

Pengembangan Sistem Pengelolaan Toko ATK Pak Eko berbasis *Payment Gateway*

Dimas Ivon Anggitama¹, Nurudin Santoso²

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹dimasivonanggitama@gmail.com, ²nurudin.santoso@ub.ac.id

Abstrak

Toko ATK Pak Eko menyediakan kebutuhan alat tulis kantor dan jasa *print* yang sangat dibutuhkan oleh kalangan anak sekolah dan mahasiswa. Kebutuhan tersebut membuat pelanggan sangat banyak sehingga diperlukan efisiensi dalam proses jual beli sehingga pelayanan yang diberikan oleh pihak toko dapat lebih optimal dan dapat menjangkau calon pembeli lebih banyak. Oleh karena itu, diperlukan sistem untuk melakukan efisiensi dalam transaksi dan juga untuk memperluas jangkauan penjualan serta pemasaran produk. Dengan adanya sistem ini memudahkan calon pembeli untuk melihat produk secara *online* dan melakukan print dokumen tanpa harus ke toko terlebih dahulu serta dapat mempermudah proses pembayaran yang menggunakan *payment gateway* sebagai metode *cashless* yang lebih populer pada masa kini. Proses pertama dari penelitian ini yakni analisis kebutuhan dan diperoleh 22 kebutuhan fungsional beserta 1 kebutuhan non fungsional yaitu *compatibility*. Kemudian dilakukan proses perancangan sistem dan implementasi sistem yang merujuk kepada kebutuhan fungsional yang sudah di definisikan sebelumnya. Implementasi aplikasi menggunakan bahasa pemrograman *Flutter* serta *Firebase* sebagai media penyimpanan data. Hasil pengujian terhadap sistem yaitu pada pengujian unit menghasilkan status valid pada setiap jalur yang diuji dalam 3 sampel unit, pengujian integrasi menghasilkan status valid pada 1 kasus uji dan pengujian validasi yang menghasilkan status valid pada 1 sampel fungsional dengan 5 skenario uji.

Kata kunci: toko atk, flutter, firebase, payment gateway

Abstract

Pak Eko's ATK provides stationery and printing services that are needed by students. It makes the number of customer are numerous so efficiency is needed in transaction process so the services provided can be more optimal and it can reach out more customer. Therefore, an this system is needed to make transactions efficient and also to expand the reach of sales and product marketing. It easier for prospective buyers to view products and print service online without go to the store first and the payment process where the cashless method with payment gateway is more popular these days. Requirements analysis obtained 22 functional requirements along with 1 non functional requirement which compatibility. Process of system design and system implement are refers to functional requirements that have been defined. Implementation used Flutter and Firebase as data storage media. Unit testing produces a valid result on each path tested in 3 unit sample, integration testing produces a valid result in 1 test case, and validation testing produces valid result on 1 functional sample with 5 test scenario.

Keywords: stationary shop atk, flutter, firebase, payment gateway

1. PENDAHULUAN

Pada Toko ATK Pak Eko sendiri, selain menyediakan kebutuhan alat tulis kantor, toko ini juga menyediakan jasa *print* yang sangat dibutuhkan oleh kalangan anak sekolah dan mahasiswa. Pada kenyataannya, permasalahan muncul ketika jasa *print* dibutuhkan pada waktu yang hampir bersamaan, terutama ketika pagi

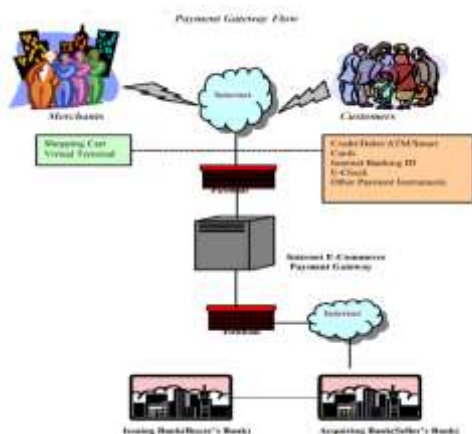
hari sebelum jam sekolah atau jam kerja dimulai, jam istirahat atau makan siang, dan di masa akhir semester. Ditambah lagi ketersediaan komputer dan *printer* sangat sedikit, yaitu hanya 3 unit. Sehingga terjadinya antrian yang sering menyebabkan rebutan antara pengguna yang hanya ingin melakukan *print* dan pengguna yang memerlukan penyuntingan terlebih dahulu. Belum lagi pengguna yang ingin melakukan

print dengan jumlah banyak seperti dokumen laporan tugas akhir, terutama dokumen PKL dan skripsi.

