahun 2012 jumlah Usaha Kecil Menengah di Indonesia mencapai 56.534.592 dan terus meningkat hingga sekarang. Permasalahan terbesar bagi Usaha Kecil Menengah di Indonesia adalah belum adanya sistem untuk memonitoring bisnis dengan mudah tanpa harus datang ke outlet yang telah ada. Penelitian ini dimaksudkan untuk mengembangkan aplikasi manajemen bagi UMKM dengan konsep *Web Apps* berbasis *Single Page Application* yang diberi nama SIKASIR. Pada aplikasi ini terdapat tiga aktor yang dapat berinteraksi dengan sistem yaitu pemilik usaha sebagai admin, manajer usaha, dan kasir. Untuk dapat mengakses aplikasi ini dibutuhkan token yang dihasilkan oleh server dan expired dalam waktu dua jam. Beberapa menu dalam aplikasi ini antara lain manajemen stok, manajemen karyawan, manajemen rekanan, dan manajemen laporan harian. Pengembangan aplikasi menggunakan konsep *Waterfall* dengan menggunakan bahasa pemrograman JSON untuk merepresentasikan data dan Angular JS sebagai *framework*. Pengujian aplikasi dilakukan dengan metode *blackbox* dan menunjukkan bahwa url aplikasi dapat diakses dan semua fitur pada sistem dapat berjalan dengan baik. Sistem *Web Apps* ini dapat digunakan sebagai sarana penyedia data dan monitoring data di manapun dan kapan pun sehingga mendapatkan informasi data dengan akurat karena informasi yang tersedia senantiasa diperbaharui.

Kata Kunci: Web Apps, JSON, Single Page Application, Angular JS

I. PENDAHULUAN

Saat ini internet dapat diakses dengan mudah juga dengan menggunakan perangkat mobile. Fasilitas internet *cloud service* bisa dimanfaatkan dengan maksimal sehingga menjadi lebih terarah dan efisien.

Salah satu pemanfaatan internet adalah membantu para pemilik Usaha Mikro, Kecil, dan Menengah (UMKM) khususnya dalam melakukan manajemen usahanya. Badan Pusat Statistik Indonesia menyatakan pada tahun 2012 jumlah UMKM di Indonesia mencapai 56.534.592 [1]. Seiring meningkatnya UKM di Indonesia masih terdapat masalah yang dihadapi para UMKM. Para penggiat UMKM sering kerepotan dalam melakukan transaksi penjualan, menghitung pendapatan, mengelola harga barang, menghitung barang terjual, menghitung sisa stok, dan membuat laporan harga. Dalam melakukan transaksi penjualan, UMKM masih menggunakan cara manual dengan menggunakan kalkulator, nota, dan buku besar.

Agar penggiat UMKM dapat melakukan manajemen usahanya dengan baik, penelitian ini dimaksudkan untuk membuat Web Service berbasis *Single Page Application* (SPA) dengan menggunakan framework *Angular JS* sebagai metodenya. Data disimpan di *cloud server* menggunakan format JSON yang ditentukan oleh server.

SPA adalah istilah untuk aplikasi berbasis web, yang menggunakan satu halaman web saja sebagai tampilan dari aplikasinya. Semua aksi klik atau pun penyajian data tidak akan membuat halaman secara utuh untuk dimuat ulang (*reload*), tetapi hanya sebagian saja yang diupdate dari server atau dari hasil proses aplikasi di sisi klien [2].

Dengan demikian, *Web Apps* dengan metode SPA dapat meringankan kerja sistem karena tidak banyak *request* data yang dipakai. Di samping itu, metode SPA dapat memberikan kenyamanan kepada pengguna karena sistem akan menampilkan data sesuai *request*. Pengguna tidak memuat (*load*) data pada satu halaman akan tetapi hanya pada beberapa layout yang diakses.

Cloud Server akan menampilkan file JSON setelah pengguna mengakses dengan link atau URL tertentu yang sudah di atur sebelumnya. Setelah mendapatkan format JSON, data akan diolah sesuai kebutuhan serta akan ditampilkan dalam bentuk *Web Apps* berbasis SPA menggunakan *Frame Work Angular JS*.

