



## Pengamanan *Restful API* menggunakan JWT untuk Aplikasi *Sales Order*

Edy<sup>1</sup>, Ferdiansyah<sup>2</sup>, Wahyu Pramusinto<sup>3</sup>, Sejati Waluyo<sup>4</sup>

<sup>1,2,3,4</sup> Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur

<sup>1</sup>flammingjr@gmail.com, <sup>2</sup>ferdiansyah@budiluhur.ac.id, <sup>3</sup>wahyu.pramusinto@budiluhur.ac.id,

<sup>4</sup>sejati.waluyo@budiluhur.ac.id

### Abstract

The company has an important role in attracting the attention of customers by promoting its products and services so that it can be widely known by the community through the sales division (sales person). The biggest challenge faced by companies is related to marketing, namely stable and sustainable income growth, and customer loyalty. So that the application of information and communication technology is needed in the business world as a tool to win the main competition in marketing products or services. In the company, order data made by the sales person is still via telephone. You need an application to speed up the product ordering process to customers. To solve these problems, each sales person will use a sales order application through a smartphone device that will make product orders. And product ordering data will go to a web-based application run by admin for the invoice creation process. This application is designed using the RESTful API which is one of the implementation models of the web service and. In terms of data exchange security using the Json Web Token. With this sales order application, it can simplify the sales person's performance in marketing their products. This application work runs well on the web and android device so the process of ordering products becomes easier. The use of json web token authentication on the RESTful API makes the application more secure because this application can not be accessed if it does not use tokens.

Keywords : Kata kunci: *RESTFUL, API, JWT, Android, Webservice*

### Abstrak

Perusahaan mempunyai peran penting didalam menarik perhatian pelanggan dengan mempromosikan produk dan layanannya sehingga dapat dikenal luas oleh masyarakat melalui divisi penjualan (sales person). Tantangan terbesar yang dihadapi oleh perusahaan sangat berkaitan dengan pemasaran, yaitu pertumbuhan pendapatan yang stabil dan berkelanjutan, serta loyalitas pelanggan. Sehingga penerapan teknologi informasi dan komunikasi diperlukan dalam dunia bisnis sebagai alat bantu memenangkan persaingan utama dalam memasarkan produk atau jasa. Di dalam perusahaan, data pemesanan yang dilakukan sales person masih melalui telephone. Dibutuhkan aplikasi untuk mempercepat proses pemesanan produk kepada pelanggan. Untuk memecahkan permasalahan tersebut, masing-masing sales person akan menggunakan aplikasi sales order melalui perangkat smartphone yang akan membuat pemesanan produk. Dan data pemesanan produk akan masuk ke aplikasi berbasis web yang dijalankan admin untuk proses pembuatan invoice. Aplikasi ini dirancang menggunakan RESTful API yang merupakan salah satu model implementasi dari web service. Keamanan dalam hal pertukaran data pada aplikasi ini menggunakan Json Web Token. Dengan adanya aplikasi sales order ini dapat mempermudah kinerja sales person dalam memasarkan produknya. Aplikasi sales order ini berjalan baik pada web dan android sehingga proses pemesanan produk menjadi lebih mudah. Penggunaan autentifikasi json web token pada RESTful API ini membuat aplikasi menjadi lebih aman karena aplikasi tidak dapat diakses jika tidak menggunakan token.

Kata kunci: *RESTFUL, API, JWT, Android, Webservice*

© 2019 Jurnal RESTI

### 1. Pendahuluan

Divisi Penjualan adalah ujung tombak perusahaan yang mempunyai tugas utama untuk memasarkan produk[1]. Tanpa pemasaran yang baik maka besarnya aset perusahaan, teknologi yang canggih dan modern, dan

produk yang berkualitas akan kurang berarti bagi kelangsungan perusahaan dalam memaksi-malkan penjualan, perolehan laba serta dalam mengembalikan investasi. Dan tantangan terbesar yang dihadapi pemimpin perusahaan di seluruh dunia sangat berkaitan

Diterima Redaksi : 15-04-2019 | Selesai Revisi : 20-05-2019 | Diterbitkan Online : 01-08-2019

dengan pemasaran, yaitu pertumbuhan pendapatan yang stabil dan berkelanjutan, serta loyalitas pelanggan.

Didalam kinerja perusahaan dibidang penjualan terdapat *sales-sales* profesional yang diharapkan dapat menunjang kepentingan pergera-kan untuk menyusun strategi dan taktik bisnis perusahaan dengan memberikan solusi atau saran kepada pelanggan atas masalah yang dihadapi. Hal ini diharapkan dapat lebih memaksimalkan kepen-tingan dalam alur kerja dalam memasarkan produk tanpa keraguan pada pelanggan.

