

## Rancang Bangun Aplikasi *E-Commerce* Kerajinan Bambu Berbasis Android untuk Desa Belega

Muhammad Abel Jollando<sup>1</sup>, Putu Wira Buana<sup>2</sup>, Fajar Purnama<sup>3</sup>

<sup>1,2,3</sup>Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana, Badung, Indonesia

Email: <sup>1</sup>abel.jollando073@student.unud.ac.id, <sup>2</sup>wbhuana@it.unud.ac.id, <sup>3</sup>fajarpurnama@unud.ac.id

### ABSTRACT

The development of information and communication technology in Indonesia has growing up significantly in the last few years. Thus, there are many opportunities to digitize the business processes of a business and business ecosystem. One of them is in the field of digital economy by adapting business processes into a digital platform, as well as designing an *E-Commerce* platform in a village that has commodities inside it and increase the product visibility. This research uses Belega Village as a case study location considering that this village has commodities in the form of bamboo crafts. Research was conducted to build an android-based bamboo craft *E-Commerce* application which is expected to increase product visibility. The application itself is built using the Agile Development method using a Sprint model. This application is built with a microservices architecture with *Express.js* technology as a backend service and *React Native* to create client applications and *PostgreSQL* as a DBMS. The results of this research are in the form of an android-based *E-Commerce* application that is expected to increase the visibility of products from bamboo crafts from Belega Village.

*Keywords: E-Commerce, Belega Village, Bamboo Crafts, React Native, Express.js.*

### ABSTRAK

Perkembangan teknologi informasi dan komunikasi di Indonesia tengah mengalami perkembangan yang signifikan pada kurun waktu beberapa tahun terakhir. Dengan demikian, banyak sekali terbuka peluang-peluang untuk melakukan digitalisasi pada proses bisnis sebuah usaha maupun ekosistem usaha. Salah satunya sendiri dalam bidang ekonomi digital dengan mengadaptasi proses bisnis ke dalam sebuah platform digital, seperti halnya perancangan platform *E-Commerce* pada sebuah Desa yang memiliki komoditas di dalamnya agar komoditas terkait dapat meningkatkan visibilitas produknya. Penelitian ini menggunakan Desa Belega sebagai lokasi studi kasus mengingat Desa ini memiliki komoditas berupa kerajinan bambu. Penelitian dilakukan untuk membangun aplikasi *E-Commerce* kerajinan bambu berbasis android yang nantinya diharapkan mampu untuk meningkatkan visibilitas produk. Aplikasi sendiri dibangun dengan menggunakan metode *Agile Development* dengan menggunakan model *Sprint*. Aplikasi ini dibangun dengan arsitektur *microservices* dengan teknologi *Express.js* sebagai *backend service* dan *React Native* untuk membuat aplikasi *client* serta *PostgreSQL* sebagai DBMS. Hasil dari penelitian ini berupa aplikasi *E-Commerce* berbasis Android yang diharapkan mampu meningkatkan visibilitas produk dari kerajinan bambu dari Desa Belega.

*Kata Kunci: E-Commerce, Desa Belega, Kerajinan Bambu, React Native, Express.js.*

### 1. Pendahuluan

Dewasa ini, perkembangan teknologi informasi dan komunikasi sangatlah masif, termasuk di Indonesia. Dengan adanya teknologi informasi dan komunikasi yang pada dasarnya ditujukan untuk melakukan digitalisasi serta mempermudah hidup manusia dalam melakukan sesuatu. Kemajuan teknologi informasi menjadi salah satu pertimbangan masyarakat dalam memenuhi kebutuhannya. Kemudahan serta kualitas yang di tawarkan oleh kemajuan teknologi informasi menjadi nilai plus dan baik di mata masyarakat. Berkat

perkembangan teknologi yang pesat, pengusaha pun sudah mulai melakukan digitalisasi pada usahanya, seperti halnya UMKM yang sudah mulai menggunakan *E-Commerce* untuk meningkatkan visibilitas produk serta untuk memperluas pangsa pasarnya, sehingga sangat penting bagi pengusaha untuk memahami pentingnya digitalisasi sebagai indikator mencapai keberhasilan bisnis di lingkungan global yang dinamis saat ini [1].

Indonesia, negara agraris yang memiliki kekayaan hayati yang melimpah, khususnya bambu, dengan

menjadi rumah untuk sekitar 176 spesies dari total 1620 spesies bambu global yang menginspirasi beragam lapisan masyarakat di Indonesia untuk menciptakan ragam kerajinan produk dari bambu yang bernilai tinggi dan unik. Salah satu contoh nyata terjadi di Desa Karangasem, Kabupaten Bantul, DIY, yang merupakan contoh nyata bagaimana masyarakat desa dapat mengoptimalkan potensi bambu untuk meningkatkan kesejahteraan ekonomi di desa. Nyaris 90% dari warga desa menggantungkan hidup pada kerajinan bambu, hal ini sudah cukup untuk menunjukkan ketergantungan yang kuat pada sumber daya alam ini. Terlebih, pengrajin dari Desa Karangasem, Kabupaten Bantul, DIY, juga telah memanfaatkan *platform E-Commerce* sebagai sarana promosi dan meningkatkan visibilitas produk.

Desa Belega, adalah desa sentra kerajinan bambu yang terletak secara persis di Kecamatan Blahbatuh, Kabupaten Gianyar, Bali, memiliki potensi yang sama dalam mengembangkan industri kreatif kerajinan bambu ini. Namun, potensi dari Desa belum dapat teroptimalkan. Hal ini terjadi disebabkan karena sebagian besar pengrajin bambu di Desa Belega masih mengandalkan metode produksi, transaksi, dan pemasaran secara konvensional. Hasil kerajinan bambu seperti meja, kursi, dan dekorasi, umumnya dipasarkan melalui toko-toko lokal, sehingga jangkauan sangat terbatas dan kurang efisien. Selain itu, pengrajin juga cukup kesulitan dalam manajemen pengelolaan pesanan, sehingga pengelolaan pesanan dan pencatatan transaksi menjadi kurang efisien. Hal ini cukup kontras jika melihat dari besarnya potensi kerajinan bambu di Indonesia, yang mana menurut Dirjen Industri Kecil, Menengah, dan Aneka Kementerian Perindustrian, Reni Yanita, industri kerajinan memiliki peran penting dalam perekonomian nasional. Nilai ekspor industri kerajinan pada tahun 2023 mencapai 724 juta dolar AS, hal ini menunjukkan adanya peluang pasar yang luas termasuk untuk kerajinan bambu.

*E-Commerce* merupakan kegiatan transaksi jual beli dengan layanan berbasis internet sebagai infrastruktur utamanya [2]. Terdapat ragam penelitian sebelumnya baik berkaitan dengan perancangan sistem maupun kebermanfaatannya *E-Commerce* bagi pelaku UMKM. Penelitian berkaitan dengan dampak *E-Commerce* bagi pelaku UMKM dilakukan pada tahun 2022 di Desa Kerinjing, hasilnya menunjukkan bahwa penggunaan *E-Commerce* dan pemasaran digital memiliki dampak positif seperti peningkatan pendapatan, kemudahan dalam meningkatkan visibilitas produk, penghematan biaya promosi, serta peningkatan kecepatan transaksi [3].

Penelitian lainnya yang juga melakukan penelitian pada dampak *E-Commerce* bagi pelaku UMKM yang bertujuan untuk mengetahui pemanfaatan e-commerce untuk pengembangan usaha kecil yang memiliki toko di Shopee, juga menunjukkan hasil positif, di mana Shopee, sebagai platform *E-Commerce* berpengaruh dan

membantu mengembangkan usaha kecil. Sebagian besar pelaku usaha kecil telah terbantu dengan melakukan penjualan melalui *E-Commerce*, sebagian besar penjual setuju bahwa *E-Commerce* sangat dibutuhkan oleh pemilik usaha kecil. Namun, masih terdapat beberapa kekurangan pada sistem kerja Shopee yang membuat sebagian penjual tidak setuju jika e-commerce menjadi kebutuhan usaha kecil [4].

Penelitian lainnya berkaitan dengan perancangan sistem informasi desa dengan sistem *E-Commerce* dengan menerapkan *agile development* juga mendapat hasil positif dengan membantu masyarakat dalam memasarkan produknya secara online sehingga meningkatkan perekonomian di desa tersebut [5].