Selain layanan *print*, kebutuhan akan alat tulis kantor juga sama populernya terlebih ketika mendekati semester baru. Toko ingin melakukan efisiensi terhadap proses transaksi sehingga dapat menjangkau lebih banyak calon pembeli dengan proses yang lebih efisien. Sistem ini bertujuan untuk melakukan efisiensi dalam transaksi dan juga untuk memperluas jangkauan penjualan serta pemasaran produk. Dengan adanya sistem ini memudahkan calon pembeli untuk melihat produk secara *online* dan melakukan print dokumen tanpa harus ke toko terlebih dahulu serta dapat mempermudah proses pembayaran yang menggunakan *payment gateway* sebagai metode *cashless* yang lebih populer pada masa kini.

2. LANDASAN KEPUSTAKAAN

2.1 Payment Gateway



Gambar 1 Flow Payment Gateway

Source: (Gulati & Srivastava, 2007)

Payment gateway berfungsi sebagai sarana yang menjembatani antara *merchant* (toko) dan lembaga keuangan yang memproses transaksi dari *buyer/customer*. Sistem *gateway* ini akan melakukan autentikasi dan merutekan detail pembayaran dalam antara berbagai pihak dan bank terkait dengan sangat aman. Bisa dikatakan aman karena ada sistem *firewall* tersendiri yang mana sistem ini sudah terlindungi dan sesuai dengan ketentuan hukum yang berlaku (Gulati & Srivastava, 2007).

2.2 Flutter

Menurut (Andrian, et al., 2022), *flutter* merupakan SDK *Mobile App* milik *Google* yang

mana dapat memungkinkan *developer* dalam *men-develop* sistem yang dapat berlajam di sistem operasi berbeda yaitu *iOS* dan *android* dengan menggunakan kode program yang sama. *Flutter* menggunakan *framework* bernama *Dart* dan dapat melakukan membangun sistem secara *native*. *Flutter* mempunyai *widget* sendiri yang dapat membentuk *interface* dan kerangka sistem.

Development pada *flutter* termasuk cepat dan ringkas. *Flutter* memiliki fitur *hot reload* yang merupakan fitur untuk menampilkan perubahan yang sedang dilakukan pada *code* secara instan pada *emulator*, *simulator* dan *hardware*. Dalam waktu singkat, kode yang sedang dirubah akan dimuat ulang ketika sistem masih berjalan tanpa perlu melakukan *restart* sehingga memudahkan dalam pembuatan *user interface* atau ketika terdapat penambahan fitur dan juga mempermudah aktivitas pada saat *bug fixing*. Kelebihan selanjutnya yaitu *flutter* maupun *Dart* jenisnya adalah *open source*.

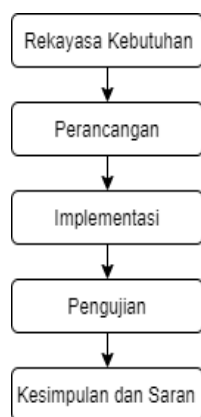
Namun, *flutter* juga memiliki beberapa kelemahan yaitu tutorial maupun dokumentasi yang tidak sebanyak bahasa pemrograman lainnya karena *flutter* merupakan produk yang masih baru dalam dunia teknologi. Kelemahan selanjutnya adalah *library third party* pada *flutter* masih tergolong sedikit dibandingkan bahasa pemrograman lainnya. Namun demikian, seiring berjalan waktu akan banyak *third party* yang melakukan *development library* dapat diimplementasikan oleh *flutter*. Walaupun masih belum bisa menyaingi pesaing *flutter* yaitu *react native* yang mana sudah lebih banyak *third party libraries* yang telah tersedia.