AngularJS adalah kerangka struktural untuk aplikasi web dinamis dan kerangka untuk membuat SPA. AngularJS memungkinkan pengguna menggunakan HTML sebagai bahasa pemrograman yang dipakai dan memungkinkan pengguna memperluas sintaks HTML untuk mengekspresikan komponen aplikasi yang dibuat oleh pengguna dengan jelas dan ringkas. Data *binding* AngularJS dan *dependency injection* AngularJS dapat memperingkas proses *coding*. Semua proses tersebut terjadi dalam browser sehingga AngularJS mampu menjadi pasangan yang ideal dengan teknologi server. AngularJS adalah salah satu bentuk dari HTML bila ingin digunakan untuk mendesain aplikasi. HTML adalah bahasa yang cukup baik untuk dokumen statis [3] dan dapat menghasilkan halaman dinamis tanpa harus membuat banyak halaman web [4].

Untuk proses mengambil data, menginput, dan mengupdate data, aplikasi ini tidak dapat langsung mengambil data dari *database*, melainkan harus melalui API (*Application Programming Interface*). API merupakan bahasa dan format pesan yang digunakan untuk berkomunikasi dengan sistem operasi atau program pengendalian lainnya seperti DBMS (*Data Base Manajemen Sistem*) atau komunikasi protokol. Format API yang digunakan pada aplikasi ini berbentuk JSON (*Javascript Object Notation*).

Aplikasi ini bertujuan sebagai sarana untuk memudahkan pembukuan transaksi agar meminimalisir terjadinya kesalahan, sebagai media untuk melakukan pembukuan transaksi, serta melakukan pengujian aplikasi pembukuan transaksi apakah lebih efisien menggunakan *cloud computing* atau dengan cara manual. Dengan adanya aplikasi ini diharapkan dapat memberi manfaat kepada pelaku bisnis UMKM, agar para pelaku bisnis UMKM dapat fokus dengan perkembangan bisnisnya dan tidak perlu repot mengurus laporan dan manajemen produk sehingga kerja menjadi lebih efisien.

JSON adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dihasilkan oleh komputer. 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, PHP, dan lain-lain. JSON adalah bentuk data asli JavaScript yang tidak membutuhkan API khusus untuk memproses data. Hal itu membuat JSON sebagai sebuah format yang menjanjikan untuk pertukaran data pada aplikasi berbasis web [5].

JavaScript adalah bahasa yang luas dan fleksibel untuk mengembangkan aplikasi cross-platform, JavaScript memiliki keterbatasan tertentu ketika mengimplementasikan pengolahan kompleks dan background processing. Kadangkadang perlu untuk menggunakan kode native untuk melakukan sebuah pekerjaan [6].

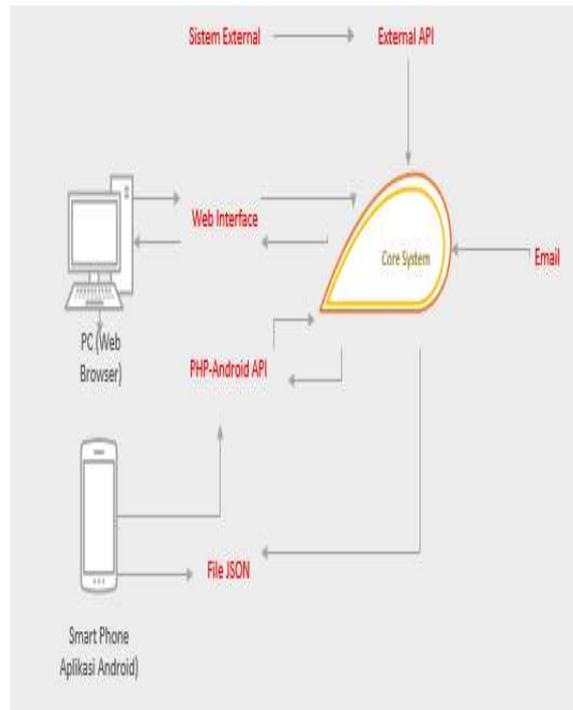
Karena tidak menggunakan format bahasa pemrograman khusus dan mudah dibaca manusia, maka JSON sering digunakan untuk pertukaran data antar komputer. Dalam sistem yang dibangun, JSON digunakan untuk menyimpan informasi dari aplikasi berbasis web yang kemudian diunduh perangkat android.

II. PERANCANGAN SISTEM

Perancangan sistem merupakan tahap awal dari perancangan perangkat lunak. Perancangan ini dilakukan untuk mengetahui kondisi sistem secara umum. Analisa yang dilakukan dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahapan pemodelan dalam analisis tersebut antara lain mengidentifikasi aktor, pembuatan diagram *use case*, diagram *sequence* dan diagram *activity*.

