

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi di era modern saat ini begitu pesat dan menyebabkan segala macam aktivitas masyarakat dipengaruhi teknologi yang sudah ada. Salah satu aktivitas yang terkena dampak teknologi adalah sistem jual beli yang saat ini dikenal dengan *E-commerce*. *E-commerce* memberikan pengalaman baru dalam transaksi jual beli produk. Kemudahan dalam memilih produk dengan harga bersaing menjadi daya tarik tersendiri bagi pembeli. Perkembangan *e-commerce* juga memberi solusi bagi pelaku usaha membuka pasar lebih luas dengan penerapan beberapa aplikasi dan layanan yang membantu pelaku usaha dalam mengelola sumber daya bisnis [1]. Namun karena kemudahan yang diberikan terdapat pelaku usaha yang melakukan kecurangan terhadap produk yang dijual, misal pada penjualan sepatu yang menyebutkan bahwa ini adalah produk asli dari *brand* ternama tetapi ketika periksa produk tersebut adalah barang tiruan.

Tukutu merupakan aplikasi titip jual beli *online* sepatu yang memberikan jaminan keaslian. Tukutu ini merupakan *marketplace* titip jual sepatu baik itu sepatu baru, bekas ataupun sepatu langka, hal tersebut memudahkan penggemar sepatu di Indonesia dapat menemukan sebuah sarana jual beli sepatu yang aman. Saat ini Tukutu menggunakan *platform Android* dan berbasis *web*, penggunaan *web* sendiri sebagai tempat untuk menyimpan data transaksi dan alur sistem jual beli di Tukutu. Kedua *platform* ini tersebut harus saling terintegrasi dan setiap transaksi yang dilakukan harus di simpan. Untuk menangani integrasi pada *platform* yang berbeda dapat mengimplementasikan teknologi *web service*.

*Web Service* merupakan mekanisme interaksi antar sistem yang menunjang interoperabilitas untuk kepentingan integrasi data yang dapat diakses oleh berbagai *platform* melalui internet [2]. *Web Service* dapat menjembatani komunikasi antar *platform* yang berbeda menggunakan protokol HTTP. Salah satu metode yang dapat digunakan ketika memakai *web service* yaitu REST (*Representational State Transfer*) atau RESTful, merupakan *standart* arsitektur komunikasi berbasis web yang kerap digunakan pada pengembangan layanan berbasis *web*. Adapun metode *web service* yang lain dengan menggunakan SOAP, terdapat penelitian tentang performa RESTful dengan SOAP yang membandingkan *response time* dan *message size* [3]:

**Tabel 1.1: Perbandingan performa REST dengan SOAP**

<b>Jumlah dari Siswa</b>	<b>Ukuran pesan (byte)</b>		<b>Waktu (ms)</b>	
	<b>SOAP / HTTP</b>	<b>REST (HTTP)</b>	<b>SOAP / HTTP</b>	<b>REST (HTTP)</b>
Informasi ringan dari 1 siswa	3370	374	29.01	13.33
Informasi berat dari 2 siswa	16055	2079	63.82	26.51
Informasi berat dari 3 siswa	29155	5281	100.06	43.38
Informasi berat dari 5 siswa	37001	7566	233.68	86.82
Informasi berat dari 6 siswa	50455	13273	286.57	117.58

Dari tabel diatas dapat dibuktikan bahwa REST lebih performa lebih baik dibanding SOAP. RESTful lebih populer karena *method request* yang digunakan mirip dengan metode web CRUD (*Create*, *Read*, *Update*, dan *Delete*). Hasil

*response* / pesan yang sering digunakan pada RESTful berupa JSON (*Javascript Object Notation*). Selain JSON ada juga XML yang digunakan sebagai response web service, namun memiliki performa yang kurang baik berdasarkan penelitian yang membandingkan kelebihan JSON dengan XML [4]:

**Tabel 1.2: Kelebihan JSON dengan XML**

JSON	XML
Teknik yang terprogram untuk deserialisasi dan serialisasi objek JavaScript, dengan sedikit coding.	Kode JavaScript ditulis oleh pengembang untuk serialisasi dan deserialisasi untuk membentuk XML.
Kebanyakan <i>browser</i> sekarang sudah mendukung JSON.	Semua browser versi terbaru sudah terpasang XML <i>parser</i> tapi sedikit rumit untuk <i>parsing</i> ke <i>cross-browser</i> .
Format ini sangat sesuai untuk memiliki pendekatan berbasis nama / nilai pasangan.	Karena <i>tag</i> dan <i>namespace</i> , formatnya sangat panjang.
Deserialisasi sangat cepat di JavaScript	Deserialisasi lambat di JavaScript.
Kebanyakan JavaScript libraries dan AJAX <i>toolkits</i> memiliki dukungan yang baik menggunakan JSON.	AJAX <i>toolkits</i> tidak cukup kuat memiliki dukungan.
Memiliki API sederhana untuk JS dan untuk bahasa lainnya.	API sangat rumit

Dari hasil perbandingan tersebut JSON memiliki banyak dukungan browser, libraries, dan memiliki API yang sederhana. Format penulisan JSON juga sederhana sehingga memudahkan pengembang dalam mengatur *response* / pesan yang akan dikembalikan.

Berdasarkan masalah yang telah diuraikan maka dilakukan penelitian yang berjudul “**PERANCANGAN DAN IMPLEMENTASI WEB SERVICE PADA SISTEM JUAL BELI TUKUTU MENGGUNAKAN REST**” sebagai pemecahan masalah integrasi *platform* yang berbeda pada aplikasi Tukutu.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka di dapatkan rumusa masalah sebagai berikut :

1. Bagaimana merancang dan mengimplementasikan REST *web service* pada sistem jual beli Tukutu

## 1.3 Batasan Masalah

Dalam penelitian ini terdapat batasan masalah sehingga pembahasan yang dilakukan tidak menyimpang dari tujuan penelitian. Berikut batasa masalah yang di dapat:

1. Web service yang dibangun menggunakan framework Laravel dan Lumen.
2. Metode yang digunakan dalam membangun web service adalah REST.
3. Implementasi yang dilakukan hanya pada aplikasi Tukutu.
4. Berfokus pada jual beli antara pembeli dan penjual di Tukutu.

## 1.4 Tujuan Penelitian

Adapun tujuan yang dilakukan penelitian ini:

1. Merancang dan implementasi *web service* dengan metode REST pada aplikasi Tukutu.

## 1.5 Manfaat Penelitian

Manfaat yang diperoleh dari penilitian ini:

### 1.5.1 Bagi Akademik

- a) Sebagai indikator sejauh mana pemahaman yang telah di dapat saat menuntut ilmu di Universitas Dian Nuswantoro.
- b) Menambah jenis penelitian pada Universitas Dian Nuswantoro yang dapat digunakan sebagai referensi penelitian lainnya.
- c) Menjalin hubungan kerja sama antara Universitas Dian Nuswantoro dan Tukutu.

### 1.5.2 Bagi Tukutu

- a) Menjalin hubungan kerja sama antara Tukutu dan Universitas Dian Nuswantoro.
- b) Dengan adanya penelitian ini, Tukutu dapat terbantu pada sistem jual belinya.

### 1.5.3 Bagi Penulis

- a) Menambah pengetahuan dan pengalaman kerja pada instansi tempat penelitian dilakukan.
- b) Penulis dapat mengembangkan ilmu yang telah di dapat selama berada di Universitas Dian Nuswantoro.

### 1.5.4 Bagi Pembaca

- a) Menambah wawasan tentang *web service*.
- b) Dapat digunakan sebagai bahan refrensi penelitian dan pengembangan selanjutnya.

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Studi

Penelitian yang dilakukan berkaitan dengan milik Dedofin dan Bagus Mulyawan membuat sistem penjualan untuk mengantikan proses pencatatan penjualan yang masih manual dengan mengimplementasikan *Restful API* sebagai *web service* dan metode Regresi Linear untuk peramalan stok barang bulan berikutnya [5].

Pada penelitian yang berjudul “RESTFul Web Service Untuk Sistem Pencatatan Transaksi Studi Kasus PT. XYZ” yang dilakukan oleh Penidas Fiodinggo Tanaem, Danny Manongga dan Ade Iriani. Menghasilkan sistem pencatatan transaksi yang saling terintegrasi dengan *platform* dan *database* yang berbeda pada PT. XYZ menggunakan *Restful API* [6]. Hasil dari penelitian yang didapat bahwa Restful API dapat menghubungkan *platform* yang berbeda dan *database* yang berbeda pula.

Sedangkan penelitian milik Alam Rahmatulloh, Heni Sulastri dan Rizal Nugroho yang berjudul “Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512” yang membandingkan algoritma SHA-512 dengan algoritma SHA-256 dalam pembuatan token JWT (JSON Web Token). Hasilnya penggunaan algoritma SHA-512 lebih cepat 1% dibandingkan dengan algoritma SHA-256 dan menghasilkan nilai hash 2% lebih besar dibandingkan algoritma SHA-256, menjadikan keamanan pertukaran data lebih baik karena token lebih panjang.

**Tabel 2.1: State of The Art**

No	Nama Penulis	Tahun	Judul	Masalah	Metode	Hasil
1	Dedofin dan Bagus Mulyawan.	2018	Perancangan Sistem Penjualan Barang Dan Jasa Untuk Toko Intikarya Aluminium	Proses pencatatan penjualan dari perusahaan tersebut menggunakan Kertas, Microsoft Word dan Excel.	Menggunakan REST API dan mengimplementasikan metode Regresi Linear.	Sistem Penjualan Barang dan Jasa yang mengimplementasikan REST API dan metode Regresi Linear untuk peramalan stok bulan berikutnya .
2	Penidas Fiodingga Tanaem, Danny Manongga dan Ade Iriani.	2016	RESTFul Web Service Untuk Sistem Pencatatan Transaksi Studi Kasus PT. XYZ	Pada PT. XYZ memiliki banyak sistem dan <i>platform</i> yang berbeda	Penerapan Restful Web Service dan JWT (JSON Web Token) sebagai akses keamanan.	Penggunaan Restful dapat mengintegrasikan data pada sistem dan <i>platform</i>

				sehingga memberikan dampak pada data gudang dan pencatatan transaksi, karena <i>database</i> yang digunakan tiap aplikasi berbeda.		yang berbeda pada PT. XYZ yang dijalankan pada jaringan local.
3	Alam Rahmatulloh, Heni Sulastri dan Rizal Nugroho.	2018	Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512	Perbandingan kinerja algoritma SHA-512 dan SHA-256 dalam proses pembuatan token.	Algoritma SHA-512 dan SHA-256	Penerapan algoritma SHA-512 lebih cepat 1% dengan algoritma SHA-256 dan menghasilkan nilai <i>hash</i> 2% lebih besar dari SHA-256

						sehingga token lebih panjang.
--	--	--	--	--	--	-------------------------------

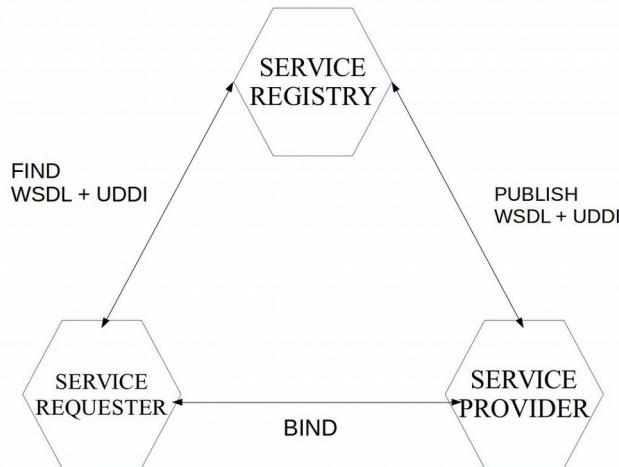
### **Kesimpulan:**

Penggunaan REST API pada *Web service* telah banyak diterapkan pada pembuatan sistem, karena kemampuanya yang dapat mengintegrasikan data pada *platform* yang berbeda dengan menggunakan protokol HTTP. Dalam keamanan integrasi data menggunakan JWT (JSON Web Token) sebagai otentikasi dalam mengakses data pada *server*.

