

PEMANFAATAN REPRESENTATIONAL STATE TRANSFER DAN JSON WEB TOKEN PADA WEB SERVICE UNTUK PENINGKATAN KINERJA DAN KEAMANAN DATA PADA APLIKASI E-COMMERCE

Yusril Fadillah ¹⁾, Agus Hermanto ²⁾

^{1,2)} Program Studi Teknik Informatika, Fakultas Teknik, Universitas 17 Agustus 1945 Surabaya, Jl. Semolowaru no. 45 Surabaya, 60118, Telp: (031) – 5931800, Fax: (031) – 5927817, E-Mail: yusril354@gmail.com ¹⁾, hermanto_if@untag-sby.ac.id ²⁾

Abstract

The development of mobile or smartphone technology with the internet access that has been provided makes a health industry in the field of marketing want to improve its business processes by building a mobile-based e-commerce application (m-commerce). In this study, implementing web service technology using the REST architecture (REpresentation State Transfer) by implementing the RESTful API to communicate data from two different platforms, namely android and website. This result is in JSON format, and by using this we can access in through a link that was created. In guaranteeing the security of data sent through a server using JSON, will implement a JWT (JSON Web Token) in the RESTful API architecture. This study also built a website applications, and expected to increase the performance of e-commerce applications can increase and can also guarantee data security when communicating with the server.

Of this study has been achived by implementing a RESTful API, the Android-based e-commerce application that is made can run according to user needs and the performance of the features contained in the application can work in accordance with the results of functional requirements analysis. And also by implementing JWT, then request the data needed to produce a response in accordance with the access rights granted. Therefore, by implementing JWT can guarantee data security in e-commerce applications that are made.

Keywords: E-Commerce, JWT, Web Service, REST, JSON.

Abstrak

Berkembangnya teknologi mobile atau smartphone dengan akses internet yang tersedia menjadikan industri kesehatan di bidang pemasaran ingin meningkatkan proses bisnisnya dengan membangun sebuah aplikasi e-commerce berbasis mobile. Pada penelitian ini menerapkan teknologi web service dengan menggunakan arsitektur REST (REpresentation State Transfer) dengan mengimplementasikan RESTful API guna mengkomunikasikan data dari dua platform yang berbeda yaitu android dan website. Penelitian ini menghasilkan sebuah format keluaran JSON agar dapat diakses melalui link/URI yang dibuat. Dalam menjamin sebuah keamanan data yang dikirim melalui server maka dalam penelitian ini menerapkan sebuah JWT (JSON Web Token) di dalam arsitektur RESTful API. Penelitian ini juga membangun aplikasi website yang bertujuan untuk menguji validitas data pada aplikasi e-commerce. Dengan menggunakan penerapan tersebut, penelitian ini mengharapkan kinerja dari aplikasi e-commerce meningkat dan juga menjamin kemanan data saat berkomunikasi dengan server.

Hasil yang dicapai dalam penelitian ini adalah dengan mengimplementasikan sebuah RESTful API maka aplikasi e-commerce berbasis android yang dibuat dapat berjalan sesuai dengan kebutuhan pengguna dan kinerja dari fitur yang terdapat pada aplikasi tersebut dapat bekerja sesuai dengan hasil analisis kebutuhan fungsional. Dengan mengimplementasikan JWT, maka request data yang dibutuhkan dapat menghasilkan sebuah response sesuai dengan hak akses yang diberikan. Maka dari itu, dengan mengimplementasikan JWT dapat menjamin keamanan data pada aplikasi e-commerce yang dibuat.

Kata Kunci: E-Commerce, JWT, Web Service, REST, JSON.

1. PENDAHULUAN

Berkembangnya akses internet saat ini tidak hanya terjangkau pada teknologi *website*, akan tetapi sudah berkembang pesat dalam perangkat *mobile*. Dengan adanya hal itu membuat penggunaan perangkat *mobile* di kalangan masyarakat menjadi meningkat. Sehingga dengan pertimbangan tersebut membuat industri kesehatan di bidang pemasaran berkeinginan untuk meningkatkan sebuah proses bisnisnya dengan mengembangkan sebuah aplikasi *e-commerce* dalam perangkat *mobile* untuk mencari dan menyebarkan sebuah informasi penjualan berupa jasa yang dapat diakses dimana saja dan kapan saja secara cepat dan tepat.

Menurut [1] secara umum definisi dari *e-commerce* adalah proses membeli, menjual, baik dalam bentuk barang, jasa ataupun informasi yang dilakukan melalui media internet. Aplikasi *e-commerce* dalam perangkat *mobile* bisa disebut juga dengan aplikasi *m-commerce* (*Mobile Commerce*). Aplikasi *m-commerce* tersebut diimplementasikan ke dalam sistem informasi berbasis Linux yaitu *android*. *Android* adalah sistem informasi berbasis Linux yang dirancang oleh Google yang biasa digunakan dalam perangkat *smartphone* atau *tablet* [14].

Dalam mengembangkan sebuah aplikasi *m-commerce* tentunya terdapat sebuah data yang harus disimpan. Data tersebut disimpan dalam sebuah server dengan menggunakan akses internet untuk diakses dimana saja dan kapan saja secara cepat dan tepat. Terdapat sebuah arsitektur pada *web service* yang dapat menghubungkan antara aplikasi *m-commerce* dengan server, yaitu arsitektur REST (*Representational State Transfer*). Dengan menggunakan arsitektur REST maka data pada aplikasi *m-commerce* dapat berkomunikasi secara cepat dan tepat ke server.

REST adalah suatu arsitektur metode telekomunikasi yang sering diterapkan dalam pengembangan layanan berbasis *website* [2]. Dengan menggunakan arsitektur REST terdapat HTTP (*Hypertext Transfer Protocol*) sebagai protokol untuk komunikasi data yang didalamnya memiliki sebuah metode permintaan (*Request Methods*). Metode tersebut merupakan sebuah akses untuk mengkomunikasikan data dari aplikasi *mobile* ke server. REST akan menampilkan sebuah *resource* yang direpresentasikan dalam format JSON (*JavaScript Object Notation*). JSON merupakan format pertukaran data yang ringan dan tidak bergantung pada bahasa pemrograman apapun, oleh karena itu menjadikannya menjadi format pertukaran yang ideal. Dengan demikian, maka akan memudahkan dalam mengembangkan sebuah aplikasi *M-Commerce* dalam mengkomunikasikan

data dari aplikasi ke server. Dalam menampilkan data pada sebuah *resource* membutuhkan sebuah sistem keamanan data yang dapat menjamin data tersebut supaya aman. JWT (*JSON Web Token*) merupakan salah satu cara untuk menjamin sebuah data yang ditampilkan pada *resource* tersebut menjadi aman.