Penelitian ini bertujuan untuk mengembangkan sebuah *platform E-Commerce* berbasis aplikasi Android dan *Website* guna meningkatkan visibilitas, memperluas pasar produk kerajinan bambu dari Desa Belega, serta memudahkan manajemen transaksi dan pencatatan pesanan bagi pengrajin bambu. *Platform* ini dirancang khusus untuk menjual produk kerajinan bambu terutama dari Desa Belega, hal ini dilakukan dengan harapan dapat membangun identitas merek yang kuat serta memberikan perbedaan dari *platform E-Commerce* umum lainnya. Secara teknis, pengembangan aplikasi akan menggunakan arsitektur *microservice* dengan antarmuka *REST API* yang memungkinkan pertukaran data antara server dan *client*. Aplikasi sepenuhnya akan dibuat dengan bahasa pemrograman *Javascript*. *Framework* yang akan digunakan untuk menyusun *logic* dari *backend* adalah *unopinionated framework* berbasis *Javascript* bernama *Express.js*. *Client side* yang berbasis Android ini selanjutnya akan dirancang menggunakan *framework React Native*.

Penelitian ini akan berfokus kepada bisnis proses yang diterapkan pada sistem yang dikembangkan. Meskipun *E-Commerce* terdapat dua entitas yang akan berinteraksi, yakni *seller* dan *buyer*, fokus dari bisnis proses pada sistem berbasis *mobile* ini akan lebih berada pada sisi *buyer*, dikarenakan *seller* lebih diutamakan pada aplikasi berbasis *website*.

Penelitian ini diharapkan dapat memberikan kontribusi terutama dalam proses pengembangan model bisnis yang berkelanjutan bagi industri kerajinan bambu di Indonesia. Selain itu, hasil penelitian ini juga dapat dijadikan referensi bagi daerah lain yang memiliki potensi serupa dalam mengembangkan produk lokalnya melalui *platform E-Commerce*.

## 2. Metode Penelitian

Metode pengembangan sistem yang selanjutnya akan digunakan pada penelitian ini ialah menggunakan *Scrum Methodology* atau biasa dikenal sebagai *Scrum*. *Scrum* merupakan sebuah metode pengembangan sistem yang secara harfiah dilakukan iteratif dan metode ini juga termasuk ke dalam metode *Agile*

*Software Development* tentang cara mengatur manajemen dan pelaksanaan proyek.

*Scrum* juga merupakan metode yang universal dan dapat diterapkan pada beragam proyek seperti halnya perancangan perangkat lunak yang mencakup web maupun *mobile*, perancangan *hardware*, *marketing*, dan bahkan *event planning*. Dalam prosesi implementasi metode *scrum* pada sebuah proyek, biasanya akan terdapat pembagian unit proses yang akan saling mendukung satu dengan lainnya, seperti halnya *Product Backlog*, *Sprint Planning*, *Daily Scrum*, *Sprint Preview*, serta yang terakhir ialah *Sprint Retrospective* [6].

### 2.1. Product Backlog

Fase ini merupakan sebuah fase melakukan pengumpulan prioritas yang berisi pekerjaan yang ada dalam proyek. Pokok yang ada di dalam *product backlog* itu sendiri dapat berupa fitur, perbaikan, ataupun perubahan lainnya yang diinginkan pada produk akhir. Fase ini biasanya dikelola langsung oleh *product owner* serta diperbaharui secara teratur.

### 2.2. Sprint Planning

Fase *Sprint Planning* merupakan pertemuan yang dilakukan oleh tim untuk melakukan perencanaan berkaitan dengan sprint berikutnya. *Sprint Planning* biasanya terbagi menjadi dua bagian, yaitu prosesi untuk memilih pokok dari *product backlog* yang akan diselesaikan selama *sprint* berlangsung, dan bagian perencanaan bagaimana pekerjaan tersebut diselesaikan.

#### a. Sprint Backlog

*Sprint Backlog* sendiri merupakan bagian dari *Sprint Planning*, di mana prosesi *Sprint Backlog* adalah proses ketika memilih *Product Backlog* yang akan dikerjakan pada proses *Sprint* yang akan dilakukan kedepannya. Proses ini bertujuan untuk membantu tim tetap fokus pada prioritas utama sprint dan memastikan semua tugas diatur dengan jelas sehingga sprint dapat selesai dengan hasil yang terukur dan berkualitas.

### 2.3 Sprint

Fase *Sprint* merupakan periode waktu terbatas yang secara umum, periode ini dipergunakan untuk melakukan pekerjaan. Pekerjaan yang dilakukan juga harus berdasarkan pokok-pokok pekerjaan yang dipilih oleh tiap anggota tim ketika prosesi *product backlog* berlangsung.

### 2.4. Daily Scrum

Fase *Daily Scrum* merupakan proses untuk *tracking* setiap hari berkaitan *progress* dari pokok pekerjaan yang dipilih ketika prosesi *product backlog* berlangsung sebelumnya. Hal tersebut bertujuan agar proses pembuatan produk dapat selesai sesuai dengan *deadline* yang telah ditentukan. Waktu dari *daily scrum* ini umumnya cukup singkat, berkisar antara 10 hingga 15 menit.

### 2.5. Sprint Review

Fase *Sprint Review* merupakan pertemuan yang diadakan ketika periode *sprint* telah usai. *Sprint Review* juga digunakan untuk melakukan presentasi hasil kerja terkait dengan pokok pekerjaan yang dipilih pada *product backlog*. Presentasi ini biasanya langsung disaksikan oleh pemangku kepentingan atau *stakeholders*.

### 2.6. Sprint Retrospective

Fase *Sprint Retrospective* merupakan fase yang digunakan untuk melakukan pertemuan reflektif setelah *sprint review* usai, di mana tim akan melakukan evaluasi pada proses pengerjaan pokok pekerjaan mereka selama *sprint*.

### 2.7. Pengembangan Sistem dengan Agile Development

Pada saat prosesi pengembangan system dan aplikasi, terdapat serangkaian prosesi untuk pembentukan tim serta pembagian tugas di dalam tim terkait. Selain pembagian tugas, terdapat juga prosesi untuk menentukan jumlah *sprint* yang dilakukan.

#### a. Komposisi Tim Scrum

Pada tahap awal implementasi metode *agile development* dalam pengembangan, tim membutuhkan *Product Owner*, *Developers*, dan *Scrum Masters* untuk melakukan proses *development* dan mengembangkan lingkungan *Scrum* agar berjalan sesuai rencana dan terlaksana dengan baik. Oleh sebab itu, tim dibentuk dengan berisikan *Products Owner* yang akan melaksanakan tugas untuk memastikan produk terimplementasi sesuai dengan target. Kemudian, terdapat *Developers* sebagai anggota *Scrum* yang akan menjadi eksekutor pengembangan aplikasi dan sistem. Terakhir, terdapat *Scrum Masters* yang akan memfasilitasi anggota untuk mulai mempelajari *Scrum*, membantu memahami *Scrum*, dan menghilangkan hambatan dalam *Scrum*.

Implementasi *Scrum* memang pada awalnya akan sangat membutuhkan *Scrum Masters* sebagai orang yang siap membantu setiap anggota dalam mengimplementasikan *Scrum* dan juga dapat memahami teori maupun praktik sehingga *Scrum* dapat berjalan secara optimal. Tugas *Scrum Masters* akan berakhir ketika *Scrum Team* dapat berjalan secara mandiri.

#### b. Tahapan Scrum

Tahapan implementasi *Scrum* terdiri dari *Product Backlog*, *Sprint Planning*, *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective* seperti yang sudah dijabarkan sebelumnya. Prosesi awal sendiri adalah melakukan tahap untuk mengatur pekerjaan yang dilakukan dalam *Sprint*. *Products Owner* bersama *Scrum Team* akan membahas *Products Backlog Item* (PBI) yang diprioritaskan untuk menjadi *Product Goal*.

Selanjutnya, *Scrum Team* akan menentukan *Sprint Goal* berdasarkan nilai urgensi yang paling penting berdasarkan pemberian *point* yang telah didiskusikan sebelumnya. Maka *Scrum Team* memilih PBI yang sesuai dengan *Sprint Goal*-nya. *Sprint Goal* sendiri akan dipermudah dengan adanya *Definition of Done* (DOD) yang memberikan tolak ukur kapan *Sprint Goal* dianggap selesai.

*Developer* akan menentukan dari proses atau pekerjaan teknis yang harus dipenuhi untuk menyelesaikan PBI. Sehingga, setiap hasil yang didapat dari penyelesaian tugas akan menjadi *Increment*. Hal ini sendiri biasanya dimulai dari proses *Sprint Planning*, PBI yang ada pada *Sprint*, beserta rencana pekerjaan akan disebut *Sprint Backlog*. Segala yang terdapat pada *Sprint Backlog* biasanya akan memiliki tenggat penyelesaian paling lambat 8 jam untuk pekerjaan yang memang membutuhkan *deadline* selama sebulan.