2.3 Firebase

Firebase Realtime Database merupakan basis data yang memungkinkan sistem untuk bisa dikoneksikan ke *cloud storage* karena *database* ini di-host pada *google cloud*. Data akan di-save dan di-synchronize secara *realtime* ke setiap *client* yang terhubung pada sistem. Ketika *developer* membangun sistem *cross platform* dengan SDK *android*, maka semua *client* akan berbagi sebuah *instance Realtime Database* dan *client* akan memperoleh *update data* terbaru secara *realtime*. Berbeda halnya dengan *SQL*, tidak ada tabel atau *record* (Firebase, 2022). Pada *media storage* ini, data akan disimpan dalam dokumen yang diatur ke dalam *collection*. Setiap dokumen memiliki pasangan *key* dan *value* seperti nama: "Dimas".

3. METODOLOGI PENELITIAN

Jenis metodologi penelitian yang digunakan adalah Model *Waterfall* yang dengan tujuan agar masing-masing langkah penelitian dijalankan sesuai urutan dan menghindari perulangan dari masing-masing langkah penelitian.



Gambar 2 Diagram *Waterfall* yang digunakan

Tahap pertama adalah rekayasa kebutuhan yang dilakukan proses *data collection* dari objek yang akan diteliti menggunakan beberapa teknik penggalan kebutuhan seperti elisitasi kebutuhan, identifikasi aktor, identifikasi kebutuhan, dan pemodelan kebutuhan. Tahap perancangan sistem atau bisa disebut sebagai tahap *system design* selanjutnya dikerjakan ketika tahap pertama sudah didefinisikan. Tahap ini akan dilaksanakan dengan *step* sebagai berikut:

1. Perancangan arsitektur merupakan kerangka kerja perangkat lunak.
2. Perancangan data yang akan dikelola sebagai pedoman untuk implementasi pembangunan basis data pada sistem.
3. Perancangan sistem *sequence diagram* dan *class diagram* (*Requirement Modelling*).
4. Perancangan antarmuka sistem (*user interface design*).

Dikarenakan pada penelitian ini sistem akan diimplementasikan menjadi sistem berbasis *mobile*, maka bahasa pemrograman yang digunakan merupakan bahasa pemrograman *Dart* di atas rangka kerja *Flutter*. Tahap *system implementation* yang akan dilakukan sebagai berikut:

1. *System specification* yang terdiri dari *hardware and software specification*.

2. Implementasi basis data pada sistem pada penelitian ini menggunakan *firebase*.
3. Implementasi kode program berdasarkan daripada *pseudocode*.
4. Implementasi antarmuka (*user interface*) yang berlandaskan pada perancangan antarmuka yang dilakukan sebelum tahap *system implementation*.

Pengujian akan dilakukan setelah menyelesaikan tahap perancangan dan implementasi sistem. Pada tahap pengujian di penelitian ini akan menggunakan metode *blackbox testing* dan *whitebox testing*. Pengujian dilakukan di setiap tingkat pada perangkat lunak yaitu pengujian unit dan integrasi. Pada pengujian integrasi menggunakan metode *incremental*. Untuk meyakinkan bahwa sistem telah sinkron dengan *requirement* yang didefinisikan, maka perlu dilakukan pengujian validasi. Pengujian unit dan integrasi pada penelitian ini menggunakan metode *whitebox testing*, sementara untuk pengujian validasi menggunakan *blackbox testing*.

Pada tahap kesimpulan dan saran, dilaksanakan sesudah segala tahapan penelitian mulai dari rekayasa kebutuhan, perancangan sistem, implementasi hasil perancangan, dan pengujian selesai dilakukan. Penulisan kesimpulan berlandaskan dari hasil kesatuan tahapan yang telah dilakukan pada penelitian ini agar dapat mengetahui bahwa penelitian dalam pemecahan permasalahan yang telah didefinisikan di awal apakah berhasil atau tidak. Penulisan saran juga diperlukan yang tujuannya untuk referensi, perawatan dari implementasi sistem, dan pengembangan penelitian ini berikutnya.