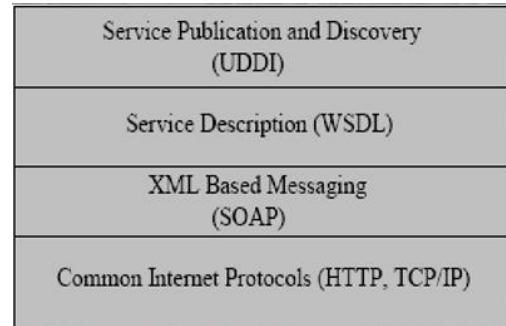
Kendala yang dialami perusahaan adalah kinerja sales person yang tidak maksimal dalam memasarkan produk kepada pelanggan. Data pemesanan produk masih dilakukan menggunakan telepon. Salah satu cara yang dapat dilakukan pihak manajemen perusahaan adalah membuat aplikasi *sales order* berbasis *mobile* dan *web* dengan menggunakan teknologi *RESTful API* dengan autentikasi keamanan menggunakan *Json Web Token*.

Tujuan pembuatan aplikasi *sales order* ini adalah dengan menerapkan teknologi dengan merancang suatu sistem dengan arsitektur *REST* yang dapat digunakan oleh berbagai *client* seperti aplikasi *mobile*, aplikasi *web* dan aplikasi *dekstop* agar membantu perusahaan dari sistem yang berjalan manual diubah menjadi suatu sistem yang terkomputerisasi sehingga dapat meningkatkan efesiensi kinerja bagi perusahaan, namun dalam pengujian aplikasi hanya dilakukan berbasis *web* dan *mobile*. Dan mempermudah dan mempercepat kinerja *sales person* dalam memasarkan produk atau barang terhadap pelanggan.

Sehingga dapat dibuat desain rancangan layar dan desain program untuk memfasilitasi fitur-fitur yang dibutuhkan pada aplikasi *sales order*. Untuk implementasi desain rancangan layar menjadi tampilan *form* menggunakan bahasa pemograman *PHP* untuk berbasis *web* dan pada *mobile smartphone* menggunakan bahasa pemograman *java*, serta *database* yang terhubung menggunakan *MariaDB*. Dan selanjutnya dilakukan tahap testing dan pengujian program dengan menggunakan *tools* yaitu *postman*.

*Web service* merupakan sebuah perangkat lunak yang tidak terpengaruh oleh *platform*, arsitektur, maupun bahasa pemograman, yang menyediakan layanan atau *method-method* untuk pertukaran data yang dapat diakses oleh *network*. Contoh implementasi dari *web service* antara lain adalah *SOAP* dan *REST*. *Web service* memiliki empat lapisan dasar seperti pada Gambar 1[2].

Gambar 1 diatas menunjukkan blok bangunan *web service* yang mana menyediakan fasilitas komunikasi jarak jauh antara dua aplikasi yang merupakan layer arsitektur *web service* [3].



Gambar 1. Lapisan Dasar *Web Service*

- 1) Layer 1 : protokol *internet* standar yang digunakan sebagai sarana transportasi adalah *HTTP* dan *TCP/IP*.
- 2) Layer 2 : *Simple Object Access Protocol* (*SOAP*) berbasis *XML* dan digunakan untuk pertukaran informasi antar sekelompok layanan.
- 3) Layer 3 : *Web service Definition Language* (*WSDL*) digunakan untuk mendiskripsikan attribute layanan.
- 4) Layer 4 : *Universal Description Discovery and Integration*, yang mana merupakan direktori pusat untuk deskripsi layanan.

Salah satu jenis dari *web service* adalah *REST* atau *RESTful* (*Representational State Transfer*). *REST* sendiri memungkinkan system request dapat mengakses dan memanipulasi teks yang direpresentasikan dari sebuah *web service*. *Web service API* yang menggunakan *REST* disebut dengan *RESTful API*. Tidak seperti jenis *web service* lainnya, *RESTful API* tidak memiliki standar yang resmi untuk notasinya dikarenakan *REST* merupakan sebuah arsitektur [4].

*REST* menentukan sekumpulan prinsip arsitektur yang mana dapat digunakan untuk merancang *web service* yang berfokus pada sumber daya sistem, termasuk bagaimana sumber daya yang dialamatkan dan ditransfer melalui *HTTP* oleh berbagai klien yang ditulis dalam Bahasa pemograman yang berbeda[4]. Dimana tujuannya adalah untuk menjadikan sistem memiliki performa yang baik, cepat, dan mudah untuk dikembangkan (*scale*) terutama dalam pertukaran dan komunikasi data. Umumnya menggunakan *HTTP* (*Hypertext Transfer Protocol*) sebagai protokol untuk komunikasi data.