Kemudian, juga terdapat prosesi pemeriksaan *Sprint Goal* dan kesesuaian dengan *Sprint Backlog* yang menjadi tujuan pada saat melakukan *Daily Scrum*. *Sprint Backlog* merupakan salah satu elemen *scrum* yang merupakan daftar tugas spesifik yang dipilih tim dari *Product Backlog* untuk diselesaikan selama *Sprint* tertentu dengan tujuan untuk membantu tim tetap focus pada prioritas utama *Sprint*. Selanjutnya, terdapat *Daily Scrum* yang biasanya dilakukan selama 15 menit dengan fokus utama pada kemajuan untuk mencapai *Sprint Goal* atau tujuan utama dari *Sprint* yang berjalan.

Selanjutnya terdapat *Sprint Review* yang akan menjadi proses untuk memeriksa hasil yang telah didapatkan selama mengerjakan *Sprint Backlog* pada fase *Sprint* yang telah usai. Proses ini mempresentasikan hasil kerja pada pemangku kebijakan terkait ketercapaian untuk menuju ke *Product Goal*. Terakhir, terdapat prosesi *Sprint Retrospective* yang bertujuan untuk meningkatkan efektivitas dan kualitas *Scrum Team*. *Scrum Team* akan melakukan kajian berkaitan dengan DoD, dan lain sebagainya termasuk interaksi antar individu. Apabila terdapat perbaikan, maka akan langsung diimplementasikan pada *Sprint Backlog* selanjutnya.

Pada pengembangan aplikasi, total *sprint* dilakukan adalah sebanyak dua kali. *Sprint* pertama, digunakan untuk melakukan prosesi pengembangan *REST API* serta melakukan integrasi baik ke dalam aplikasi *web* maupun ke aplikasi, selanjutnya pada *Sprint* yang kedua, *Developers* melakukan proses optimasi pada integrasi yang telah dilakukan.

## 2.8. Tools dan Teknologi

Beragam *tools* dan teknologi yang digunakan untuk mengembangkan aplikasi ini diantaranya adalah *framework React Native* dan *Express.js*. Alasan dari penggunaan kedua *framework Javascript* ini adalah agar proses *development* baik untuk *backend* maupun *frontend* dapat berjalan dengan lebih cepat dikarenakan bahasa pemrograman yang digunakan sama, sehingga

tidak membutuhkan waktu adaptasi kembali dan mampu berjalan secara beriringan. Berikut ini adalah beberapa *tools* dan teknologi yang digunakan beserta versinya secara lengkap.

### a. REST API

*REST API* sendiri merupakan sebuah arsitektur untuk berkomunikasi antar aplikasi melalui internet. *REST API* sendiri cukup populer digunakan, terutama saat ini dengan menggunakan bahasa pemrograman *Javascript* yang dibantu oleh *runtime environment Node JS* [7]. *Node JS* yang digunakan saat pengembangan menggunakan versi 18.18.2. *Node JS* sendiri digunakan agar bahasa pemrograman *Javascript* dapat digunakan sebagai *backend* yang menjadi pusat logika dari program yang dibangun. Selain itu, arsitektur *REST API* digunakan karena memang memiliki komunitas yang luas serta lebih mudah diimplementasikan dibandingkan arsitektur yang lainnya.

### b. Javascript

*Javascript* sendiri merupakan bahasa pemrograman yang sering kali digunakan dalam pengembangan aplikasi berbasis *web* baik untuk *backend* maupun *frontend* dikarenakan memiliki komunitas yang luas [8]. Selain memiliki komunitas yang luas, penggunaan *Javascript* bertujuan untuk mempermudah proses pengembangan sistem, dikarenakan menawarkan *development experience* yang cukup baik karena lebih ringan dibandingkan bahasa pemrograman yang lain untuk proses pengembangan aplikasi *mobile*.

### c. Express.js

*Express.js* merupakan *framework unopinionated* yang digunakan untuk mengembangkan *backend* menggunakan bahasa pemrograman *Javascript*. *Express.js* sendiri sangat populer dikalangan para *backend developer* dikarenakan fleksibilitas yang ditawarkan [9]. *Express.js* yang digunakan pada saat pengembangan menggunakan versi 4.18.2. *Framework* ini sendiri digunakan dikarenakan fleksibilitasnya serta jauh lebih ringan dibandingkan dengan *framework backend* lain seperti *Django* dan *Laravel*.

### d. React Native

*React Native* sendiri merupakan *framework* untuk *hybrid mobile development* yang dikembangkan menggunakan bahasa pemrograman *Javascript* dengan menggunakan *library React* [10]. *React Native* yang digunakan pada proses *development* aplikasi menggunakan versi 0.71.14, sedangkan *React* yang digunakan adalah versi 18.2.0. *React Native* sendiri digunakan dikarenakan alasan subjektif dari *developer* dikarenakan menawarkan pengalaman pengembangan yang jauh lebih mudah dan ringan dibandingkan menggunakan *Kotlin* maupun *Java* untuk pengembangan aplikasi *mobile* berbasis android.

f. Postman

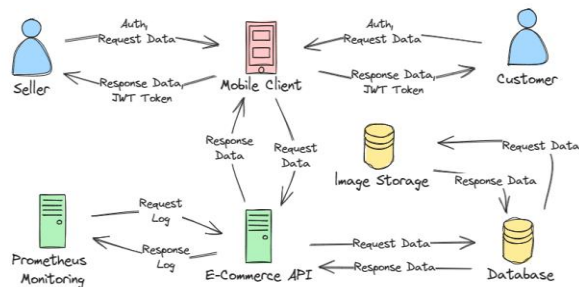
Postman sendiri merupakan aplikasi yang digunakan untuk melakukan pengujian pada *backend* serta dapat digunakan untuk melakukan *load testing* pada *endpoint* yang diinginkan. Postman yang digunakan sendiri merupakan versi 9.4 dan *tools* ini dipilih dikarenakan cukup fleksibel dan memiliki *user experience* yang lebih baik dibandingkan aplikasi sejenis lainnya.

### 3. Hasil dan Pembahasan

Deskripsi umum dari rancangan sistem yang akan dibangun menggunakan arsitektur *REST API* dengan menggunakan metode pengembangan *Scrum Methodology* akan di jabarkan pada Gambaran Umum Sistem. Pada aplikasi berbasis *mobile*, *seller* hanya dapat melakukan konfirmasi pemesanan atas order yang dilakukan oleh *buyer*.

#### 3.1. Gambaran Umum Sistem

Aplikasi *E-Commerce* berbasis Android untuk Desa Belega memiliki gambaran umum sistem yang bertujuan untuk pemodelan secara umum dari sistem yang dibuat serta mempermudah pemberian informasi tentang bagaimana sistem akan dirancang. Gambaran umum sistem pada penelitian ini akan dijabarkan pada gambar berikut ini.



Gambar 1. Gambaran Umum Sistem Arsitektur Microservice

Gambar 1 di atas merupakan gambaran umum sistem dari Rancang Bangun Aplikasi *E-Commerce* Kerajinan Bambu Berbasis Android untuk Desa Belega. Pada gambaran umum sistem penelitian ini berfokus terhadap dua entitas yang saling terlibat secara intens, yaitu *seller* dan *buyer (customer)*.

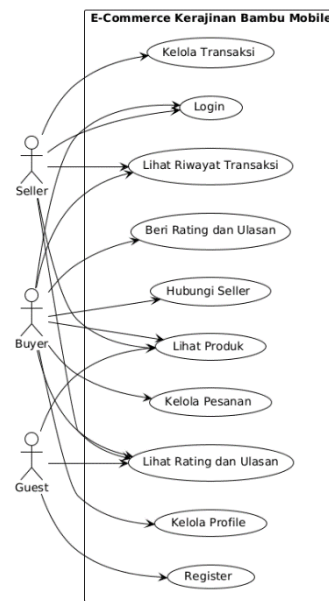
Proses akan dimulai dari pembuatan database dengan PostgreSQL serta terdapat prosesi linking ke *Image Storage*. Kemudian akan dilakukan pembuatan API menggunakan *unopinionated framework* Express.js. Terakhir, *end user* utama pada aplikasi mobile yaitu *buyer (customer)* dan *seller* mengakses data login sesuai dengan role yang diberikan pada RBAC yang ada pada *server-side* atau biasa disebut *backend*.

