

**Article history**

Received September 06, 2022

Accepted February 15, 2023

***REST API BACKEND APLIKASI E-COMMERCE SECONDHAND MENGGUNAKAN FRAMEWORK SPRING BOOT*****Tania Prilsafira<sup>1</sup>, Yesi Novaria Kunang<sup>2,\*</sup>, Muhammad Hatta Putra<sup>3</sup>**<sup>1</sup> Sistem Informasi, Universitas Bina Darma<sup>2</sup> Ilmu Komputer, Universitas Bina Darma<sup>3</sup> Binar Academy

email: tania.prilsafira18@gmail.com, yesinovariakunang@binadarma.ac.id, hatta.putraaa@gmail.com

***Abstract***

*The advancement in information technology is currently rapidly. Information technology can make it easier for humans to work or interact with other people. Technology applications have many various uses. One of the uses is that it can be a means of buying and selling goods or what we often call e-commerce. The SECONDHAND application developed in this study focuses on API Backend selling used goods using the Spring Boot framework. The method used is the SCRUM method which is a type of System Development Life Cycle (SDLC) design method. This research produces a REST API-based backend system.*

***Keywords:*** application, e-commerce, API, Spring Boot.

***Abstrak***

Perkembangan teknologi informasi saat ini berkembang pesat. Dengan adanya teknologi informasi dapat mempermudah manusia dalam bekerja atau berinteraksi dengan orang lain. Banyak sekali kegunaan aplikasi teknologi, salah satu kegunaan yang ada adalah dapat menjadi sarana jual beli barang atau yang sering kita sebut dengan *e-commerce*. Aplikasi SECONDHAND yang dikembangkan pada penelitian ini berfokus pada pengembangan *API Backend* penjualan barang bekas dengan menggunakan *framework Spring Boot*. Metode yang digunakan adalah metode SCRUM yang merupakan jenis metode perancangan *System Development Life Cycle (SDLC)*. Penelitian ini menghasilkan system *backend* berbasis REST API.

***Kata Kunci:*** aplikasi, e-commerce, API, Spring Boot

## 1. PENDAHULUAN

Perkembangan teknologi informasi yang makin pesat mempermudah manusia untuk bekerja ataupun untuk berinteraksi dengan orang lain [1]. Salah satu jenis teknologi informasi yang berkembang pesat dalam beberapa tahun terakhir berupa aplikasi. Aplikasi sendiri merupakan penerapan proses penyimpanan data, pekerjaan atau transaksi ke dalam suatu sarana atau media yang bisa mengimplementasikan hal atau pekerjaan tersebut sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari hal data, permasalahan, dan pekerjaan itu sendiri [2].

Dunia bisnis sekarang ini mengalami perkembangan pesat terutama dalam hal teknologi informasi dan komunikasi. Salah satu bentuk aplikasi bisnis yang sangat penting di era perkembangan internet adalah aplikasi penjualan *online* yang dikenal dengan istilah *e-commerce*. *E-commerce* sendiri merupakan proses transaksi jual beli dengan menggunakan media elektronik. Pengaplikasian *e-commerce* dapat memudahkan konsumen untuk melakukan pembelian produk secara *online* [3]. Selain itu dengan *e-commerce* juga memungkinkan para penjual saling bersaing untuk mendapatkan konsumen dengan mencari tahu pengalaman konsumen dalam membeli suatu produk. Di sisi produsen juga memerlukan perubahan strategi pasar yang bisa mempengaruhi perilaku konsumen. *E-commerce* merupakan pilihan alternatif untuk mengikuti zaman yang serba cepat, instan dan mudah [4].

Potensi penjualan *online* juga mulai dilirik untuk menawarkan barang bekas. Selama ini penjualan barang bekas pakai biasanya dipasarkan kepada tetangga dan kerabat dengan respon yang cukup baik. Sebagian penjual juga sudah memasarkan produknya melalui media situs jejaring sosial. Komunikasi antara pengelola dan pembeli dilakukan melalui sms atau melalui komentar yang diberikan oleh situs jejaring sosial. Kemudian jika ada kesepakatan, pembeli dapat mentransfer sejumlah uang yang disepakati kedua belah pihak ke nomor rekening pengelola dan barang dikirim melalui jasa kurir. Akan tetapi resiko transaksi penjualan menggunakan media sosial sangat rentan dengan kasus penipuan [5].