Teknologi pengembangan *webservice* selain *REST* terdapat salah satu *webservice* yang familia bernama *Simple Object Access Protocol* (*SOAP*) dimana pada seperti halnya *REST* dapat melakukan pertukaran data yang sama untuk beberapa platform yang berbeda, namun *SOAP* hanya terbatas pada keluaran berupa *XML* (*Extensible Markup Language*) [5]

Pada arsitektur *REST*, *REST server* menyediakan *resources* (sumber daya / data) dan *REST client* mengakses dan menampilkan *resource* tersebut untuk

penggunaan selanjutnya. Setiap *resource* diidentifikasi oleh URIs (*Universal Resource Identifiers*) atau global ID. *Resource* tersebut direpresentasikan dalam bentuk format teks, JSON atau XML.

*Json Web Token* adalah sebuah token berbentuk *string* JSON yang sangat padat (ukurannya), informasi mandiri yang gunanya sendiri untuk melakukan sistem autentikasi dan pertukaran informasi. Karena bentuknya kecil, token JWT dapat dikirim melalui URL, parameter HTTP *POST* atau di dalam *header* HTTP, dan juga karena ukurannya yang kecil maka dapat ditransmisikan dengan lebih cepat. Disebut informasi mandiri karena isi dari token yang dihasilkan memiliki informasi dari pengguna yang dibutuhkan, sehingga tidak perlu query ke basis data lebih dari satu kali [6].

JWT bekerja dimana token ini seperti *password*, jadi ketika *user* berhasil melakukan *login* maka *server* akan memberikan sebuah token. Nanti token tersebut akan disimpan oleh *user* pada *local storage* atau *cookies browser* dan bila *user* ingin mengakses halaman halaman tertentu maka harus menyertakan token tersebut. Untuk itu *user* akan mengirim balik token yang dikasih diawal tadi sebagai bukti bila *user* ini, sudah melakukan *login*.

Struktur JWT terdiri dari tiga bagian yang dipisahkan oleh titik (.), yaitu *header*, *payload*, dan *signature*. Oleh karena itu, JWT biasanya terlihat seperti berikut ini (xxxxx.yyyyyy.zzzzz).

*Hash* adalah fungsi yang menerima masukan *string* yang panjangnya sembarang dan mengkonversinya menjadi *string* keluaran yang panjangnya tetap (*fixed*) umumnya berukuran jauh lebih kecil daripada ukuran *string* semula. Fungsi *hash* dapat menerima masukan *string* apa saja [7].

*Keyed-hash Message Authentication Code* adalah teknik *Message Authentication Code* yang dibuat oleh Mihir Bellare, Ran Canetti, Hugo Krawczyk pada tahun 1996. Hal ini didasari karena fungsi *hash* itu sendiri tidak memungkinkan penggunaan kunci ketika menghitung nilai digest-nya. Tujuan dari dibangunnya algoritma HMAC ini adalah [7] Untuk menggunakan fungsi *hash*, tanpa modifikasi, yang telah tersedia dan mudah didapatkan. Selain itu digunakan juga untuk mempertahankan performa dari algoritma *hash* yang sudah ada. Ditambah ntuk menggunakan dan mengatur kunci secara mudah. Seta untuk mendapatkan pengertian yang lebih dalam dari analisis kriptografi mengenai kekuatan mekanisme autentikasi yang berdasarkan fungsi *hash* sehingga nantinya digunakan untuk mempermudah pengubahan atau penggantian fungsi *hash* yang digunakan apabila algoritma *hash* baru yang lebih cepat atau lebih aman ditemukan.

SHA-256 merupakan salah satu fungsi hash satu arah, karena tidak mungkin menemukan pesan dari message digest yang dihasilkan. SHA-256 menggunakan enam fungsi logika, di mana setiap fungsi beroperasi pada 32-bit, yang direpresentasikan sebagai *x*, *y*, dan *z* [7]. Berikut adalah enam fungsi logika seperti pada persamaan :

$$Ch(x, y, z) = (x \wedge y) \oplus (\neg x \wedge z) \quad (1)$$

$$Maj(x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \quad (2)$$

$$\sum_0^{(256)}(x) = ROTR^2(x) \oplus ROTR^{13}(x) \oplus ROTR^{22}(x) \quad (3)$$

$$\sum_1^{(256)}(x) = ROTR^6(x) \oplus ROTR^{11}(x) \oplus ROTR^{25}(x) \quad (4)$$

$$\sigma_0^{(256)}(x) = ROTR^7(x) \oplus ROTR^{18}(x) \oplus SHR^3(x) \quad (5)$$

$$\sigma_1^{(256)}(x) = ROTR^{17}(x) \oplus ROTR^{19}(x) \oplus SHR^{10}(x) \quad (6)$$

*Smartphone* merupakan sebuah *device* yang memungkinkan untuk melakukan komunikasi, didalamnya juga terdapat fungsi PDA (Personal Digital Assistant) dan berkemampuan seperti layaknya komputer. Beberapa sistem operasi yang dapat ditemukan pada *smartphone* antara lain, *Symbian OS*, *iPhone OS*, *RIM Blackberry*, *Windows mobile*, *Linux*, *Palm WebOS*, dan *Android*. *Android* merupakan suatu sistem operasi mobile yang berbasis pada sistem operasi Linux [8].