## **2.2 Tinjauan Pustaka**

### **2.2.1 Web Service**

Web Service adalah sebuah entitas komputasi yang dapat diakses melalui jaringan internet maupun intranet dengan standar protokol tertentu dalam platform dan antarmuka bahasa pemrograman yang independen [2]. Web Service mampu menjembatani komunikasi antar program karena tidak dipengaruhi oleh platform, arsitektur, maupun bahasa pemrograman. Sehingga setiap platform yang berada pada satu jaringan sama atau pada jaringan yang berbeda dapat saling berkomunikasi dengan menggunakan protokol yang ditetapkan oleh web service seperti HTTP.



**Gambar 2.1: Arsitektur Web Service**

Peran service pada arsitektur:

1. Service Provider

Service provider adalah penyedia layanan web. Penyedia layanan mengimplementasikan layanan dan membuatnya tersedia di Internet.

2. Service Requester

Ini adalah konsumen dari web Service. Peminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.

3. Service Registry

Berfungsi sebagai lokasi central yang mendeskripsikan semua layanan/service yang telah di-register.

### 2.2.2 REST

Konsep REST pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000. REST merupakan standar arsitektur komunikasi yang digunakan pada pengembangan layanan berbasis web. Sistem yang menggunakan prinsip-prinsip dari REST dapat disebut dengan “RESTful”. Cara kerja RESTful dimulai dari client melakukan *request* melalui *HTTP Request*, kemudian server merespon melalui *HTTP Response* [7]. Komponen dari *HTTP Request* adalah sebagai berikut:

- **Verb. HTTP Method**, yang digunakan di antaranya GET, PUT, POST, DELETE.
- **Uniform Resource Identifier (URI)**, untuk mengidentifikasi lokasi *resource* pada *server*.
- **HTTP Version**, menjelaskan versi dari HTTP yang akan digunakan, contohnya HTTP v1.1.
- **Request Header**, berisi metadata untuk HTTP Request. Contohnya adalah tipe client/browser, format yang didukung oleh client, format dari body pesan, dan setting cache.
- **Request Body**, konten data yang dikirimkan.

Sedangkan komponen pada HTTP Response berisikan:

- **Status/Response Code**, menjelaskan status *server* pada resource yang di *request*.
- **HTTP Version**

- **Response Header**, berisi metadata untuk HTTP Response. Contohnya tipe *server*, tipe content, panjang content, dan waktu *response*.
- **Response Body**, berisi konten data dari hasil *request*.

### 2.2.3 JSON

JSON (JavaScript Object Notation) merupakan suatu format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan oleh komputer. Sebelum adanya JSON, XML biasa digunakan untuk saling bertukar data. Format penulisan JSON tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh bahasa pemrograman keluarga C seperti C, C++, C#, Perl, Java, JavaScript, Python dan sebagainya. Oleh karena sifat – sifat tersebut, hal ini menjadikan JSON ideal sebagai bahasa pertukaran data [5].

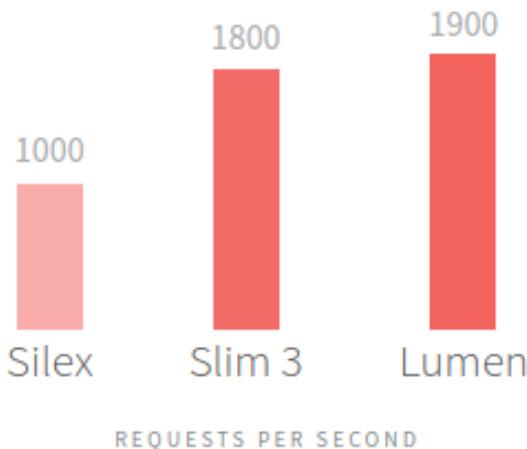
### 2.2.4 Laravel

Laravel dikembangkan oleh Taylor Otwell, dirilis pada tanggal 5 Juni 2011 dan masih terus berkembang sampai sekarang. Laravel merupakan web application framework berbasis PHP yang open source dan menggunakan konsep MVC (Model, View, Controller) [8]. MVC memisahkan aplikasi berdasarkan komponen – komponen aplikasi, seperti manipulasi data, controller dan user interface.

- Model merupakan bagian yang berfungsi untuk mengakses database.
- View merupakan desain antarmuka yang langsung berinteraksi dengan pengguna.
- Controller merupakan bagian yang memproses permintaan dari pengguna.

### 2.2.5 Lumen

Lumen merupakan *micro-framework* yang dibuat dari Laravel. Lumen merupakan versi kecilnya Laravel yang difokuskan pada pengembangan web dengan menggunakan konsep Rest API. Terdapat beberapa micro-framework yang dapat digunakan untuk membuat Rest API pada PHP seperti Slim Framework, Silex, dan lain – lain. Perbedaan Lumen dari micro-framework lainnya adalah kemampuannya dalam menangani Request hingga 1900 Request/detik.



**Gambar 2.2: Benchmark Request Lumen**

### 2.2.6 API

*Application Programming Interface* atau disingkat API yang memungkinkan developer untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan. API terdiri dari berbagai elemen seperti fungsi, protokol dan tools lainnya yang memungkinkan pengembang untuk membuat aplikasi. Tujuan penggunaan API yaitu untuk mempercepat proses development dengan menyediakan function secara terpisah sehingga pengembang

tidak perlu membuat fitur yang serupa [2]. Pada pengembangan web integrasi data menggunakan format penulisan yang sering digunakan seperti XML atau JSON, pada penelitian ini menggunakan format JSON sebagai format integrasi data yang digunakan.

#### 2.2.7 JSON Web Token (JWT)

JWT adalah sebuah token dengan format JSON yang dienkripsi, isi dari token tersebut berupa informasi pengguna yang gunanya untuk melakukan sistem autentikasi dan pertukaran informasi [7]. Sebuah token umumnya diterbitkan oleh penyedia layanan dan dikonsumsi oleh pihak yang mengandalkan konten untuk mengidentifikasi subjek token dengan tujuan yang berhubungan dengan keamanan [6].

```
xxxxxxxx.yyyyyyyy.zzzzzzzz
header.payload.signature
```

*Gambar 2.3: Struktur JSON Web Token (JWT)*

JSON Web Token terdiri dari tiga bagian yang dipisahkan oleh titik “.” yaitu Header, Payload dan Signature. Header biasanya terdiri dari dua bagian: jenis token , yaitu JWT, dan algoritma hashing seperti HMAC SHA256.

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

*Gambar 2.4: JWT Header*

Bagian kedua dari JWT adalah payload, yang berisi klaim. Klaim adalah pernyataan tentang suatu entitas (biasanya, pengguna) dan *metadata* tambahan. Ada tiga jenis klaim, yaitu *registered*, *public*, dan *private claims*.

- Registered claims

Ini adalah seperangkat klaim yang telah ditentukan sebelumnya yang tidak wajib tetapi direkomendasikan, untuk memberikan serangkaian klaim yang berguna dan dapat dioperasionalkan. Beberapa di antaranya adalah:

- iss (Penerbit) mengidentifikasi pokok yang menerbitkan JWT. Pemrosesan klaim ini umumnya khusus untuk aplikasi. Nilai "iss" adalah string case-sensitive yang berisi nilai String atau URI. Penggunaan klaim ini adalah OPTIONAL atau tidak wajib.
- exp (Waktu kedaluwarsa) mengidentifikasi waktu kedaluwarsa pada atau setelah mana JWT tidak harus diterima untuk diproses. Pemrosesan klaim "exp" mensyaratkan bahwa tanggal / waktu saat ini harus sebelum tanggal / waktu kedaluwarsa yang tercantum dalam klaim "exp".
- sub (Subjek) mengidentifikasi pokok yang menjadi subjek JWT. Klaim dalam JWT biasanya merupakan pernyataan tentang subjek. Nilai subjek harus dicakup untuk menjadi unik secara lokal dalam konteks penerbit atau unik secara global. Pemrosesan klaim ini umumnya khusus untuk aplikasi. Nilai "sub" adalah string case-sensitive yang berisi nilai String atau URI. Penggunaan klaim ini adalah optional atau tidak wajib.
- aud (Audiens) mengidentifikasi penerima yang dimaksudkan untuk JWT. Setiap pelaku dimaksudkan untuk memproses JWT harus mengidentifikasi dirinya dengan nilai dalam klaim audiens. Jika pelaku

memproses klaim tidak mengidentifikasi dirinya dengan nilai dalam klaim "aud" ketika klaim ini hadir, maka JWT harus ditolak. Dalam kasus umum, nilai "aud" adalah array dari string *case-sensitive*, masing-masing berisi nilai String atau URI. Dalam kasus khusus ketika JWT memiliki satu pemirsa, nilai "aud" mungkin menjadi *case-sensitive* huruf tunggal yang berisi nilai String atau URI. Penafsiran nilai-nilai audiens umumnya spesifik aplikasi. Penggunaan klaim ini adalah optional atau tidak wajib.

- nbf (Bukan sebelumnya) mengidentifikasi waktu sebelum JWT tidak harus diterima untuk diproses. Pemrosesan klaim "nbf" mensyaratkan bahwa tanggal / waktu saat ini HARUS setelah atau sama dengan tanggal / waktu tidak-sebelum yang tercantum dalam klaim "nbf". Pelaksana mungkin menyediakan beberapa kelonggaran kecil, biasanya tidak lebih dari beberapa menit, untuk memperhitungkan kemiringan jam. Nilainya harus berupa angka yang berisi nilai Tanggal Numerik. Penggunaan klaim ini adalah optional atau tidak wajib.
- iat (Dikeluarkan pada) mengidentifikasi waktu ketika JWT diterbitkan. Klaim ini dapat digunakan untuk menentukan usia JWT. Nilainya harus berupa angka yang berisi nilai Tanggal Numerik. Penggunaan klaim ini adalah optional atau tidak wajib.
- jti (JWT ID) menyediakan pengidentifikasi unik untuk JWT. Nilai pengidentifikasi harus ditetapkan dengan cara yang memastikan bahwa ada kemungkinan yang dapat diabaikan bahwa nilai yang sama akan secara tidak sengaja ditetapkan ke objek data yang berbeda; jika aplikasi menggunakan beberapa emiten, tabrakan harus dicegah di antara nilai yang dihasilkan oleh emiten yang berbeda juga. Klaim "jti" dapat digunakan untuk mencegah JWT diputar ulang. Nilai "jti" adalah

string case-sensitive. Penggunaan klaim ini adalah optional atau tidak wajib.

- Public claims

Ini dapat didefinisikan sesuka hati oleh mereka yang menggunakan JWT. Tetapi untuk menghindari tabrakan mereka harus didefinisikan dalam *IANA JSON Web Token Registry* atau didefinisikan sebagai URI yang berisi *namespace* tahan tabrakan.

- Private claims

Ini adalah klaim khusus yang dibuat untuk membagikan informasi antara pihak-pihak yang setuju untuk menggunakannya dan bukan merupakan klaim terdaftar atau publik.

```
{  
  "iss": "Tukutu",  
  "exp": 12345678,  
  "sub": 123,  
  "jti": "elns-slkn-ss",  
  "name": "elfahos",  
  "merchant": true  
}
```

**Gambar 2.5: JWT Payload**

Bagian ketiga dari JWT adalah signature, berisi hash dari komponen-komponen header, payload, dan kunci rahasia. Contoh JWT Signature ini menggunakan algoritme HMAC SHA-256.

```
HMACSHA256(  
    base64UrlEncode(header) + "." +  
    base64UrlEncode(payload),  
    secret)
```

*Gambar 2.6: JWT Signature*

Hasil keluaran berupa token yang berisi struktur data JSON yang dienkripsi dengan panjang 256 bit dari hasil enkripsi menggunakan algoritme HMAC SHA-256. Penerapan JWT dapat digunakan pada 2 kondisi yaitu kondisi saat autentikasi dan pertukaran informasi. Autentikasi merupakan kondisi yang umum ditemukan untuk menggunakan JWT, setelah pengguna login. Setiap permintaan yang dilakukan oleh client harus menyertakan JWT, yang dapat memungkinkan pengguna untuk akses router, service, dan sumber daya yang diizinkan dengan token yang dibuat.

#### 2.2.8 Firebase Cloud Messaging (FCM)

Firebase Cloud Messaging (FCM) adalah solusi perpesanan lintas platform yang memungkinkan Anda mengirimkan pesan secara andal tanpa biaya.

Menggunakan FCM, Anda dapat memberi tahu aplikasi klien bahwa email baru atau data lain tersedia untuk disinkronkan. Anda dapat mengirim pesan notifikasi untuk mendorong keterlibatan dan retensi pengguna. Untuk kasus penggunaan seperti olahpesan cepat, pesan dapat mentransfer muatan hingga 4KB ke aplikasi klien [12].