4. HASIL DAN PEMBAHASAN

Hasil penelitian yang didapatkan serta pembahasan yang dapat dijabarkan dari penelitian ini dibagi dalam 4 poin, yaitu rekayasa kebutuhan, perancangan, implementasi, dan pengujian sistem.

4.1 Rekayasa Kebutuhan

Rekayasa kebutuhan pada penelitian ini memiliki aktivitas yang mencakup analisis proses bisnis dan kebutuhan beserta penjabarannya sebagai basis landasan dalam pengembangan sistem. Pada tahap ini dilakukan

untuk mem-*built* dasar sistem. Aktivitas yang dilakukan adalah analisis prosedur bisnis yang sekarang berlaku di Toko ATK Pak Eko untuk memperoleh kebutuhan yang diperlukan yang dilakukan dengan cara wawancara kepada Bapak Eko selaku pemilik toko dari Toko ATK Pak Eko. Identifikasi layanan toko dilakukan untuk mencari tahu layanan apa saja yang tersedia pada Toko ATK Pak Eko.

Tahap identifikasi layanan ini dilakukan setelah melakukan wawancara kepada pihak terkait dan hasilnya adalah layanan *print* dan pembelian ATK. Gambaran umum dari sistem ini dalam penelitian adalah aplikasi yang menyediakan layanan untuk melakukan transaksi *online* yang ditujukan khusus kepada calon pembeli di Toko ATK Pak Eko. Layanan yang disediakan meliputi pemesanan, pembayaran dan *online delivery*. Untuk memudahkan proses pembayaran secara *online*, sistem pada penelitian ini menggunakan *third party payment gateway*.

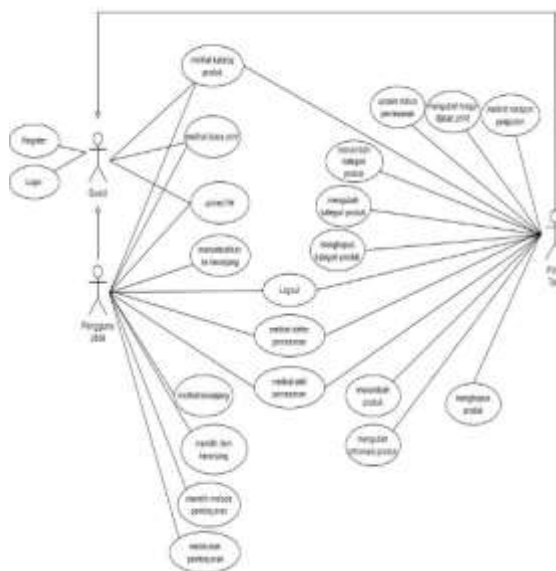
Hal selanjutnya yang perlu diidentifikasi adalah aktor. Daftar aktor yang teridentifikasi dapat dilihat pada Tabel 1.

Tabel 1 Identifikasi Aktor

Aktor	Deskripsi Aktor
Guest	Guest hanya dapat melakukan <i>register</i> , <i>login</i> , melihat katalog. Guest tidak dapat melakukan proses transaksi.
Pihak Toko	Pihak Toko merupakan aktor yang digeneralisasi dari <i>guest</i> . Pihak Toko dapat mengelola <i>control panel</i> sistem layaknya admin.
Pengguna Jasa	Pengguna jasa merupakan aktor yang digeneralisasi dari <i>guest</i> dan dapat melakukan proses transaksi.

Pemodelan semua kebutuhan dari hasil identifikasi kebutuhan dan spesifikasinya direpresentasikan ke dalam bentuk diagram *use case*. Setiap aktor dapat melakukan sesuatu pada sistem sesuai dengan perannya masing – masing.

Sebagai contoh, *guest* dan pengguna jasa memiliki kebutuhan yang beririsan yaitu melihat daftar ATK namun *guest* tidak dapat memilih barang maupun melakukan transaksi. Pihak toko dapat melakukan pengelolaan terhadap produk, metode pembayaran dan lainnya. Gambar 3 merupakan diagram *use case* yang diimplementasikan pada penelitian ini.