*Android* pertama kali dikembangkan oleh perusahaan startup bernama *Android, Inc* yang berpusat di Palo Alto, California, Amerika Serikat., yang digawangi oleh Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Pada bulan Agustus 2005, Google membeli *Android* dan mengambil ahli proses pengembangannya hingga pada saat ini. Google merilis versi beta *Android SDK* (*System Development Kit*) pada November 2007 [8].

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman open source yang sangat cocok atau dikhususkan untuk pengembangan web dan dapat ditanamkan pada sebuah skripsi HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti *C*, *Java*, dan *Perl* serta mudah untuk dipelajari. PHP merupakan bahasa *scripting server-side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, *server* lah yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada client yang melakukan permintaan [9].

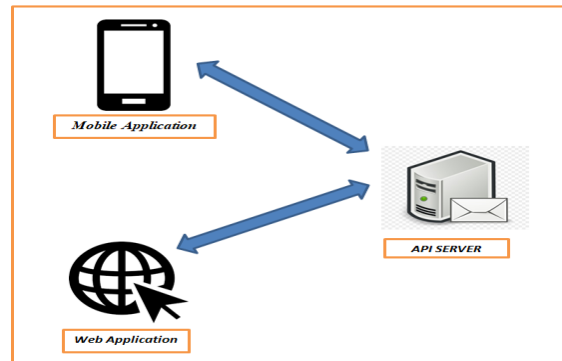
Pada prinsipnya *server* akan bekerja apabila ada permintaan dari *client*. Dalam hal ini *client* menggunakan kode-kode PHP untuk mengirimkan permintaan ke *server*. Dalam hal ini *client* menggunakan kode-kode PHP untuk mengirimkan permintaan ke *server*. Sistem kerja dari PHP diawali dengan permintaan yang berasal dari halaman *website* oleh *browser*. Berdasarkan URL atau alamat *website* dalam jaringan internet, *browser* akan menemukan sebuah alamat dari *webserver*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *webserver*.

*Database Management System* (DBMS) merupakan perangkat lunak untuk mengendalikan pembuatan, pemeliharaan, pengolahan, dan penggunaan data yang berskala besar. Penggunaan DBMS saat ini merupakan hal yang sangat penting dalam segala aspek, baik itu dalam skala yang besar atau kecil. Beberapa DBMS yang digunakan adalah MySQL dan MariaDB. MariaDB merupakan salah satu *database server* yang digunakan untuk menyimpan dan manajemen data. MariaDB tidak jauh berbeda dengan MySQL, karena MariaDB merupakan versi pengembangan terbuka dan mandiri dari MySQL. Sejak diakuisisinya MySQL oleh Oracle pada September 2010, Monty Program sebagai penulis awal kode sumber MySQL memisahkan diri dari pengembangan dan membuat versi yang lebih mandiri yakni MariaDB [10].

## 2. Metode Penelitian

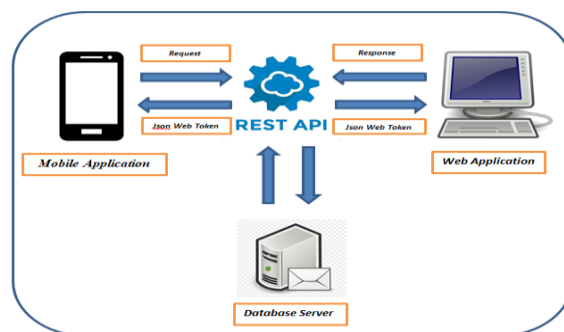
Metode dalam pengembangan penelitian yang digunakan adalah metode *waterfall* dimana tahap awal metode ini adalah melakukan Analisis yakni dimana cara yang digunakan adalah melalui wawancara dengan beberapa karyawan. Setelah didapatkan data yang dibutuhkan, tahap selanjutnya adalah desain program dimana pada tahap ini dibuat desain rancangan sistem menggunakan *use case*, *flowchart*, algoritma, dan rancangan layar. Serta membuat perancangan program untuk memfasilitasi fitur-fitur yang dibutuhkan pada aplikasi sales order. Tahap pengkodean dilakukan setelah tahap desain program dimana pada tahap ini adalah proses implementasi desain rancangan layar menjadi tampilan *form* menggunakan bahasa pemrograman PHP untuk berbasis *web* dan pada *mobile smartphone* menggunakan bahasa pemrograman *java*, serta *database* yang terhubung menggunakan MariaDB. Tahap terakhir yaitu tahapan Pengujian dimana tahapan melakukan beberapa *testing* atau pengujian program, baik dengan menggunakan *browser* maupun perangkat telepon selular berbasis android. Jika masih ada hal yang masih belum berjalan maka akan dilakukan perbaikan baik dari layar maupun program.