Dalam pengembangannya penjualan barang bekas berpotensi untuk dilakukan transaksi secara *online*. Konsep pengembangan aplikasi penjualan bertujuan menciptakan, mengelola, dan memperluas hubungan komersial *online*. Proses

bisnis berbasis teknologi elektronik bisa menghubungkan perusahaan, konsumen dan masyarakat dalam bentuk transaksi elektronik dan pertukaran atau penjualan barang, jasa, dan informasi secara elektronik. Untuk menjawab masalah tersebut maka pada penelitian ini dilakukan perancangan dan pengembangan sistem pelayanan donasi berbasis *Application Programming Interface* (API) yang berfungsi sebagai *Backend* dan diimplementasikan pada Aplikasi *e-commerce* SECONDHAND. *Backend* yang dikembangkan bekerja di sisi server dan database untuk melayani transaksi permintaan dari aplikasi *client*. Untuk mempermudah pengembangan *backend* maka peneliti menerapkan arsitektur *Representational State Transfer Application Programming Interface* (REST API).

REST yang diciptakan oleh Roy Fielding dari University of California merupakan layanan web yang sangat sederhana dan ringan dibandingkan dengan arsitektur sejenis seperti SOAP [6]. REST yang memisahkan antara klien dan server memungkinkan pengembangan aplikasi yang independen baik dari sisi platform, bahasa pemrograman, dan memiliki visibilitas, keandalan dan skalabilitas. Sedangkan *Application Programming Interface* (API) merupakan *interface* yang memungkinkan aplikasi *backend* dapat berinteraksi dan berbagi data. API sendiri terdiri dari *elemen function*, *protocols*, dan *tools* lainnya digunakan pengembang untuk membuat sebuah aplikasi. Web API sendiri berjalan di semua jenis server seperti Apache atau web server lainnya. Web API mendukung banyak bahasa pemrograman yang digunakan [7].

Penelitian ini memanfaatkan *framework spring* yang dikenal dengan *spring boot* untuk memudahkan pengembangan aplikasi *micro service backend* berbasis Java. Arsitektur *micro services* ini digunakan untuk membagi fungsionalitas aplikasi menjadi banyak bagian, atau banyak layanan mikro berdasarkan proses bisnis dan layanan yang saling berhubungan, menjadi aplikasi tunggal dengan proses bisnis yang lengkap. Keuntungan dari arsitektur *spring boot* ini memungkinkan penambahan banyak layanan mikro dapat tanpa mempengaruhi kerja layanan lainnya [8].

Dengan *framework spring boot* dikembangkan *e-commerce* yang melayani penjualan barang bekas. Aplikasi ini diberi nama SECONDHAND yang sedikit banyak terinspirasi dari OLX sebagai

*benchmark*. Platform aplikasi ini digunakan untuk jual beli barang secara *online*, khususnya barang bekas yang menyediakan berbagai kategori untuk kebutuhan pengguna. Pengguna di platform ini dapat bertindak sebagai penjual dan pembeli di akun yang sama. Selain itu, platform ini memiliki keunggulan fitur yang digunakan untuk mempertemukan penjual dan pembeli untuk menegosiasikan produk dan transaksi langsung di luar *platform*.

## 2. METODE PENELITIAN

Fokus penelitian yang dikembangkan adalah pembuatan REST API aplikasi *e-commerce* SECONDHAND (*benchmark* dari OLX) dengan menggunakan *framework* *Spring Boot*. Metode yang digunakan dalam perancangan REST API pada penelitian adalah *System Development Life Cycle (SDLC)*, yang mempunyai tahapan sebagai berikut:

- (1) **Identifikasi:** identifikasi dilakukan untuk mendapatkan kebutuhan informasi terkait transaksi jual beli khususnya untuk .
- (2) **Analisis:** pada tahapan ini dilakukan pengumpulan data dan informasi yang berhubungan dengan data apa saja yang harus ditampilkan oleh *Backend*.
- (3) **Perancangan:** pada tahapan ini peneliti merancang arsitektur sistem dan merancang REST API. Selain itu juga merancang database yang akan digunakan.
- (4) **Implementasi:** pada tahapan ini dikembangkan REST API dengan *framework* *Spring Boot* dan *frontend* dengan menggunakan bahasa pemrograman *Java*. Dilakukan pengujian dengan data *dummy* untuk pengiriman data API berbentuk JSON ke pihak *frontend*. Skema basis data dibuat menggunakan *Dewa.io Visual Studio Code* dengan ekstensi *Live Server*. GIT sebagai *Version Control System*, *web browser* menggunakan *Chrome* dan *postman* untuk mempermudah *testing* REST API. Tipe database *PostgreSQL* serta *Sequelize* untuk mempermudah lalu lintas data ke dalam database
- (5) **Uji Coba/Testing:** tahap terakhir melaksanakan uji coba terhadap masing-masing REST API yang telah dibuat pada situs URL (*Uniform Resource Locator*) website menggunakan bantuan *postman*. Tujuan dilakukan uji coba adalah agar API dapat berjalan menghantar-kan data secara optimal.

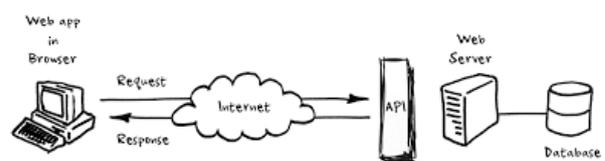
## 3. HASIL DAN PEMBAHASAN

REST API yang sudah dikembangkan digunakan dengan memanfaatkan aplikasi *WeShare*. Proses dalam perancangan yang ditempuh *Backend* pada *WeShare* dijelaskan mulai dari perancangan, konfigurasi hingga pengujian *backend* dari aplikasi SECONDHAND.

### 3.1. Hasil Rancangan Backend dan Arsitektur Sistem

*Backend* merupakan proses sistem yang berjalan di belakang layar sebuah *website* atau aplikasi. Pada penelitian ini *backend* dikembangkan dengan menggunakan bahasa pemrograman *Java*. *Backend* membantu mengkomunikasikan informasi atau data yang terdapat pada basis data kepada user melalui tampilan *User Interface (UI)* dari *frontend*. Aplikasi *client* dalam versi android dan website pada penelitian ini menggunakan metode REST untuk melakukan integrasi *Application Programming Interface*.

Gambar 1 merupakan arsitektur sistem yang menggambarkan skema pengambilan data menggunakan REST API. Data yang dikirim pengguna digabungkan dan dikirim dalam bentuk JSON. Data tersebut akan dikirim ke server API, dan server API akan melakukan proses permintaan ke database.



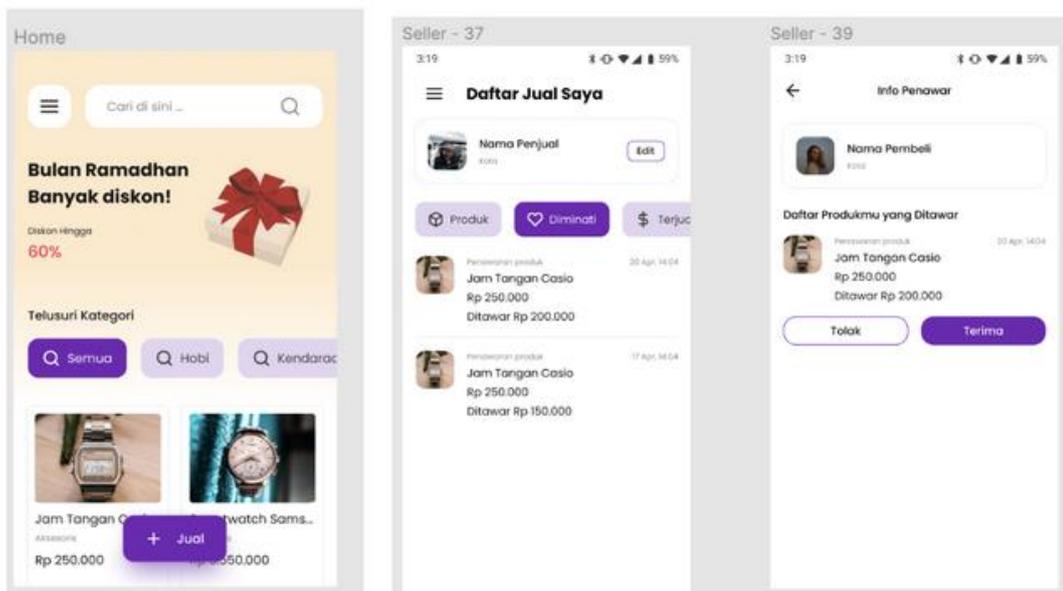
Gambar 1. Arsitektur Sistem.