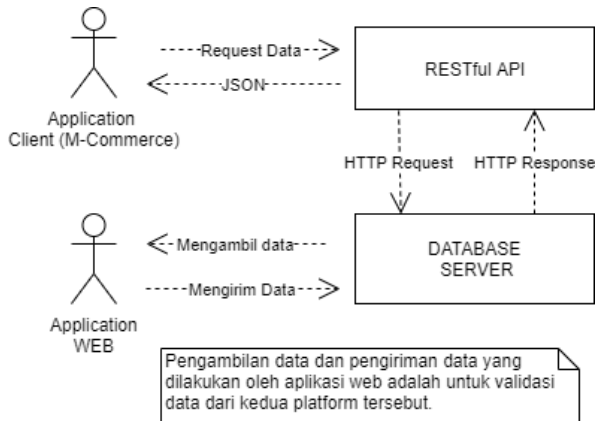
Hal ini menjadikan penulis untuk menggunakan arsitektur REST pada *Web Service* dengan mengimplementasikan sebuah RESTful API (*Application Program Interface*) yang akan direpresentasikan dalam format JSON guna memperlihatkan atau memberikan informasi data dari server untuk mengkomunikasikan aplikasi *M-Commerce* dengan server dan juga mengimplementasikan sebuah JWT untuk menjamin data yang dikirim melalui server dalam bentuk *resource*. Penulis juga mengembangkan sebuah aplikasi *website* sebagai uji validitas data atau monitoring transaksi pada aplikasi *M-Commerce*. Jadi arsitektur REST juga dapat digunakan untuk uji validitas data dari aplikasi *M-Commerce* ke aplikasi *website*. Sehingga dengan arsitektur REST kedua aplikasi tersebut dapat saling berkomunikasi ke server. Arsitektur REST juga digunakan untuk pertukaran data antara aplikasi *M-Commerce* dengan aplikasi *website*. Dengan demikian, memungkinkan kinerja aplikasi *E-Commerce* dapat meningkat dan dapat diakses dimana saja dan kapan saja secara cepat dan tepat dan juga data yang dikirim dari server dapat terjamin keamanannya.

2. METODOLOGI PENELITIAN

Pada penelitian ini menerapkan tiga tahapan yang dilakukan, yaitu analisis kebutuhan untuk mengkomunikasikan data antar *platform*, desain dan coding untuk menghasilkan sebuah keluaran berupa format JSON, dan yang terakhir adalah melakukan implementasi dan pengujian sistem informasi yang sudah dibangun.

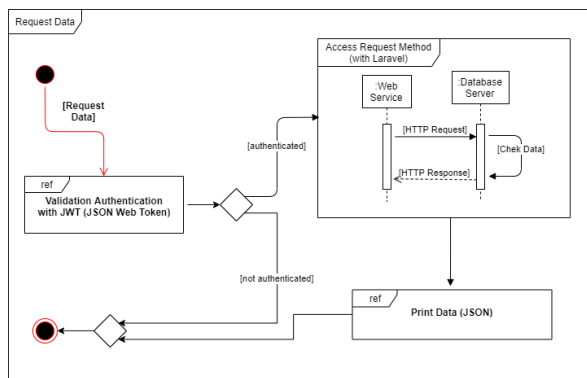
2.1. Analisis Kebutuhan

Pada tahap ini melakukan analisis kebutuhan mulai dari kebutuhan data dan informasi dan kebutuhan fungsional dari identifikasi masalah yang terjadi dengan melakukan sebuah observasi ke perusahaan. Tahap ini dilakukan dengan cara wawancara terhadap *client* dan juga *development* dari aplikasi *m-commerce* dalam mengidentifikasi masalah. Adapun informasi *flow diagram* yang digunakan adalah sebagai berikut:



Gambar 2.1. Information Flow Diagram Web Service.

Gambar 2.1 memberikan sebuah informasi dari alur yang digunakan dalam mengimplementasikan arsitektur RESTful API. Gambar tersebut mengacu ke alur komunikasi data ke server. Terdapat sebuah perbedaan dalam proses komunikasi data ke server, yaitu apabila aplikasi *client* ingin berkomunikasi dengan server harus melalui *web service* (RESTful API) yaitu dengan menggunakan *endpoint* (alamat URI atau *link* dari sebuah *resource*), sedangkan aplikasi *web* bisa langsung berkomunikasi dengan server, dalam artian bisa menggunakan *session*. *Session* adalah sebuah cara yang digunakan aplikasi *web* untuk menyimpan data ke server dalam bentuk *variable* super global.



Gambar 2.2. Interaction Overview Diagram Web Service.

Gambar 2.2. adalah sebuah interaksi *overview* diagram dari arsitektur yang dibuat, yaitu *client* melakukan *request* data ke server untuk melakukan komunikasi atau pertukaran data. Ketika melakukan *request* data, nantinya aplikasi *client* akan melakukan validasi autentikasi dengan menggunakan objek JSON yaitu JWT (JSON Web Token). JWT adalah sebuah objek JSON yang didefinisikan dalam RFC 7519 sebagai cara yang aman untuk mewakili kumpulan informasi antara dua pihak [15]. JWT akan menghasilkan sebuah token yang bisa memastikan keamanan aplikasi

yang dibuat, meliputi data pengguna dengan cara yang aman. Token yang dihasilkan oleh JWT meliputi *header*, *payload*, *signature*.

Apabila *client* terverifikasi maka *web service* akan mengakses *request* data dari *client* dengan mengimplementasikan RESTful API yang dibuat melalui *framework* Laravel. *Web service* akan mengirimkan HTTP *request* ke server dan server akan melakukan pengecekan data didalamnya, kemudian data akan dikembalikan lagi ke *web service* dengan menggunakan HTTP *response*. Apabila *web service* sudah menerima *response* dari server, maka *web service* akan memberikan keluaran berupa JSON yang berisikan data yang dibutuhkan oleh *client*.

Dari kedua diagram tersebut memberikan kesimpulan bahwa kebutuhan fungsional yang nantinya akan dijadikan bahan penelitian ini adalah kebutuhan *client* dalam melakukan komunikasi ke server, data apa saja yang dibutuhkan dan juga *request method* apa saja yang dibutuhkan oleh *client* yang nantinya akan diberikan sebuah alamat URI atau *link* yang nantinya akan memberikan sebuah *response* berupa JSON. Dan juga kebutuhan fungsional dari aplikasi *web* yang nantinya akan menjadi uji validitas data pada aplikasi *client* yaitu *m-commerce*. Kebutuhan fungsional dari aplikasi *web* tersebut adalah:

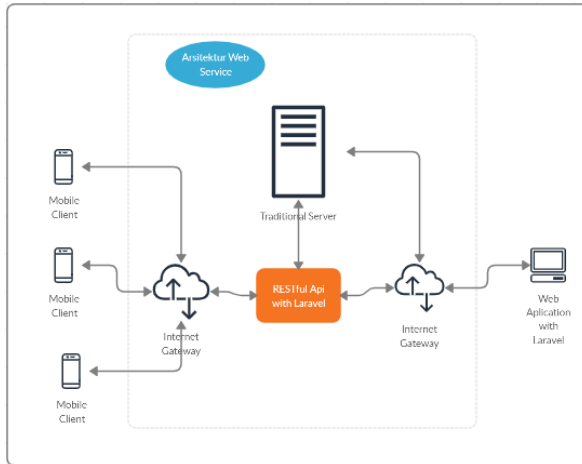
- Sistem dapat manajemen pengguna dari aplikasi *M-Commerce* dan *website*.
- Sistem menampilkan riwayat transaksi pengguna dari aplikasi *M-Commerce*.
- Sistem dapat manajemen berkas sertifikasi dari pengguna dalam penggunaan saldo pada *M-Commerce*.
- Sistem menampilkan riwayat penggunaan saldo pengguna dari aplikasi *M-Commerce*.
- Sistem dapat manajemen layanan beserta mengatur harga setiap layanan yang ada pada aplikasi *M-Commerce*.
- Sistem dapat mengatur jadwal sertifikasi.
- Sistem dapat manajemen pendaftaran pengguna aplikasi *M-Commerce*.