3.1. Arsitektur Sistem

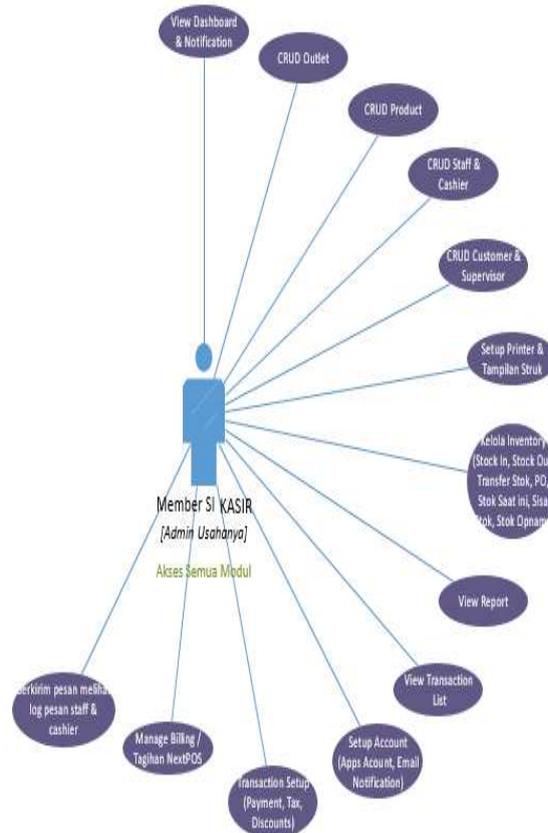
Arsitektur sistem disajikan pada Gambar 1. Berdasarkan Gambar 1, pengguna dapat mengirim dan menerima informasi melalui website yang diakses menggunakan PC atau melalui aplikasi android. Aplikasi ini dibuat menggunakan *Frame Work AngularJS* untuk metode SPA. Proses Ambil, Input, Perbaharui, dan Hapus data menggunakan format JSON yang diberikan melalui server, dan *client* menggunakan format JSON tersebut untuk selanjutnya akan di proses melalui server.



Gambar 1. Rancangan Arsitektur Sistem

3.2. Use Case

Diagram *use case* merupakan pemodelan untuk kelakuan sistem yang dibangun. Diagram ini menggambarkan interaksi antara satu atau lebih aktor terhadap sistem. Diagram *use case* untuk aplikasi SIKASIR disajikan pada Gambar 2.



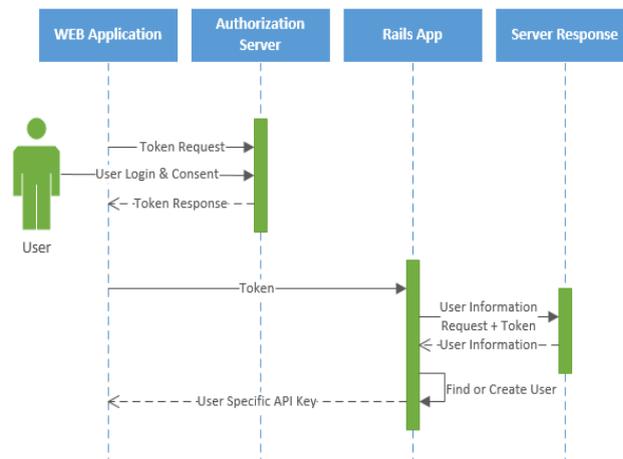
Gambar 2. Use Case Diagram

Dalam sistem ini terdapat 3 aktor yang berinteraksi dengan sistem, antara lain:

1. Admin (Owner), adalah orang yang berhak untuk manajemen pengguna sistem, View dashboard & Notifacation, CRUD outlet, CRUD Product, CRUD Staf dan cashier ,CRUD Customer & Supervisor, Setup Printer & Tampilan Struk, Kelola Inventory, View Report, View Transaction List, Setup Account, Transaction Setup, Manage Billing, dan berkirim pesan melihat log pesan staf & cashier.
2. Manager, adalah orang yang berhak untuk View dashboard & Notifacation , CRUD Product, CRUD Staf dan cashier ,CRUD Customer & Supervisor, Setup Printer & Tampilan Struk, Kelola Inventory, View Report, View Transaction List, Transaction Setup dan berkirim pesan kepada Staff Admin dan Staff cashier.
3. Cashier, adalah orang yang berhak untuk View dashboard & Notifacation, melakukan Point Of Sale, CURD Order List, CRUD Customer, Kelola Inventrory, kelola kas, View Transaction List, Export laporan harian, berkirim pesan kepada cashier admin dan cashier staff.