### 3.2. Konfigurasi Aplikasi

Tahap awal pembuatan *project* *Spring Boot* adalah dengan menambahkan dependensi sesuai dengan kebutuhan yang akan digunakan. *Dependency* merupakan salah satu fitur penulisan *coding* dalam pengembangan aplikasi *Spring Boot*. Selanjutnya adalah mengkonfigurasi *Application Properties*. *Application Properties* pada *Spring Boot* berisi file konfigurasi yang berisi pengaturan yang digunakan dalam aplikasi. Gambar 2 merupakan konfigurasi *Application Properties* yang digunakan.

```
application.properties 1.25 KiB
1 spring.datasource.url=jdbc:mysql://localhost:3306/secondhand
2 spring.datasource.username=root
3 spring.datasource.password=
4 server.port=8081
5 # status="staging"
6 # logging.level.org.hibernate.type=TRACE
7
8 spring.jpa.hibernate.ddl-auto=create-drop
9 #spring.jpa.database-platform=org.hibernate.dialect.MySQL5Dialect
10 spring.jpa.properties.hibernate.format_sql=true
11
12
```

Gambar 2. *Application Properties* yang digunakan.



Gambar 3. Contoh sampel desain *Home* dan *Filter* yang dirancang.

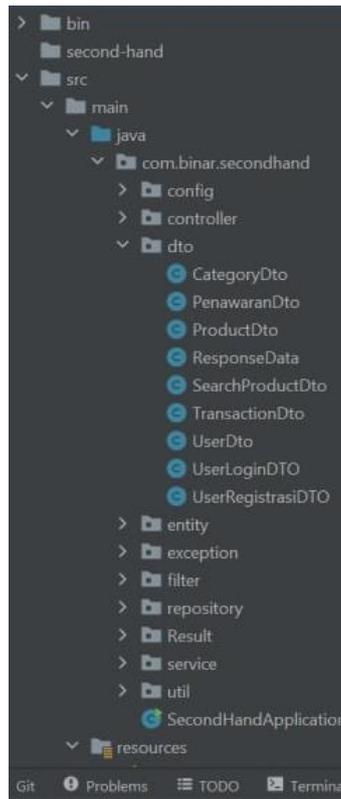
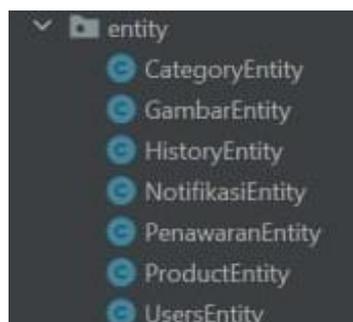
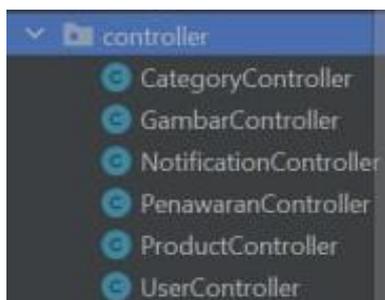
### 3.3. Hasil Perancangan API

Implementasi perancangan *REST API* dalam aplikasi *WeShare*, tim pengelola *Backend* melakukan analisa kebutuhan *API* dengan menggunakan desain gambar. Desain Sampel mempermudah pekerjaan tim *Backend* dalam proses analisa *API* dan kebutuhan data yang perlu disediakan.