2.2. Desain dan Coding

Pada tahap ini akan melakukan sebuah desain dari hasil perancangan yang sudah dibuat, dan akan melanjutkan dengan meng-coding program aplikasi yang dibuat.

2.2.1. Desain Perancangan Sistem

Penelitian ini menggunakan sebuah arsitektur dari *web service* yaitu dengan mengimplementasikan RESTful API di dalamnya.



Gambar 2.3. Arsitektur Diagram Web Service.

Dalam analisis sistem informasi ini mengacu pada arsitektur diagram pada gambar 2.3. Dari arsitektur tersebut membutuhkan beberapa objek dalam mengkomunikasikan data ke server. Dengan menggunakan akses internet maka sebuah sistem informasi dapat berkomunikasi langsung dengan server. Akan tetapi dalam mengkomunikasikan sebuah aplikasi *client* ke server membutuhkan sebuah akses dalam mengirim atau mengambil data dari server.

Dari arsitektur tersebut menghasilkan desain perancangan sistem yang berisikan sebuah pemodelan sistem seperti ERD (Entity Relationship Diagram) yang merupakan sebuah cara yang dapat digunakan dalam menggambarkan kebutuhan data pada sebuah sistem informasi, *use case diagram* yang nantinya akan menjelaskan tentang peran apa saja yang dilakukan oleh *user* (kebutuhan fungsional), *sequence diagram* yang menjelaskan tentang cara kerja dari objek dalam sistem informasi yang dibuat, dan juga terdapat *activity diagram* yang berupa BPMN (*Business Process Model and Notation*) model yang menjelaskan tentang kejadian yang terjadi dalam dunia nyata dari proses kerja yang berlangsung.

Dengan menggunakan model tersebut, maka dapat memudahkan untuk mendapatkan sebuah solusi dalam memecahkan masalah dalam membangun/mengembangkan sistem informasi pada penelitian ini.

2.2.2. Coding Program

Dalam membangun sistem informasi ini menggunakan *framework* Laravel, dan dengan mengimplementasikan sebuah arsitektur dari *web service* yaitu RESTful API dan juga memberikan sebuah keluaran berupa *format* JSON melalui *link* atau URI yang sudah dibuat. Dalam membangun arsitektur ini diberikan sebuah sistem dalam menjamin keamanan data dari server. Dengan mengimplementasikan sebuah objek JSON (JWT), maka keamanan data yang dikirim ke *client* menjadi aman/terlindungi.

Penelitian ini juga membangun sebuah sistem informasi berbasis *web* yang menjadi wadah untuk uji validitas data yang terdapat pada aplikasi *client* (*m-commerce*). Aplikasi *web* ini hanya dapat diakses oleh pemilik atau bagian admin dari perusahaan.

2.3. Implementasi

Dalam penelitian ini, diimplementasikan ke sebuah perusahaan di bidang kesehatan dalam meningkatkan proses bisnisnya. Arsitektur RESTful API ini digunakan oleh pengembang aplikasi *client* dalam mengembangkan sebuah sistem informasi *e-commerce* berbasis *android*. Aplikasi *web* yang dibangun, juga diimplementasikan secara langsung ke perusahaan tersebut.

2.4. Uji Testing

Pada hasil dari penelitian ini dilakukan tahap uji *testing* untuk menguji kesesuaian aplikasi yang sudah dibangun. Pada tahap pengujian ini dengan menggunakan metode *back box testing*. *Black box testing* adalah metode pengujian perangkat lunak atau aplikasi dengan menguji fungsionalitas aplikasi yang sudah dibangun dengan menyesuaikan dari hasil analisis kebutuhan pengguna. Pengujian ini menggunakan tipe dari metode ini yaitu dari aspek fungsional dan non-fungsional. Tipe fungsional terkait dengan aspek persyaratan kebutuhan fungsional sistem. Dan non-fungsional terkait dengan kinerja, skalabilitas dan kegunaan dari sistem yang sudah dibangun.

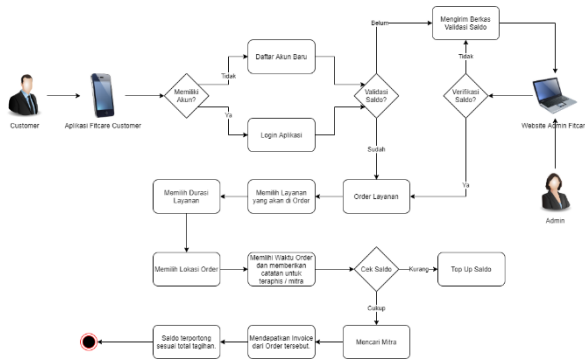
3. HASIL DAN PEMBAHASAN

Pada hasil dan pembahasan ini akan memberikan sebuah hasil yang sudah dicapai dalam membangun arsitektur restful API dan

aplikasi website. Hasil tersebut berupa sebuah hasil perancangan dan implementasi dari hasil perancangan itu sendiri.

3.1. Proses Bisnis Aplikasi

Proses bisnis yang didapat adalah bagaimana jalannya aplikasi yang akan dibuat. Berikut adalah salah satu proses bisnis dari aplikasi yang sudah dibuat yaitu proses bisnis *order* layanan *customer*



Gambar 3.1. Proses Bisnis.

Penjelasan dari proses bisnis diatas adalah ketika *customer* ingin melakukan sebuah *order* layanan yaitu *customer* harus masuk ke aplikasi Fitcare untuk *customer*. Apabila *customer* belum memiliki akun, maka *customer* harus daftar terlebih dahulu, akan tetapi apabila *customer* sudah memiliki akun maka *customer* langsung login dengan nomor HP yang sudah didaftarkan. Ketika *customer* ingin melakukan order, maka *customer* tersebut harus melakukan validasi saldo. Dengan cara mengirimkan berkas melalui aplikasi *customer* yang nantinya akan diverifikasi oleh admin apabila berkas tersebut valid, jika tidak maka sebaliknya.

Ketika sudah melakukan validasi saldo, maka *customer* dapat melakukan *order* layanan dengan memilih layanan yang akan di-order. Kemudian *customer* memilih durasi layanan, lokasi *order*, waktu *order* dan *customer* juga dapat memberikan sebuah catatan untuk mitra. Setelah mengisi semua form maka *customer* melakukan *order*, kemudian sistem akan melakukan pengecekan saldo *customer* tersebut. Apabila saldo mencukupi maka *order* akan dilanjutkan dan *customer* akan mendapatkan *invoice* dengan mitra yang didapat. Tapi jika sebaliknya, maka *customer* harus *top up* saldo melalui menu *top up* yang sudah disediakan pada aplikasi *customer*.