Berdasarkan dari diagram yang dipaparkan pada Gambar 1, arsitektur *microservice* yang digunakan pada proses Rancang Bangun Aplikasi *E-Commerce* Kerajinan Bambu Berbasis Android untuk Desa Belega ini, dapat terlihat jelas bahwa arsitektur ini mendekomposisi aplikasi *E-Commerce* menjadi

layanan-layanan kecil, seperti halnya memisahkan antara *Database* utama dengan dengan *Image Storage*, serta prosesi monitoring dengan *Prometheus*, yang mana masing-masing layanan memiliki tanggung jawab spesifik dan dapat dikembangkan, di-*deploy*, dan di-*scale* secara mandiri.

#### 3.2. Use Case Diagram

*Use Case Diagram* Aplikasi *E-Commerce* berbasis Android untuk Desa Belega di bawah ini memiliki beberapa fitur yang dapat digunakan oleh aktor. Aktor terkait dapat mengakses masing-masing fitur yang diberikan sesuai dengan role yang dimiliki seperti yang digambarkan pada Gambar 2 di bawah ini.



Gambar 2. Usecase Diagram Sistem

Pada proses berjalannya aplikasi, pengguna aplikasi dibagi menjadi beberapa aktor, diantaranya adalah sebagai *Seller*, *Buyer*, dan *Guest* yang memiliki akses fitur yang berbeda-beda. *Seller* sendiri, memiliki fitur untuk melakukan pengelolaan transaksi, melihat riwayat transaksi yang terjadi, melihat produk, dan dapat melihat *rating* dan ulasan. Kemudian *buyer* memiliki fitur yang sedikit lebih kompleks, seperti halnya melakukan pengelolaan pesanan (ini termasuk dengan pembayaran maupun menolak pesanan yang telah dikonfirmasi oleh *seller*), pengelolaan *profile*, melihat produk, menghubungi *seller*, memberikan *rating* dan ulasan, melihat riwayat transaksi, serta melihat *rating* dan ulasan. Terakhir, *Guest* merupakan *user* anonim yang belum memiliki akun. *Guest* sendiri hanya dapat melihat produk, melihat *rating* dan ulasan, serta *register*.

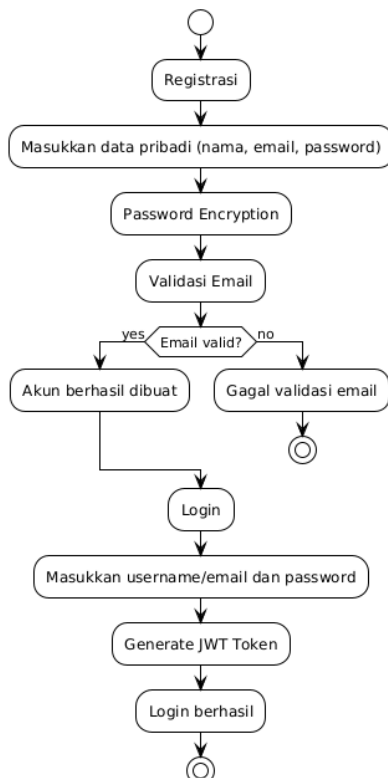
#### 3.3. Bisnis Proses

Secara harfiah, bisnis proses merupakan serangkaian aktivitas maupun tugas yang saling terkait dan dilakukan penyusunan secara sistematis untuk memberikan pencapaian pada tujuan yang dicanangkan pada sebuah bisnis.

Alur dari bisnis proses yang ada di Desa Belega sudah terimplementasi melalui aplikasi yang dikembangkan pada penelitian ini. Aplikasi ini juga sudah dapat menjembatani interaksi yang diperlukan antara *buyer* dan *seller* untuk melakukan transaksi jual beli.

#### a. Bisnis Proses Manajemen Pengguna

Bisnis proses manajemen pengguna ini lebih berfokus kepada manajemen akun dari *buyer*. Prosesi registrasi akan memberikan *role* sebagai *buyer* secara langsung bagi pengunjung atau *guest* dari aplikasi. Ilustrasi lengkap dari alur bisnis proses manajemen pengguna akan ditampilkan pada Gambar 3.



Gambar 3. Flowchart Bisnis Proses Manajemen Pengguna

Proses utama dalam melakukan manajemen pengguna adalah registrasi akun pengunjung agar dapat memiliki akses ke dalam aplikasi *E-Commerce*. Berikut ini merupakan penjelasan mendetail berkaitan dengan bisnis proses manajemen pengguna.

Pengunjung aplikasi melakukan akses ke dalam aplikasi *E-Commerce*. Setelah berada di dalam aplikasi, pengunjung dapat melakukan registrasi akun dengan mengisi data diri yang diperlukan. Setelah selesai melakukan registrasi dengan mengisi data diri, serta *password* akun, pengunjung dapat melakukan verifikasi *email* menggunakan kode OTP yang akan dikirimkan ke *email* yang digunakan pada saat registrasi sebelumnya.

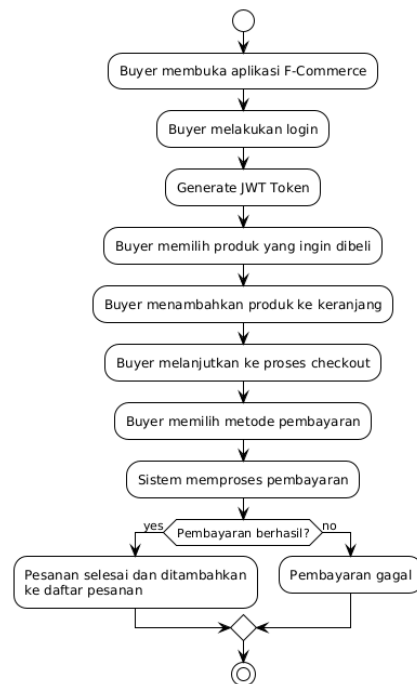
*Password* dari pengunjung ketika melakukan registrasi akan dilakukan proses enkripsi. Proses enkripsi sendiri dilakukan dengan menggunakan algoritma *hash256* untuk mengamankan akun yang terdaftar di dalam sistem.

Seluruh fitur yang terdapat di dalam aplikasi *E-Commerce* ini hanya dapat di akses secara menyeluruh apabila pengunjung telah diverifikasi. Kemudian, setelah akun yang dilakukan pendaftaran sebelumnya telah terverifikasi, pengunjung mendapat status baru sebagai seorang *buyer*.

*Buyer* wajib *login* terlebih dahulu menggunakan akun yang dimiliki sebelumnya dan wajib telah terverifikasi. Hal ini cukup penting, mengingat aplikasi *E-Commerce* menggunakan keamanan berlapis dengan mengandalkan token *JWT* yang hanya bisa didapatkan oleh *buyer* apabila *login* menggunakan akun yang telah diverifikasi.

#### b. Bisnis Proses Order Produk

Bisnis proses yang wajib ada di dalam sistem *E-Commerce* tentunya ialah proses melakukan order produk, yang menjadi cikal bakal proses transaksi akan terjadi. Alur dari bisnis proses dari order produk akan diperlihatkan oleh Gambar 4.



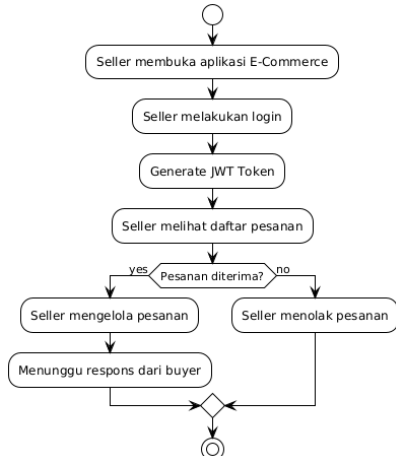
Gambar 4. Flowchart Bisnis Proses Order Produk

Prosesi order produk di dalam aplikasi dapat dilakukan melalui *cart* maupun secara langsung pada produk untuk melakukan *checkout* produk yang diinginkan oleh *buyer*.

Proses ketika melakukan *checkout*, *buyer* dapat memilih layanan jasa untuk pengantaran barang melalui aplikasi. Jasa ekspedisi menggunakan *third party* API bernama Rajaongkir API, yang menyediakan tiga pilihan jasa ekspedisi, yakni JNE, POS, dan TIKI. Harga dari ongkos kirim yang dibebankan kepada *buyer* bergantung dari alamat yang diberikan oleh *buyer* ketika melakukan *checkout* produk pada aplikasi *E-Commerce* serta berat dari produk terkait.

c. Bisnis Proses Konfirmasi Pesanan

Bisnis proses konfirmasi pesanan, merupakan salah satu bisnis proses yang aksesnya dimiliki oleh *seller* di aplikasi *E-Commerce* Android ini, selain bisnis proses menolak pesanan. Ilustrasi terkait alur dari bisnis proses konfirmasi pesanan ini akan ditampilkan secara lebih rinci melalui Gambar 5.