Secara umum aplikasi yang akan dibuat terdiri dari dua bagian besar yaitu aplikasi yang ada pada bagian *client* dan aplikasi yang ada pada bagian *server*. Aplikasi yang ada di *server* bertugas untuk menyediakan data yang dapat dikonsumsi oleh aplikasi *client*. Sedangkan aplikasi *client* digunakan untuk meminta data dari aplikasi yang ada di *server*.



Gambar 2. Arsitektur Aplikasi

Pada gambar 2 menunjukkan arsitektur aplikasi yang akan digunakan pada sistem yang akan dibuat. Pada gambar sisi sebelah kiri dapat dilihat bahwa aplikasi *client* yang akan dibuat terdiri dari aplikasi yang berbasis *web*, dan aplikasi berbasis *mobile*. Sedangkan di sisi *server* aplikasi yang akan dibuat adalah aplikasi *web service* yang menggunakan arsitektur REST.

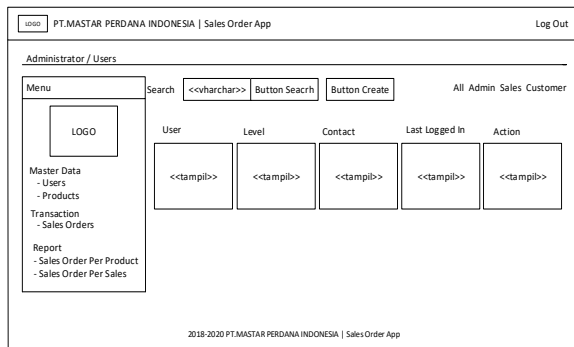


Gambar 3. Arsitektur Sistem

Pada gambar 3 diatas menunjukkan skema pengambilan data menggunakan RESTful API. Data barang yang sudah dipesan oleh pelanggan pada *mobile* akan dikirimkan dalam bentuk *Json Web Token* (JWT). Kemudian data tersebut akan dikirim ke *server API* yang dibangun. *Server API* akan melakukan proses permintaan ke *database* melalui *web*. Yang kemudian hasil dari proses permintaan akan dikembalikan lagi ke pengguna dalam bentuk *json* dan diuraikan oleh aplikasi sehingga data hasil proses bisa digunakan dan ditampilkan.

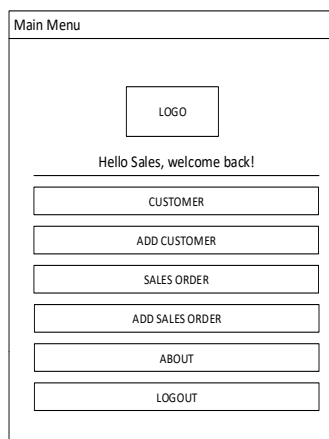
Perancangan layar user pada aplikasi *web* merupakan tampilan yang berisi semua dash dari halaman utama setelah *admin login* yang dapat mengakses aplikasi *web* ini. Terdapat beberapa menu tampilan yang bisa digunakan untuk melihat, mengubah, mengganti, dan

menghapus data baik itu *admin* maupun produk beserta *sales order*. Rancangan layar *dash* utama seperti pada gambar 4.



Gambar 4. Rancangan Dash Web

Sedangkan tampilan rancangan pada *smartphone* untuk *sales person* dalam mengakses aplikasi ini yang seperti pada gambar 5.



Gambar 5. Rancangan Dash Mobile

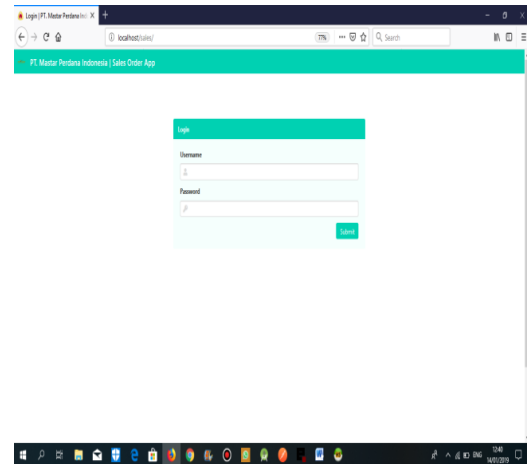
### 3. Hasil Dan Pembahasan

Dalam pelaksanaan dan penerapan program yang akan dilakukan dalam perusahaan dibagian penjualan, maka akan dilakukan tindakan untuk mencapai sasaran yang efektif. Didalam pembahasan ini akan dijelaskan mengenai tampilan aplikasi *web service RESTful API* pada *web* dan *android*.