3.2. Perancangan Interaksi Sistem

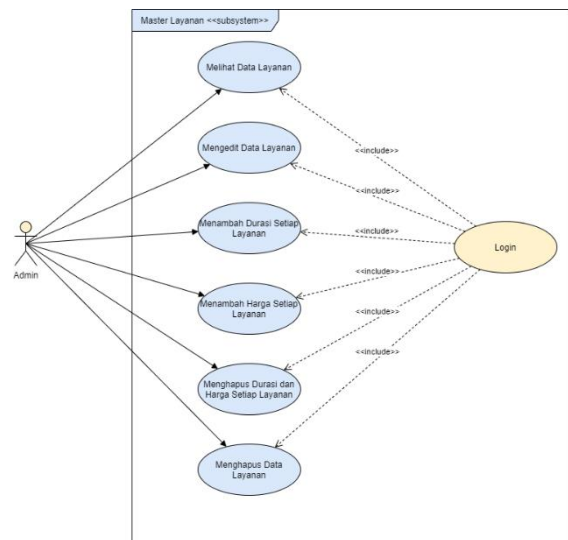
Pada hasil perancangan interaksi sistem memberikan sebuah gambaran perancangan sistem berupa usecase diagram, class diagram, sequence diagram dan activity diagram.

3.2.1. Use Case Diagram

Use case diagram merupakan sebuah interaksi antara pengguna aplikasi dengan sistem yang digambarkan dalam bentuk diagram, sehingga mempermudah dalam menentukan fitur dari aplikasi yang dibuat. Berikut merupakan use case diagram dari aplikasi m-commerce dan aplikasi website yang dibagi menjadi beberapa sub sistem, diantaranya:

a. Sub Sistem Master Layanan

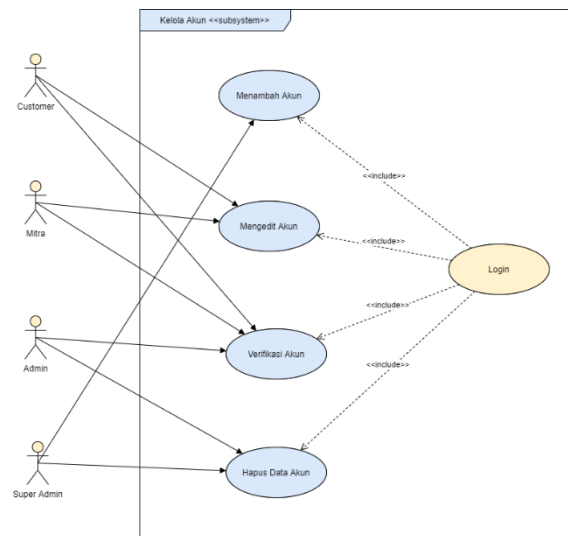
Berikut adalah use case dari sub sistem master layanan.



Gambar 3.2. Use Case Sub Sistem Master Layanan.

b. Sub Sistem Kelola Akun

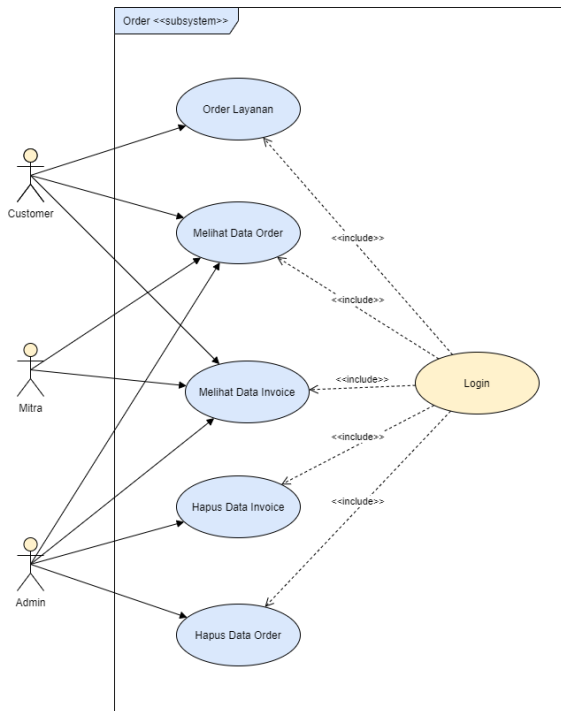
Sub sistem ini akan menggambarkan sebuah interaksi yang berhubungan dengan kelola akun yang terdapat pada aplikasi m-commerce dan aplikasi website.



Gambar 3.3. Use Case Sub Sistem Kelola Akun.

c. Sub Sistem Order

Berikut adalah use case dari sub sistem master order.



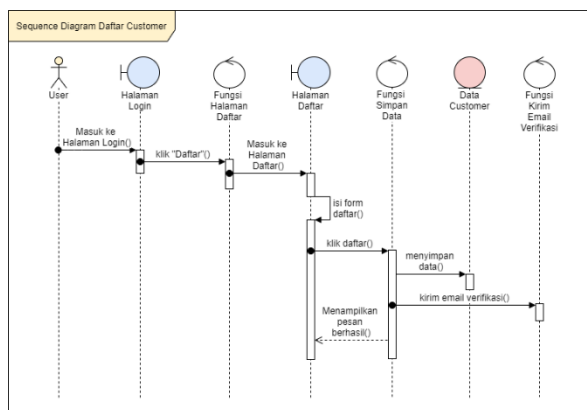
Gambar 3.4. Use Case Sub Sistem Order.

3.2.2. Sequence Diagram

Berikut adalah hasil dari sequence diagram dalam merancang aplikasi Website dan Aplikasi M-Commerce.

a. Daftar Customer

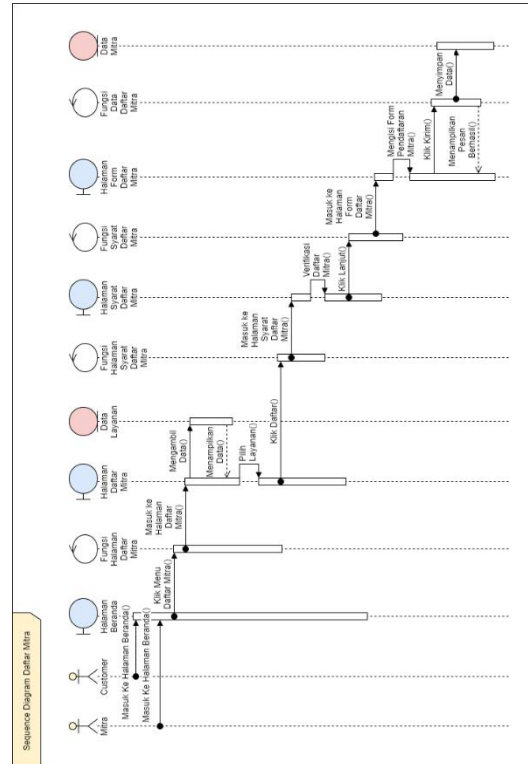
Berikut adalah hasil dari sequence diagram daftar customer dengan menggunakan aktor user.