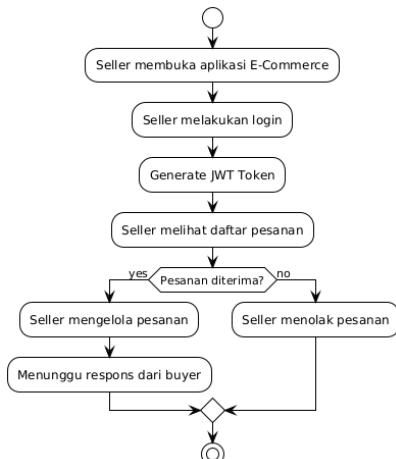
Gambar 5. Flowchart Bisnis Proses Konfirmasi Pesanan

Alur lengkap dari bisnis proses konfirmasi pesanan dimulai ketika *seller* melakukan *login* melalui aplikasi *mobile*. Hal ini dapat dilakukan apabila akun *seller* sudah terverifikasi sebelumnya melalui aplikasi *website*.

*Seller* dapat melakukan konfirmasi pesanan terhadap order yang dilakukan oleh *buyer* sebelumnya pada produk yang dimiliki oleh *seller*. Dengan demikian, *buyer* dapat melakukan pembayaran atas produk yang telah di *checkout* sebelumnya melalui modul pembayaran.

d. Bisnis Proses Menolak Pesanan

Bisnis proses menolak pesanan ini, adalah salah satu bisnis proses yang bisa dilakukan *seller* selain melakukan konfirmasi pesanan di aplikasi *E-Commerce* Android ini. Ilustrasi terkait alur dari bisnis proses konfirmasi pesanan ini akan ditampilkan secara lebih rinci melalui Gambar 6.



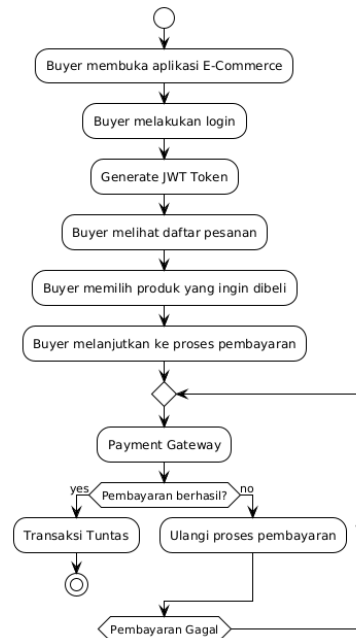
Gambar 6. Flowchart Bisnis Proses Menolak Pesanan

Alur lengkap dari bisnis proses konfirmasi pesanan dimulai ketika *seller* melakukan *login* melalui aplikasi *mobile*. Hal ini dapat dilakukan apabila akun *seller* sudah terverifikasi sebelumnya melalui aplikasi *website*.

Pada bisnis proses ini, *seller* dapat menolak pesanan yang masuk dari *buyer*. Dengan demikian, status pesanan dari *buyer* akan dilakukan proses *update* menjadi ditolak, sehingga *buyer* tidak bisa melakukan pembayaran.

e. Bisnis Proses Pembayaran Pesanan

Bisnis proses pembayaran pesanan ini dapat digunakan oleh *buyer* ketika pesanan yang diterima oleh *seller* telah di konfirmasi. Setelah pesanan dilakukan proses *konfirmasi*, *buyer* dapat melakukan pembayaran pesanan. Ilustrasi lengkap berkaitan dengan bisnis proses pembayaran pesanan dapat dilihat melalui Gambar 7.



Gambar 7. Flowchart Bisnis Proses Pembayaran Pesanan

Pada bisnis proses ini, *buyer* dapat melakukan pembayaran terhadap pesanan yang telah dikonfirmasi *seller*. Kemudian, *buyer* melakukan pembayaran dengan menggunakan *payment gateway* yang telah disediakan aplikasi *E-Commerce*, yakni dengan *payment gateway* Midtrans.

Midtrans sendiri, mengakomodir banyak sekali metode pembayaran, seperti halnya *E-Wallet*, Debit, *Over the Counter*, serta melalui VA dari Bank seperti Bank BCA, BNI, BRIVA, ATM Bersama, dan lain-lain. Setelah melakukan pembayaran, transaksi *buyer* akan diubah statusnya menjadi sukses.

Dengan demikian, proses transaksi jual beli antar entitas *buyer* dan *seller* telah usai apabila *buyer* telah menuntaskan kewajibannya dengan melakukan pembayaran melalui *payment gateway*.

f. Bisnis Proses Pembatalan Pesanan

Bisnis proses pembatalan pesanan, merupakan salah satu bisnis proses yang bisa dilakukan oleh *buyer* untuk membatalkan pesanan yang telah dikonfirmasi oleh *seller*. Ilustrasi mendetail dari bisnis proses ini akan ditampilkan pada Gambar 8.

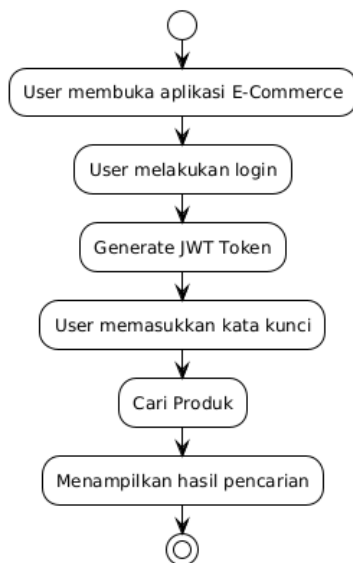


Gambar 8. Flowchart Bisnis Proses Pembatalan Pesanan

Pada bisnis proses pembatalan pesanan ini, *buyer* harus memasuki laman *list order*, kemudian menekan tombol batal yang disediakan. Setelah itu, akan muncul *pop up* yang mewajibkan *buyer* untuk memasukkan alasan pembatalan. Terakhir, *buyer* akan menerima *email* bahwa pesanan telah berhasil dibatalkan.

g. Bisnis Proses Pencarian Produk

Bisnis proses pencarian produk ini dapat digunakan oleh *buyer* maupun *seller*, apabila ingin mencari sebuah produk. Ilustrasi lengkap berkaitan dengan bisnis proses untuk melakukan pencarian produk dapat dilihat melalui Gambar 9.

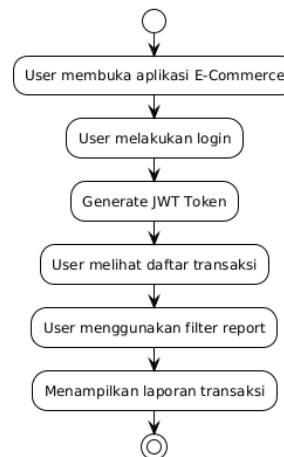


Gambar 9. Flowchart Bisnis Proses Pencarian Produk

Pada bisnis proses pencarian produk, *buyer* maupun *seller* dapat langsung beralih ke laman pencarian untuk mencari produk. Nantinya, produk akan langsung ditampilkan di dalam aplikasi sesuai dengan keyword yang dicari oleh pengguna, baik *buyer* maupun *seller*.

h. Bisnis Proses Report Transaksi

Bisnis proses *report* transaksi ini dapat digunakan oleh *buyer* untuk melakukan rekapitulasi berkaitan dengan daftar transaksi yang dilakukan. Ilustrasi berkaitan dengan bisnis proses ini akan ditampilkan pada Gambar 10.

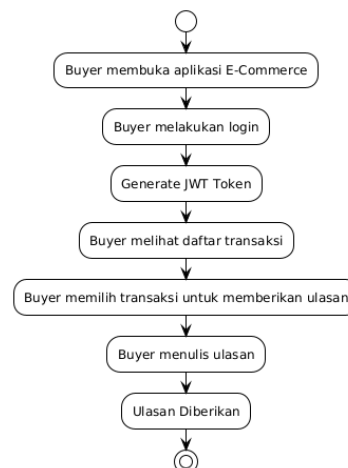


Gambar 10. Flowchart Bisnis Proses Report Transaksi

Pada bisnis proses *report* transaksi ini, *seller* maupun *buyer* dapat melakukan proses *filtering* terlebih dahulu dengan menggunakan fitur *filter* yang disediakan. Fitur *filter* menyediakan ragam proses sortir data untuk ditampilkan sesuai dengan kebutuhan, seperti halnya *sort by date* maupun *sort by price*.

i. Bisnis Proses Ulasan Produk

Bisnis proses ulasan produk ini akan digunakan oleh *buyer* ketika transaksi pembelian produk telah dilakukan. Ilustrasi berkaitan dengan bisnis proses untuk mengulas produk dapat dilihat melalui Gambar 11.