### 3.4. Desain *Coding Foundation*

Proses desain *coding* merupakan hal yang penting diperhatikan dalam mendesain *REST API* guna menghindari kesalahan pada saat penggabungan *REST API* ke *cloud server*.

Kesalahan yang sering terjadi biasanya karena susunan folder yang berbeda pada server yang didesain oleh masing-masing tim pengembang *Backend*. Untuk itu perlu dipergunakan satu *coding foundation*. *Coding Foundation* yang dipergunakan menggunakan konsep pola desain arsitektur *MVC (Model, Control, View)*. Konsep *MVC* memiliki bagian-bagian tersendiri dan dapat dikembangkan secara individual. Proses desain *API* bisa dilakukan lebih cepat karena setiap pengembang akan lebih fokus mengerjakan satu bagian saja. Gambar 4 memperlihatkan konsep pengaturan folder *MVC* di aplikasi *WeShare* yang digunakan tim pengembang.

Gambar 4. Desain *Coding Foundation*Gambar 5. Model *Entity*Gambar 6. *Controllers*

Gambar 5 dan 6 masing-masing memperlihatkan model *entity* dan model *controller* pada arsitektur *MVC* yang bisa dikembangkan secara terpisah oleh tim pengembang.

### 3.5. Handling API Contracts

*Contract* merupakan tempat mendeklarasikan bagaimana *API* akan berperilaku. Di dalam *API contract* diatur URL, tindakan, argumen, contoh respons, dan detail lainnya (gambar 7). *API contract* juga diimplementasikan dengan aplikasi *WeShare* ke sesama tim pengembang.

```

{
  "produkId": 0,
  "produkname": "string",
  "hargaProduk": 0,
  "categoryId": 0,
  "categoryName": "string",
  "statusTerjual": "string",
  "statusfishlist": "NO",
  "deskripsi": "string",
  "image": [
    "string"
  ],
  "penawaran": [
    {
      "penawaranId": 0,
      "notifikasiId": 0,
      "userId": 0,
      "produkId": 0,
      "messageNotifikasi": "string",
      "time": "string",
      "imageUrl": "string",
      "productName": "string",
      "harga": 0,
      "hargaPenawaran": 0,
      "statusTerima": "string",
      "statusTawaran": "NEGOSIASI"
    }
  ],
  "historyId": 0,
  "userId": 0,
  "statusProduk": "PUBLISH"
}

```

Gambar 7. *API Contract*

*Contract* ditentukan oleh penyedia layanan dan ditujukan untuk konsumen *API* yaitu pengembang yang akan menggunakan *API*.

### 3.6. Desain Infrastruktur Backend

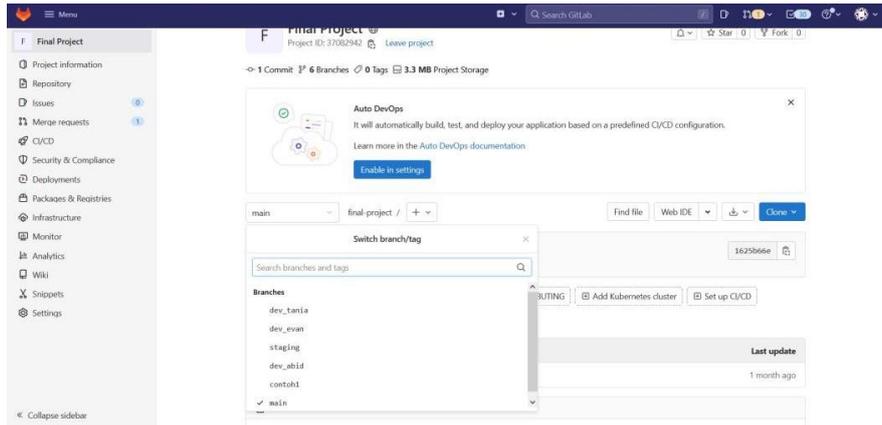
Rancangan Infrastruktur *REST API* pada aplikasi *WeShare* terbagi menjadi 3 (tiga) poin utama yaitu:

#### a) Version Control System

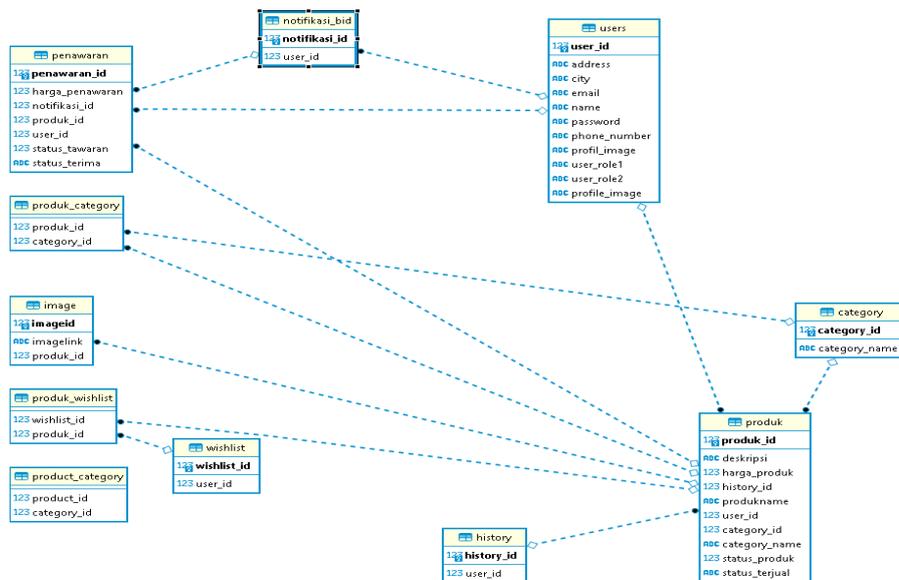
*Version Control System* yang dipergunakan dalam perancangan *REST API* menggunakan *GIT*. Pada penelitian ini *GITLAB* dimanfaatkan untuk akses jarak jauh ke layanan repositori yang mendukung sistem kontrol. Dengan kedua *tools* ini memudahkan tim untuk berkolaborasi. Sistem kontrol menjadi perantara layanan untuk proses *push* ke *server cloud*. Dengan demikian *REST API* dapat dimanfaatkan oleh *Frontend* dan *React Native*. Gambar 8 merupakan contoh *version control system* yang memperlihatkan proses *push* menggunakan *GIT* ke *remote* layanan repositories *GTILAB*.

Repositories *REST API WeShare* dapat diakses melalui link *GITLAB*

<https://gitlab.com/abidsutisna/final-project.git>



Gambar 8. Version Control System.



Gambar 9. Model basis data ORM yang dikembangkan.

b) Model Basis Data

Model basis data yang digunakan adalah ORM (*Object Relation Mapping*). Model ORM pada memperlihatkan relasi antar tabel satu dengan tabel yang lainnya. Gambar 9 memperlihatkan skema tabel ORM yang dikembangkan untuk memberikan gambaran alur lalu lintas objek data dalam basis data. Jenis basis data yang digunakan dalam penelitian adalah *PostgreSQL*.

c) Cloud Server

Langkah selanjutnya menentukan *cloud server* sebagai pusat untuk menyediakan *REST API*. URL server *cloud* menyediakan layanan bagi *Frontend* dan *React Native*. *Cloud server* yang digunakan disini menggunakan *HEROKU*.