### 2.2.9 MySQL

MySQL dikembangkan oleh sebuah perusahaan Swedia yang bernama MySQL AB, yang pada saat itu bernama TcX DataKonsult AB, sejak sekitar tahun 1994-1995. MySQL mempunyai fitur yang mudah dipelajari bagi para penggunanya dan dikembangkan untuk menangani database yang besar dengan waktu yang lebih singkat. Kecepatan, konektivitas, dan keamanannya yang lebih baik membuat MySQL sangat dibutuhkan untuk mengakses database di internet [13].

MySQL merupakan perangkat lunak dibawah lisensi GPL (General Public License). Pengguna dapat secara bebas menggunakan MySQL, tetapi dengan batasan perangkat lunak tersebut, pengguna tidak boleh dijadikan produk turunan yang bersifat komersial MySQL dapat berjalan di atas banyak sistem operasi seperti Solaris, MAC OS X, linux, dan FreeBSD [14].

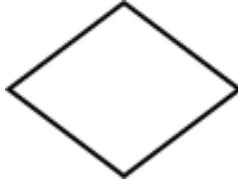
MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *database* yang terbagi menjadi tiga bagian, yaitu DDL, DML dan DCL [15].

1. Data Definition Language (DDL) merupakan kelompok perintah yang berfungsi untuk mendefinisikan atribut – atribut basis data, tabel, atribut (kolom), batasan – batasan terhadap suatu atribut, serta hubungan antar tabel.
2. Data Manipulation Language(DML) adalah kelompok perintah yang berfungsi untuk memanipulasi data dalam *database*, misalnya untuk pengambilan, penyisipan, pengubahan, dan penghapusan data.
3. Data Control Language(DCL) berisi perintah-perintah untuk mengendalikan pengaksesan data. Pengendalian dapat dilakukan berdasarkan pengguna, tabel, kolom maupun operasi yang boleh dilakukan.

### 2.2.10 Flowchart

Flowchart yaitu penggambaran secara grafik dari setiap langkah – langkah dan urutan prosedur dari suatu program. Dengan membuat *flowchart* dapat membantu *analyst* dan *programmer* untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasiannya. *Flowchart* di visualkan dengan bentuk gambar diagram yang memiliki aliran satu atau dua arah secara sekuensial.

**Tabel 2.2: Komponen Flowchart**

Simbol	Nama	Keterangan
	Terminator	Menunjukkan awal atau akhir sebuah proses.
	Process	Menyatakan kegiatan yang akan terjadi dalam diagram
	Decision	Proses / langkah di mana perlu adanya keputusan atau adanya kondisi tertentu. Di titik ini selalu ada dua keluaran untuk melanjutkan aliran kondisi yang berbeda.

	Input/Output Data	Digunakan untuk mewakili data masuk, atau data keluar.
	Flow Line	Menunjukkan arah aliran algoritme, dari satu proses ke proses berikutnya.

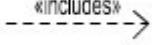
#### 2.2.11 Use Case Diagram

*Unified Modeling Language* (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML adalah sebuah tool yang dikembangkan dengan paradigma Object – Oriented. UML menyediakan beberapa diagram untuk memodelkan aplikasi berorientasi objek, salah satunya yaitu use case diagram.

*Use Case* mewakili bagaimana sistem berinteraksi dengan lingkungannya dengan menggambarkan aktivitas yang dilakukan oleh pengguna dan respon sistem. *Use case* diagram merupakan representasi interaksi pengguna dengan sistem dan menggambarkan spesifikasi dari kasus penggunaan. Elemen pada use case meliputi *actor*, *use case*, *system boundary*, dan *relationship*.

**Tabel 2.3: Komponen Use Case Diagram**

Gambar	Nama	Keterangan
	Actor	Mempresentasikan seseorang/himpunan peran yang dimainkan ketika berinteraksi dengan use case.

	Use Case	Deskripsi dari urutan aksi - aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	Association	Sebagai penghubung antara satu objek dengan objek lainnya.
	Includes	Kelakuan yang harus terpenuhi agar sebuah event dapat terjadi, dimana pada kondisi ini sebuah <i>use case</i> adalah bagian dari <i>use case</i> lainnya.
	Boundary	Batasan yang dari sistem terhadap lingkungan.

### 2.3 Deskripsi Tempat KKI

PT. Dian Nuswantoro Teknologi dan Informasi merupakan perusahaan di bidang teknologi komputerisasi yang berdiri sejak tahun 2006. PT Dian Nuswantoro Teknologi dan Informasi ini lebih dikenal dengan nama PT Dinustek berfokus pada pelayanan tentang informasi dan teknologi (IT) , Layanan jaringan internet (Internet Service Provider) dan layanan pembuatan aplikasi (Software Development).

Informasi tentang PT Dinustek :

Alamat : Jl. Arjuna No 36 ,Kota Semarang, Jawa Tengah, Indonesia.  
Telepon : (024) 3568492  
Email : marketing@dinustek.com  
Website : www.dinustek.com

### 2.3.1 Logo dan Makna Tempat KKI



*Gambar 2.7: Logo PT Dian Nuswantoro Teknologi dan Informasi*

Logo PT Dinustek terbagi menjadi 3 (tiga) bagian yaitu :

1. Simbol Api

Simbol api berwarna oranye melambangkan solusi untuk klien yang bermaksud memberikan penyelesaian masalah terbaik dari sisi teknologi dan informasi bagi bisnis klien.

2. Warna Biru

Warna biru melambangkan solusi teknologi informasi yang kami tawarkan dapat dipercaya kehandalan bagi klien.

### 3. Bentuk Huruf

Huruf-huruf yang tersusun berbentuk small caps melambangkan keramah-tamahan perusahaan kepada klien, dan nuansa kekeluargaan di perusahaan. Bentuk huruf yang dinamis namun kuat menggambarkan tim perusahaan yang kreatif dan solid.

#### 2.3.2 Struktur Organisasi Tempat KKI

Struktur organisasi PT Dinustek:

##### 1. Direktur Utama

Mohamad Sidiq, S.Si., M. Kom

##### 2. Direktur

Dr. Pulung Nurtantio Andono, ST, M. Kom

##### 3. Manager of Software Department

Abu Salam, M. Kom

##### 4. Kepala Divisi Stratup Software

Moh. Tofa Nurzaki, S. Kom

#### 2.3.3 Visi dan Misi Tempat KKI

Visi : Menjadi perusahaan teknologi informasi dan komunikasi kelas dunia.

Misi : Memberikan solusi keseluruhan teknologi informasi dan komunikasi untuk klien.

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Instrumen Penelitian**

Penyusunan suatu proyek penelitian, sesuai dengan data-data permasalahan pokok yang dihadapi. Data dapat dikatakan baik apabila data tersebut dapat mewakili objek yang sedang diteliti dan untuk mendapatkan data yang baik diperlukan metode atau serangkaian cara yang sesuai dengan kebutuhan penelitian.

#### **3.2 Prosedur Pengambilan Data**

Untuk mendapatkan data yang benar, akurat dan relevan serta sesuai dengan sumber data dan tujuan penyusunan Laporan Kuliah Kerja Industri ini, maka penulis dalam pengumpulan data menggunakan beberapa teknik antara lain adalah sebagai berikut :

##### **3.2.1 Observasi**

Teknik pengumpulan data dengan cara melakukan pengamatan dan terlibat secara langsung di dalam sistem pengembangan yang telah ada di dalam Tim Pengembang Aplikasi Tukutu yang berada di bawah pengawasan PT Dinustek.

##### **3.2.2 Studi Literatur**

Melakukan berbagai pencarian terhadap teori-teori dan literatur yang dapat digunakan sebagai dasar melakukan penelitian. Serta mempelajari penelitian yang pernah dilakukan sebelumnya sehingga dapat lebih memahami teori yang ada.

### 3.2.3 Forum Diskusi dan Tutorial

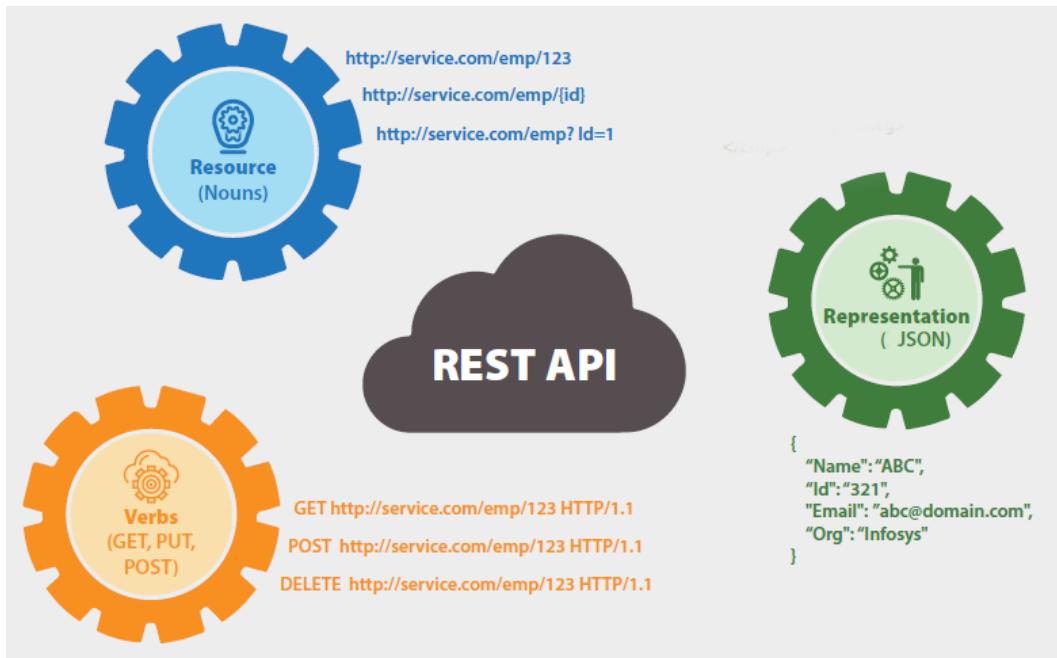
Teknik pengumpulan data dengan cara membuat sebuah forum diskusi, yang berisikan para pengembang aplikasi. Peneliti melakukan diskusi dengan sesama tim pengembang aplikasi Tukutu, guna untuk mengetahui fitur yang dibutuhkan dalam sistem.

## 3.3 Teknik Analisis Data

Data yang telah diperoleh kemudian dikumpulkan dan dianalisa. Hasil analisa data yang didapatkan akan dilakukan rumusan masalah dengan pengembangan fitur yang cocok sebagai solusi dari masalah tersebut. Solusi yang telah ditemukan diubah menjadi sistem yang terdiri dari fitur – fitur yang telah ditentukan, kemudian dari hasil tersebut dibuat alur sistem pembelian pada aplikasi Tukutu.

## 3.4 Metode

Dalam pembuatan web service pada sistem pembelian Tukutu, peneliti menggunakan metode REST/RESTful API. Metode ini cocok digunakan dalam pengembangan web yang menggunakan HTTP verb yang biasa digunakan untuk membuat CRUD (Create Read Update Delete), karena fungsi yang serupa pada HTTP verb yaitu POST, PUT, GET, dan DELETE. Proses yang diterapkan pada pembuatan web service yaitu:



**Gambar 3.1: REST Web Service**

### a) Resource

*Resources* adalah elemen dasar dari *web*. Ketika bekerja didalam REST, hal pertama yang di identifikasi adalah *resources* dan menemukan bagaimana *resource - resource* dihubungkan dengan yang lain. Setiap resource memiliki identifikasi yang unik di *web* platform yang disebut *Universal Resource Identifier* (URI) atau pada *web* berupa *Uniform Resource Locator* (URL). Pada sistem tukutu URI yang digunakan menggunakan route milik Lumen yang diberi *prefix* untuk *versioning* API yang digunakan. Tiap resource yang akan diakses memerlukan autentikasi JWT berupa token yang perlu disertakan pada *Request Header* untuk bisa masuk kedalam sistem.

b) Verbs

Verb adalah sebuah aksi HTTP seperti POST, GET, PUT, DELETE. Berikut keterangan mengenai method-method yang digunakan dalam RESTful web service [16]:

- a) GET digunakan untuk mengambil data *resource* dari *server*.
  - b) POST digunakan untuk mengirim data *resource* ke *server* atau membuat *resource* yang baru.
  - c) PUT digunakan untuk memodifikasi *resource* yang ada di server.
  - d) DELETE digunakan untuk menghapus *data resource* yang ada di server.
- c) Representation

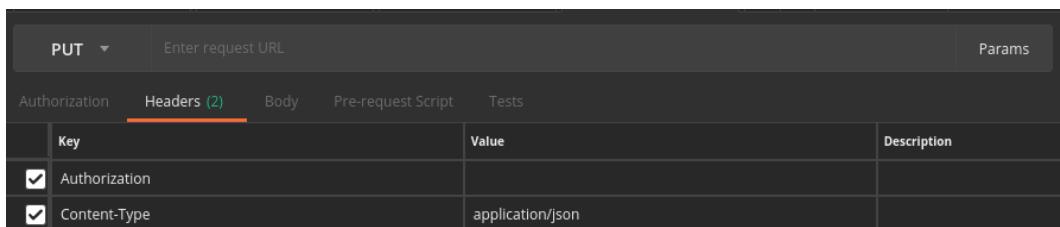
Representasi digunakan untuk menentukan jalan untuk menampilkan *resource – resource* ke *client*. Pada Tukutu memakai *platform* Android yang digunakan oleh pengguna dalam melakukan transaksi yang terhubung dengan *web service*. Pada sistem pembelian Tukutu menggunakan format JSON untuk menerima *resource* dari *web service* pada *platform* Android yang digunakan.