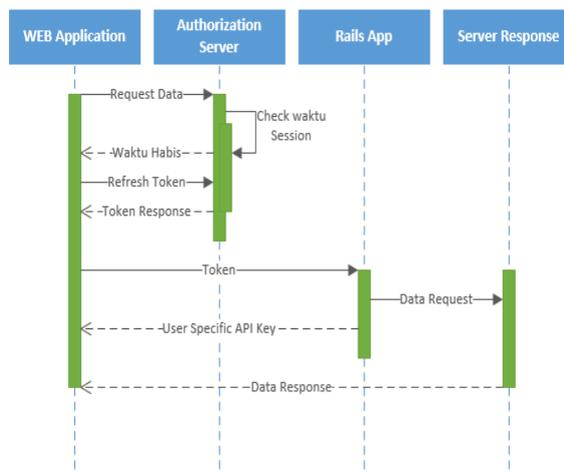
3.3. Squence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan pesan yang diletakkan di antara obyek-obyek di dalam *use case*. Komponen utama *sequence diagram* terdiri dari obyek yang dituliskan dengan kotak segi empat bernama. Message diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progres vertikal. *Sequence diagram* untuk aplikasi ini disajikan pada Gambar 3.



Gambar 3. Squence Diagram alur aplikasi login

Sequence diagram login aplikasi ini menggambarkan pengguna sedang melakukan login menggunakan *username* dan *password*. Ketika *username* dan *password* ini benar terdapat di dalam database, server akan mengirimkan token yang berbentuk ID unik. Token ini harus disimpan di dalam header aplikasi yang berfungsi sebagai identitas masuk agar dapat mengakses data yang dibutuhkan oleh aplikasi SIKASIR. Token ini dapat *expired* dalam setiap waktu 2 jam, dan akan terus berganti ID unik. *Sequence diagram* ntuk tetap bisa mengakses data setelah login disajikan pada Gambar 4.

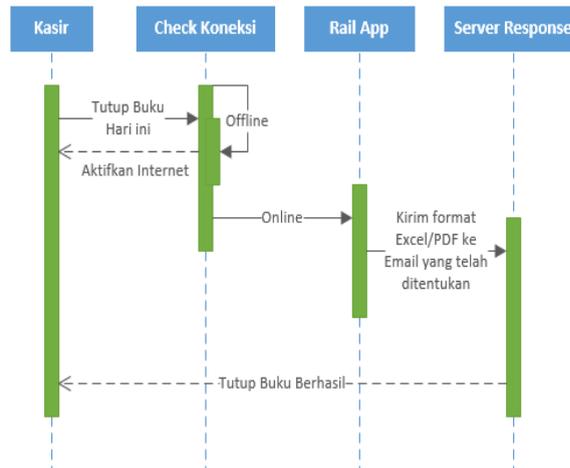


Gambar 4. Squence Diagram refresh token

Token yang diberikan oleh server waktunya dibatasi setiap 2 jam sekali. Ketika lebih dari waktu tersebut dimulai dari pertama kali akses maka token tersebut akan bersifat *expired*. Sehingga aplikasi harus *refresh* ulang

token tersebut untuk disimpan kembali kedalam header aplikasi dan untuk keperluan get, post, input, delete data dari server.

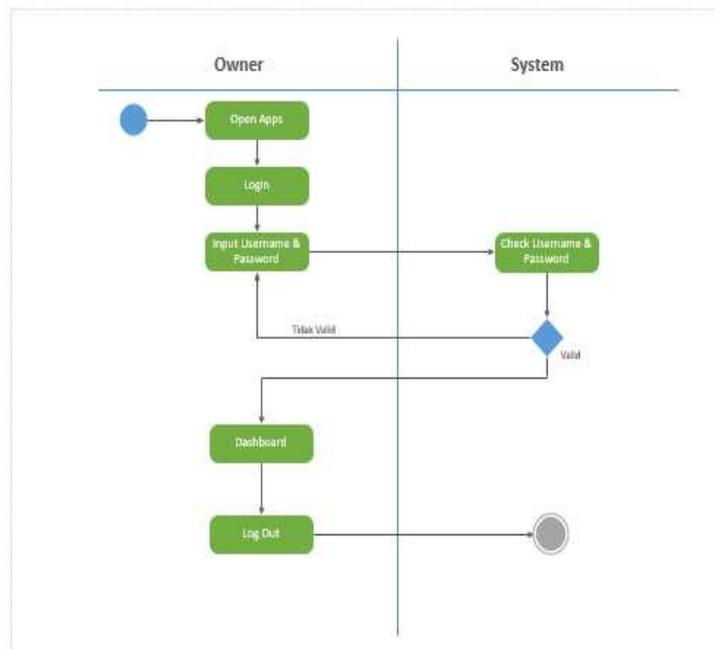
Setiap Pemilik Usaha dapat melihat daftar laporan harian transaksi yang telah dilakukan. Kasir yang bertugas menutup transaksi mengirimkan laporan ketika *outlet* atau toko sudah ditutup dan melakukan *export* atau kirim laporan harian kepada email yang *account* emailnya sudah diatur di dalam sistem. Aplikasi ini akan mengecek koneksi internet yang terhubung, apabila tidak ada koneksi internet, maka tutup buku atau tutup transaksi harian tidak dapat di lanjutkan sampai terdapat koneksi internet. *Sequence diagram* laporan harian disajikan pada Gambar 5.