Gambar 3.5. Seq. Diagram Daftar Customer.

b. Daftar Mitra

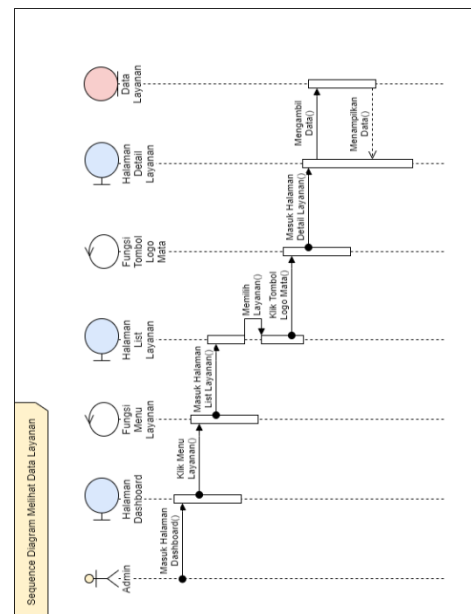
Berikut adalah hasil dari sequence diagram daftar mitra, dengan menggunakan dua aktor yaitu customer dan mitra:



Gambar 3.6. Seq. Diagram Daftar Mitra.

c. Melihat Master Data Layanan

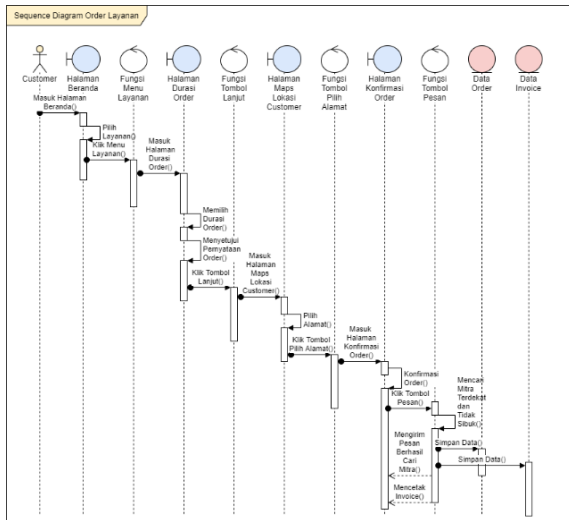
Berikut adalah sequence diagram melihat master data layanan yang terdapat pada aplikasi website:



Gambar 3.7. Sequence Diagram Melihat Master Data Layanan.

d. Order Layanan

Berikut adalah hasil sequence diagram order layanan dengan mengimplementasikan restful API:



Gambar 3.8. Sequence Diagram Order Layanan.

3.3. Implementasi Hasil Perancangan

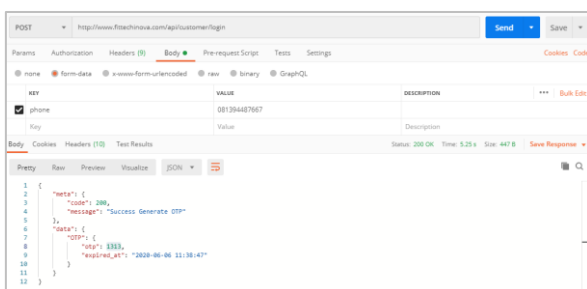
Dalam implementasi hasil perancangan (IHP) dibagi menjadi dua, yaitu hasil perancangan yang dihasilkan dalam mengembangkan sebuah arsitektur REST yang mengimplementasikan RESTful API dan juga hasil perancangan sistem informasi berbasis *website* yang digunakan untuk mengelola data yang terdapat pada database. Berikut adalah hasil perancangan yang sudah dibuat:

3.3.1. Arsitektur RESTful API

Dalam implementasi hasil perancangan arsitektur REST ditujukan untuk aplikasi *e-commerce* yang dikembangkan oleh pihak pengembang aplikasi *e-commerce* berbasis *android*. Berikut adalah beberapa implementasi hasil perancangan arsitektur RESTful API:

a. Login Customer

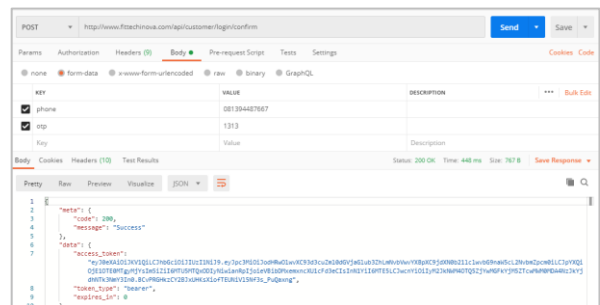
Berikut adalah hasil dari arsitektur REST yang digunakan dalam aplikasi *e-commerce* untuk *login* dengan menggunakan parameter *phone*. *Response JSON* yang dihasilkan adalah sebuah kode OTP dan juga waktu *expired* OTP.



Gambar 3.9. IHP REST Login Customer.

b. Konfirmasi Kode OTP

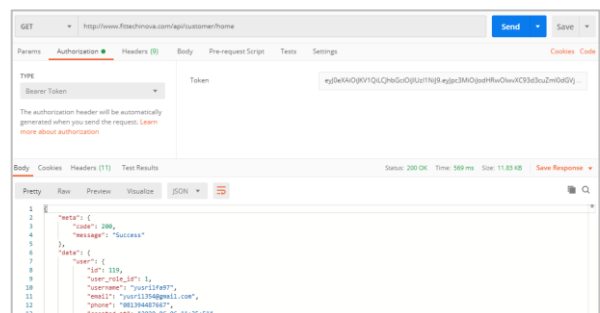
Berikut adalah hasil dari konfirmasi kode OTP ketika proses *Login*.



Gambar 3.10. IHP REST Konfirmasi OTP Login.

c. Get Data Beranda

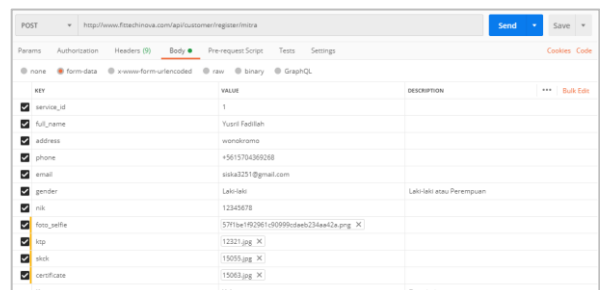
Berikut adalah hasil dari arsitektur REST yang digunakan dalam mengambil sebuah data untuk halaman *home* pada aplikasi *e-commerce*. Dengan autentikasi *bearer token* yang sudah digenerate ketika proses konfirmasi kode otp ketika *register* atau *login*.



Gambar 3.11. IHP REST Get Data Home.

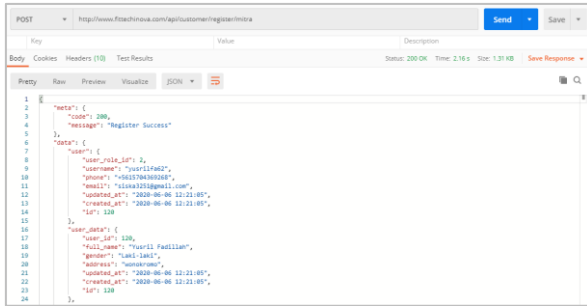
d. Daftar Jadi Mitra

Berikut adalah hasil dari arsitektur REST yang akan digunakan dalam mengirim sebuah data untuk mendaftar menjadi mitra dari *Fitcare*. Sehingga *customer* harus menunggu verifikasi akun dari admin untuk menjadi mitra.