**Gambar 3.2: Alur Web Service TUKUTU**

### 3.5 Eksperimen dan Cara Pengujian Metode

Eksperimen yang dilakukan dengan memanfaatkan *build-in web server* bawaan PHP untuk menjalankan service dan menggunakan POSTMAN sebagai ujicoba REST API yang telah dibuat. Pada saat melakukan pengujian REST API diperlukan *Request Header* tambahan yang berisi **Authorization** dan **Content-Type** yang akan digunakan untuk validasi Request resource pada server. Isi dari **Authorization** berisikan token JWT yang dibuat dari *request login* pada sistem yang kemudian disimpan pada *platform* Android Tukutu yang kemudian akan terus digunakan sebagai hak akses kedalam sistem Tukutu. Sedangkan **Content-Type** digunakan pada *method* PUT HTTP yang digunakan untuk memodifikasi resource, isi dari header tersebut berupa nilai **application/json** sebagai identifikasi data yang dikirim berformat JSON.



*Gambar 3.3: Header pada POSTMAN*

Pada saat *request* method POST atau PUT untuk mengirim resource ke *web service* membutuhkan *Request Body* yang berisi **PARAMS** atau elemen *key* dan *value*. Untuk method PUT resource yang dikirim dalam bentuk format JSON dengan tambahan header **Content-Type**. *Key* dan *value* berisikan nilai yang akan digunakan untuk membuat atau memodifikasi *resource*, *key* yang dimasukan harus sesuai dengan yang telah ditetapkan oleh *web service*.

The screenshot shows the 'Body' tab in Postman with the 'form-data' option selected. There are two key-value pairs: 'nama' with value 'Tukutu' and 'panggilan' with value 'Tukutu'. A third row 'New key' is present but empty.

Key	Value
<input checked="" type="checkbox"/> nama	Tukutu
<input checked="" type="checkbox"/> panggilan	Tukutu
New key	

Gambar 3.4: Request Body pada method POST

The screenshot shows the 'Body' tab in Postman with the 'raw' option selected and the 'JSON (application/json)' type dropdown open. The JSON content is a single object with 'nama' and 'panggilan' keys.

```

1 {  

2   "nama": "Tukutu",  

3   "panggilan": "Tukutu"  

4 }

```

Gambar 3.5: Request Body pada method PUT

Sebagai response dari hasil request menetapkan standar format yang digunakan pada platform Android untuk mengetahui response balik dari web service. Berikut standar format yang digunakan:

```

[{"status": {
    "success": true,
    "code": 200,
    "message": "OK"
},
"result": {}]

```

Gambar 3.6: Standar format yang digunakan

Penjelasan dari hasil response yang didapat:

- **status**

Berisi status dari hasil request resource yang terdiri dari:

- **success** berisi nilai dengan tipe boolean yang menandakan jika request berhasil atau gagal.
- **code** memiliki nilai dari Response code dari hasil request. Contoh 4xx yang menandakan pada kesalahan request atau 5xx yang menandakan terjadi kesalahan pada server.
- **message** berupa pesan dari hasil request yang berisi pesan kesalahan atau pesan sukses.

- **result**

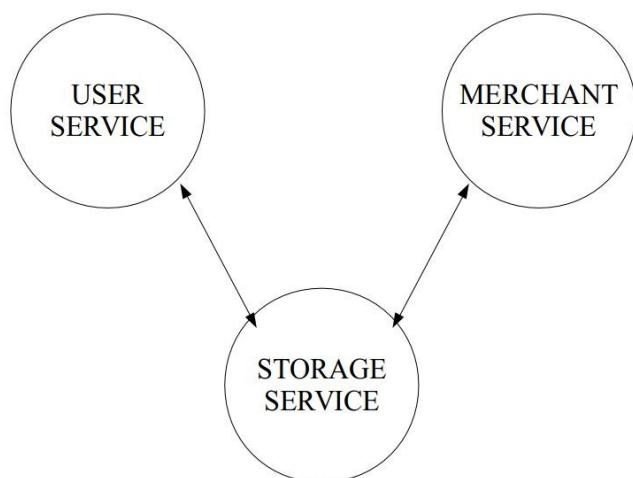
Merupakan resource dari hasil request yang dapat berupa object atau array yang kemudian ditampilkan.

## BAB IV

### IMPLEMENTASI DAN PEMBAHASAN

#### 4.1 Desain Web Service

Pada sistem Tukutu terdapat 3 service yang digunakan oleh platform Android, tiap *service* memiliki fungsionalitas tersendiri dalam menangani *resource*. Berikut service yang digunakan:



**Gambar 4.1: Desain Web Service**

##### 4.1.1 User Service

Service ini memiliki fungsionalitas yang digunakan oleh pembeli, karena memuat fitur – fitur yang diperlukan dalam transaksi pada aplikasi Tukutu. Service ini bertugas memberikan informasi yang ada pada aplikasi Tukutu seperti daftar sepatu yang dijual, brand sepatu, akun pengguna, dan transaksi. *User service* dibangun menggunakan framework Lumen. Berikut fitur yang ada:

- Order (pembelian)
- Bid (penawaran)

- Diskusi
- Manajemen Transaksi

#### 4.1.2 Merchant Service

*Merchant service* memiliki fungsionalitas untuk memanajemen penjualan sepatu pada aplikasi Tukutu, *service* ini hanya bisa digunakan oleh akun yang telah terdaftar sebagai penjual kecuali pada fitur pengajuan sebagai *merchant* (penjual) pada aplikasi. *Merchant service* dibangun menggunakan framework Lumen. Berikut fitur yang ada:

- Gabung jadi penjual
- Menajemen Sepatu
- Manajemen Transaksi

#### 4.1.3 Storage Service

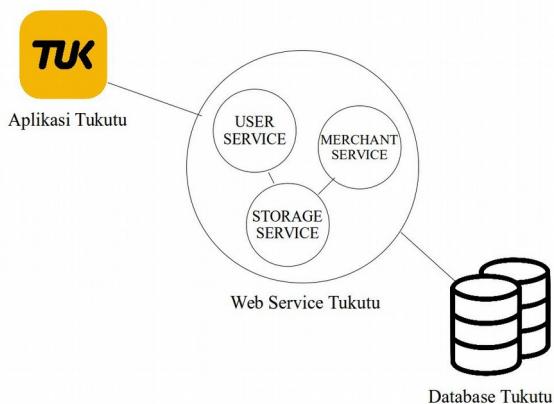
*Storage service* berguna sebagai tempat menyimpan gambar dan menjalankan fungsi tambahan yang bersangkutan dengan plugin pihak ketiga atau bawaan milik framework Laravel. *Storage service* terhubung dengan *user service* dan *merchant service*, ketika resource yang di akses pada *user service* atau *merchant service* mengandung fungsi pihak ketiga yang harus dijalankan maka akan langsung menghubungi *storage service* untuk menjalankannya. Hal ini dilakukan karena framework Laravel lebih mudah dalam pemasangan plugin pihak ketiga. Berikut fungsionalitas yang terdapat pada *storage service*:

- Menyimpan gambar
- Mengirim email SMTP kepada pembeli dan penjual
- Mengirim notifikasi FCM (Firebase Cloud Messaging) kepada pembeli dan penjual

- Menjalankan sistem *auto cancel* (pembatalan otomatis) berdasarkan waktu yang telah ditetapkan yaitu
  - Waktu pembayaran untuk pembelian : 3 Jam
  - Pengisian form pembelian jika penawarannya diterima : 12 Jam
  - Waktu kirim merchant (penjual) ke warehouse Tukutu : 2 Hari
  - Waktu merchant (penjual) mencairkan dana : 1 Hari

## 4.2 Alur Kerja Web Services

Pada aplikasi Tukutu *service* yang diakses pertama adalah *user service* karena pada *service* ini memuat fitur utama pada Tukutu. Kemudian pada *merchant service* digunakan pada saat ada akun yang ingin mendaftar menjadi penjual dan menjajen penjualan pada aplikasi Tukutu. Kedua *service* tersebut terhubung pada satu *service* yaitu *storage service* yang merupakan tempat untuk menyimpan *resource* dan *plugin* pihak ketiga. Data *resource* yang disimpan tidak hanya gambar, tapi juga *database migrations* yang merupakan bawaan milik Laravel. *Database migrations* kemudian di *generate* menjadi tabel pada database yang kemudian diakses *user service*, *merchant service*, dan *storage service*.



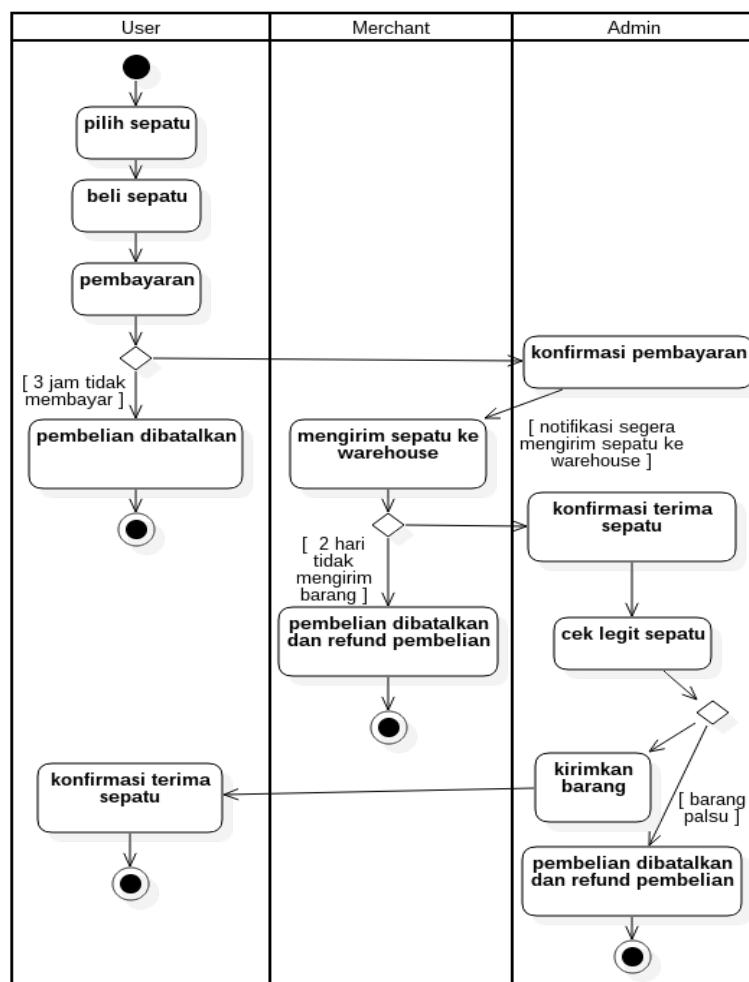
**Gambar 4.2: Alur Web Service Tukutu**

### 4.3 Proses Transaksi

Pada Tukutu terdapat 2 cara dalam transaksi, yaitu pembelian dan penawaran. Berikut penjelasannya:

#### 4.3.1 Pembelian

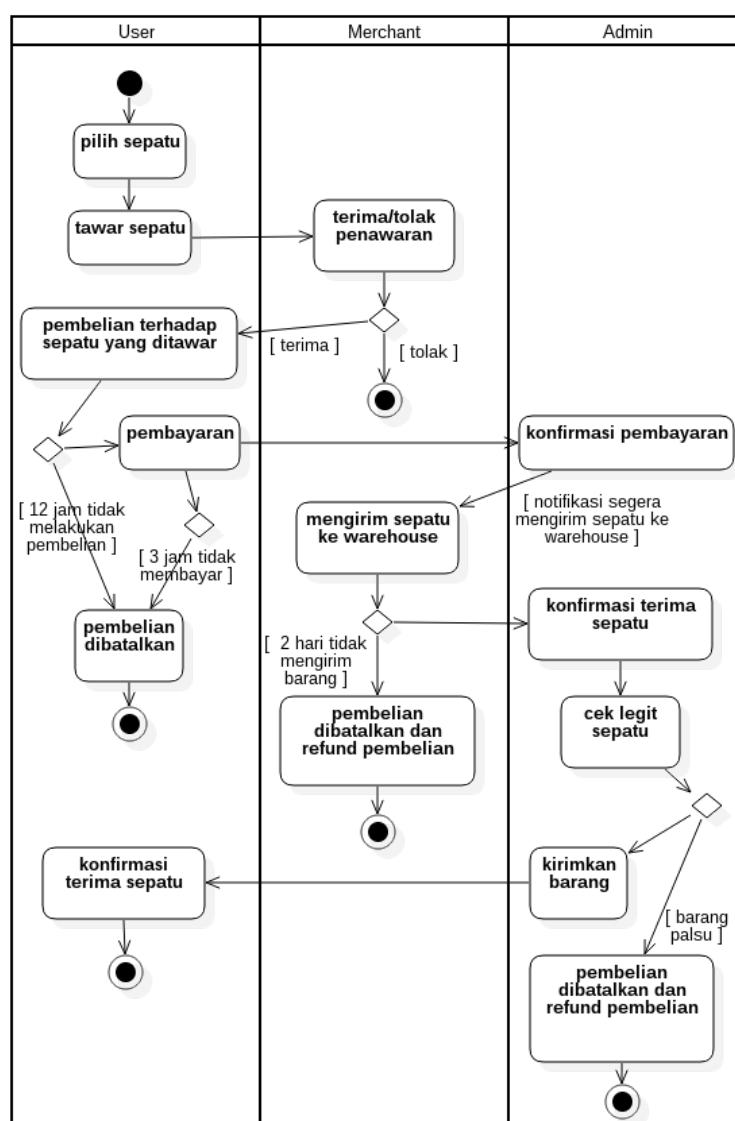
Pembelian dilakukan pada sepatu yang telah dipilih, kemudian pembeli mengisi data pembelian. Pembelian dapat dibatalkan jika pembeli tidak segera membayar dalam waktu 3 jam, penjual tidak mengirim barang ke warehouse dalam waktu 2 hari, dan barang terbukti palsu.



Gambar 4.3: Alur Pembelian

#### 4.3.2 Penawaran

Pada penawaran pembeli dapat menawar sepatu yang dipilih jika sepatu dapat ditawar, jika penawaran diterima maka pembeli harus melakukan pembelian sebelum 12 jam jika tidak ingin kehilangan hasil tawarnya. Alur pembelian setelah tawar sama seperti alur pembelian sebelumnya.



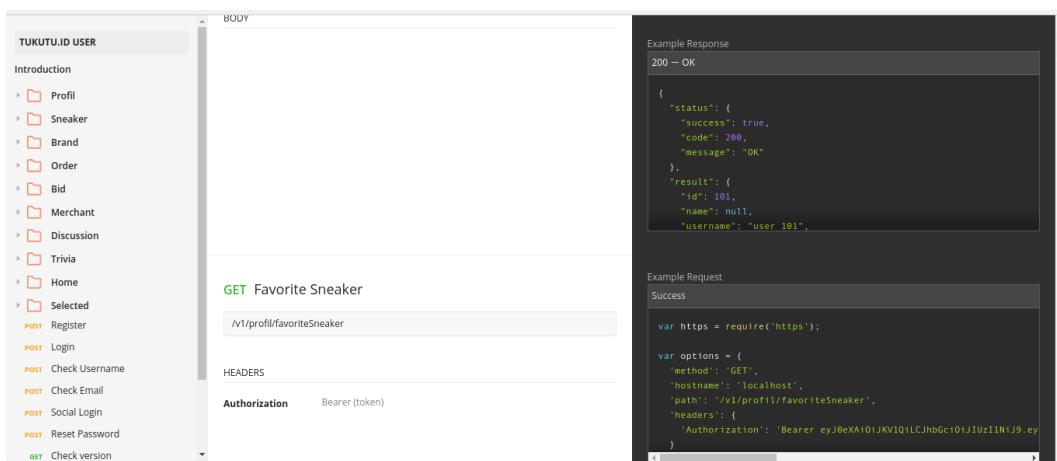
Gambar 4.4: Alur Penawaran

## 4.4 Implementasi dan Pembahasan

Berikut hasil dari implementasi *web service* pada Tukutu.

### 4.4.1 Dokumentasi API

Untuk memudahkan integrasi *web service* ke aplikasi Tukutu, dibutuhkan dokumentasi API yang memuat daftar *endpoint API* dan data *response* yang diterima.



Gambar 4.5: Dokumentasi API

Dari dokumentasi tersebut bagian developer platform Android Tukutu dapat mengetahui data yang diterima. Data yang diterima dalam bentuk JSON sesuai dengan standar format teks yang telah ditentukan sebelumnya pada **sub bab 3.5** di atas. Jika hasil *request* sukses data *resource* terdapat pada objek *result* yang berisi data *object* atau *array* yang berisikan informasi untuk ditampilkan pada sisi *client*.

### 4.4.2 Response

Standar format JSON yang digunakan yang sesuai dengan gambar (gambar 3.6) yang berisikan informasi objek status dan result. Objek status berisi informasi dari hasil request yang terdiri dari success, code, dan message yang

telah dideskripsikan pada sub bab 3.5, sedangkan objek result berisikan data resource yang dibutuhkan oleh aplikasi Android.

```
{  
    "status": {  
        "success": true,  
        "code": 200,  
        "message": "message response"  
    },  
    "result": {  
        "object_1": "value object 1",  
        "object_2": "value object 2",  
        "object_3": "value object 3",  
        "object_4": "value object 4"  
    }  
}
```

*Gambar 4.6: Contoh response*

```
{  
    "status": {  
        "success": true,  
        "code": 200,  
        "message": "message response"  
    },  
    "result": {  
        "object_1": "value object 1",  
        "object_2": "value object 2",  
        "object_3": "value object 3",  
        "object_4": "value object 4",  
        "objcet_5": {  
            "object_6": "value object 6"  
        }  
    }  
}
```

*Gambar 4.7: Contoh response objek didalam objek*

```
{  
    "status": {  
        "success": true,  
        "code": 200,  
        "message": "message response"  
    },  
    "result": [  
        {  
            "object_1": "value object 1"  
        },  
        {  
            "object_2": "value object 2"  
        }  
    ]  
}
```

**Gambar 4.8: Contoh response dengan data array**

#### 4.4.3 Autentikasi

Proses autentikasi Tukutu menggunakan JSON Web Token (JWT), dimana pembutan token terjadi saat aplikasi Tukutu (*client*) mengakses fungsi login pada web service. Hasil generate token kemudian disimpan pada sisi *client*, token tersebut digunakan pada setiap kali request/mengakses web service untuk meminta, mengirim atau memanipulasi data resource.

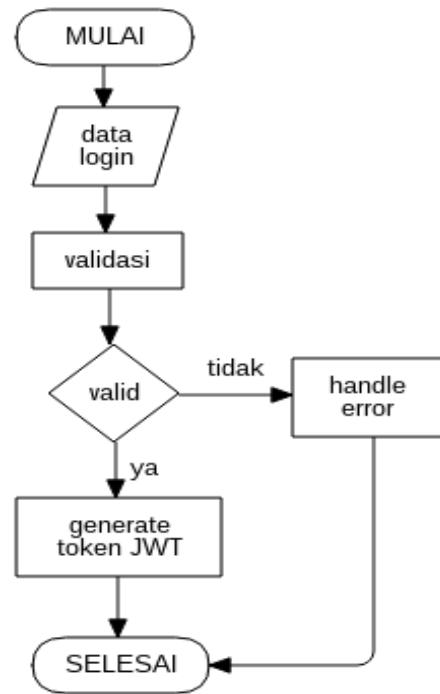
```
"token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9  
.eyJpc3MiOiJ0dWt1dHUVSUQlCJpYXQiOjE1MzI5NDA4NjcsImV4cCI6MTU2NDQ3Njg2Nyic3ViIjoxMDUsImp0aSI6IkZsT21mckN0ODIifQ  
.fbgb20mnSFPncbGC2qdvPH51Xa5Pou0_Oct4K18svrQ"
```

**Gambar 4.9: Contoh token JWT**

Hasil token yang dibuat kemudian disimpan juga pada database untuk pengecekan jika hasil token tersebut milik Tukutu karena digunakan sebagai hak akses ke data resource. Token JWT yang didapat hanya bisa digunakan pada *user service* dan *merchant service*.

