

IMPLEMENTASI *WEB SERVICE* DENGAN METODE RESTFUL PADA SISTEM PENJUALAN BARANG DI PT. WELLCOMM GROUP

Dimas Eko Nugroho^{1*}, Joko Christian Chandra², Utomo Budiyanto³, Titin Fatimah⁴

^{1,2,3,4}Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, DKI Jakarta, Indonesia

Email: ^{1*}1811510336@student.budiluhur.ac.id, ²joko.christian@budiluhur.ac.id, ³utomo.budiyanto@budiluhur.ac.id, ⁴titin.fatimah@budiluhur.ac.id
(* : corresponding author)

Abstrak-Teknologi *web* dan *mobile* saat ini sangat banyak digunakan dalam berbagai kalangan bisnis dan masyarakat. Perusahaan PT Wellcomm Group menjual aksesoris dan gadget yang saat ini sudah menggunakan sistem penjualan *offline* melalui sistem *Point of Sales*. Namun, untuk proses transaksi *online* masih membutuhkan konfirmasi melalui sales. Masalah pada penelitian ini adalah *customer* tidak bisa melakukan pembelian barang *online* secara mandiri. Oleh karena itu, tujuan penelitian ini adalah diperlukannya suatu aplikasi seperti *e-commerce* dan penerapan *web service* dengan metode RESTful API untuk memudahkan pelanggan dalam melakukan pembelian barang secara *online*. Web service dibutuhkan untuk mempermudah integrasi sistem dan fleksibilitas dalam pengembangan bisnis serta mempermudah akses informasi produk oleh pelanggan secara *online*. Penelitian ini menggunakan kaidah metodologi *waterfall* dengan menggunakan *Framework* Laravel untuk membuat *web service* dengan metode RESTful API. Metode RESTful API menggunakan protokol HTTP sebagai standar pertukaran data dan memudahkan komunikasi antar sistem. Dalam menjaga keamanan data, penggunaan autentikasi token Laravel Sanctum dapat menjaga keamanan data sehingga informasi pelanggan dan transaksi tetap aman. Kesimpulan penelitian ini adalah dihasilkan *web service* yang membantu proses transaksi *online*. Pelanggan dapat melakukan pembelian barang secara *online* dengan mudah dan aman tanpa memerlukan konfirmasi melalui sales.

Kata Kunci: Laravel Sanctum, *Mobile Application*, Penjualan Barang, RESTful API, *Web Service*

IMPLEMENTATION OF *WEB SERVICE* WITH RESTFUL METHOD ON SALES SYSTEM IN PT. WELLCOMM GROUP

Abstract-*Web and mobile technology are currently very widely used in various business circles and society. PT Wellcomm Group, a company engaged in accessories and gadgets, currently uses an offline sales system through the Point of Sales system. However, the online transaction process still requires confirmation through sales. The problem in this research is that customers cannot make online purchases independently. Therefore, the purpose of this research is the necessity of an application such as e-commerce and the application of web services with the RESTful API method is needed to facilitate customers in purchasing goods online. Web services are needed to facilitate system integration and flexibility in business development and facilitate access to product information by customers online. This research uses the waterfall method, which utilizes Laravel framework to create a web service using the RESTful method. The RESTful API method uses the HTTP protocol as a data exchange standard and facilitates communication between systems. In maintaining data security, the use of Laravel Sanctum token authentication can maintain data security so that customer and transaction information remains safe. The conclusion of this research is that a web service has been produced that assists in the online transaction process. Customers can purchase goods online easily and safely without requiring confirmation through sales.*

Keywords: *Laravel Sanctum, Mobile Application, Goods Sales, RESTful API, Web Service*

1. PENDAHULUAN

Teknologi web saat ini merupakan salah satu bidang teknologi Informasi yang sangat banyak digunakan di berbagai kalangan bisnis, mulai dari pengusaha kecil sampai pada perusahaan perusahaan besar. Hal ini sangat terlihat jelas pada banyaknya aplikasi-aplikasi berbasis web. Teknologi informasi merupakan bidang yang berkembang pesat dan mempunyai peran yang sangat penting dalam kehidupan sehari-hari, khususnya pada bidang bisnis [1]. PT Wellcomm Group terdiri dari beberapa perusahaan yang berdiri di atas nama Wellcomm merupakan perusahaan yang bergerak dibidang aksesoris dan gadget. Masalah dalam penelitian ini adalah dalam kesehariannya terdapat kegiatan jual beli yang dilakukan. Pada saat ini proses penjualannya sudah menggunakan sistem penjualan POS (*Point of Sales*) untuk transaksi *offline store*. Sedangkan jika pelanggan ingin memesan barang secara *online* dan mandiri maka pelanggan harus konfirmasi terlebih dahulu melalui sales, baru barang akan diproses dan dikirim secara tersendiri tanpa melalui sistem. Oleh karena itu dibutuhkan sistem aplikasi *mobile* dan *web* semacam aplikasi *e-commerce* agar memudahkan pelanggan dalam melakukan pembelian barang secara *online* tanpa melalui sales.

Teknologi yang tepat untuk menghubungkan setiap *platform* adalah *web services* [2]. *Web service* merupakan sebuah sistem perangkat lunak yang memungkinkan aplikasi yang berbeda untuk berkomunikasi dan berinteraksi dengan satu sama lain melalui jaringan Internet [3]. Dimana dalam aplikasi tersebut menyediakan metode dengan tujuan di gunakan untuk interaksi aplikasi satu dengan aplikasi yang lain di akses dengan URL dan menerima *response* dalam bentuk JSON, XML, TXT, CSV dan lainnya. *Representational State Transfer*, disingkat REST adalah jenis arsitektur untuk mengimplementasikan *web service* yang menggunakan konsep perpindahan antar *state*. *State* disini dapat digambarkan seperti *browser* yang meminta suatu halaman *web*, di sisi server mengirimkan *state* halaman *web* saat ini ke *browser*. Memindahkan URL yang ditentukan sama dengan mengganti *state* dari halaman *web*. REST memungkinkan sitem yang berbeda untuk berkomunikasi dan mengirim atau menerima data dengan sangat mudah. Setiap panggilan REST memiliki hubungan antara HTTP dan URL, sehingga *database* dapat dihubungkan ke titik akhir API di REST [4].