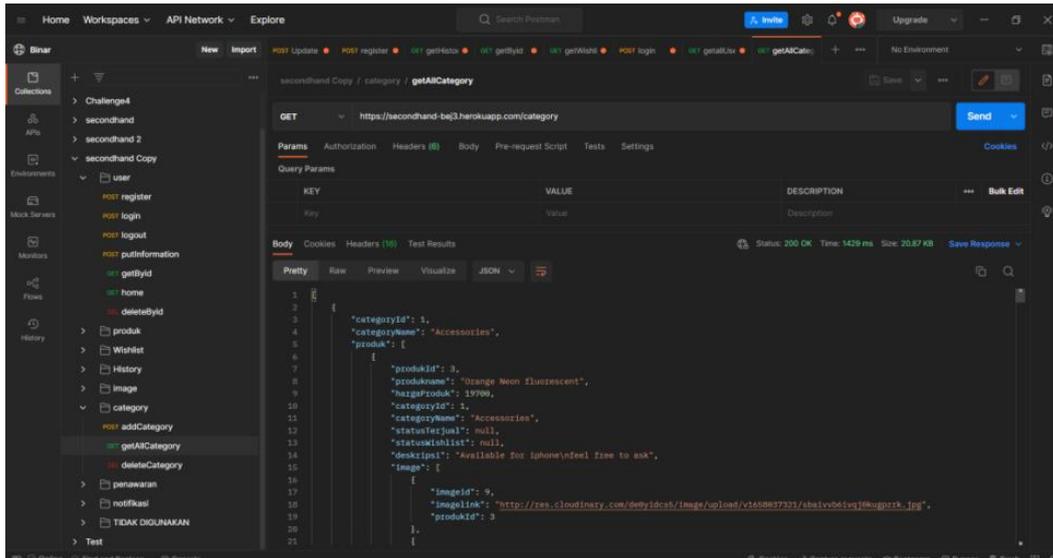
Berikut link *cloud server HEROKU REST API* :

<https://secondhand-bej3.herokuapp.com>

3.7. Metode Pengembangan

Metode pengembangan yang digunakan menggunakan metode *SCRUM*. Dengan metode *SCRUM* pengembangan program bisa dilakukan secara paralel oleh tim. Dengan metode ini mempermudah tim mengadopsi perubahan dalam mendesain aplikasi *WeShare SECONHAND* yang dikembangkan. Untuk menerapkan aplikasi *WeShare* ini digunakan *tools* *Trello*. Gambar 10 memperlihatkan aplikasi *WeShare SECONHAND* yang dimanajemen dengan *Trello*.





**Gambar 14.** Contoh output JSON Object

Pada *REST API Backend* untuk aplikasi WeShare yang dikembangkan terdapat beberapa fungsi dari metode *response request, endpoint* sebagai alamat API untuk mempermudah penggunaan API oleh *Frontend*. Layanan metode respon yang disediakan pada *WeShare API* antara lain berupa metoda GET untuk mengambil atau membaca data, metoda POST untuk mengembalikan output, metoda PUT adalah memperbarui item/sumber daya yang ada dan metoda DELETE untuk menghapus item/*resource* yang ada.

Layanan *Backend* yang dikembangkan untuk aplikasi *SECONDHAND* terdiri atas layanan user registrasi di mana user bisa memiliki dua role yaitu *seller* dan *buyer* agar dapat menggunakan seluruh fitur yang ada di *SECONDHAND*. Sebagai layanan *seller*, user dapat memposting produk sehingga dapat mulai menjual dan mendapatkan uang. Jika sudah memposting *seller* akan mendapatkan notifikasi pada setiap update produk, sehingga dapat merespon dengan cepat, apabila *seller* mendapatkan notifikasi bahwa ada *buyer* yang tertarik dengan produknya dan melakukan penawaran. *Seller* dapat mengaksesnya melalui halaman notifikasi dan daftar jual saya di halaman diminati. *Seller* dapat mengklik notifikasi tersebut untuk melihat detail info penawar, *seller* merespon penawaran dengan mengklik tombol Tolak atau Terima. Jika *seller* terima, maka dia akan melihat lembar tombol *product match* dan dapat menghubungi *buyer* melalui *WhatsApp* dan jika *seller* sudah bertransaksi di *WhatsApp* *seller* akan masuk ke

info penawar dari tab yang di minati di halaman daftar jual, dan *seller* dapat mengupdate status “Berhasil Dijual” atau “Batalkan Transaksi”. Sedangkan layanan *buyer* dapat menjelajahi beranda, memfilter daftar produk dengan tombol kategori, *buyer* dapat melihat detail produk dengan mengklik salah satu kartu produk. Memilih barang sesuai minat *buyer* dan melakukan tawar menawar dengan *seller*, *buyer* melihat *bottom sheet*/modal untuk memasukkan harga tawarannya, dan mengirimkannya ke penjual, jika *buyer* berhasil mengirimkan tawarannya dan melihat halaman *update* produk dengan CTA yang dinonaktifkan, *buyer* mendapatkan notifikasi jika *seller* menanggapi penawarannya, maka penawaran produk diterima, *buyer* tinggal menunggu *seller* menghubunginya dan jika penawaran produk ditolak, *buyer* dapat mengajukan kembali penawarannya di halaman detail.