Gambar 3.12. IHP REST Daftar Jadi Mitra (1).

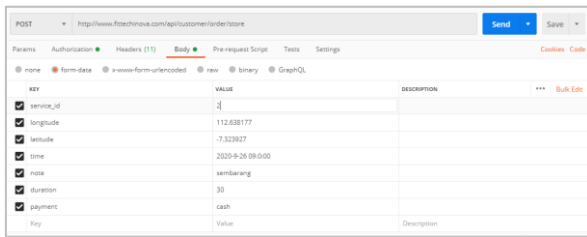
Sehingga mendapatkan sebuah *response* berupa *JSON* sebagai berikut.



Gambar 3.13. IHP REST Daftar Jadi Mitra (2).

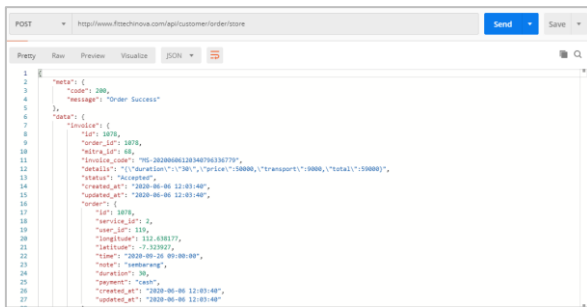
e. Order Layanan

Pada proses *order* layanan terdapat sebuah kecerdasan buatan di dalamnya yaitu pada proses pencarian mitra dengan jarak paling jauh 15 kilometer dan ketika mitra sedang sibuk. Apabila terdapat sebuah *order* dengan waktu yang sama maka tidak akan mendapatkan mitra yang sama juga. Pada proses *order* menggunakan autentikasi bearer token dan dengan data sebagai berikut.



Gambar 3.14. IHP REST Order Layanan.

Maka akan menghasilkan sebuah *response* JSON sebagai berikut.



Gambar 3.15. IHP REST Hasil Rresponse Order Layanan.

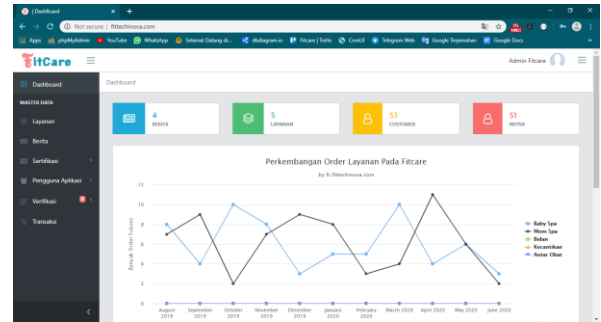
3.3.2. Aplikasi Website

Aplikasi *website* yang sudah dikembangkan, ditujukan untuk menguji validitas data yang ada pada aplikasi *e-commerce* berbasis *android*. Sehingga, berikut adalah IHP aplikasi *website*.

a. Halaman Dashboard

Berikut adalah hasil dari implementasi hasil perancangan halaman *dashboard* pada aplikasi *website*. Yaitu terdapat sebuah informasi jumlah

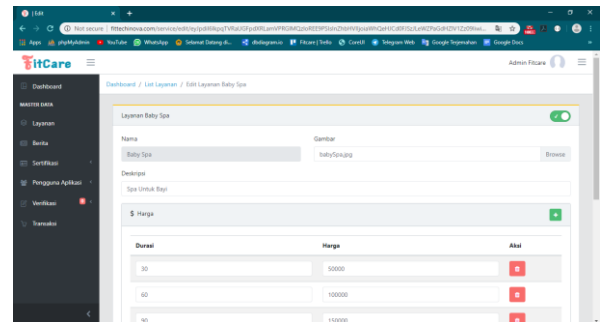
berita, layanan, dan pengguna aplikasi *e-commerce*. Pada halaman ini juga terdapat diagram yang menunjukkan sebuah grafik peningkatan *order* dari tiap layanan per bulan dan juga terdapat sebuah *list* mitra yang paling banyak mendapatkan sebuah *order*. Untuk sementara hasil dari halaman *dashboard* adalah sebagai berikut.



Gambar 3.16. IHP Website Hal. Dashboard.

b. Halaman Edit Layanan

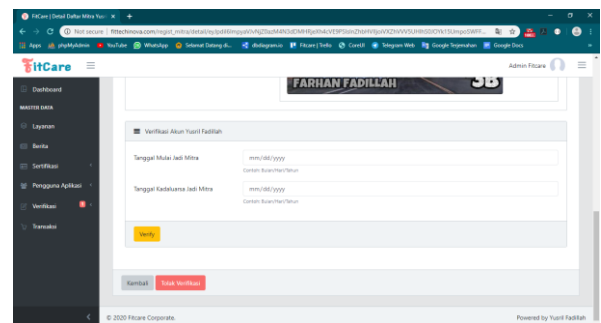
Berikut adalah hasil dari implementasi hasil perancangan halaman edit layanan pada aplikasi *website*. Yaitu memberikan sebuah *form* untuk mengedit sebuah data layanan pada database.



Gambar 3.17. IHP App. Website Hal. Detail Layanan.

c. Halaman Verifikasi Mitra

Berikut adalah hasil dari implementasi hasil perancangan halaman verifikasi daftar mitra pada aplikasi *website*. Yaitu memberikan sebuah *form* verifikasi daftar mitra. Halaman ini terletak pada halaman detail daftar mitra.



Gambar 3.18. IHP Website Hal. Verifikasi Daftar Mitra.

3.4. Hasil Pengujian

Dalam penelitian ini pengujian black box *testing* yang digunakan adalah dengan menggunakan 3 tipe pengujian, yaitu pengujian persyaratan fungsional dengan menguji keberhasilan fungsi pada RESTful API dan aplikasi *website*, pengujian kinerja aplikasi dengan menguji kecepatan proses *request* data pada RESTful API, dan juga pengujian usability dengan menguji tingkat kepuasan pengguna aplikasi *website*. Berikut adalah hasil pengujian dengan menggunakan metode black box *testing*:

3.4.1. Pengujian Persyaratan Fungsional

Pada pengujian persyaratan fungsional dilakukan dengan 2 bagian yaitu pada RESTful API dan pada aplikasi *web* yang sudah dibuat. Berikut adalah hasil pengujian persyaratan fungsional yang sudah dilakukan.

a. Pengujian Persyaratan Fungsional RESTful API

Berikut adalah beberapa hasil pengujian persyaratan fungsional RESTful API:

Tabel 3.1. Pengujian Persyaratan Fungsional RESTful API.