Laravel Sanctum adalah *package* yang disediakan laravel untuk menangani proses autentikasi dan otorisasi pengguna, sanctum menyediakan sistem autentikasi ringan untuk *Single Page Application* (SPA), aplikasi *mobile*, dan API berbasis token sederhana. Sanctum memungkinkan setiap pengguna menghasilkan beberapa token API untuk akun mereka. Token ini dapat diberikan kemampuan yang menentukan tindakan mana apa yang diizinkan untuk dilakukan oleh token. Berdasarkan latar belakang di atas, penelitian ini dilakukan dengan bertujuan mengimplementasikan metodologi *waterfall* untuk mencapai beberapa tujuan yakni layanan *web service* akan dikirimkan melalui API kepada client agar bisa diakses melalui *multiplatform*; membuat *web service* RESTful API yang mengirimkan informasi produk PT Wellcomm dengan API pada aplikasi *mobile* dan *web client* yang digunakan oleh pelanggan dan aplikasi web admin yang hanya bisa diakses oleh *staff* admin; dan mengimplementasikan Laravel Sanctum untuk pengamanan pertukaran datanya menggunakan autentikasi token pada *web service* RESTful API pada saat pelanggan melakukan *login* maupun *register* aplikasi.

2. METODE PENELITIAN

2.1 Data Penelitian

Data yang dipakai saat pengujian yaitu data produk dari PT. Wellcomm Group. Data tersebut berupa nama produk, foto produk, kategori produk, harga, dan *rate* produk yang akan di-*inputkan* melalui *web admin* dan akan ditampilkan datanya di *mobile apps wellcomm-store*, dan *website client wellcomm-store*. Teknik Pengumpulan data menggunakan teknik observasi, wawancara dan studi pustaka.

2.2 Metode Pemandangan

Berikut adalah metode perbandingan dari berbagai sumber yang digunakan sebagai referensi pada penelitian ini sebagai berikut:

Penelitian ini berkaitan dengan perkembangan pariwisata di pulau Lombok yang menyebabkan tumbuhnya profesi tukang dan kebutuhan akan kemudahan dalam menjalankan usaha. Penelitian ini mengembangkan aplikasi *multiplatform* untuk mempermudah pengguna dalam memilih jasa tukang, mengetahui harga, dan biaya lainnya. Metode pengembangan yang digunakan adalah metode *waterfall*. Aplikasi telah diuji pada beberapa tukang dan hasil pengujian menunjukkan 51.4% tukang sangat setuju dengan aplikasi, sehingga dapat disimpulkan bahwa aplikasi memenuhi kebutuhan tukang [5].

Penelitian ini proses pemesanan jasa *make up* di Pontianak masih mengalami kendala seperti kesulitan mengetahui jenis dan harga, menyesuaikan jadwal, dan melihat laporan pemesanan. Penelitian ini mengembangkan aplikasi *mobile* untuk mempermudah proses pemesanan jasa *make up* dengan restful *web service* untuk terintegrasi antara admin dan pelanggan. Aplikasi ini dapat digunakan untuk menambahkan jenis jasa, melakukan pemesanan, *check out*, dan meng-*upload* bukti pembayaran. Admin dapat melihat pesanan dan *history* laporan pesanan. Penelitian ini menggunakan metode *black box testing* dengan teknik *graph-based* [6].

Penelitian ini adalah sebuah aplikasi *sales order* yang dirancang untuk membantu *sales person* dalam memasarkan produk perusahaan dan mempercepat proses pemesanan produk. Aplikasi ini memiliki kemampuan untuk melakukan pemesanan produk dan mengirimkan data pemesanan ke aplikasi berbasis web yang dijalankan oleh admin [7].

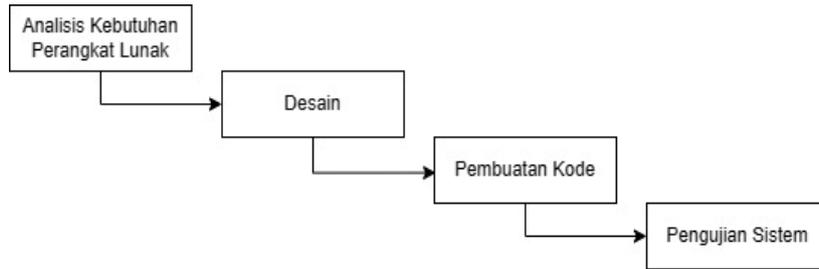
Penelitian ini dapat disimpulkan bahwa aplikasi fasilitas keuangan memiliki peran penting dalam meningkatkan kualitas keterbukaan informasi dan mempermudah transaksi data antar aplikasi, terutama antara aplikasi pencarian fasilitas keuangan. Keunggulan dari penerapan *web service* dalam aplikasi fasilitas keuangan ini adalah dapat berjalan pada berbagai *platform* yang berbeda dan dapat ditulis dalam berbagai bahasa pemrograman, sehingga memudahkan proses integrasi dan transaksi data [8].

Penelitian ini berbeda dengan penelitian sebelumnya diantaranya penelitian ini menggunakan Laravel Sanctum untuk pertukaran data secara aman, menggunakan Midtrans API untuk memproses pembayaran agar

terotomatisasi dan juga tujuan dari penelitian *web services* ini diimplementasikan untuk mendukung proses bisnis. yaitu sebagai alat penyimpanan data pemesanan barang berupa aplikasi *mobile* dan *website* di PT Wellcomm Group.

2.3 Penerapan Metode

Metode *Waterfall* adalah model pengembangan sistematis dan berurutan untuk membuat sebuah *software*. Sehingga masih sering digunakan dalam pengembangan *software* karena dapat membantu *developer* untuk memahami keseluruhan *project* dan melihatnya sebagai sebuah kesatuan yang terorganisir [9]. Berikut penjelasan tahapan metode *waterfall* yang dipaparkan pada gambar 1.



Gambar 1. Metodologi *Waterfall*

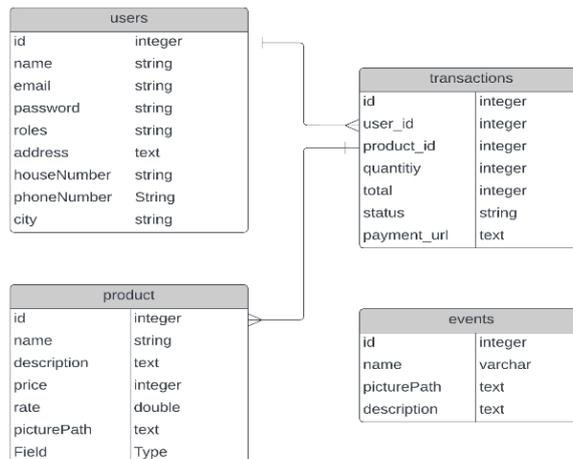
- Analisis kebutuhan perangkat lunak dimulai dari Analisa kebutuhan sistem yang membutuhkan *web service* untuk menghubungkan *client* dan *server*. Tujuan tahap ini untuk membuat aplikasi sistem penjualan yang bisa langsung diakses oleh pada *customer*.
- Desain fokus pada merancang desain perangkat lunak pada struktur data, arsitektur perangkat lunak, *interface*, dan tahap koding yang akan digunakan.
- Tahap koding secara mekanis. Setelah tahapan desain selesai maka dibuatnya *function coding* untuk tiap-tiap *form* dari aplikasi sistem penjualannya.
- Pengujian sistem ini dilakukan untuk mencari tahu apakah tahapan dari langkah sebelumnya sudah sesuai.