Hasil pengujian dengan memanfaatkan *Postman* memperlihatkan Backend API yang dikembangkan bekerja dengan baik dengan memberikan output JSON sesuai dengan yang ditentukan. Untuk mempermudah pengembangan *Frontend* maka aplikasi API backend *SECONDHAND* ini dilengkapi dengan data JSON kategori objek WeShare secara lengkap yang dapat diakses melalui tautan <https://secondhand-bej3.herokuapp.com/category>. Selain itu juga disertakan dokumentasi API WeShare untuk pengembangan aplikasi *client* (*frontend*).

#### 4. PENUTUP

##### Kesimpulan

Penelitian ini menghasilkan rancangan Aplikasi *Backend Representatif State Transition Application Programming Interface (REST API)* untuk aplikasi penjualan barang bekas yang diberi nama *SECONDHAND*. Pada sisi aplikasi backend ini menyediakan layanan user registrasi, layanan seller dan layanan buyer.

Untuk memudahkan pengembangan aplikasi Frontend, maka aplikasi backend ini dilengkapi juga dengan dokumentasi API berupa layanan dan data kategori JSON untuk tiap layanan.

##### Saran

Untuk pengembangan lebih lanjut maka diperlukan pengujian aplikasi Backend yang dikembangkan langsung dengan aplikasi Frontend atau *React Native* yang diperlukan untuk melakukan transaksi.

#### 5. REFERENSI

- [1] S. Syamsuar and R. Reflianto, "Pendidikan dan tantangan pembelajaran berbasis teknologi informasi di era revolusi industri 4.0," *E-Tech J. Ilm. Teknol. Pendidik.*, vol. 6, no. 2, 2019.
- [2] E. Sany, "Aplikasi eVoting Pada Pemilihan Presiden Badan Eksekutif Mahasiswa (BEM) Universitas Nurdin Hamzah," in *Seminar Nasional Informatika (SENATIKA)*, 2021, pp. 398–408.
- [3] E. P. Fitriyadi, "Inovasi E-Commerce dan Starup Sebagai Tantangan Masyarakat Industri 4.0," *J. ESIT E-Bisnis Sist. Inf. Teknol. Inf.*, vol. 14, no. 3, 2021.
- [4] A. Roliansyah, A. P. Utami, A. Asnaini, and N. Afrianty, "Analisis Transaksi Jual Beli Online di Marketplace Shopee Menurut Konsep Bisnis Islam pada Masa Pandemi Covid 19," *Juremi J. Ris. Ekon.*, vol. 1, no. 5, pp. 443–456, 2022.
- [5] N. S. B. Maria and T. Widayati, "Dampak perkembangan ekonomi digital terhadap perilaku pengguna media sosial dalam melakukan transaksi ekonomi," *JKBM J. Konsep Bisnis Dan Manaj.*, vol. 6, no. 2, pp. 234–239, 2020.
- [6] R. Choirudin and A. Adil, "Implementasi Rest Api Web Service dalam Membangun Aplikasi Multiplatform untuk Usaha Jasa," *MATRIK J. Manaj. Tek. Inform. Dan Rekayasa Komput.*, vol. 18, no. 2, pp. 284–293, 2019.
- [7] S. N. Yanti and E. Rihyanti, "Penerapan Rest API Untuk Sistem Informasi Film Secara Daring," *J. Inform. Univ. Pamulang*, vol. 6, no. 1, pp. 195–201, 2021.
- [8] H. Suryotrisongko, D. P. Jayanto, and A. Tjahyanto, "Design and development of backend application for public complaint systems using microservice spring boot," *Procedia Comput. Sci.*, vol. 124, pp. 736–743, 2017.