Gambar 5. Squence Diagram laporan harian

3.4. Activity Diagram

Diagram aktivitas menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, keputusan yang mungkin terjadi, dan bagaimana suatu aktivitas berakhir. Setiap langkah adalah keadaan dalam melakukan sesuatu. Gambar 6 adalah *activity diagram* dari aplikasi yang dibuat dengan login sebagai pemilik usaha.

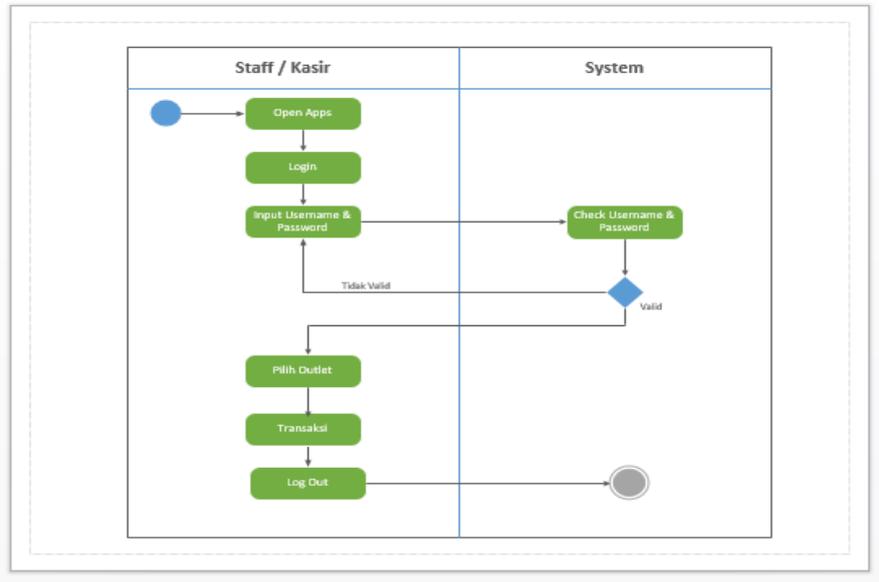


Gambar 6. Activity Diagram Login Pemilik Usaha

Saat pemilik usaha atau admin login kedalam aplikasi, sistem akan mengecek apakah *username* dan *password* yang dimasukkan sudah benar atau belum. Ketika *username* dan *password* yang dimasukkan sudah benar, admin akan langsung menampilkan *dashboard*. *Dashboard* berfungsi untuk memonitoring laporan harian, mingguan,

ataupun bulanan serta untuk manajemen produk, stok, karyawan, dan keuangan. Transaksi berfungsi apabila admin bertindak sebagai kasir untuk melakukan input transaksi.

Berbeda dengan login admin, login staf atau kasir ini langsung masuk ke menu *point of sale* atau transaksi. Jika staf atau kasir ini mempunyai hak akses lebih dari 1 outlet, maka staff atau kasir sebelum masuk ke menu transaksi harus memilih outlet terlebih dahulu. Diagram aktivitas login staf disajikan pada Gambar 7.



Gambar 7. Activity Diagram Login Staff

III. IMPLEMENTASI DAN PENGUJIAN

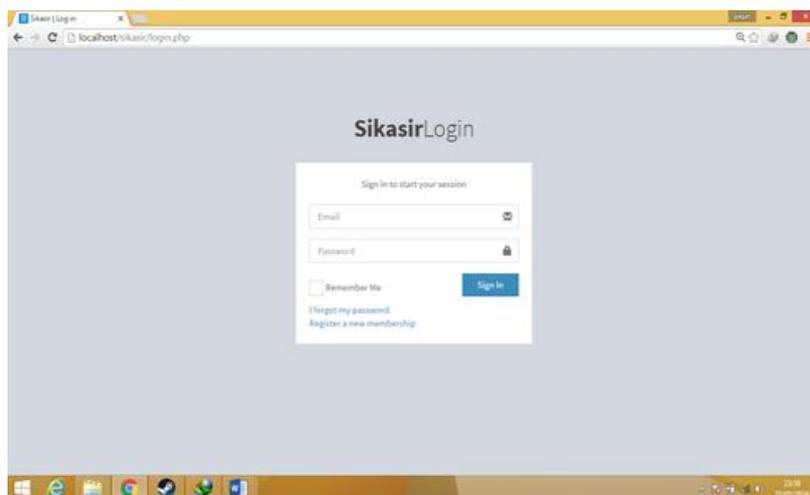
Setelah dilakukan proses perancangan kemudian dilakukan implementasi dan pengujian.