2.4 Rancangan Pengujian

Metode pengujian yang akan digunakan perangkat lunak yaitu metode *Black Box*, Metode ini memungkinkan pengujian yang lebih objektif dan terstruktur, dengan fokus pada input dan output sistem tanpa memperhatikan struktur internal dari sistem [10]. Metode pengujian *black box* berfokus pada spesifikasi fungsional dari perangkat lunak, kumpulan kondisi input dan melakukan pengetestan pada fungsional program.

2.5 Rancangan Basis Data

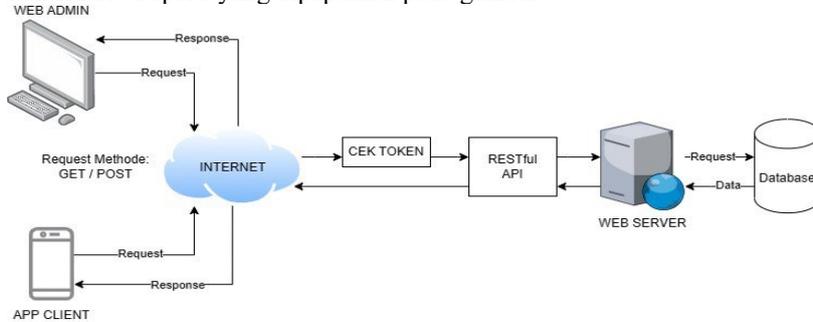
Dalam proses pembuatan aplikasi ini membutuhkan basis data yang berisikan semua informasi yang diperlukan untuk menjalankan aplikasi. Berikut bentuk *Entity Relationship Diagram* (ERD) dari *database* yang seperti yang dipaparkan pada gambar 2.



Gambar 2. Rancangan *Entity Relationship Diagram*

2.6 Rancangan Sistem

Dari hasil Analisis sistem didapatkannya gambaran bahwa *client* mengirimkan pesan *request* yang berisi HTTP *method* yang akan diperbarui, lokasi *resource* berupa URL, setelah itu server akan mengirimkan *response* sesuai dengan spesifikasi yang dibutuhkan oleh *client*. Dengan REST, aplikasi *client* bisa berupa apa saja hanya dengan memanfaatkan HTTP. Seperti yang dipaparkan pada gambar 3.



Gambar 3. Arsitektur Umum Sistem

2.7 Rancangan Web Service

Rancangan *web service* ini meliputi *service* yang digunakan dalam aplikasi dan penjelasan dan masing-masing *service* seperti nama layanan, nama fungsi, *endpoint*, *response code*, dan *status* untuk *detail*-nya terdapat pada Tabel 1.

Tabel 1. Hasil Rancangan *Web Service*

No	Nama Layanan	Fungsi	Endpoint	Response Code	Status
1	User Login	POST	/login	200	OK
2	User Register	POST	/login	200	OK
3	Upload Photo User	POST	/user/photo	200	OK
4	Update Profile User	POST	/user	200	OK
5	Get User Data	GET	/user	200	OK
6	Get Data Product	GET	/product	200	OK
7	Get Data Event	GET	/event	200	OK
8	Get Data Transactions	GET	/transactions	200	OK
9	Checkout	POST	/checkout	200	OK
10	User Logout	POST	/logout	200	OK

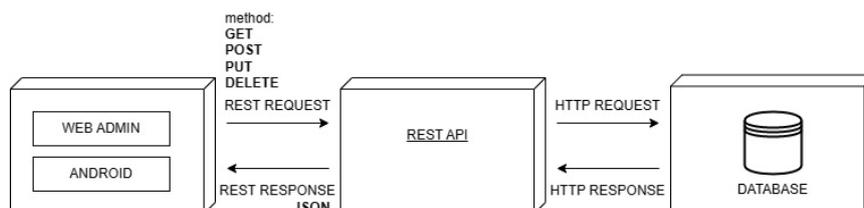
Untuk mengirimkan informasi yang diminta oleh *client*, *response code* HTTP Web Service dijelaskan sebagai berikut ini:

- 200, berarti permintaan berhasil.
- 401, berarti permintaan ditolak dan dikembalikan karena tidak terotorisasi.
- 404, berarti sumber daya yang diminta tidak dapat ditemukan.
- 500, berarti permintaan yang diminta gagal untuk di akses

3. HASIL DAN PEMBAHASAN

3.1 Deployment Diagram

Berikut ini menjelaskan gambaran dari lingkungan percobaan yang dibuat dalam bentuk *deployment diagram* implementasi dari sistem yang dirancang menggunakan *web service* metode RESTful API yang dipaparkan pada gambar 4 di bawah ini.



Gambar 4. Deployment Diagram

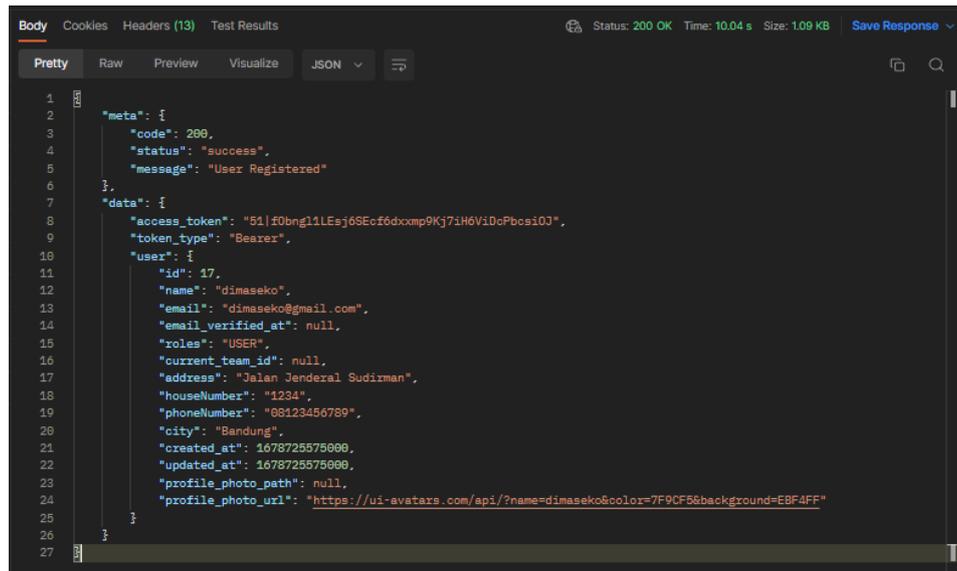
3.2 Pengujian Sistem

3.2.1 Pengujian *Web Service*

Untuk pengujian *web service* disini dapat menggunakan *tools* yang umum digunakan seperti *postman*.