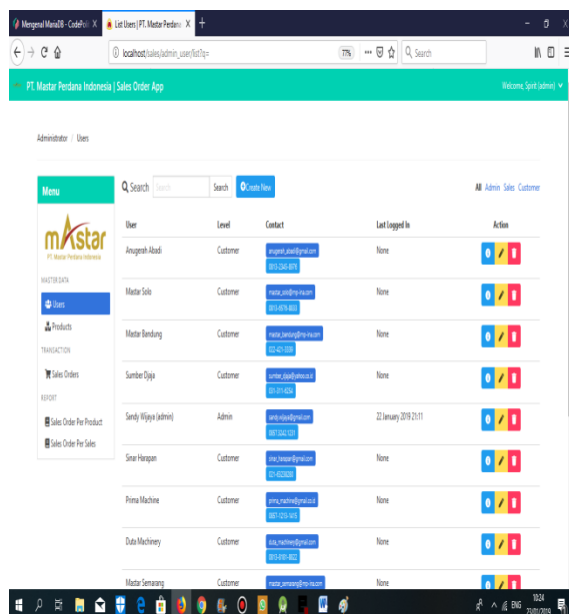
Tampilan layar *form login* merupakan yang pertama kali muncul saat *user* ketika membuka aplikasi ini. Pada tampilan ini *user* harus melakukan *login* terlebih dahulu untuk dapat mengakses ke halaman utama. Tampilan layar *form login* seperti pada gambar 6.

Setelah berhasil melakukan proses login maka akan keluar tampilan layar halaman utama *dash user*. Pada tampilan layar *main menu* ini akan menampilkan semua data *user* baik data sebagai *admin*, *customer* maupun *sales person* yang telah terdaftar pada aplikasi. Dan pada tampilan ini juga terdapat halaman produk barang

apa aja yang dijual dan pemesanan barang beserta laporan penjualan barang. Tampilan layar *form main menu* seperti pada gambar 7.



Gambar 6 Tampilan Layar Form Login Aplikasi Web

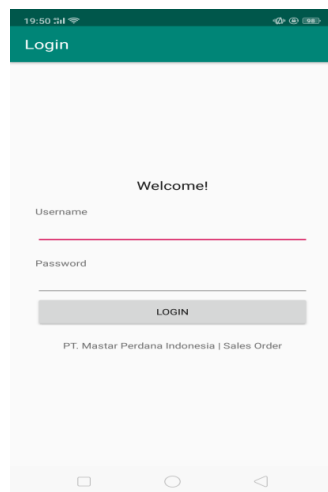


Gambar 7. Tampilan Layar Form Main Menu Aplikasi Web

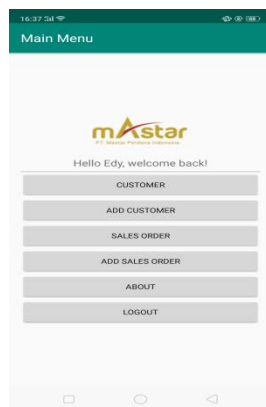
Untuk aplikasi *mobile* tampilan layar *login* merupakan yang pertama kali muncul saat *sales person* ketika membuka aplikasi *mobile*. Pada tampilan ini *sales person* harus melakukan *login* terlebih dahulu untuk dapat mengakses ke halaman utama. Tampilan layar *form login* seperti pada gambar 8.

Setelah melakukan login maka akan keluar tampilan layar *main menu* yang merupakan halaman utama *dash*. Pada tampilan layar ini akan menampilkan menu-menu pada aplikasi *mobile* dari melakukan pendataan pelanggan baru dan melakukan proses *sales order* pelanggan. Tampilan layar *form main menu* seperti pada gambar 9.





Gambar 8. Tampilan Layar Form Login Pada Mobile



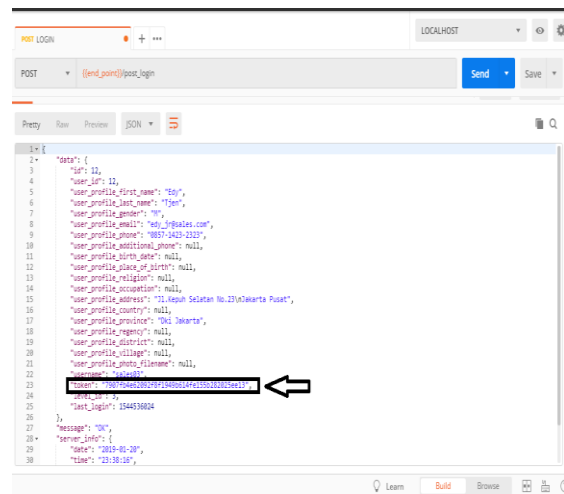
Gambar 9. Tampilan Layar Main Menu Aplikasi Mobile

Untuk melakukan testing program RESTful API menggunakan tool yaitu *postman* seperti pada gambar 10. Fungsi utama *postman* ini adalah sebagai *GUI API Caller* namun sekarang *postman* juga menyediakan beberapa fitur yaitu *Sharing Collection API for Documentation, Testing API, Real Time, Monitoring API, Integration*.