Gambar 11. Flowchart Bisnis Proses Ulasan Produk

Pada bisnis proses ini, *buyer* dapat memberikan ulasan terhadap produk yang dibeli sebelumnya. *Buyer* juga wajib menyertakan bukti berupa gambar ketika memberikan ulasan kepada produk yang baru dibeli.

### 3.4. Antarmuka Aplikasi

Antarmuka aplikasi merupakan tampilan yang akan disuguhkan kepada pengguna ketika menggunakan aplikasi yang telah dirancang. Antarmuka aplikasi yang ditampilkan merupakan beberapa bagian yang krusial dan penting untuk menjalankan proses bisnis yang ada di dalam aplikasi.

#### a. Antarmuka Login

Antarmuka *login* pada Aplikasi E-Commerce Kerajinan Bambu berbasis Android untuk Desa Belega merupakan tampilan dari antarmuka *login* yang nantinya dapat diakses oleh *guest* untuk melakukan masuk ke dalam akun miliknya. Berikut merupakan antarmuka *login* pada aplikasi yang akan ditampilkan pada Gambar 12.

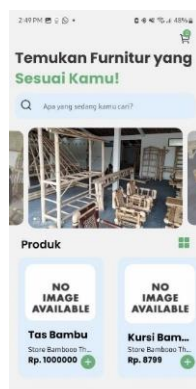


Gambar 12. Tampilan Antarmuka Login

Pada antarmuka *login* yang ditampilkan, pengguna dapat memberikan *input* berupa *email* dan *password* yang dimiliki.

#### b. Antarmuka Landing Page

Antarmuka *Landing Page* sendiri merupakan antarmuka yang akan disuguhkan kepada *guest*, *buyer*, maupun *seller* ketika telah berhasil *login* ke dalam aplikasi. Antarmuka ini akan ditampilkan pada Gambar 13.

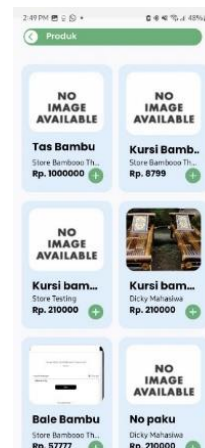


Gambar 13. Antarmuka Landing Page

Pada antarmuka ini ditampilkan beberapa komponen yang ada seperti daftar produk dan *carousel* yang berisikan gambar pembuatan kerajinan bambu di desa. Antarmuka ini juga memiliki beberapa bagian lain yang menampilkan keseluruhan produk

#### c. Antarmuka List Produk

Antarmuka *List* Produk sendiri merupakan antarmuka yang menampilkan daftar produk yang ada di dalam aplikasi. Antarmuka ini akan ditampilkan pada Gambar 14.

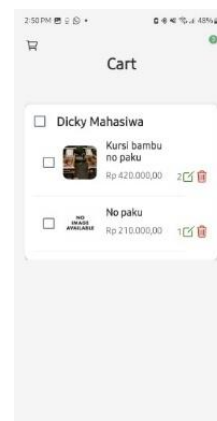


Gambar 14. Tampilan Antarmuka List Produk

Pada antarmuka *list* produk yang ditampilkan, pengguna dapat melihat secara langsung seluruh produk yang tersedia dari beragam toko. Pengguna dapat langsung membuka detail produk untuk mengetahui lebih lanjut terkait ulasan dan *rating* produk.

#### d. Antarmuka Cart

Antarmuka ini memiliki peran penting untuk melakukan *checkout* barang yang diinginkan oleh *buyer*. Antarmuka *Cart* akan ditampilkan melalui Gambar 15 di bawah ini.

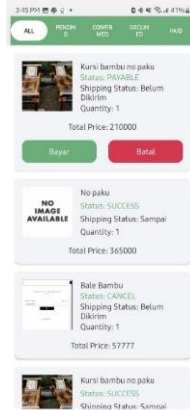


Gambar 15. Antarmuka Cart

Antarmuka *Cart* ini digunakan untuk menampung sementara barang yang ingin dibeli oleh *buyer*. *Buyer* dapat melakukan *checkout* beberapa barang sekaligus baik dari toko yang sama maupun toko yang berbeda secara sekaligus.

e. Antarmuka Order

Antarmuka Order ini digunakan untuk menampilkan order yang dilakukan oleh *buyer*. Antarmuka ini memiliki beberapa filter seperti untuk menampilkan order *pending* dan sebagainya. Antarmuka order akan ditampilkan pada Gambar 16 di bawah ini.



Gambar 16. Antarmuka Order

Antarmuka Order ini memiliki kemiripan dengan antarmuka transaksi, hal ini wajar karena dibuat dengan komponen yang sama. Antarmuka ini nanti berfungsi untuk menampilkan daftar order dari *buyer* serta digunakan untuk melakukan pembayaran order dari *buyer*.

f. Antarmuka Detail Order

Antarmuka Detail Order ini digunakan untuk menampilkan detail dari order yang dimiliki oleh *buyer*. Antarmuka Detail Order akan ditampilkan pada Gambar 17 di bawah ini.



Gambar 17. Antarmuka Detail Order

Antarmuka Detail Order yang akan ditampilkan kepada *buyer*. Antarmuka ini nantinya berfungsi untuk menampilkan data mendetail terkait order yang dilakukan oleh *buyer*.

g. Antarmuka Bayar Order

Antarmuka Bayar Order ini digunakan oleh pengguna untuk melakukan pembayaran melalui *payment gateway* Midtrans. Antarmuka Bayar Order ini akan ditampilkan melalui Gambar 18.

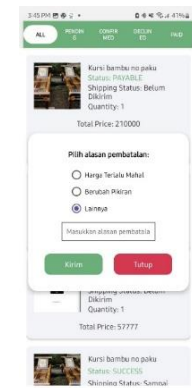


Gambar 18. Antarmuka Bayar Order

Antarmuka Bayar Order ini akan ditampilkan kepada *buyer* ketika melakukan prosesi pembayaran. Antarmuka ini sendiri merupakan *webview* yang ditampilkan *payment gateway* Midtrans.

h. Antarmuka Pembatalan Order

Antarmuka Pembatalan Order ini nantinya digunakan pengguna untuk melakukan pembatalan order pada transaksi yang diinginkan. Antarmuka Pembatalan Order ini akan ditampilkan melalui Gambar 19.



Gambar 19. Antarmuka Pembatalan Order

Antarmuka ini akan ditampilkan ketika *buyer* menekan batal pada daftar order. *Buyer* juga dapat mengisikan alasan secara langsung untuk menyampaikan perihal alasan pembatalan.

3.5. Load Testing

Pengujian *Load Testing* sendiri merupakan bagian penting dalam prosesi pengujian perangkat lunak yang bertujuan untuk memberikan kepastian terkait performansi dan keandalan sebuah sistem atau sebuah aplikasi dalam menghadapi *workload* atau beban kerja hingga ke skala tertentu. Tujuan dari adanya *load testing* sendiri adalah untuk memberikan kepastian, sejauh mana sistem maupun aplikasi dalam menangani beban yang tinggi sebelum mengalami galat atau kegagalan dan penurunan kinerja yang signifikan.

Prosesi *load testing* ini akan menggunakan *tools Postman* dengan parameter yang diukur berupa total waktu, rata-rata *requests* tiap detik, rata-rata waktu *response* dari *endpoint* terhadap *requests* yang masuk,

serta persentase *error* yang terjadi. Berikut ini merupakan pengujian *load testing* beberapa *endpoint* yang digunakan pada aplikasi berbasis *mobile*.

#### a. Endpoint Lihat Semua Produk

*Load Testing* yang dilakukan untuk mengetahui sejauh mana ketahanan API pada *workload* tinggi dan intens dengan mencoba salah satu *endpoint* untuk menampilkan seluruh produk. Pengujian akan dilakukan dengan skenario 100 *request* secara bersamaan dan akan naik secara eksponensial hingga ke 1000 *requests*. Gambar 20 merupakan hasil dari pengujian *load testing* yang dilakukan pada *endpoint* terkait.