a. Pengujian *Web Service Register*

Pada pengujian *web service register*, *request register* dilakukan untuk melakukan pembuatan akun untuk *user* pada aplikasi *mobile*, jika *request register* berhasil maka akan membuat *access token* dan mendapatkan *response status code success 200*, Seperti yang dipaparkan pada gambar 5.



```

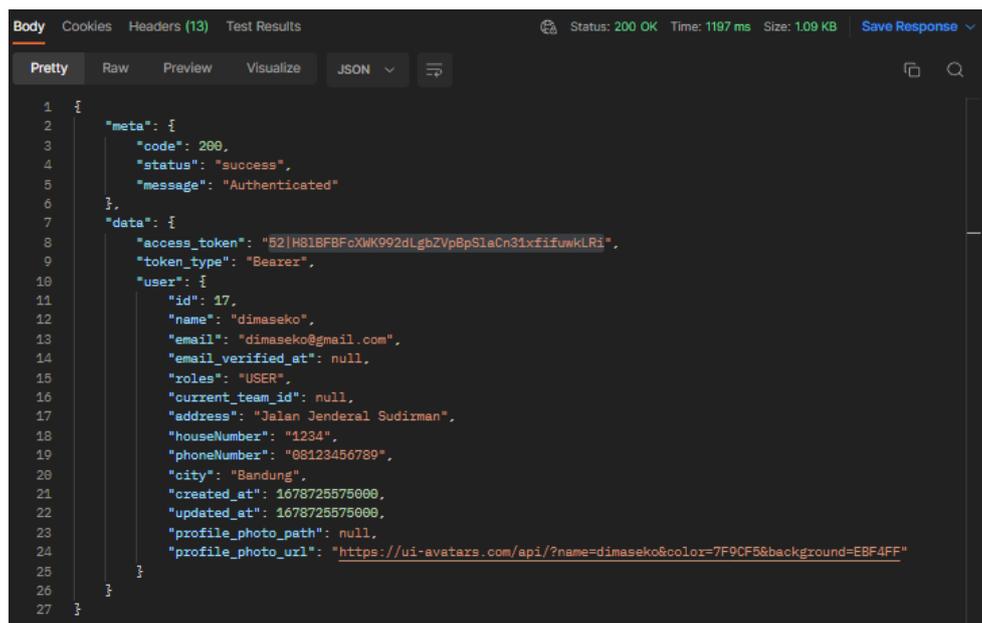
1  {
2    "meta": {
3      "code": 200,
4      "status": "success",
5      "message": "User Registered"
6    },
7    "data": {
8      "access_token": "51|f0bng11lEsj6SEcf6dxmp9Kj7iH6V1DcPbos10J",
9      "token_type": "Bearer",
10     "user": {
11       "id": 47,
12       "name": "dimaseko",
13       "email": "dimaseko@gmail.com",
14       "email_verified_at": null,
15       "roles": "USER",
16       "current_team_id": null,
17       "address": "Jalan Jenderal Sudirman",
18       "houseNumber": "1234",
19       "phoneNumber": "08123456789",
20       "city": "Bandung",
21       "created_at": "1678725575000",
22       "updated_at": "1678725575000",
23       "profile_photo_path": null,
24       "profile_photo_url": "https://ui-avatars.com/api/?name=dimaseko&color=7F9CF5&background=EBF4FF"
25     }
26   }
27 }

```

Gambar 5. Pengujian *Web Service Register*

b. Pengujian *Web Service Login*

Pada pengujian *web service login*, *request login* dilakukan untuk melakukan akses terhadap aplikasi dengan memasukan *email* dan *password*, jika *request login* berhasil maka akan membuat *access token* dan mendapatkan *response status code success 200*, Seperti yang dipaparkan pada gambar 6.



```

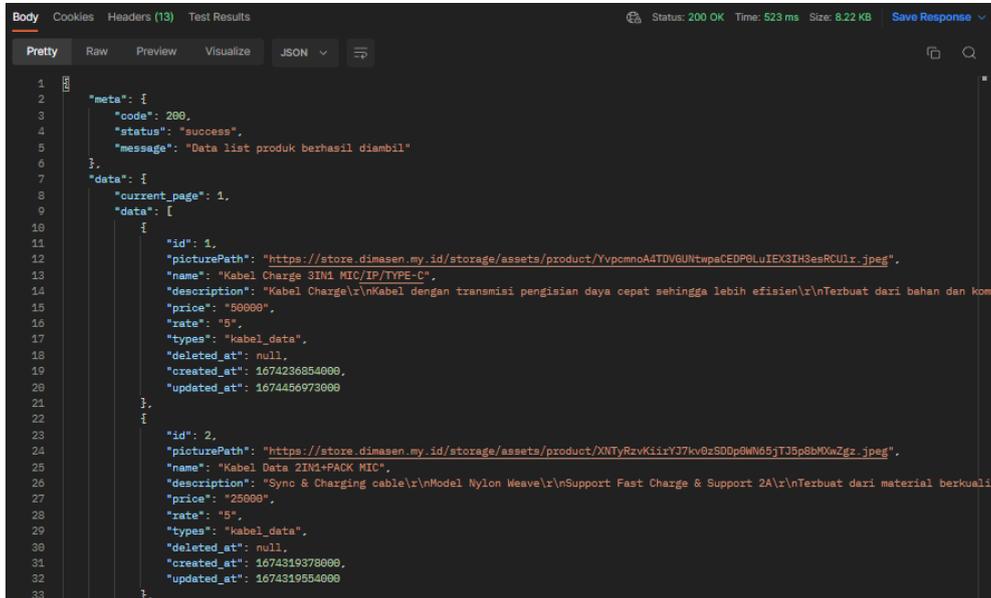
1  {
2    "meta": {
3      "code": 200,
4      "status": "success",
5      "message": "Authenticated"
6    },
7    "data": {
8      "access_token": "52|H818FBFoXWk992dLgbZVpBpSlaCn31xfifuwkLRi",
9      "token_type": "Bearer",
10     "user": {
11       "id": 47,
12       "name": "dimaseko",
13       "email": "dimaseko@gmail.com",
14       "email_verified_at": null,
15       "roles": "USER",
16       "current_team_id": null,
17       "address": "Jalan Jenderal Sudirman",
18       "houseNumber": "1234",
19       "phoneNumber": "08123456789",
20       "city": "Bandung",
21       "created_at": "1678725575000",
22       "updated_at": "1678725575000",
23       "profile_photo_path": null,
24       "profile_photo_url": "https://ui-avatars.com/api/?name=dimaseko&color=7F9CF5&background=EBF4FF"
25     }
26   }
27 }