ID	Deskripsi atau Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian
01-RA	Pre-kondisi: mengisi sebuah parameter <i>form login</i> .	Menghasilkan sebuah <i>response</i> JSON berupa kode OTP dan mengirimkan sebuah email berupa kode OTP.	Sesuai dengan harapan .
02-RA	Pre-kondisi: mengisi parameter <i>form konfirmasi login</i> dengan kode otp yang dihasilkan atau dikirim melalui email.	Menghasilkan sebuah <i>response</i> JSON berupa JWT untuk digunakan sebagai <i>authorisasi</i> .	Sesuai dengan harapan .
03-RA	Pre-kondisi: input <i>authorisasi bearer token</i> JWT yang sudah	Menghasilkan sebuah <i>response</i> berupa JSON yaitu berisikan sebuah data	Sesuai dengan harapan .

	dihasilkan untuk mengambil sebuah data untuk halaman beranda.	yang dibutuhkan pada halaman beranda, yaitu <i>user data</i> , layanan, berita dan saldo.	
04-RA	Pre-kondisi: mengisi parameter <i>form register mitra</i> .	Data tersimpan dan mengirimkan email berhasil daftar mitra. Dan menghasilkan <i>response</i> JSON data yang didaftarkan.	Sesuai dengan harapan .
05-RA	Pre-kondisi: input sebuah <i>authorisasi bearer token</i> JWT yang sudah dihasilkan ketika konfirmasi kode OTP dan mengisi parameter <i>form</i> yang dibutuhkan untuk melakukan <i>order</i> layanan.	Menghasilkan sebuah <i>response</i> berupa JSON yaitu berisikan sebuah data <i>order</i> dan menghasilkan sebuah <i>invoice</i> yang berisikan terdapat sebuah id mitra yang didapat terdekat dari lokasi <i>order</i> . Dan berisikan sebuah data mitra yang didapat.	Sesuai dengan harapan .

b. Pengujian Persyaratan Fungsional Aplikasi Website

Berikut adalah beberapa hasil pengujian persyaratan fungsional aplikasi *website*:

Tabel 3.2. Pengujian Persyaratan Fungsional Aplikasi Website.

ID	Deskripsi atau Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian
01-AW	Pre-kondisi: Masuk halaman <i>login</i> . Mengisi <i>form</i> dengan	Masuk ke halaman <i>dashboard</i> .	Sesuai dengan harapan .

	<i>username</i> atau <i>password</i> dengan benar.		
02-AW	Pre-kondisi: Masuk halaman <i>list</i> layanan. Pilih salah satu layanan kemudian klik button berwarna kuning dengan logo pensil.	Masuk ke halaman edit layanan.	Sesuai dengan harapan .
03-AW	Pre-kondisi: Masuk halaman edit layanan. Isi <i>form</i> sesuai atau dengan benar kemudian klik tombol <i>save</i> .	Data <i>terupdate</i> dan masuk ke halaman <i>list</i> layanan.	Sesuai dengan harapan .
04-AW	Pre-kondisi: masuk ke halaman detail calon mitra. Isi <i>form</i> tanggal untuk verifikasi calon mitra kemudian klik tombol berwarna kuning dengan tulisan <i>verify</i> .	Masuk ke halaman <i>list</i> daftar mitra dan calon mitra terverifikasi menjadi mitra.	Sesuai dengan harapan .

3.4.2. Pengujian Kinerja Aplikasi

Pada bab ini akan menjelaskan beberapa tentang hasil dari pengujian kinerja aplikasi yang ditujukan untuk menguji kinerja dari RESTful Api

yang sudah dikerjakan yaitu diukur dari kecepatan *response* yang dihasilkan.

Tabel 3.3. Pengujian Kinerja Aplikasi.

ID	Proses	Waktu Eksekusi	
		Offline (ms)	Online (ms)
01-PKA	<i>Register</i> Mitra	3950	8370
02-PKA	<i>Login</i>	2170	1455
03-PKA	Konfirmasi Kode OTP	310	300
04-PKA	<i>Get Data</i> Halaman <i>Home</i>	281	311
15-PKA	<i>Order</i> Layanan	9130	3120

3.4.3. Pengujian Usabilitas

Dalam bab ini akan menjelaskan tentang hasil dari pengujian usabilitas yang ditujukan pada aplikasi *website* dengan menguji tingkat kepuasan pengguna dalam menggunakan aplikasi *website* tersebut. Proses pengumpulan data untuk menguji tingkat kepuasan pengguna adalah dengan menyebarkan sebuah *form* kuisioner secara online yang berisikan sebuah *link* dari *website* yang sudah dibuat, akses *login website*, informasi pengguna dan soal kuisioner tentang tingkat kepuasan setelah menggunakan *website* tersebut.

Pada pengujian penulis melakukan penyebaran kuisioner yang ditujukan kepada rekan atau teman dari penulis yang dianggap tidak keberatan dalam mengisi sebuah kuisioner dengan menyebarkan kuisioner melalui pesan pada aplikasi whatsapp dan instagram secara satu per-satu. Hasil tanggapan yang didapat dalam penyebaran kuisioner tersebut adalah dengan respon 88 tanggapan. Dari hasil tanggapan tersebut populasi mahasiswa terdapat 57 responden dan non mahasiswa 31 responden. Berikut adalah hasil tanggapan dari kuisioner yang sudah disebarkan:

Tabel 3.4. Hasil Pengujian Usabilitas Aplikasi *Website*.

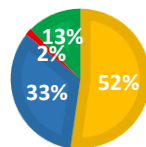
No	Keterangan			
1.	Reliabilitas Data yang Digunakan			
	Kurang	Cukup	Baik	Memuaskan
	0%	4.6%	60.2%	35.2%
2.	Navigasi atau Tata Letak Fitur			
	Kurang	Cukup	Baik	Memuaskan
	0%	5.7%	51.1%	43.2%
3.	Daya Tanggap atau Respon Sistem			
	Kurang	Cukup	Baik	Memuaskan
	0%	9.1%	53.4%	37.5%

4.	Keterpaduan Komunikasi Data			
	Kurang	Cukup	Baik	Memuaskan
	0%	9.1%	61.4%	29.5%
5.	Pengendalian Penggunaan Fitur			
	Kurang	Cukup	Baik	Memuaskan
	6.8%	22.7%	51.1%	19.3%
6.	Penjaminan Keamanan			
	Kurang	Cukup	Baik	Memuaskan
	2.3%	18.2%	53.4%	26.1%
7.	Kualitas Informasi Data			
	Kurang	Cukup	Baik	Memuaskan
	0%	10.2%	48.9%	40.9%
8.	Efisiensi Proses atau Fitur			
	Kurang	Cukup	Baik	Memuaskan
	8%	36.4%	37.5%	18.2%
9.	Efektifitas Penggunaan Fitur			
	Kurang	Cukup	Baik	Memuaskan
	0%	6.8%	55.7%	37.5%
10.	Tingkat Kepuasan Penggunaan Aplikasi			
	Kurang	Cukup	Baik	Memuaskan
	0%	2.3%	50%	47.7%
Total				
	Kurang	Cukup	Baik	Memuaskan
	17.1%	125.1%	522.7%	335.1%
Persentase Keseluruhan				
	Kurang	Cukup	Baik	Memuaskan
	1.7%	12.5%	52.3%	33.5%

Hasil persentase keseluruhan dari opsi atau keterangan kuisioner dapat dilihat pada diagram pie chart berikut ini.

HASIL KUISIONER

■ Baik ■ Memuaskan ■ Kurang ■ Cukup



Gambar 3.19. Diagram Pie Chart Hasil Kuisioner.