Gambar 3 Use Case Diagram

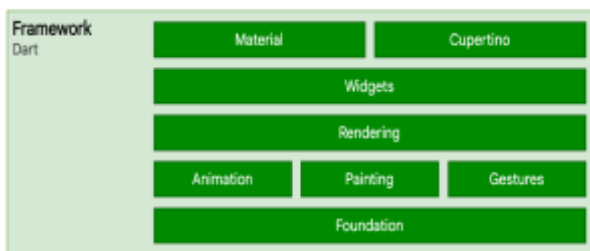
Penjabaran *use case* pada skenario *use case* memiliki tujuan untuk dapat mengenali dan menjabarkan lebih *detail* dari langkah-langkah yang dilakukan aktor terhadap sistem serta bagaimana *system response* kepada aksi yang dilakukan oleh aktor yang terlibat untuk mencapai *objectivity* sebuah *use case*. Alur dari skenario *use case* akan menjadi landasan dalam pembuatan *sequence diagram*.

4.2 Perancangan Sistem

Tahapan perancangan (*system design*) dapat disebut sebagai tahap yang mana sistem di-*design* sebelum kemudian dilakukan engimplementasian. Penjabaran atas *system design* yang dilakukan pada tahap ini diantaranya akan dijelaskan dan divisualisasikan dalam perancangan arsitektur, perancangan data, pemodelan *sequence diagram*, pemodelan *class diagram*, perancangan algoritma, dan perancangan antarmuka pengguna.

Perancangan arsitektur berlandaskan pada *framework* yang digunakan dalam *software development* pada penelitian ini, yang mana adalah *Framework Dart*. Gambar 4

menunjukkan serangkaian lapisan yang terdapat pada *framework* tersebut.



Gambar 4 Flutter

Menurut (Flutter, 2022) arsitektur *Framework Dart* adalah sebagai berikut:

1. *Foundational* yang berarti *framework* tersebut mem-*provide function* dan *class* dasar yang digunakan untuk *application development* menggunakan *flutter*.
2. *Animation* yang berarti kemampuan untuk memberikan tampilan interaktif, *Painting* memiliki fungsi untuk dapat mengubah warna *widget*, dan *Gestures* yang berfungsi menangkap interaksi pengguna terhadap sistem seperti menekan, menyeret dan lainnya.
3. *Rendering* merupakan komponen perender tingkat rendah dan merender semua di aplikasi *flutter*.
4. *Widgets* merupakan *component* yang sifatnya adalah *immutable* dan menjadi bagian dari *user interface*. Tidak seperti *framework* lain yang membagi sistem menjadi *views*, *controllers*, dan *models*. *Flutter* memiliki sebuah *unit object* yang konsisten yang disebut *widgets*.
5. *Material* dan *Cupertino* merupakan dua *widget* yang disesuaikan dengan bahasa desain tertentu. *Widget Material Design* mengaplikasikan bahasa desain Google, sementara *Widget Cupertino* desain iOS milik Apple.

Selanjutnya tahap perancangan data dilakukan dilakukan terlebih dahulu perancangan pada data yang akan dikelola sebagai pedoman untuk implementasi pembangunan basis data pada sistem. Data di-*design* dengan berlandaskan pada hasil analisis data pada saat melakukan rekayasa kebutuhan, serta disesuaikan terhadap kebutuhan *framework*