```

Gambar 6. Pengujian *Web Service Login*

c. Pengujian Web Service Product

Pada pengujian *web service product*, *request product* bertujuan untuk menampilkan informasi pada data *product*, jika *request product* berhasil maka akan mendapatkan *response status code success 200*, Seperti yang dipaparkan pada gambar 7.



```

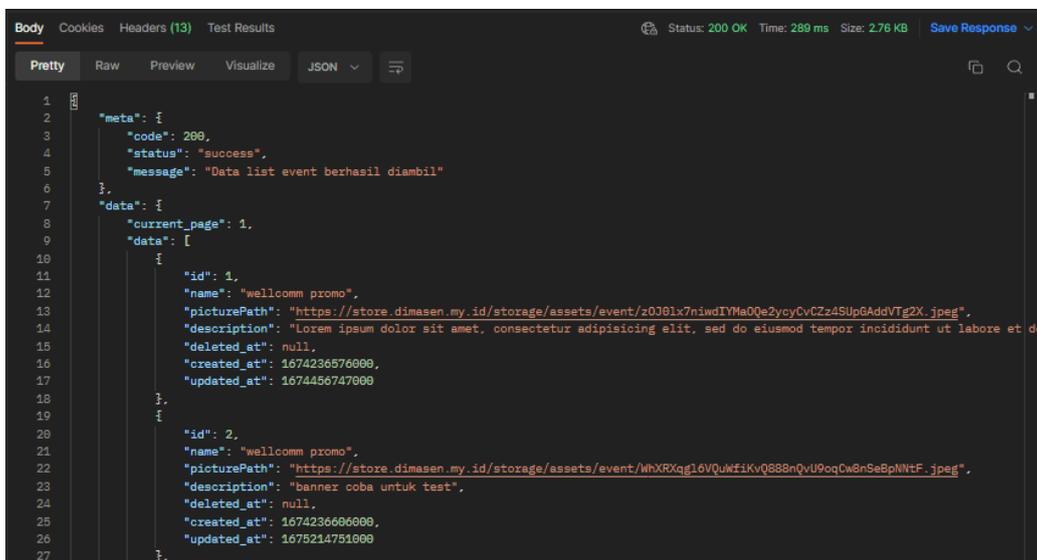
1  {
2    "meta": {
3      "code": 200,
4      "status": "success",
5      "message": "Data list produk berhasil diambil"
6    },
7    "data": {
8      "current_page": 1,
9      "data": [
10     {
11       "id": 1,
12       "picturePath": "https://store.dimasen.my.id/storage/assets/product/YvpcmoA4TDVGUNtpaCE0P8LuIEX3IH3eRCUzr.jpeg",
13       "name": "Kabel Charge 3IN1 MIC/IP/TYPE-C",
14       "description": "Kabel Charge\r\nKabel dengan transmisi pengisian daya cepat sehingga lebih efisien\r\nTerbuat dari bahan dan kom
15       "price": "58000",
16       "rate": "5",
17       "types": "kabel_data",
18       "deleted_at": null,
19       "created_at": 1674236854000,
20       "updated_at": 1674456973000
21     },
22     {
23       "id": 2,
24       "picturePath": "https://store.dimasen.my.id/storage/assets/product/XNTyRzvKiirY37kv0zSD0pM65jT35p8bMwZgz.jpeg",
25       "name": "Kabel Data 2IN1+PACK MIC",
26       "description": "Sync & Charging cable\r\nModel Nylon Weave\r\nSupport Fast Charge & Support 2A\r\nTerbuat dari material berkuali
27       "price": "25000",
28       "rate": "5",
29       "types": "kabel_data",
30       "deleted_at": null,
31       "created_at": 1674319378000,
32       "updated_at": 1674319554000
33     }
34   ]
35 }

```

Gambar 7. Pengujian Web Service Product

d. Pengujian Web Service Event

Pada pengujian *web service event*, *request event* bertujuan untuk menampilkan informasi pada data *event*, jika *request event* berhasil maka akan mendapatkan *response status code success 200*, Seperti yang dipaparkan pada gambar 8.



```

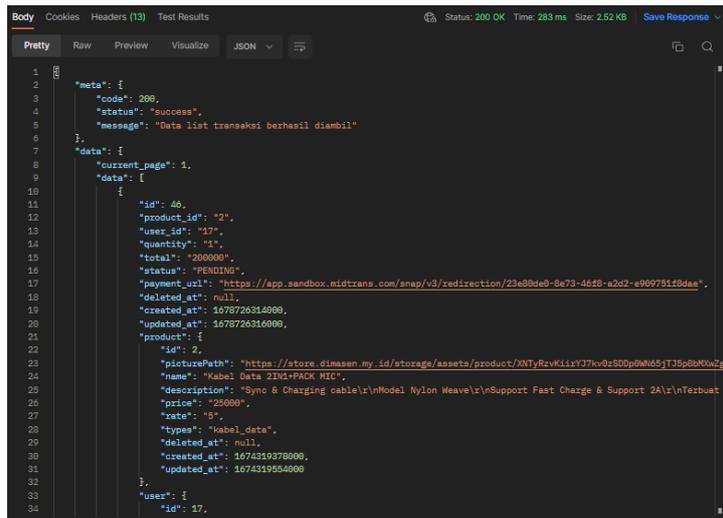
1  {
2    "meta": {
3      "code": 200,
4      "status": "success",
5      "message": "Data list event berhasil diambil"
6    },
7    "data": {
8      "current_page": 1,
9      "data": [
10     {
11       "id": 1,
12       "name": "welocmm promo",
13       "picturePath": "https://store.dimasen.my.id/storage/assets/event/z0J0lx7n1wdIYMa0qe2ycyCvCZz4SUpGAddVTg2X.jpeg",
14       "description": "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et d
15       "deleted_at": null,
16       "created_at": 1674236576000,
17       "updated_at": 1674456747000
18     },
19     {
20       "id": 2,
21       "name": "welocmm promo",
22       "picturePath": "https://store.dimasen.my.id/storage/assets/event/WHXRXqg16VQuWfiKvQ888nQvU9cqCw8nSeBpNHF.jpeg",
23       "description": "bannex coba untuk test",
24       "deleted_at": null,
25       "created_at": 1674236606000,
26       "updated_at": 1675214751000
27     }
28   ]
29 }

```

Gambar 8. Pengujian Web Service Event

e. Pengujian Web Service Transaction

Pada pengujian *web service transaction*, *request transaction* bertujuan untuk menampilkan informasi pada data transaksi, jika *request transactions* berhasil maka akan mendapatkan *response status code success 200*, Seperti yang dipaparkan pada gambar 9.