Dilihat dari diagram pie chart hasil kuisioner yang telah dilakukan, maka dapat disimpulkan bahwa hasil penilaian dari responden mengenai tingkat kepuasan dalam pengujian usability mendapatkan hasil yang baik dan memuaskan. Hasil tersebut diambil dari hasil respon kuisioner responden dalam menggunakan aplikasi *website* fitcare dengan menggunakan variabel yang terdapat pada tabel hasil pengujian usability.

4. KESIMPULAN DAN SARAN

Pada bab ini menjelaskan tentang hasil dari kesimpulan dan saran yang diberikan penulis pada penelitian ini. Berikut adalah kesimpulan dan saran yang diberikan oleh penulis:

4.1. Kesimpulan

Berdasarkan hasil perancangan dan implementasi yang sudah dilakukan mendapatkan sebuah kesimpulan sebagai berikut:

- Dengan tahap penelitian yang dilakukan oleh penulis adalah dengan melakukan observasi atau wawancara langsung ke perusahaan maka penulis dapat menganalisis kebutuhan data pada aplikasi *website* dan aplikasi *e-commerce* yang menggunakan RESTful API sebagai penghubung antara aplikasi *e-commerce* dengan server.
- Dengan menggunakan arsitektur RESTful API dapat meningkatkan kinerja aplikasi *m-commerce* berbasis *android* dalam menyimpan sebuah data ke database, yaitu dengan URI yang menggunakan metode *request* HTTP yang digunakan untuk kebutuhan yang terdapat pada aplikasi *m-commerce* berbasis *android*. Arsitektur RESTful API tersebut menghasilkan sebuah keluaran atau *response* berupa JSON yang lebih ringkas dan lebih mudah dipahami oleh pengembang aplikasi *m-commerce*.
- Berdasarkan hasil implementasi arsitektur RESTful API pada proses *login* yang membutuhkan sebuah kode OTP untuk memverifikasi sebuah nomor HP menghasilkan sebuah *response* JSON berupa kode token dengan menggunakan tipe token bearer token. Kode token tersebut adalah sebuah JWT yang memiliki 3 karakter, yaitu header, payload dan signature. Dari hasil pengujian yang sudah dilakukan, dengan mengimplementasikan JWT dapat menjamin keamanan data yang ada ketika proses *request/response* data pada aplikasi *e-commerce* yaitu dengan autentikasi menggunakan bearer token tersebut.
- Dalam penelitian ini untuk menguji RESTful API yang sudah dikerjakan adalah dengan menggunakan metode pengujian black box *testing* dengan tipe pengujian adalah persyaratan fungsionalitas dan kinerja aplikasi. Pada pengujian dengan tipe persyaratan fungsionalitas adalah dengan

menguji keberhasilan seluruh URI yang sudah dihasilkan dengan metode *request* masing-masing API dengan *response* JSON dengan kode HTTP 200 yaitu kode berhasil atau sukses. Dan pada pengujian kinerja aplikasi adalah dengan menguji kecepatan *response* yang dihasilkan pada saat memanggil sebuah URI yang sudah dihasilkan. Dengan kedua tipe pengujian metode black box *testing* tersebut maka penulis sudah merasa cukup dengan pengujian terhadap RESTful API yang dikerjakan.

4.2. Saran

Dalam penelitian ini, penulis menemukan sebuah masalah yaitu dalam mengirimkan sebuah kode OTP melalui email. Dalam penggunaan proses tersebut, waktu pengiriman email tidak efisien. Jadi dalam proses konfirmasi kode OTP yang dilakukan banyak terjadi keterlambatan input kode OTP tersebut. Jadi adapun saran yang dapat diberikan dalam penelitian ini adalah dalam mengirimkan sebuah kode OTP untuk aplikasi *android* sebaiknya melalui SMS. Karena dengan menggunakan SMS waktu pengiriman kode OTP akan menjadi lebih efisien.

DAFTAR PUSTAKA

- [1] G. B. Santoso, M. R. Fadilla, and A. Ibrahim, "Implementasi E-Commerce Berbasis Open Source Pada Usaha Online Shopping Fashion," *Semin. Nas. Pakar ke 2 Tahun 2019*, pp. 1–6, 2019.
- [2] A. Firdaus, S. Widodo, A. Sutrisman, S. G. F. Nasution, and R. Mardiana, "Rancang Bangun Sistem Informasi Perpustakaan Menggunakan Web Service Pada Jurusan Teknik Komputer POLSRI," vol. 5, no. 2, pp. 81–87, 2019.
- [3] N. M. Widani, A. P. Abiyasa, G. S. Darma, and A. F. Maradona, "Menguji Ketajaman Implementasi E-Commerce Dalam Penjualan Kamar Hotel di Bali," vol. 16, no. 2, pp. 79–98, 2019.
- [4] R. Rizal and A. Rahmatulloh, "Restful Web Service Untuk Integrasi Sistem Akademik Dan Perpustakaan Universitas Perjuangan," *J. Ilm. Inform.*, vol. 7, no. 01, p. 54, 2019.
- [5] R. C. Buwono, "Web Services Menggunakan Format JSON," vol. XIV, pp. 1–10, 2019.
- [6] A. Warda, P. Putra, A. Bhawiyuga, and M. Data, "Implementasi Autentikasi JSON Web Token (JWT) Sebagai Mekanisme Autentikasi Protokol MQTT Pada Perangkat NodeMCU," vol. 2, no. 2, pp. 584–593, 2018.
- [7] A. Aminuddin, "Cara Efektif Belajar Framework LARAVEL," 2018.
- [8] R. Choirudin and A. Adil, "Implementasi Rest Api Web Service dalam Membangun Aplikasi Multiplatform untuk Usaha Jasa," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 18, no. 2, pp. 284–293, 2019.
- [9] H. Sy and I. Intan, "Implementasi Restful Api Portal Akademik Stmik Dipanegara Berbasis Android," vol. VIII, no. 2, pp. 63–70, 2019.
- [10] S. A. L. Ghozaly, "implementasi rest api pada pusat informasi mahasiswa universitas teknologi yogyakarta," 2019.
- [11] A. Hermanto, S. Supangat, and F. Mandita, "Evaluasi Usabilitas Layanan Sistem Informasi Akademik Berdasarkan Kombinasi ServQual dan Webqual Studi Kasus : SIAKAD Politeknik XYZ," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 3, no. 1, p. 33, 2017.
- [12] A. Hermanto and S. E. Sartono, "PEMANFAATAN WEB USAGE MINING PADA E-COMMERCE UNTUK PENINGKATAN PENJUALAN PADA RETAILER ONLINE," vol. 105, no. 3, pp. 129–133, 2016.
- [13] JSON.org, "Pengenalan JSON". [Online]. Tersedia pada: <https://www.json.org/json-id.html>, diakses pada 28 November 2019.
- [14] Hermawan, Achmed Islamic, "Apa itu Android? Penjelasan Super Lengkap Android ada di Sini", 29 Oktober 2018. [Online]. Tersedia pada: <https://windowsku.com/apa-itu-android-adalah/>, diakses pada 18 Oktober 2019.
- [15] Garuda Cyber Indonesia, "Langkah Untuk Memahami JSON Web Token Dengan Mudah", 04 Oktober 2018. [Online]. Tersedia pada: <https://garudacyber.co.id/artikel/1110-langkah-untuk-memahami-json-web-token-dengan-mudah>, diakses pada 5 Desember 2019.