3.1 Implementasi

Untuk mengimplementasi sistem yang telah dirancang, maka data yang dapat diakses dalam bentuk JSON haruslah stabil dan sudah diletakkan ke dalam *cloud server*. Apabila terjadi kesalahan dalam sisi server maka yang bertugas pada bagian web services inilah yang harus memperbaiki sampai format data JSON yang diberikan stabil kembali. Implementasi sistem merupakan tahap meletakkan sistem sehingga siap untuk dioperasikan.

3.1.1 Menu Login

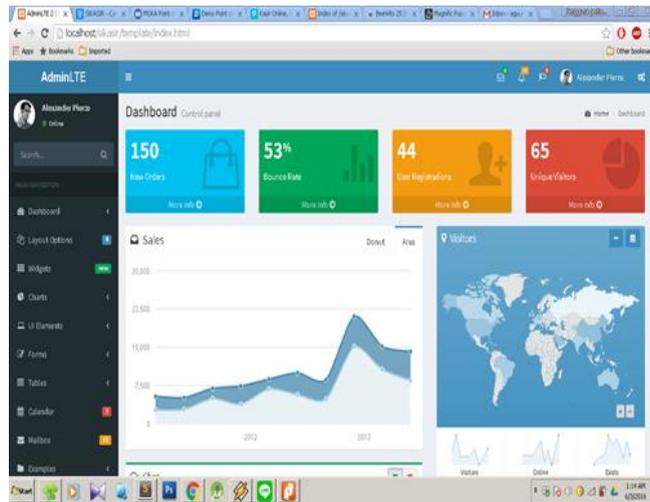
Form Login pada Web Base ini disajikan pada Gambar 8. Form login dijalankan oleh pemilik/karyawan/kasir dengan menggunakan Browser. Untuk dapat mengakses aplikasi ini, pemilik/karyawan/kasir melakukan proses login terlebih dahulu dengan memasukkan *email* dan *password* sesuai dengan *user id* yang dimilikinya kemudian sistem akan merespon berdasarkan hak akses yang sudah ditentukan.



Gambar 8. Login

3.1.2 Dashboard

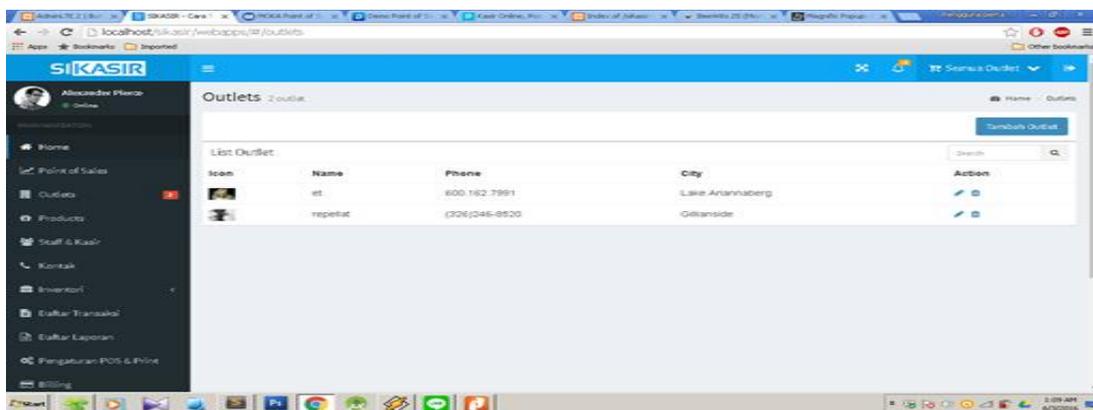
Tampilan Dashboard ini adalah tampilan awal ketika user berhasil login dan berisi laporan-laporan dalam bentuk grafik. Tampilan Dashboard disajikan pada Gambar 9.



Gambar 9. Dashboard

3.1.3 Manajemen Outlet

Tampilan Outlet seperti disajikan pada Gambar 10 adalah untuk manajemen outlet tampil, tambah, perbaharui, dan hapus outlet.