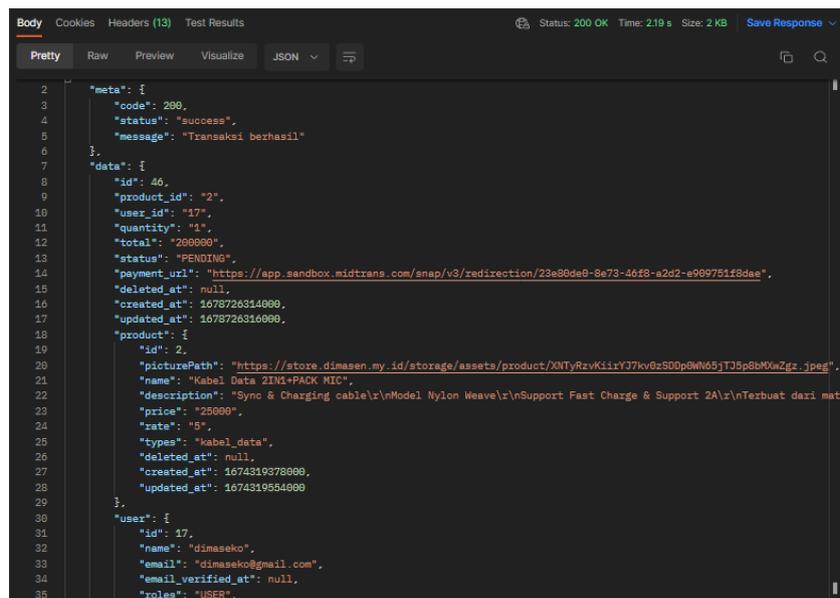
```

1 200
2  {
3    "meta": {
4      "code": 200,
5      "status": "success",
6      "message": "Data list transaksi berhasil diambil"
7    },
8    "data": {
9      "current_page": 1,
10     "data": [
11       {
12         "id": 46,
13         "product_id": "2",
14         "user_id": "17",
15         "quantity": "1",
16         "total": "290000",
17         "status": "PENDING",
18         "payment_url": "https://app.sandbox.midtrans.com/snap/v3/redirection/23e88de9-8e73-46f8-a2d2-e909751f3dae",
19         "deleted_at": null,
20         "created_at": 1678726314898,
21         "updated_at": 1678726316098,
22         "product": {
23           "id": 2,
24           "picturePath": "https://store.dimasen.my.id/storage/assets/product/XNTyRzvKiiyJ7kvQzS0Dp9Wl65jT35p8MxwZg",
25           "name": "Kabel Data 2IN1+PACK MIC",
26           "description": "Sync & Charging cable\l\nModel Nylon Weave\l\nSupport Fast Charge & Support 2A\l\nTextbuat",
27           "price": "290000",
28           "rate": "5",
29           "types": "kabel_data",
30           "deleted_at": null,
31           "created_at": 1674319378000,
32           "updated_at": 1674319354900
33         },
34         "user": {
35           "id": 17,
  
```

Gambar 9. Pengujian *Web Service Transaction*

f. Pengujian *Web Service Checkout*

Pada pengujian *web service checkout*, *request checkout* bertujuan untuk melakukan transaksi dan data transaksi akan disimpan kedalam *database*, jika *request checkout* berhasil maka akan mendapatkan *response status code success 200*, Seperti yang dipaparkan pada gambar 10.



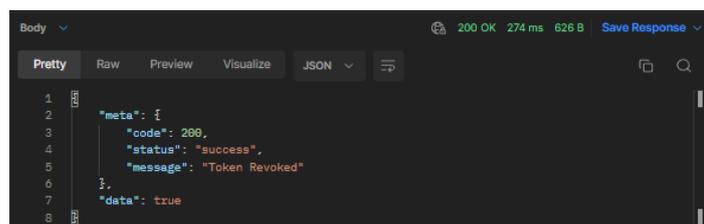
```

2 200
3  {
4    "meta": {
5      "code": 200,
6      "status": "success",
7      "message": "Transaksi berhasil"
8    },
9    "data": {
10     "id": 46,
11     "product_id": "2",
12     "user_id": "17",
13     "quantity": "1",
14     "total": "290000",
15     "status": "PENDING",
16     "payment_url": "https://app.sandbox.midtrans.com/snap/v3/redirection/23e88de9-8e73-46f8-a2d2-e909751f3dae",
17     "deleted_at": null,
18     "created_at": 1678726314898,
19     "updated_at": 1678726316098,
20     "product": {
21       "id": 2,
22       "picturePath": "https://store.dimasen.my.id/storage/assets/product/XNTyRzvKiiyJ7kvQzS0Dp9Wl65jT35p8MxwZgz.jpeg",
23       "name": "Kabel Data 2IN1+PACK MIC",
24       "description": "Sync & Charging cable\l\nModel Nylon Weave\l\nSupport Fast Charge & Support 2A\l\nTextbuat dari mat",
25       "price": "290000",
26       "rate": "5",
27       "types": "kabel_data",
28       "deleted_at": null,
29       "created_at": 1674319378000,
30       "updated_at": 1674319354900
31     },
32     "user": {
33       "id": 17,
34       "name": "dimaseko",
35       "email": "dimaseko@gmail.com",
36       "email_verified_at": null,
37       "roles": "USER",
  
```

Gambar 10. Pengujian *Web Service Checkout*

g. Pengujian *Web Service Logout*

Pada pengujian *web service logout*, *request logout* bertujuan untuk *merevoke token* dari aplikasi, jika *request logout* berhasil maka akan mendapatkan *response status code success 200*, Seperti yang dipaparkan pada gambar 11.



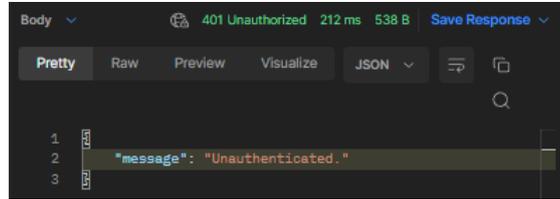
```

1 200 OK
2  {
3    "meta": {
4      "code": 200,
5      "status": "success",
6      "message": "Token Revoked"
7    },
8    "data": true
  
```

Gambar 11. Pengujian *Web Service Logout*

h. Pengujian *Web Service* Gagal

Pada pengujian kali ini ditunjukkan apabila token tidak terbentuk, maka *response* dari *web service* akan menjadi 401, dan data *message* dikembalikan dengan status *unauthenticated*. Seperti yang dipaparkan pada gambar 12.



Gambar 12. Pengujian *Web Service* Gagal

3.2.2 Pengujian Website Admin

Berikut ini merupakan hasil dari pengujian sistem dengan metode *black box testing* aplikasi *website admin* pada tabel 2 berikut ini.