Gambar 20. Hasil Load Testing pada Endpoint Lihat Semua Produk

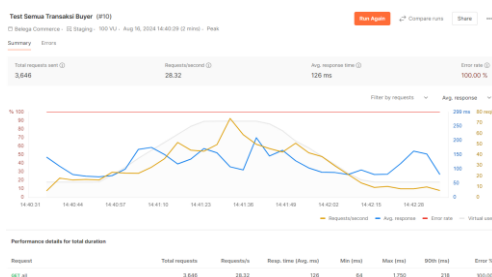
Skenario dari pengujian sendiri sudah dijalankan sesuai prosedur sehingga mendapatkan hasil dengan detail seperti berikut:

1. Total waktu yang dibutuhkan sebesar 5 menit.
2. Rata-rata *request* setiap detik 23,25 *request*/detik.
3. Rata – rata waktu respon 0,154 detik.
4. Presentase terjadinya *error* sebesar 0,00%.

Berdasarkan hasil pengujian *endpoint* terkait, dapat disimpulkan bahwa *endpoint* API dapat berjalan dengan cukup baik dengan tidak adanya galat atau *error* yang terjadi dan mampu menangani beban kerja berat tanpa mengalami penurunan performa yang signifikan.

#### b. Endpoint Lihat Semua Order Buyer

*Load Testing* yang dilakukan untuk mengetahui sejauh mana ketahanan API pada *workload* tinggi dan intens dengan mencoba salah satu *endpoint* untuk menampilkan data semua *order buyer*. Pengujian akan dilakukan dengan skenario 100 *requests* secara bersamaan dan akan naik secara eksponensial hingga ke 1000 *requests*. Gambar 21 merupakan hasil dari pengujian *load testing* yang dilakukan pada *endpoint* terkait.



Gambar 21. Hasil Load Testing pada Endpoint Lihat Semua Order Buyer

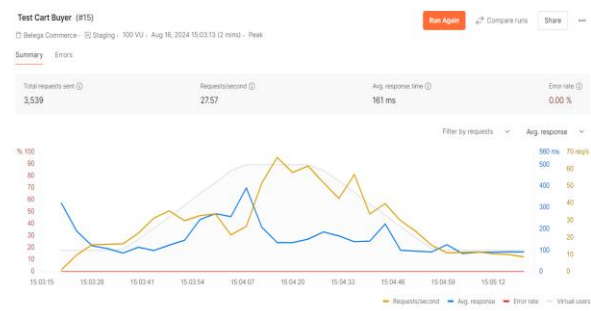
Skenario dari pengujian sendiri sudah dijalankan sesuai prosedur sehingga mendapatkan hasil dengan detail seperti berikut:

1. Total waktu yang dibutuhkan sebesar 2 menit.
2. Rata-rata *request* setiap detik 28,32 *request*/detik.
3. Rata – rata waktu respon 0,126 detik.
4. Presentase terjadinya *error* sebesar 0,00%.

Berdasarkan hasil pengujian *endpoint* terkait, dapat disimpulkan bahwa *endpoint* API dapat berjalan dengan cukup baik dengan tidak adanya galat atau *error* yang terjadi dan mampu menangani beban kerja berat tanpa mengalami penurunan performa yang signifikan.

#### c. Endpoint Lihat Cart Buyer

*Load Testing* yang dilakukan untuk mengetahui sejauh mana ketahanan API pada *workload* tinggi dan intens dengan mencoba salah satu *endpoint* untuk menampilkan data *cart* milik *buyer*. Pengujian akan dilakukan dengan skenario 100 *requests* secara bersamaan dan akan naik secara eksponensial hingga ke 1000 *requests*. Gambar 22 merupakan hasil dari pengujian *load testing* yang dilakukan pada *endpoint* terkait.



Gambar 22. Hasil Load Testing pada Endpoint Lihat Cart Buyer

Skenario dari pengujian sendiri sudah dijalankan sesuai prosedur sehingga mendapatkan hasil dengan detail seperti berikut:

1. Total waktu yang dibutuhkan sebesar 2 menit.
2. Rata-rata *request* setiap detik 27,57 *request*/detik.
3. Rata – rata waktu respon 0,161 detik.
4. Presentase terjadinya *error* sebesar 0,00%.

Berdasarkan hasil pengujian *endpoint* terkait, dapat disimpulkan bahwa *endpoint* API dapat berjalan dengan cukup baik dengan tidak adanya galat atau *error* yang terjadi dan mampu menangani beban kerja berat tanpa mengalami penurunan performa yang signifikan.

#### 3.6 Blackbox Testing

Dalam membangun sebuah aplikasi maupun sistem yang komprehensif, tentu diperlukan ragam pengujian yang mampu membuktikan bahwa sistem yang

dibangun sudah dapat memenuhi proses bisnis yang telah dicanangkan. Berikut adalah hasil dari *Blackbox Testing* dari Aplikasi *E-Commerce* Kerajinan Bambu Berbasis Android.

a. Modul Autentikasi

Berikut ini merupakan hasil dari *Blackbox Testing* yang telah dilakukan pada modul autentikasi aplikasi yang telah dilakukan. Hasil dari *Blackbox Testing* akan ditampilkan melalui Tabel 1 berikut ini.

Tabel 1. Blackbox Testing Modul Autentikasi

No	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
1	Mendaftar sebagai <i>buyer</i> dengan mengisi data lengkap	Sistem mengirimkan OTP ke email yang digunakan, mengarahkan pengguna untuk mengisi formulir OTP, dan memverifikasi pengguna sebagai <i>buyer</i> .	Valid
2	Mendaftar sebagai <i>buyer</i> dengan mengisi data yang kurang lengkap	Sistem mengirimkan notifikasi berupa <i>alert</i> kepada pengguna untuk melengkapi data yang kurang	Valid

b. Modul Pemesanan

Berikut ini merupakan hasil dari *Blackbox Testing* yang telah dilakukan pada modul pemesanan pada aplikasi yang meliputi bisnis proses order produk, konfirmasi pesanan, menolak pesanan, pembayaran pesanan, serta pembatalan pesanan. Hasil dari *Blackbox Testing* akan ditampilkan melalui Tabel 2 berikut ini.

Tabel 2. Blackbox Testing Modul Pemesanan

No	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
1	<i>Buyer login</i> , serta melakukan <i>checkout</i> pada item yang diinginkan.	Sistem memberikan <i>alert</i> bahwa <i>checkout</i> berhasil dan <i>buyer</i> harus menunggu seller untuk mengkonfirmasi pesannya.	Valid
2	Pengguna <i>login</i> dan pergi ke halaman <i>profile</i> untuk beralih ke status	Sistem mengirimkan notifikasi berupa <i>alert</i> kepada <i>seller</i> bahwa pesanan telah	Valid

sebagai *seller*, kemudian pergi ke halaman daftar

dikonfirmasi dan status berubah menjadi *payable*.

pesanan, lalu menekan tombol konfirmasi

3

*Buyer login*, pergi ke halaman *profile* serta membuka halaman order. *Buyer* dapat menekan data transaksi untuk melakukan kalkulasi ongkos kirim, lalu kembali ke halaman order dan melakukan pembayaran

Sistem mengarahkan pengguna ke *payment gateway Midtrans* dan pengguna melakukan pembayaran sesuai dengan metode yang dipilih, setelahnya status transaksi berubah menjadi sukses.

Valid

4

*Buyer login*, pergi ke halaman *profile* serta membuka halaman order serta menekan tombol batal.

Sistem mengirimkan *pop-up* alasan pembatalan serta *buyer* mengisi alasan pembatalan tersebut. Sistem mengirimkan email kepada *buyer* bahwa proses pembatalan berhasil dan status berubah menjadi *cancel*

Valid

5

Pengguna *login* dan pergi ke halaman *profile* untuk beralih ke status sebagai *seller*, kemudian pergi ke halaman daftar pesanan, lalu menekan tombol tolak

Sistem mengirimkan notifikasi berupa *alert* kepada *seller* bahwa pesanan telah dikonfirmasi dan status berubah menjadi *cancel*.

Valid

c. Modul Pencarian

Berikut ini merupakan hasil dari *Blackbox Testing* yang telah dilakukan pada modul pencarian pada aplikasi. Hasil dari *Blackbox Testing* akan ditampilkan melalui Tabel 3 berikut ini.

Tabel 3. Blackbox Testing Modul Pencarian

No	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
1	<i>Buyer</i> melakukan pencarian melalui halaman pencarian produk dengan mengetik nama produk maupun kata kunci yang berkaitan dengan produk	Sistem menampilkan item yang dicari berdasarkan kata kunci maupun nama produk secara utuh.	Valid

#### d. Modul List Transaksi

Berikut ini merupakan hasil dari *Blackbox Testing* yang telah dilakukan pada modul *list* transaksi pada aplikasi. Hasil dari *Blackbox Testing* akan ditampilkan melalui Tabel 4 berikut ini.