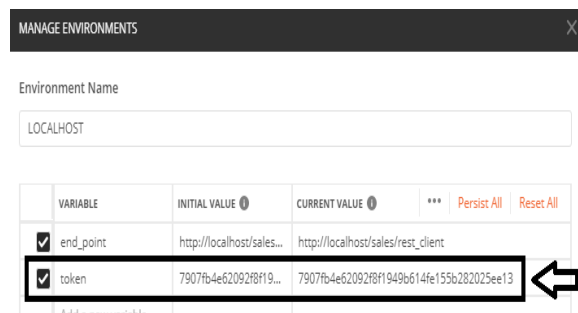
Gambar 10. Tool Postman

Pada saat *postman* dijalankan yang dilakukan *sales person* dalam melakukan *login* dengan menggunakan *method POST* maka akan keluar token untuk membuka tampilan menu pada aplikasi *mobile* seperti pada gambar 11.



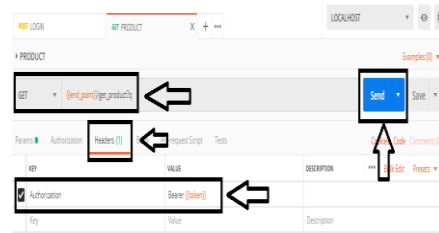
Gambar 11. Tampilan Token Pada Saat Login

Token tersebut akan disimpan oleh *user* pada *local storage* dan bila *user* mengakses halaman tertentu maka harus menyertakan token tersebut. Token tersebut akan dimasukkan ke *manage environments* untuk melakukan testing pada data yang terdapat di dalam aplikasi pada seperti gambar 12.



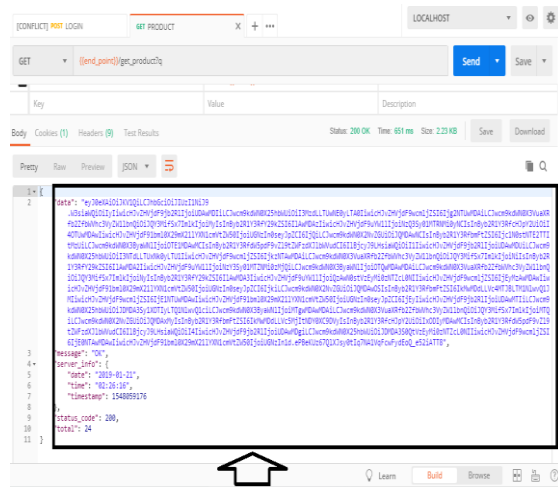
Gambar 12. Tampilan Pengisian Token Dalam Variable

Langkah lanjutan dalam pengujian testing *RESTful API* adalah memanggil data produk dengan menggunakan *method GET* yang ditampilkan seperti pada gambar 13. Diharuskan mengisi *url* yang akan dipanggil data produk dengan mengisi *header* dengan token pada saat melakukan *login*.



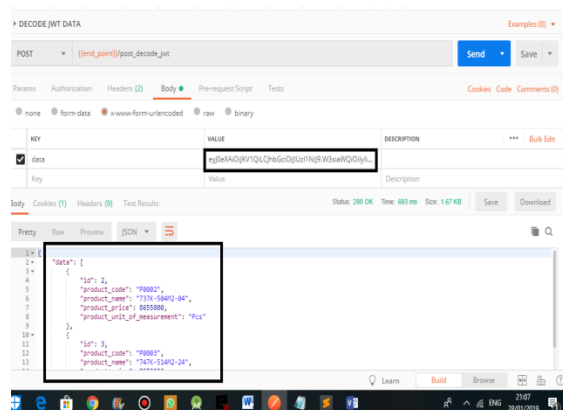
Gambar 13. Tampilan Untuk Memanggil Data Produk

Setelah menekan tombol *send* maka akan keluar tampilan data produk yang telah dienkripsi oleh *json web token*, seperti pada gambar 14.



Gambar 14. Tampilan Data Produk Dalam Bentuk JWT

Untuk hasil dekripsi dalam bentuk *json web token* maka akan keluar data asli barang seperti pada gambar 15.



Gambar 15. Tampilan Data Barang Hasil Dekripsi

Uji coba dan evaluasi dilakukan agar dapat melihat hasil, apakah ada kekurangan dalam aplikasi tersebut.

Tabel 1. Login Aplikasi *Sales Person*

No	Id	User	Name	Token	Login	is_active
1	21	Sales01	Hendy	7907b4e62092f8f1949b614fe155b282025ee13	09/01/2019 09:14	A
2	23	Sales03	Edy	a2644c6b1ba8901545009c7e7b40fe382518b05	09/01/2019 11:11	A
3	30	Sales05	Alvian	af232de78bae102126fbc6be3863140549966248	09/01/2019 10:13	A

Tabel 1 diatas merupakan aktifitas *sales person* dalam menggunakan aplikasi *mobile*.