Tabel 2. Hasil Pengujian *Black Box Testing* Website Admin

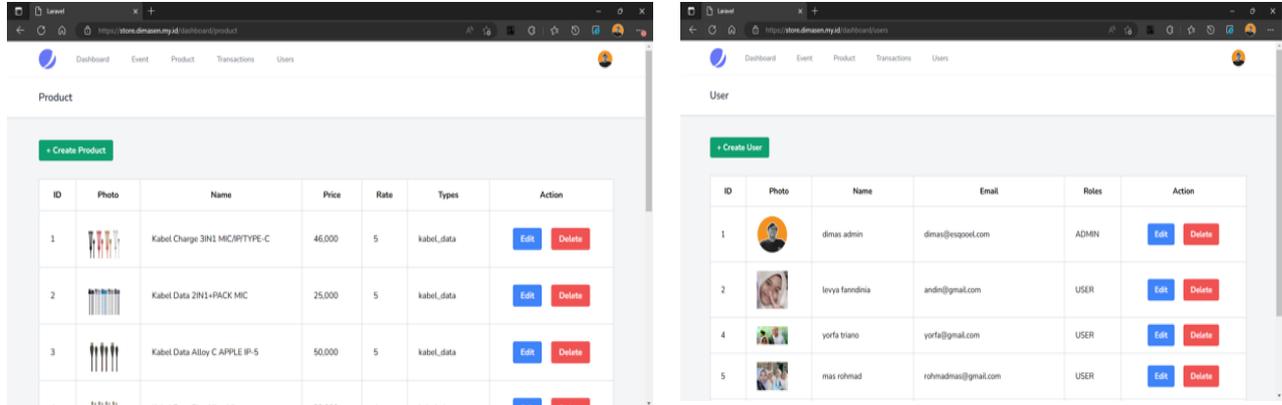
Skenario pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
Login aplikasi web admin	Login username dan password	Aplikasi memproses login dan pindah ke halaman <i>dashboard</i>	Sesuai Harapan	valid
Pilih menu <i>product</i>	Klik <i>product</i> di <i>dashboard</i> menu	Pindah ke halaman <i>index product</i>	Sesuai Harapan	valid
Menambah <i>product</i> baru	Klik tombol <i>create product</i>	Masuk ke <i>form</i> pembuatan <i>product</i> baru	Sesuai Harapan	valid
Merubah data <i>product</i> yang tersedia	Klik tombol <i>edit</i> pada kolom <i>action</i> di halaman <i>index product</i>	Masuk ke <i>form</i> perubahan <i>product</i> yang tersedia	Sesuai Harapan	valid
Menghapus <i>product</i>	Klik tombol <i>delete</i> pada kolom <i>action</i> di halaman <i>index product</i>	Data <i>product</i> berhasil dihapus dari <i>database</i>	Sesuai Harapan	Valid
Menambah <i>event</i> baru	Klik tombol <i>create event</i>	Masuk ke <i>form</i> pembuatan <i>event</i> baru	Sesuai Harapan	valid
Merubah data <i>event</i> yang tersedia	Klik tombol <i>edit</i> pada kolom <i>action</i> di halaman <i>index event</i>	Masuk ke <i>form</i> perubahan <i>event</i> yang tersedia	Sesuai Harapan	valid
Menghapus <i>event</i>	Klik tombol <i>delete</i> pada kolom <i>action</i> di halaman <i>indeks</i>	Data <i>event</i> berhasil dihapus dari <i>database</i>	Sesuai Harapan	Valid
Menambah <i>users</i> baru	Klik tombol <i>create users</i>	Masuk ke <i>form</i> pembuatan <i>users</i> baru	Sesuai Harapan	valid
Merubah data <i>users</i> yang tersedia	Klik tombol <i>edit</i> pada kolom <i>action</i> di halaman <i>index users</i>	Masuk ke <i>form</i> perubahan <i>users</i> yang tersedia	Sesuai Harapan	valid
Menghapus <i>users</i>	Klik tombol <i>delete</i> pada kolom <i>action</i> di halaman <i>index users</i>	Data <i>users</i> berhasil dihapus dari <i>database</i>	Sesuai Harapan	Valid
Melihat <i>detail transactions</i>	Klik tombol <i>view</i> pada kolom <i>action</i> di halaman <i>index transactions</i>	Pindah ke halaman <i>detail transactions</i>	Sesuai Harapan	Valid
Merubah status <i>transaction</i>	Klik salah satu tombol <i>on delivery/delivered/cancelled</i>	Status transaksi berubah	Sesuai Harapan	Valid
Menghapus <i>Transaction</i>	Klik tombol <i>delete</i> pada kolom <i>action</i> di halaman <i>index transactions</i>	Data <i>Transactions</i> berhasil dihapus dari <i>database</i>	Sesuai Harapan	Valid
Keluar aplikasi	Klik <i>dropdown</i> di <i>profile</i> pada menu <i>navigasi</i> bar lalu pilih <i>log out</i>	Keluar aplikasi, dan berpindah ke halaman <i>login</i> awal	Sesuai Harapan	valid

3.3 Implementasi Aplikasi

Implementasi aplikasi berguna untuk mengetahui apakah program yang sudah selesai dapat berfungsi dengan baik atau jika mengalami kesalahan yang tidak diinginkan. Maka program tersebut harus diuji terlebih dahulu agar berjalan Sesuai Harapan. Implementasi ditunjukkan pada gambar 14, 15, dan 16 di bawah ini.

3.3.1 Implementasi Website (Admin)

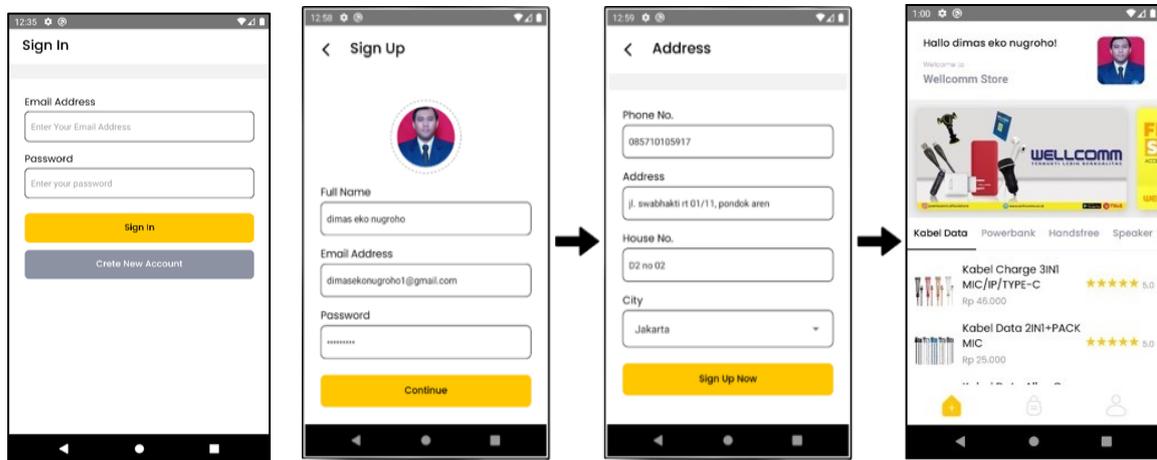
Implementasi ini ditunjukkan hanya bisa diakses oleh admin, yang memiliki fungsi untuk bisa menambahkan data yang akan bisa ditampilkan oleh *mobile* dan *web* aplikasi pada *customer*. Dapat dilihat pada gambar 14.