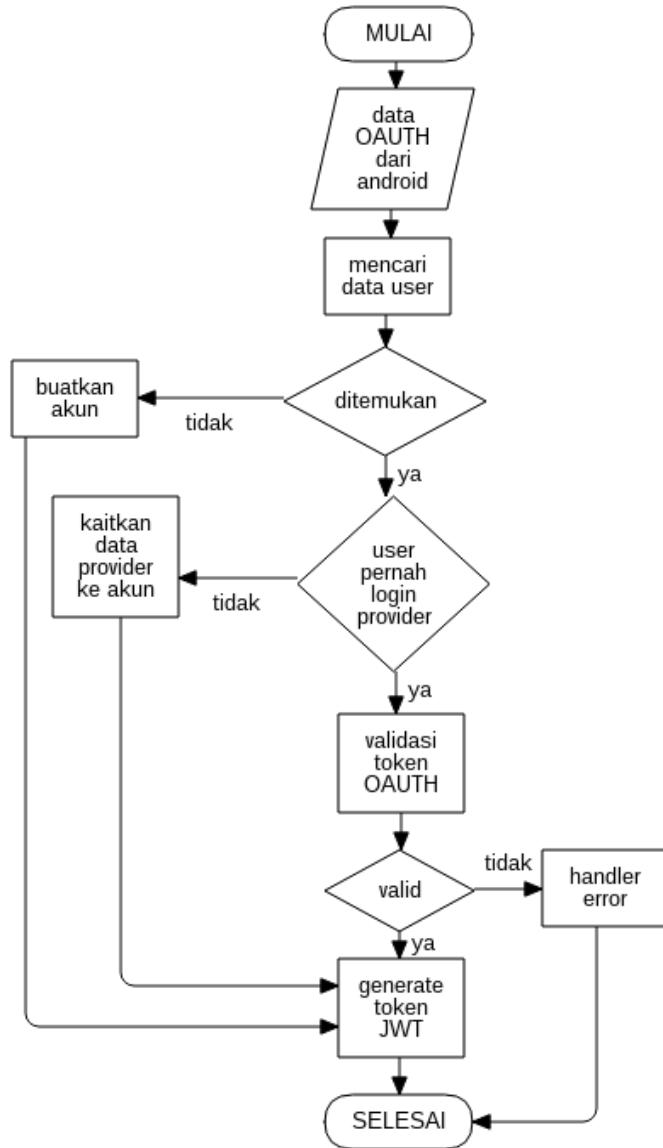
#### 4.4.4 Flowchart

##### a) Login



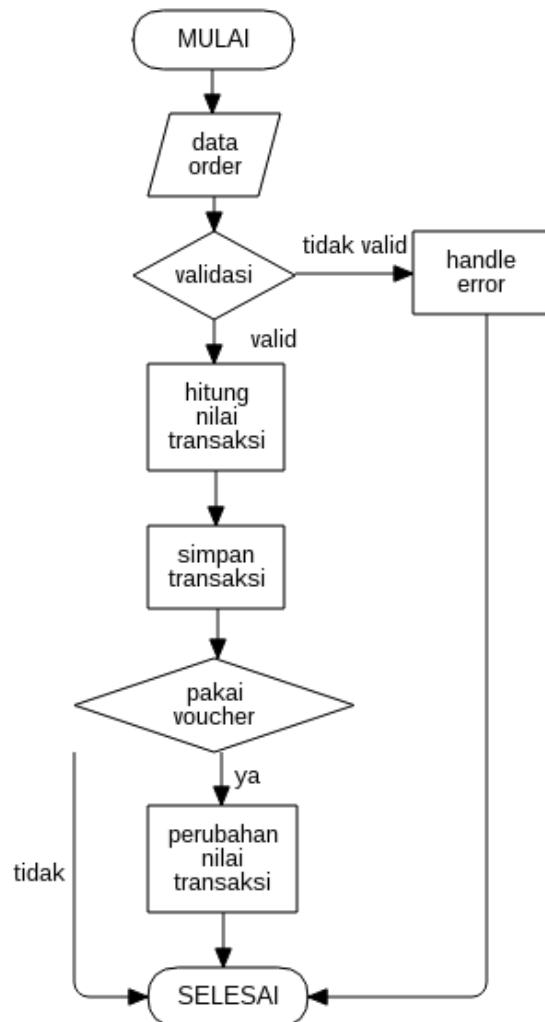
*Gambar 4.10: Flowchart Login*

b) Login Provider



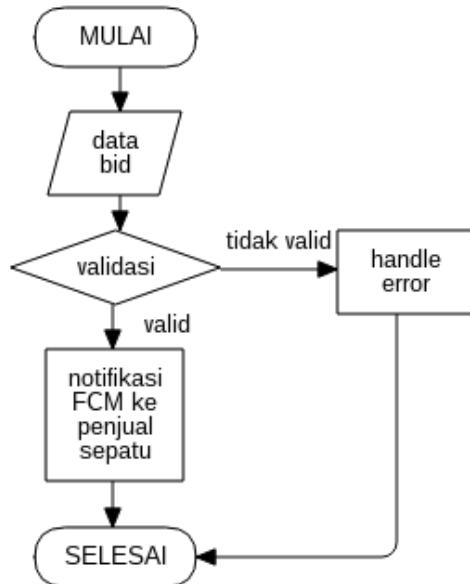
*Gambar 4.11: Flowchart Login Provider*

c) Order (pembelian)



*Gambar 4.12: Flowchart Order*

d) Bid (penawaran)



**Gambar 4.13: Flowchart Bid**

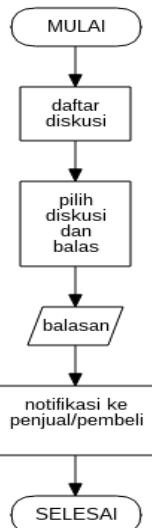
Jika nilai tawar diterima oleh penjual maka pembeli dapat melakukan pembelian dengan alur pembelian yang sama pada *flowchart* sebelumnya yaitu *flowchart order*.

e) Diskusi



**Gambar 4.14: Flowchart Diskusi**

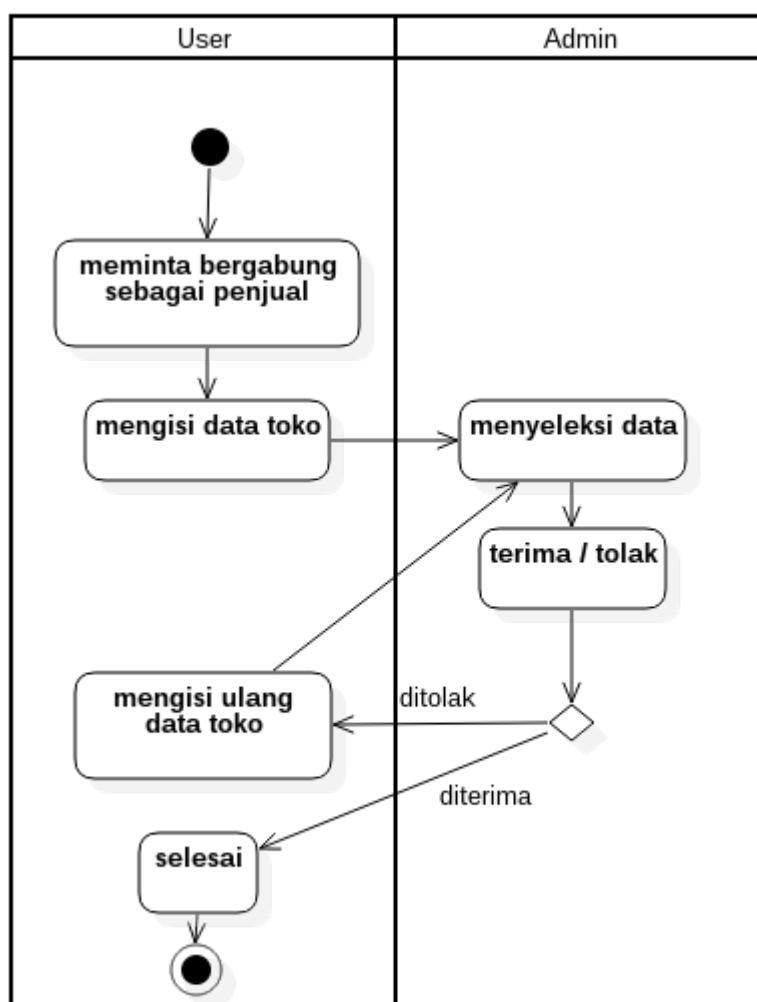
f) Balas Diskusi



**Gambar 4.15: Flowchart Balas Diskusi**

#### 4.4.5 Merchant

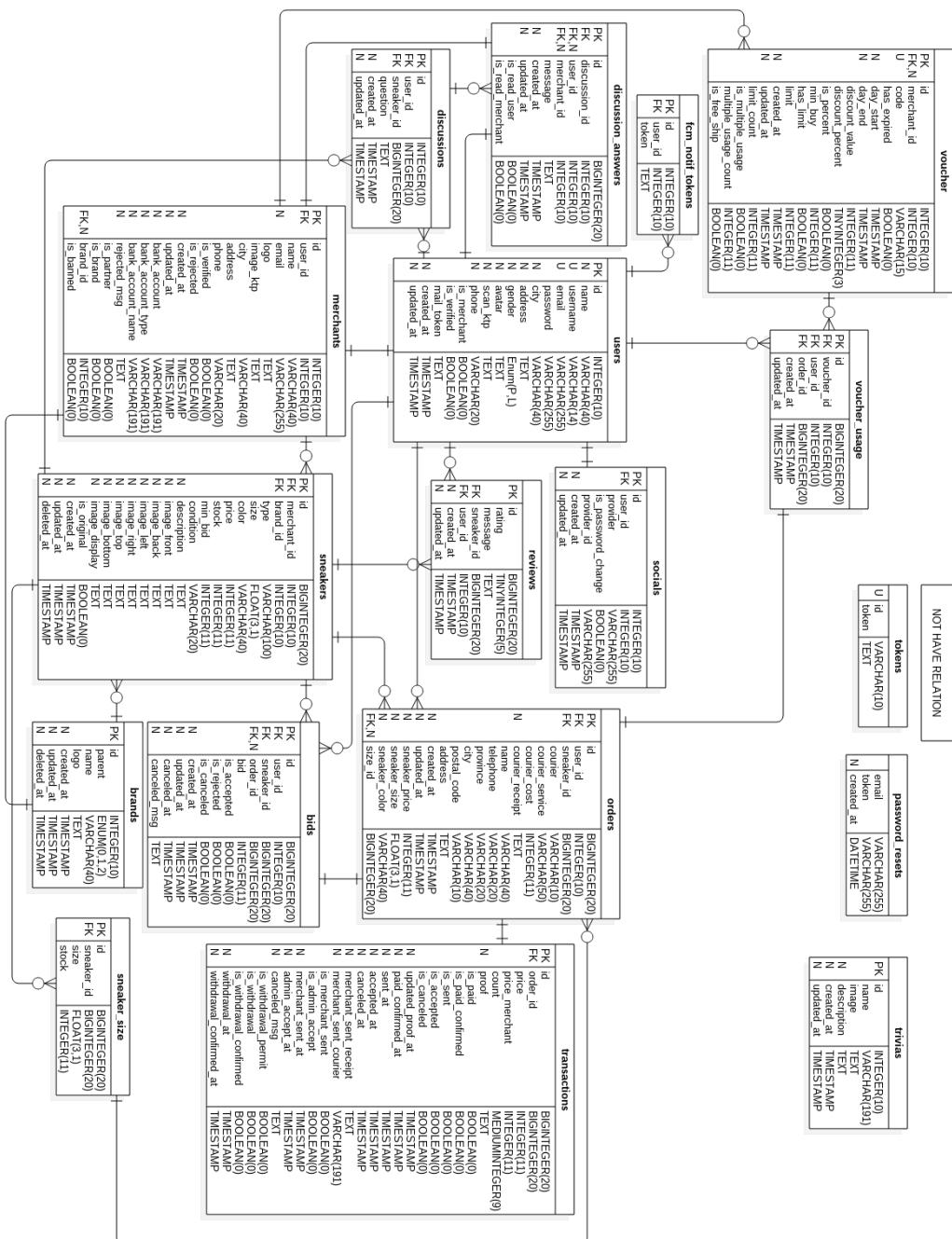
Penjual pada Tukutu disebut merchant, setiap akun dapat menjadi penjual dengan syarat dan kententuan yang berlaku. Untuk menjadi penjual akun yang terdaftar harus sudah terferifikasi, kemudian mengajukan data – data yang diperlukan untuk diseleksi oleh Admin Tukutu.



**Gambar 4.16: Alur jadi Merchant**

Setelah menjadi Merchant user dapat mulai menambah sepatunya di Tukutu dan menjualnya.

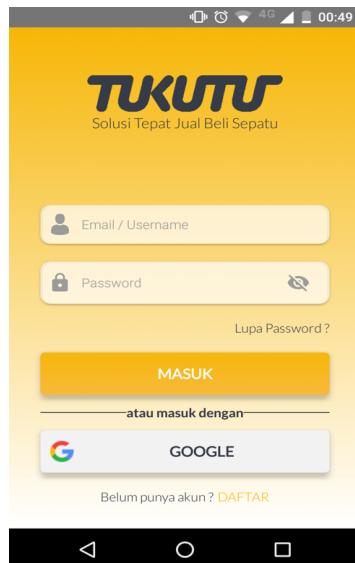
## 4.5 Perancangan Database



**Gambar 4.17:** ERD Tukutu

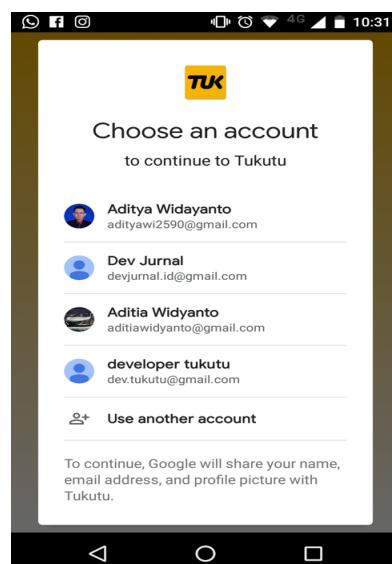
## 4.6 Implementasi Aplikasi Tukutu (Client)