Tabel 2 di atas merupakan hasil dari *sales person* dalam menginput data pemesanan produk kepada pelanggan yang akan di proses oleh *admin* untuk dibuatkan *invoice*. Proses *sales order* sampai tahap pembuatan *invoice* pada status *close*. Pada status *open*, jika

pelanggan ada penambahan dalam pemesanan produk, dan status *reject* jika pelanggan membatalkan order produk.

Tabel 2. Hasil Uji *Sales Order*

No	SO_Code	Customer	Sales_id	Ship_Datetime	Status
1	7HMDUVFWW	Duta Machinery	Sales01	09/01/2019 09:30	Close
2	FT6P25TVCW	MPI Solo	Sales03	09/01/2019 11:25	Close
3	PXJ7YRW2X3	MPI Bandung	Sales05	09/01/2019 10:23	Close
4	URH2HRRY4R	Sinar Harapan	Sales03	09/01/2019 14:22	Open
5	US625EYEW	Toko Subur	Sales01	09/01/2019 11:28	Close
6	DH6XZYEWU	Anugerah	Sales03	09/01/2019 15:11	Close
7	SH2TXPHJNT	MPI Semarang	Sales05	09/01/2019 12:19	Close
8	GKL30WEPDL	Sinar Jaya	Sales01	09/01/2019 13:09	Close
9	PL4ASAFICV	Cahaya Sewing	Sales05	09/01/2019 12:30	Reject
10	4ERFSADSC2	Toko Mentari	Sales05	09/01/2019 13:41	Close
11	9FRASLL3LA	Abadi Perkasa	Sales03	09/01/2019 15:41	Close
12	PRLS2SD5AA	Jaya Abadi	Sales05	09/01/2019 14:11	Close
13	0PSJDIS2YU	Surya Agung	Sales01	09/01/2019 15:11	Reject
14	TS5AQZMDL	Lima Jaya	Sales05	09/01/2019 15:21	Close
15	ER2PLEMMZ	KNP Garindo	Sales03	09/01/2019 16:53	Open

#### 4. Kesimpulan

Aplikasi sales order dengan RESTful API ini berjalan baik pada web dan android sehingga proses pemesanan produk menjadi lebih mudah. Penggunaan autentifikasi json web token pada RESTful API ini membuat aplikasi menjadi lebih aman karena aplikasi tidak dapat diakses jika tidak menggunakan token.

#### Daftar Rujukan

- Kurniawan, E., 2014. Implementasi Rest Web Service Untuk Sales Order Dan Sales Tracking Berbasis Mobile. *Eksis*, vol. 7, no. 1, pp. 1–12.
- Perwira, R.I, Santosa, B., 2017. Implementasi Web Service Pada Integrasi Data Akademik Dengan Replika Pangkalan Data Dikti. *Telematika*, vol. 14, no. 1, pp. 1–11.
- Rulloh dkk, 2017. Implementasi REST API pada Aplikasi Panduan Kepaskibraan Berbasis Android. *Tek. Vol. 1 No. 2*, vol. 1, no. 2, pp. 85–89.
- Tanaem, P.F., Manongga, D., dan Iriani, A., 2016. RESTful Web Service Untuk Sistem Pencatatan Transaksi. *J. Tek. Inform. dan Sist. Inf.*, vol. 2, no. April, pp. 2443–2229.
- Nurdiyanto, W. , 2012. Perbandingan SOAP dan REST sebagai Web Service [Online]. Available at : <http://pusdiklat.bps.go.id/index.php?r=artikel/cetak&id=206>. [Accessed: 21 Feb 2019].
- Rahmatulloh, A., Sulastr, H., dan Nugroho, R. 2018. Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512". *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 7, no. 2.
- Situmorang, J.D., 2013. Implementasi Algoritma Keyed-Hash Message Authentication Code ( HMAC ) Pada Pesan Teks Berbasis Chatting,. *Pelita Inform. Budi Darma*, vol. III, no. April, pp. 89–95.
- Yusmantoro, S., Hermansyah, E., dan Efendi, R., 2014. Pengamanan Keaslian Surat Elgama Dan Secure Hash Algorithm 256 Studi Kasus: Badan Pelayanan Perizinan Terpadu ( Bppt ) Kota Bengkulu. *J. Rekursif*, vol. 2, no. 1, pp. 28–36.
- Firman, A., Wowor, H., dan Najoran, X., 2016. Sistem Informasi Perpustakaan Berbasis Web Application. *J. Sist. Inf. Bisnis*, vol. 1, no. 2, pp. 66–77.
- Warman, I., Ramdaniyansyah, R., 2019. Analisis Perbandingan Kinerja Query Database Management System (Dbms) Antara Mysql 5.7.16 Dan Mariadb 10.1. *J. Teknoif*, vol. 6, no. 1, pp. 32–41.