Gambar 10. Manajemen Outlet

3.1.4 Manajemen Produk

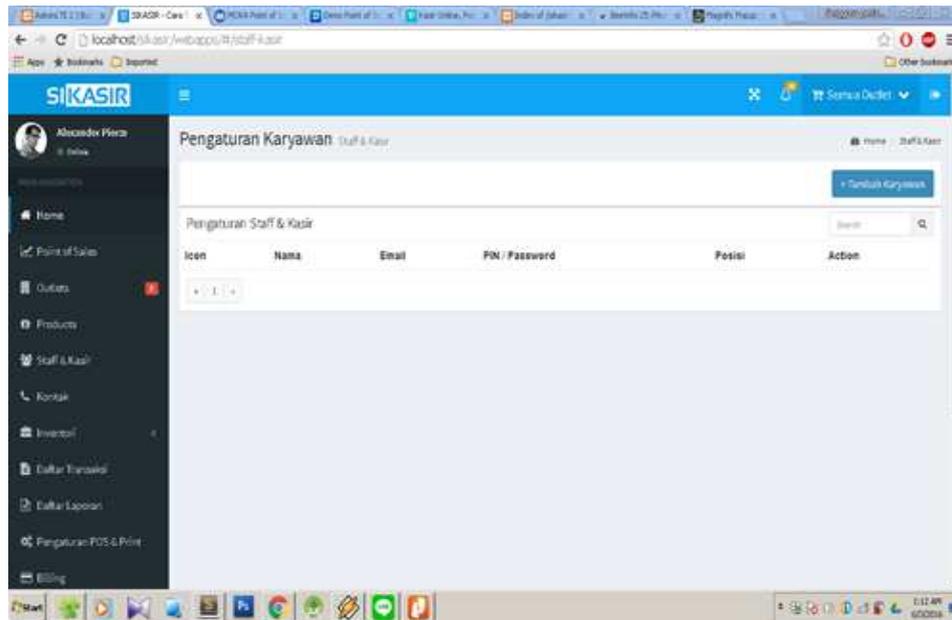
Tampilan Manajemen Produk disajikan pada Gambar 11 berisi tentang apa saja produk, kategori, dan variant yang di jual di setiap outletnya serta admin dapat menambahkan produk baru.



Gambar 11. Manajemen Produk

3.1.5 Manajemen Karyawan

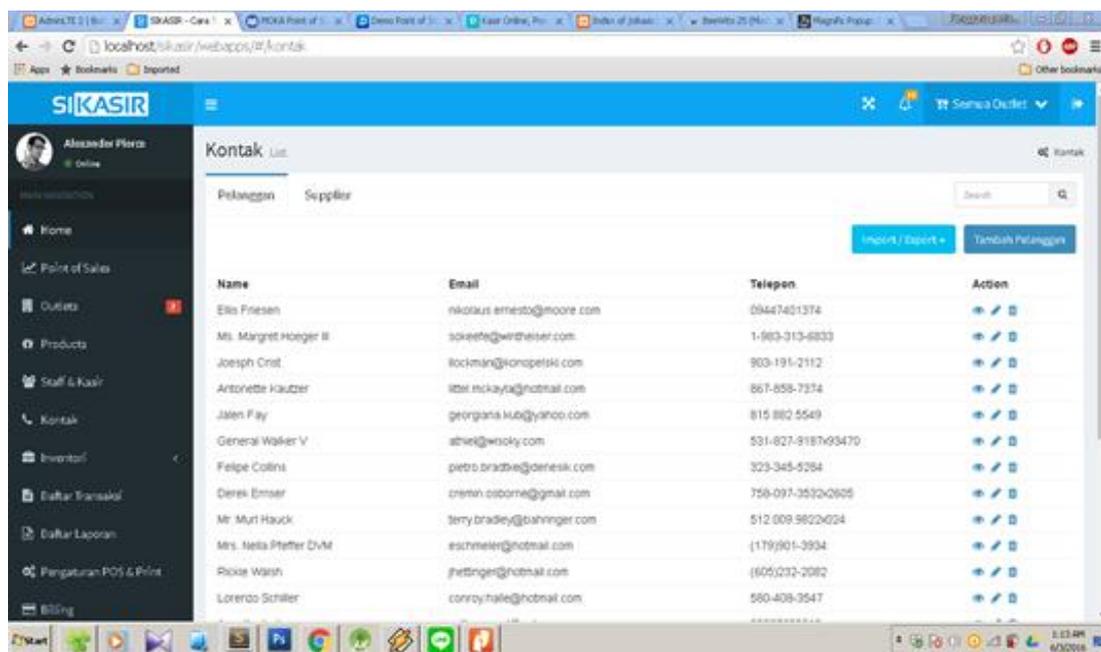
Tampilan Manajemen Karyawan seperti disajikan pada Gambar 12 berisi tentang data staf dan kasir yang bekerja dan admin dapat *mengcreate, update, dan delete* data staf dan kasir yang ada.