Gambar 13. Implementasi Form Users dan Products

3.3.2 Implementasi Android (Client)

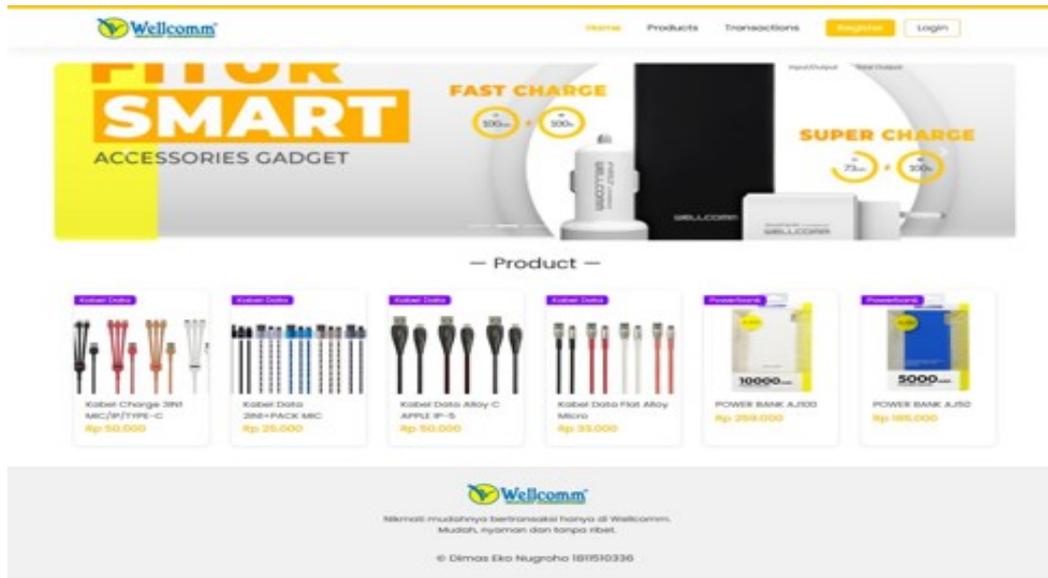
Implementasi aplikasi *mobile* ini ditunjukkan untuk *customer* wellcommstore, sebelum memulai aplikasi *customer* diwajibkan sudah membuat akun, jika sudah memiliki akun *customer* bisa mencoba untuk *login* aplikasi. Setelah berhasil *login* maka *customer* dapat melihat *product* dan bisa juga untuk langsung membeli *product* yang diinginkan. Seperti yang ditunjukkan pada gambar 15.



Gambar 14. Implementasi login User dan Form Register User

3.3.3 Implementasi Website (Client)

Implementasi aplikasi web ini ditunjukkan untuk *customer* apabila ingin melihat *product* wellcommstore dari *website*, jika hanya ingin melihat *product* saja maka *customer* tidak perlu melakukan *login* aplikasi terlebih dahulu. Namun jika ingin membeli *product* maka *customer* diwajibkan untuk *login* aplikasinya. Seperti yang dipaparkan pada gambar 16.



Gambar 15. Implementasi Menu Utama Web Client

4. KESIMPULAN

Berdasarkan hasil dari analisis penelitian, maka kesimpulan dari penelitian ini adalah aplikasi ini bisa membantu *customer* dalam membeli barang secara *online*, menyediakan catatan transaksi untuk admin sebagai dokumentasi. Selanjutnya, *Web service* sudah menggunakan laravel sanctum untuk mengimplementasikan autentikasi API pada aplikasi Laravel. Pada penelitian berikutnya akan menambahkan fitur *add to cart* agar mempermudah *customer* jika ingin membeli lebih dari satu tipe *product*. Serta memberikan notifikasi terhadap *admin* dan *customer* jika adanya transaksi yang baru dibuat dan juga notifikasi pada *customer* jika status pengiriman sudah dirubah oleh *admin*. Selanjutnya juga bisa langsung terintegrasi dengan jasa pengiriman agar dapat melacak barang secara langsung dari aplikasi.

DAFTAR PUSTAKA

- [1] F. M. Alharbi, T. A. Alshammari, S. M. Alotaibi, and A. M. Aldakhil, "The Role of Information Technology in the Development of Society," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 7, no. 2, p. 81, 2021.
- [2] R. Yogiswara and D. R. Astriyanto, "Penerapan Web Service dan Firebase Notification Pada Pengubahan Aplikasi Gerakan Nasi bungkus Jember Berbasis Android" JIP (Jurnal Informatika Polinema), vol. 4, no. 2, p. 161, Aug. 2018.
- [3] M. S. Rahman, T. Sarker, and M. R. Hasan, "Web Services: Overview, Architectures, Standards, and Research Issues," in *IEEE Access*, vol. 9, pp. 47953-47970, 2021.
- [4] K. Rizki and A. Adil, "Implementasi Google Maps API Berbasis Android untuk Lokasi Fasilitas Umum di Kabupaten Sumbawa," *J. Matrik*, vol. 17, no. 2, pp. 34–44, 2018.
- [5] R. Choirudin and A. Adil, "Implementasi REST API Web Service dalam membangun aplikasi multiplatform untuk jasa usaha," *Jurnal Manajemen, Teknik Informatika & Rekayasa Komputer*, vol. 18, no. 2, pp. 284–293, May 2019.
- [6] V. Kay Gow, S. Kosasi, T. Wijaya, David, and R. Laipaka, "Penerapan Teknologi RESTFUL Web Service Aplikasi Mobile Jasa Make Up," *InfoSys Journal*, vol. 6, no. 2, pp. 127–140, 2022.
- [7] Baharuddin, H. Wakkang, and B. Irianto, "Implementasi Web Service Dengan Metode REST API Untuk Integrasi Data Covid 19 di Sulawesi Selatan," *Jurnal Sintaks Logika*, vol. 2, no. 1, pp. 236–241, 2022.
- [8] Herfandi, M. Julkarnaik, and M. Hanif, "Desain Dan Implementasi RESTful Web Services Untuk Integrasi Data dan Aplikasi," *Jurnal Informatika Teknologi dan Sains*, vol. 4, no. 1, pp. 36–41, 2022.
- [9] R. Kusumawati, "Analisis Perbandingan Metode Waterfall dan Agile pada Pengembangan Perangkat Lunak," *Jurnal Sains dan Seni ITS*, vol. 8, no. 2, pp. A1-A6, 2019.
- [10] A. A. Wibowo, R. Sarno and R. Wardoyo, "Penguujian Perangkat Lunak Berbasis Web dengan Metode Black Box di Indonesia," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 3, no. 4, pp. 921-926, 2019.