### 4.6.1 Halaman Login



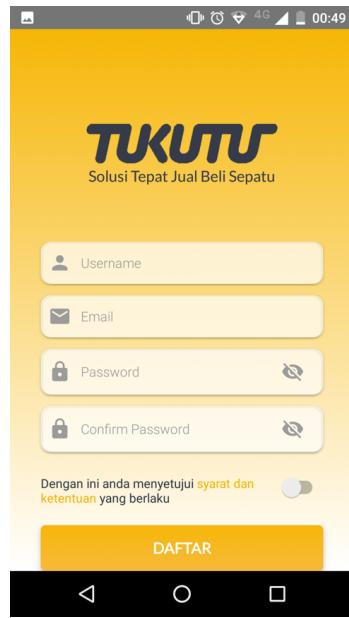
Gambar 4.18: Halaman Login

### 4.6.2 Halaman Login Provider



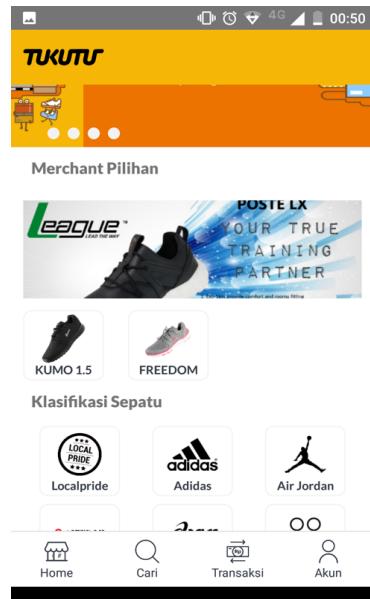
Gambar 4.19: Halaman Login Provider

#### 4.6.3 Halaman Register



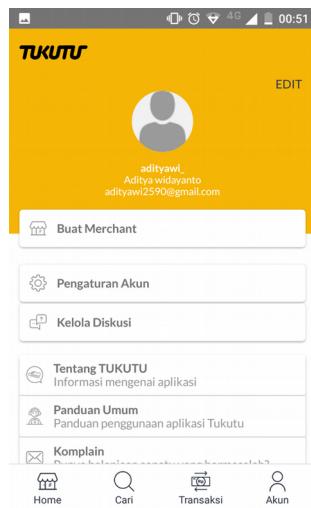
Gambar 4.20: Halaman Register

#### 4.6.4 Halaman Utama



Gambar 4.21: Halaman Utama

#### 4.6.5 Halaman Profil



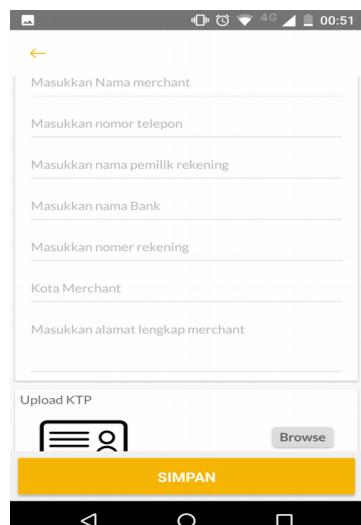
Gambar 4.22: Halaman Profil

#### 4.6.6 Halaman Daftar Transaksi



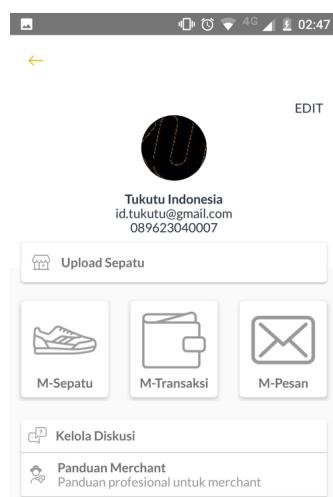
Gambar 4.23: Halaman Daftar Transaksi

#### 4.6.7 Halaman Gabung Jadi Penjual



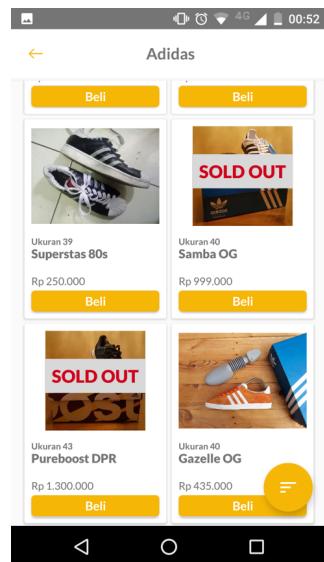
**Gambar 4.24: Halaman Gabung Jadi Penjual**

#### 4.6.8 Halaman Manajemen Penjual



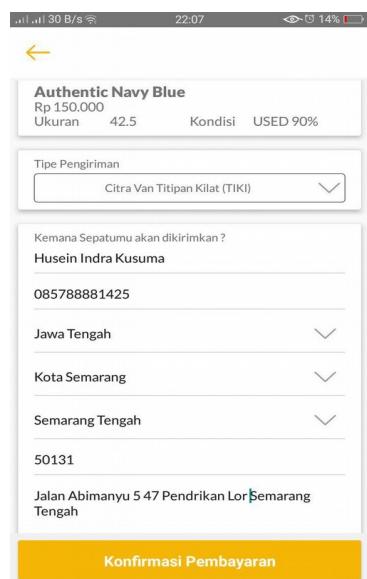
**Gambar 4.25: Halaman Manajemen Penjual**

#### 4.6.9 Halaman Daftar Sepatu



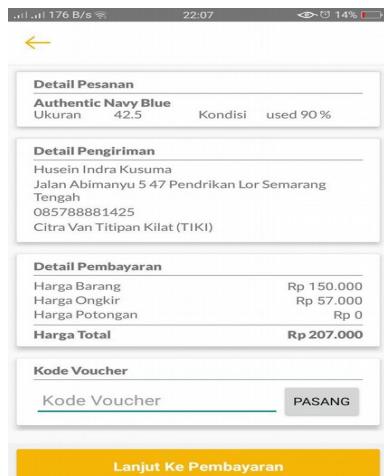
**Gambar 4.26: Halaman Daftar Sepatu**

#### 4.6.10 Halaman Pembelian



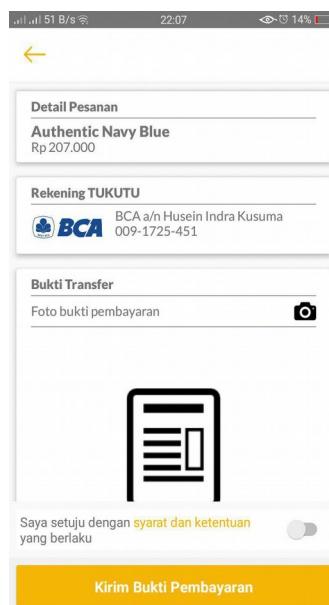
**Gambar 4.27: Halaman Pembelian**

#### 4.6.11 Halaman Konfirmasi Pembelian



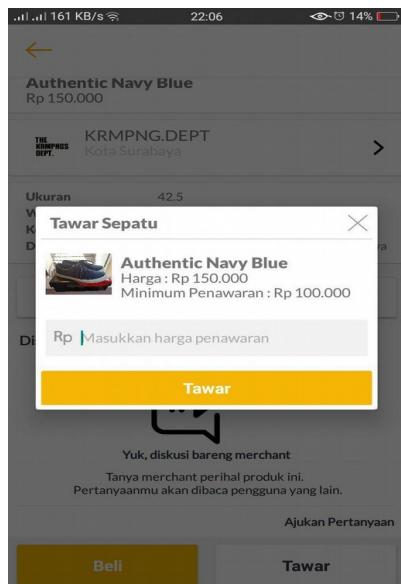
**Gambar 4.28: Halaman Konfirmasi Pembelian**

#### 4.6.12 Halaman Pembayaran



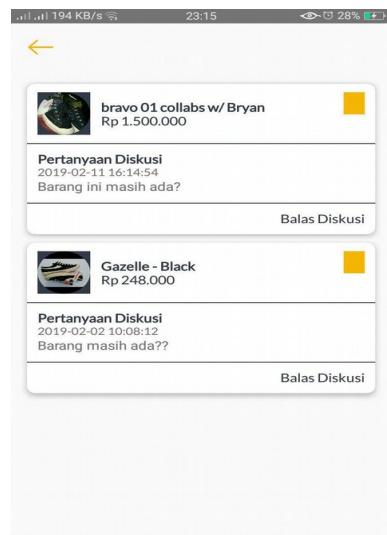
**Gambar 4.29: Halaman Pembayaran**

#### 4.6.13 Halaman Penawaran



Gambar 4.30: Halaman Tawar

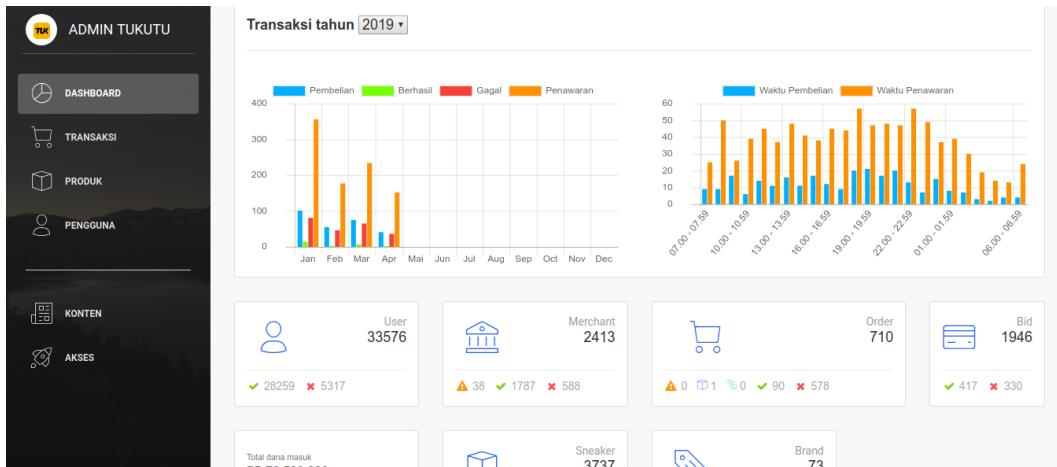
#### 4.6.14 Halaman Diskusi



Gambar 4.31: Halaman Daftar Diskusi

## 4.7 Implementasi Dashboard Admin (Mimin)

### 4.7.1 Halaman Utama



Gambar 4.32: Halaman Utama Admin

### 4.7.2 Halaman Transaksi

The figure shows the Admin Tukutu transaction history page. The sidebar is identical to the main dashboard. The main area displays a table of transactions. The columns are: ID, BARANG IT, PENJUAL IT, PEMBELI IT, WAKTU PEMBELIAN IT, and STATUS IT. The table lists 9 transactions, all marked as 'Sudah diterima' (Received). The transactions include items like 'Losing Grip: casual', 'Vans: Old Skool Gum Sole', 'Nike: Vapormax Triple Black 3.0', 'Adidas: Continental 80', 'Converse: Chuck Taylor', 'Converse: CHUK TAYLOR II HI CAMO REFLECTIVE', and 'Vans: Vault Slip-On OG True White Canvas'.

Gambar 4.33: Halaman Transaksi

### 4.7.3 Halaman Produk

The screenshot shows the 'ADMIN TUKUTU' dashboard with a sidebar containing links for Dashboard, Transaksi, Produk (selected), Pengguna, Konten, and Akses. The main content area is titled 'Sepatu' and displays a table of products. The table has columns for ID, MERCHANT, BRAND, TIPE, UKURAN, WARNA, and HARGA. The data includes various brands like wafflesfootwear, pairshoes Malang, Monkeysquad, ZEEGOODS, Rowley Store, Hayes Footwear, Betenstore.id, BLESSINGFOOT, salvok store official, and another entry for BLESSINGFOOT. Each row shows the product name, brand, type, size, color, and price.

ID	MERCHANT	BRAND	TIPE	UKURAN	WARNA	HARGA
4	wafflesfootwear	Waffles	SNAKE toddler slip on	22	green/black	RP 399.000
5	pairshoes Malang	Vans	SK8hi Tumble	42	white	RP 1.199.000
6	Monkeysquad	Vans	Oldskool OG LX Forest Night	42	Hijau	RP 1.850.000
7	ZEEGOODS	Vans	Slip On Classic Checkerboard	40.5	Hitam / Putih	RP 799.000
8	Rowley Store	Vans	Old Skool V Side Stripe Black	9	Hitam	RP 1.200.000
9	Hayes Footwear	Hayes	Phantom Black White	39	Hitam dan Putih	RP 300.000
10	Betenstore.id	Vans	Vans Era Pro 'Anniversary 50th' Stripe Checkerboard	40.5	Stripes Checkerboard	RP 1.100.000
11	BLESSINGFOOT	Vans	Vans Slip On Og Vault True White	9	Putih	RP 850.000
12	salvok store official	Vans	old skool	39	white green jade	RP 1.300.000
13	BLESSINGFOOT	Vans	Vans Slip On Og Vault True White	9	Putih	RP 850.000

Gambar 4.34: Halaman Produk

### 4.7.4 Halaman Pengguna

The screenshot shows the 'ADMIN TUKUTU' dashboard with a sidebar containing links for Dashboard, Transaksi, Produk, Pengguna (selected), Konten, and Akses. The main content area displays a table of users. The table has columns for ID, NAMA, KOTA, TELP/HP, EMAIL, STATUS, and WAKTU BERGABUNG. The data includes various users like Converse 70s OX, Bandot Shop, threéal sneakers, MixedUp.Hype, Kelvin Riandy, kid\_flip, wishlist, Jefri Store, prelovedck, and DealBuy. Each row shows the user's name, city, contact number, email, status (Tertunda), and the time they joined.