Gambar 12. Manajemen Karyawan

3.1.6 Manajemen Kontak

Tampilan kontak disajikan pada Gambar 13 berisi tentang daftar kontak pelanggan dan supplier.



Gambar 13. Manajemen Kontak

3.2 Pengujian dan Fungsionalitas Web Apps

Proses pengujian ini dilakukan dengan cara membuka browser dan menguji menu-menu yang ada. Pengujian dengan menggunakan metode *blackbox*.

3.2.1 Lingkungan Pengujian

Integrasi dan sinkronisasi program SIKASIR sebagai proses monitoring menggunakan platform web diuji coba di browser Google Chrome 50.0.2661.103.

3.2.2 Hasil Pengujian

Hasil pengujian disajikan pada Tabel 1. Beberapa item yang diuji meliputi akses ke situs SIKASIR, login sesuai hak akses, memilih dashboard, memilih menu outlet, memilih menu produk, memilih menu karyawan, memilih menu kontak, dan memilih menu stok. Semua fitur dapat berjalan dengan baik dengan hasil sesuai seperti yang diharapkan.

TABLE 1
HASIL PENGUJIAN

Nama Pengujian	Hasil yang diharapkan	Keterangan
Membuka Browser lalu ketikkan url sikasir.com/sikasir	Tampil Halaman Login	Sesuai
Login ke dalam Aplikasi	Menampilkan pilihan sesuai hak akses	Sesuai
Memilih Dashboard	Menampilkan data-data laporan, dan menu untuk manajemen.	Sesuai
Memilih Menu Outlet	Menampilkan data Outlet yang sudah ada dan menambahkan outlet yang baru di buat.	Sesuai
Memilih Menu Produk	Menampilkan data produk / varian serta menambahkan produk baru.	Sesuai
Memilih Menu Karyawan	Menampilkan data karyawan dan meng create,update,dan delete data karyawan	Sesuai
Memilih Menu kontak	Menampilkan kontak pelanggan dan supplier serta dapat meng create,update,delete daftar kontak pelanggan dan suplier.	Sesuai
Memilih Menu Stock	Menampilkan data Stok barang yang tersedia,stok masuk,stok keluar, transfer stok, puchase stok, dan stoko opname	Sesuai

IV. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Kesimpulan yang di dapat dari perancangan, Integrasi dan sinkronisasi program SIKASIR sebagai proses monitoring menggunakan platform web ini adalah sistem ini dapat menghasilkan output berupa memonitoring laporan penjualan, sisa stok barang, menyimpan data kontak pelanggan dan supplier, memonitoring karyawan, menambahkan outlet baru, dan melihat produk yang masih ada di outlet dengan mudah dimana pun dan kapanpun melalui jaringan internet.

4.2 Saran

Adapun saran yang dapat di berikan untuk pengembang adalah aplikasi mendatang sebaiknya terdapat fitur chatting antara pemilik dengan staff atau dengan kasir sehingga dapat mempermudah pengguna dalam bertukar informasi dan lebih cepat serta efisien.

REFERENSI

- [1] Badan Pusat Statistik. "Tabel Perkembangan UMKM pada Periode 1997 -2012". Diambil pada tanggal 19 Mei 2016 dari <http://www.bps.go.id/linkTabelStatis/view/id/1322>.
- [2] Single Page Application. Diakses pada tanggal 20 April 2016. Available at : <https://msdn.microsoft.com/en-us/magazine/dn463786.aspx>.
- [3] AngularJS. What Is Angular? . Diakses pada tanggal 14 mei 2016. Avaliable at :<https://docs.angularjs.org/guide/introduction>.
- [4] Wahlin, D. 2013. AngularJS. Wahlin Consulting.
- [5] Peng, Dunlu.; Cao, Lidong.; Xu, Wenjie. 2011. Using JSON for Data Exchanging in Web Service Applications. Journal of Computational Information System 7. Vol. 16 (I) Hlm. 5883-5890.
- [6] Ghatol, R. dan Patel, Y. 2012. Beginning PhoneGap: Mobile Web Framework for JavaScript and HTML5. California: Apress Media LLC.