Tabel 4. Blackbox Testing Modul List Transaksi

No	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
1	<i>Buyer login</i> dan pergi ke halaman <i>profile</i> , kemudian pergi ke halaman order dan <i>buyer</i> dapat melihat semua transaksi, transaksi berhasil, transaksi gagal, dan transaksi pending	Sistem menampilkan semua transaksi dan menampilkan transaksi berdasarkan status transaksi milik <i>buyer</i> .	Valid
2	<i>Seller login</i> dan pergi ke halaman <i>profile</i> , kemudian mengubah status sebagai <i>seller</i> , dan <i>seller</i> dapat pergi ke halaman daftar pesanan, maka sistem akan menampilkan semua transaksi, transaksi berhasil, transaksi gagal, dan transaksi pending	Sistem menampilkan semua transaksi dan menampilkan transaksi berdasarkan status transaksi milik <i>seller</i> .	Valid

#### e. Modul Pemberian Ulasan

Berikut ini merupakan hasil dari *Blackbox Testing* yang telah dilakukan pada modul pencarian pada aplikasi. Hasil dari *Blackbox Testing* akan ditampilkan melalui Tabel 5 berikut ini.

Tabel 5. Blackbox Testing Modul Pemberian Ulasan

No	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
1	<i>Buyer</i> pergi ke halaman <i>profile</i> , kemudian mencari transaksi yang memiliki status sukses dan menekan tombol Beri Ulasan.	Sistem menampilkan <i>pop-up</i> untuk <i>buyer</i> mengisi ulasan berkaitan dengan produk yang telah dibeli sebelumnya.	Valid

#### 3.7 Tantangan Implementasi

Penelitian ini mengidentifikasi dua tantangan utama dalam implementasi *E-Commerce* di Desa Belega, yaitu kesenjangan digital antara pengrajin dan teknologi yang sudah dikembangkan, serta kendala finansial dari pemerintah desa akibat biaya pengembangan arsitektur *microservice* yang cukup tinggi dikarenakan menggunakan server dengan sistem *pay as you go*.

#### 4. Kesimpulan

Dalam penelitian ini, dapat disimpulkan bahwa dengan adanya layanan *E-Commerce*, terdapat kemungkinan terjadinya peningkatan pada visibilitas produk kerajinan dari Desa Belega dapat terjadi selaras dengan adanya digitalisasi berupa pengembangan aplikasi layanan *E-Commerce* berbasis Android ini dengan menawarkan sistem pengelolaan pesanan secara sistematis seperti dengan adanya pembayaran digital, pencatatan pesanan, dan pencatatan agen pengiriman serta aplikasi ini juga dapat dijadikan wadah untuk *branding* serta pemasaran secara digital melalui aplikasi berbasis android maupun melalui *website* berkaitan dengan hasil kerajinan bambu dari Desa Belega, sehingga nantinya diharapkan mampu untuk meningkatkan *visibilitas* kerajinan dan memperluas pangsa pasar dari kerajinan bambu dari Desa Belega. Selain itu, untuk membangun identitas pasar yang kuat, aplikasi ini juga hanya akan menjual kerajinan bambu saja. Dengan demikian, pengrajin bambu tidak perlu mengeluarkan *extra cost* untuk biaya promosi, karena sudah terdapat aplikasi yang mewadahi proses promosi dan *branding* yang langsung terafiliasi oleh pemerintah desa.

Prosesi penambahan fitur *payment gateway* serta perhitungan ongkos kirim ini untuk memaksimalkan potensi dari layanan *E-Commerce* dengan acuan dari

penelitian sebelumnya yang berfokus pada pengembangan aplikasi berbasis *website* dan tidak menggunakan *payment gateway* maupun skema penghitungan ongkos pengiriman. Kelemahan utama dari aplikasi yang dikembangkan ini ialah pada biaya operasional. Hal ini disebabkan oleh arsitektur *REST API* yang memang cukup memakan biaya dikarenakan perhitungan dari biaya ialah setiap *request* yang dilakukan oleh *user* aplikasi. Selain itu, harapan kedepannya adalah, terdapat pelatihan dan pendampingan dari *stakeholder* terkait, dalam hal ini adalah pemerintah Desa Belega untuk menyediakan pelatihan yang intensif dan berkelanjutan bagi pengrajin untuk meningkatkan literasi digital serta membangun kemitraan dengan berbagai pihak seperti pemerintah, LSM, dan perusahaan teknologi, untuk mendapatkan dukungan finansial maupun teknis.

Pengembangan lanjutan dalam aplikasi ini ialah menambahkan fitur tambahan untuk *seller*, seperti halnya *dashboard*, *payout*, serta fitur pengelolaan produk, sehingga aplikasi ini bisa menjadi *superapps* yang mana semua role dapat menggunakan fungsionalitas secara utuh.

#### Ucapan Terimakasih

Saya ingin mengucapkan terima kasih yang sebesar-besarnya kepada kedua dosen pembimbing saya yang telah memberikan bimbingan dan saran dalam proses penulisan jurnal ini. Karena dengan bantuan kedua dosen pembimbing saya, saya bisa menyelesaikan penulisan artikel ilmiah ini tanpa kendala yang berarti. Tidak lupa juga, saya mengucapkan terima kasih yang sebesar-besarnya kepada keluarga saya yang selalu memberikan dukungan dan semangat dalam proses penulisan jurnal ini. Dukungan kalian menjadi salah satu faktor penting dalam penulisan ini

#### SUMBER RUJUKAN

##### Referensi

- [1] L. Yana Siregar, M. Irwan Padli Nasution Prodi Manajemen, and U. Negeri Islam Sumatera Utara, "HIRARKI Jurnal Ilmiah Manajemen dan Bisnis DEVELOPMENT OF INFORMATION TECHNOLOGY ON INCREASING BUSINESS ONLINE," vol. 2, no. 1, pp. 71–75, 2020, doi: 10.30606/hjimb.
- [2] Alwendi, "Penerapan E-Commerce Dalam Meningkatkan," vol. 17, no. 3, 2020, [Online]. Available: <http://journal.undiknas.ac.id/index.php/magister-manajemen/>
- [3] B. Burhanudin, Y. Yusnaini, N. Khamisah, and S. F. Kartasari, "Utilization of E-Commerce and Digital Marketing to Increase Value Added Products for MSMEs and Home Businesses in Kerinjing Village," *Sricommerce: Journal of Sriwijaya Community Services*, vol. 3, no. 1, pp. 57–64, Mar. 2022, doi: 10.29259/jscs.v3i1.177.
- [4] E. Suhayati, S. Renali, S. V. Suherman, I. F. Matulesy, and H. Saputra, "Perkembangan e-commerce untuk usaha kecil," *Journal of Information System, Applied, Management, Accounting and Research*, vol. 5, no. 3, p. 626, Aug. 2021, doi: 10.52362/jisamar.v5i3.477.
- [5] L. Fitriani, P. Hakim, and R. M. Al Haq, "E-Commerce For Village Information System Using Agile Methodology," *Jurnal Online Informatika*, vol. 7, no. 1, p. 89, Jun. 2022, doi: 10.15575/join.v7i1.825.
- [6] H. Noprisson, "Implementasi Metodologi Agile Software Development pada Proyek Perangkat Lunak," vol. 5, no. 2, 2023, [Online]. Available: <https://jurnal.ikhafi.or.id/index.php/jusubi>
- [7] I. Kurniawan, Humaira, and F. Rozi, "REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android," 2020. [Online]. Available: <http://jurnal-itsi.org>
- [8] S. Mariko, "APLIKASI WEBSITE BERBASIS HTML DAN JAVASCRIPT UNTUK MENYELESAIKAN FUNGSI INTEGRAL PADA MATA KULIAH KALKULUS," *Jurnal Inovasi Teknologi Pendidikan*, vol. 6, no. 1, pp. 80–91, 2019, doi: 10.21831/jitp.v6.1.22280.
- [9] F. Widyoutomo, H. Ajie, and Widodo, "PENGEMBANGAN WEB SERVICE MODUL MAHASISWA PADA SISTEM INFORMASI AKADEMIK UNIVERSITAS NEGERI JAKARTA," 2021.
- [10] Novrina, R. Nurdiani Sari, and A. Rangga Kusuma, "PEMANFAATAN REACT NATIVE DAN FIREBASE UNTUK PEMBUATAN APLIKASI JUAL-BELI ITEM GAME ONLINE," 2021. [Online]. Available: <http://jurnal.bsi.ac.id/index.php/larik>