ID	NAMA	KOTA	TELP/HP	EMAIL	STATUS	WAKTU BERGABUNG
2355	Converse 70s OX	Kota Jakarta Selatan	087881961237	irsyad.gmb@gmail.com	Tertunda	6 hari yang lalu
2357	Bandot Shop	Kota Serang	081247951285	starfox212222@gmail.com	Tertunda	6 hari yang lalu
2358	threéal sneakers	Kota Yogyakarta	087825485522	ardianurokhan@gmail.com	Tertunda	6 hari yang lalu
2359	MixedUp.Hype	Kota Semarang	081315157488	ste phs.detailshop@gmail.com	Tertunda	6 hari yang lalu
2363	Kelvin Riandy	Kabupaten Bogor	+6287832753083	kelvinriandy21@gmail.com	Tertunda	5 hari yang lalu
2366	kid_flip	Kota Jakarta Timur	081234345666	firlyflip@gmail.com	Tertunda	5 hari yang lalu
2367	wishlist	Kabupaten Karanganyar	085336442860	lowstore.official@gmail.com	Tertunda	5 hari yang lalu
2369	Jefri Store	Kota Surabaya	083847732722	rm.jefri@gmail.com	Tertunda	4 hari yang lalu
2373	prelovedck	Kota Bandung	08112069100	tropicalvibes@gmail.com	Tertunda	3 hari yang lalu
2376	DealBuy	Kota Pekalongan	089619446735	fauzanspnda@gmail.com	Tertunda	3 hari yang lalu

Gambar 4.35: Halaman Pengguna

#### 4.7.5 Halaman Seleksi Penjual

The screenshot shows the 'Detail Merchant' page of the Admin Tukutu application. On the left is a sidebar with icons for Dashboard, Transaksi, Produk, Pengguna, Konten, and Akses. The main area has tabs for 'Kembali' and 'Detail Merchant'. At the top right are buttons for 'Lihat Sepatu', 'TERIMA' (in green), and 'TOLAK' (in red). Below these are two images: one of a blue Converse sneaker on an orange box, and another of a product label. To the right is a detailed merchant information card:

User	bdcTFZls4P5r	Potongan	10% <input checked="" type="checkbox"/>
Nama	Converse 70s OX	Telp/HP	087881961237
Bank	BNI	No. Rekening	0728343124
Kota	Kota Jakarta Selatan	A/N	Irsyad Ramadhan
Alamat jl. durentiga selatan VIII no. 19 rt:09/002			
Bergabung pada 18 April 2019, 21:02:07 WIB			
Terakhir diperbarui 18 April 2019, 21:02:07 WIB			

Gambar 4.36: Halaman Seleski Penjual

### 4.8 Pengujian

Pengujian dilakukan untuk mengetahui apakah aplikasi yang telah dibangun dapat berjalan dengan baik dan memenuhi spesifikasi yang telah ditentukan.

#### 4.8.1 White Box Testing

Pada pengujian *white box* disini akan menggunakan teknik *Basic Path*. Pada *Basic path*, dilakukan pengujian dengan membuat test case yang didapatkan dari jalur independen. Jalur independen tersebut didapatkan dari menghitung nilai *cyclomatic complexity* dari flowgraph dengan rumus berikut:

$$V(G) = E - N + 2$$

Dimana:

E = jumlah *edge* pada grafik

N = jumlah *node* pada grafik

Untuk melakukan pengujian *white box* digunakan fungsi utama yaitu order (pembelian).

```
if ($invalid = $this->response->validate($req, Rule::ORDER)) return $invalid;
try {
    $sneaker = Sneaker::findOrFail($sneaker_id);

    // SIZE SNEAKER
    if ($req->filled('size_id')) {
        $size = $sneaker->sizes()->find($req->size_id);
        if (!$size) return $this->response->messageError('Size not found', 404);
    }
    else $size = $sneaker->sizes()->first();

    // VALIDATE CONDITION SNEAKER
    if ($notValid = $this->validateOrder($req, $sneaker)) return $notValid;

    // IF STOCK = 1. THE SNEAKER HAS RECENTLY BID, CANCEL ALL BID
    if ($sneaker->stock == 1) $this->isStockIsOne($sneaker);

    // START cost calculation
    $price = $sneaker->price * $req->count;
    $fee_tukutu = ceil(round($price * $sneaker->merchant->cut_fee, 2));
    $price_merchant = $price - $fee_tukutu;
    $price_transaction = $price + $req->courier_cost;
```

**Gambar 4.37: Source Code Order bagian a**

```
$req->merge([
    'user_id' => $this->user->id,
    'sneaker_id' => $sneaker->id,
    'sneaker_price' => $sneaker->price,
    'sneaker_size' => $size->size,
    'sneaker_color' => $sneaker->color,
    'price' => $price_transaction,
    'price_merchant' => $price_merchant,
    'size_id' => $size->id
]);
// END cost calculation

$order = OrderModel::create($req->all());
$order->transaction()->create($req->only('price', 'price_merchant', 'count'));
$sneaker->decrement('stock', $req->count);
$size->decrement('stock', $req->count);
```

**Gambar 4.38: Source Code Order bagian b**

```

/// VOUCHER USAGE
if ($req->filled('code')) {
    $voucher = $this->validateOrderVoucher($req->code, $price);

    if ($voucher->valid) {
        $usageVoucher = VoucherUsage::create([
            'user_id' => $this->user->id,
            'voucher_id' => $voucher->id,
            'order_id' => $order->id
        ]);

        if ($usageVoucher->voucher->has_limit) $usageVoucher->voucher()->decrement('limit');
        if (!$voucher->is_free_ship) {
            $price = $price - (($voucher->is_percent) ? (int)((($voucher->value/100) * $price):$voucher->value) + $req->courier_cost;
        }
        $order->transaction()->update(['price' => $price]);
    }

    $order->voucher_usage = $voucher;
}

```

**Gambar 4.39: Source Code Order bagian c**

```

// Auto Cancel Call
$this->callAutoCancel($order->transaction->id);

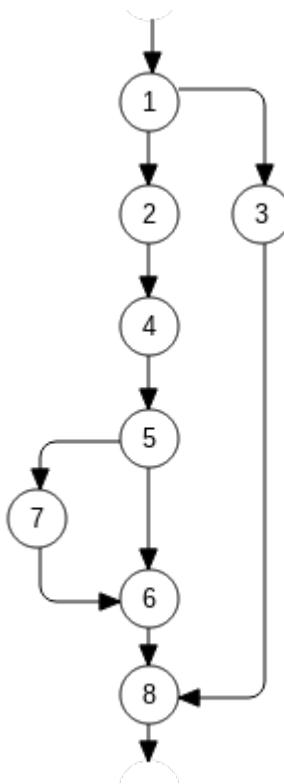
LogTukutu::create($req, $this->user, ['activity', 'process'], 'order sneaker id:'.$sneaker_id, ['order' => $order, 'sneaker' => $sneaker])
return $this->send($order);
} catch (Exception $e) {
    if ($e instanceof \Illuminate\Database\Eloquent\ModelNotFoundException)
        return $this->response->messageError('Sneaker not found', 404);

    return $this->response->serverError();
}

```

**Gambar 4.40: Source Code Order bagian d**

Dari source coder tersebut kemudian dibuat flowgraph untuk menghitung nilai *cyclomatic complexity* dan menentukan jalur,



*Gambar 4.41: Flowgraph Order (pembelian)*

Dari flowgraph kemudian menghitung *cyclomatic complexity* dengan rumus sebelumnya  $V(G) = E - N + 2$ .

$$V(G) = E - N + 2$$

$$V(G) = 9 - 8 + 2$$

$$V(G) = 3$$

Dari hasil perhitungan *cyclomatic complexity* dibuat jalur independen yang diketahui memiliki 3 jalur.

**Jalur 1 = 1 – 3 – 8**

Keterangan : Terjadi pembelian tapi gagal karena data yang dikirim tidak valid atau syarat pembelian tidak terpenuhi.

**Jalur 2 = 1 – 2 – 4 – 5 – 6 – 8**

Keterangan : Terjadi pembelian dan berhasil.

**Jalur 3 = 1 – 2 – 4 – 5 – 7 – 6 – 8**

Keterangan : Terjadi pembelian yang menggunakan voucher dan berhasil.

Hasil pengujian test case dapat dilihat ditabel berikut:

**Tabel 4.1: Test Case White Box**

No	Test Case	Ketercapaian
1	<b>1 – 3 – 8</b>	Berhasil
2	<b>1 – 2 – 4 – 5 – 6 – 8</b>	Berhasil
3	<b>1 – 2 – 4 – 5 – 7 – 6 – 8</b>	Berhasil

#### 4.8.2 Black Box Testing

Pengujian *Black Box* adalah suatu cara yang digunakan untuk menguji spesifikasi fungsional dari perangkat lunak (*software*). Pada pengujian *Black Box* menggunakan Postman untuk mengetahui hasil response jika berhasil, pengujian dinyatakan berhasil jika objek *success* bernilai *true*. Hasil pengujian dibuat dalam tabel berikut:

**Tabel 4.2: Black Box Testing**

No	Proses	Hasil yang diharapkan	Hasil
1	Register	status : - success : true	Berhasil

		<ul style="list-style-type: none"> <li>- code : 201</li> <li>- message : “user has been created, please verified your account”</li> <li>result : null</li> </ul>	
2	Login	<ul style="list-style-type: none"> <li>status :</li> <li>- success : true</li> <li>- code : 200</li> <li>- message : “OK”</li> <li>result :</li> <li>- token</li> </ul>	Berhasil
3	Login Provider	<ul style="list-style-type: none"> <li>status :</li> <li>- success : true</li> <li>- code : 200</li> <li>- message : “OK”</li> <li>result :</li> <li>- token</li> </ul>	Berhasil
4	Order	<ul style="list-style-type: none"> <li>status :</li> <li>- success : true</li> <li>- code : 200</li> <li>- message : “OK”</li> <li>result :</li> <li>- data transaksi</li> </ul>	Berhasil
5	Bid	<ul style="list-style-type: none"> <li>status :</li> <li>- success : true</li> <li>- code : 201</li> <li>- message : “new bid added”</li> <li>result : null</li> </ul>	Berhasil
6	Diskusi	<ul style="list-style-type: none"> <li>status :</li> <li>- success : true</li> </ul>	Berhasil

		<ul style="list-style-type: none"> <li>- code : 200</li> <li>- message : “OK”</li> <li>result :</li> <li>- data diskusi</li> </ul>	
7	Balas Diskusi	<ul style="list-style-type: none"> <li>status :</li> <li>- success : true</li> <li>- code : 200</li> <li>- message : “OK”</li> <li>result :</li> <li>- data diskusi</li> </ul>	Berhasil

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Setelah melakukan analisis, perancangan dan melakukan implementasi web service dengan metode RESTful pada Sistem Jual Beli Tukutu, dapat diperoleh kesimpulan sebagai berikut :

1. Terciptanya *web service* yang digunakan pada aplikasi Tukutu *marketplace* jual beli sepatu brand lokal dengan jaminan keaslian pada brand – brand yang dijualnya.
2. Restful *Web Service* yang dibangun berhasil mengintegrasikan *service – service* yang telah dibuat dan platform Android (client) sehingga menjadi *marketplace* jual beli sepatu.

#### **5.2 Saran**

Adapun saran yang dapat dipergunakan untuk pengembangan penelitian selanjutnya adalah sebagai berikut :

1. Pada fitur diskusi belum menggunakan sistem realtime sehingga perlu request ulang untuk mendapatkan data yang baru, untuk pengembangan selanjutnya diharapkan dapat menggunakan *socket.io* atau *library* lainnya.
2. Desain struktur JSON masih menggunakan Model Laravel tanpa adanya cast data hasil query ke database. Ini kurang aman karena data yang dikirim hampir sama dengan yang ada di database.
3. *Web service* yang dibangun menggunakan Laravel dari keluarga PHP, untuk pengembangan selanjutnya diharapkan dapat dimigrasi ke pengembangan *web modern* dengan menggunakan NodeJS atau pemrograman lainnya.