yang

digunakan

```

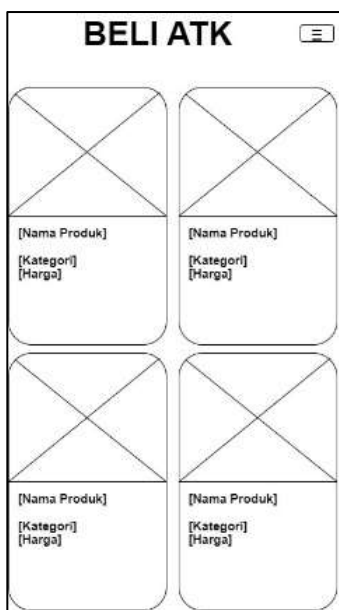
e-business-atk
├── - cart
│   ├── - documentID : string
│   │   ├── - isItemChecked : boolean
│   │   ├── - orderID : string
│   │   ├── - productID : string
│   │   ├── - productQuantity : int
│   │   └── - userID : string
│   └── - categories
│       ├── - documentID : string
│       │   └── - id : string
│       └── - category : string
├── - orders
│   ├── - documentID : string
│   │   ├── - status : string
│   │   ├── - totalPrice : int
│   │   ├── - totalPrintItem : int
│   │   ├── - totalProductItem : int
│   │   └── - userID : string
├── - products
│   ├── - documentID : string
│   │   ├── - brand : string
│   │   ├── - category : string
│   │   ├── - image : string
│   │   ├── - name : string
│   │   ├── - price : int
│   │   └── - stockQuantity : int
├── - users
│   ├── - documentID : string
│   │   ├── - address : string
│   │   ├── - email : string
│   │   ├── - name : string
│   │   ├── - password : string
│   │   ├── - phoneNumber : string
│   │   └── - status : string
├── - printPrice
│   ├── - documentID : string
│   │   ├── - colorfulPrice : int
│   │   └── - greyScalePrice : int

```

Gambar 5 Perancangan Data

Penelitian ini juga dilengkapi dengan perancangan *sequence diagram* yang dipergunakan untuk landasan yang menjelaskan alur pertukaran informasi atau pesan antar objek. Pemodelan diagram *sequence* didasarkan pada skenario *use case* yang telah dibuat. Pada bagian perancangan diagram *class*, digambarkan *object* – *object* yang termuat pada sistem beserta *relationship*-nya. Diagram kelas dari sistem ATK ini memiliki total 50 *class* dengan rincian 7 *class* diantaranya merupakan kelas BLOC (*business logic / controller*), 7 *class* diantaranya merupakan *state class* data, 22 *class* diantaranya merupakan *event class* (fungsi), 1 *class* diantaranya merupakan *main class*, dan 8 *class* diantaranya merupakan *class* tampilan antar muka (*view*). Seluruh *object* kelas pada diagram *class* tersebut ber-*associate* satu dan lainnya sesuai dengan garis penghubungnya.

Perancangan algoritma menghasilkan sampel tiga *pseudocode* yang berisi algoritma yang akan diimplementasikan pada sistem ini. *Pseudocode* yang dijadikan rancangan algoritma dalam penelitian ini merupakan salah satu fitur utama yaitu melihat katalog produk, menambahkan ke keranjang dan menambah produk. Tahap terakhir dari perancangan dalam penelitian ini adalah perancangan *user interface* (antarmuka). Gambar 6 menunjukkan antarmuka katalog ATK yang masih berbentuk rancangan.



Gambar 6 Perancangan Antarmuka Katalog Produk

4.3 Implementasi Sistem

Proses implementasi sistem pada penelitian ini mencakup 4 *step* yaitu *step* spesifikasi sistem, implementasi kode program, implementasi basis data dan implementasi antarmuka. Sistem ini dikembangkan menggunakan bahasa pemrograman *flutter* dan *firebase* sebagai tempat penyimpanan data. Implementasi basis data aplikasi dikerjakan sesuai dengan perancangan sistem yang sudah dilakukan terlebih dahulu yaitu perancangan data. Kode program pada aplikasi ini dikerjakan sesuai dengan perancangan sistem sebelumnya pada tahap perancangan algoritma. Antarmuka pada sistem juga dikerjakan sesuai dengan perancangannya. Berikut antarmuka dari sistem Toko ATK Pak Eko yang dapat dilihat pada Gambar 7.



Gambar 7 Implementasi Antarmuka Katalog Produk

4.4 Pengujian Sistem

Tahapan selanjutnya dari penelitian yang dilakukan setelah implementasi yaitu pengujian sistem. Pengujian yang akan dijalankan terhadap sistem ini terbagi menjadi pengujian unit dan pengujian integrasi dengan menggunakan metode *whitebox testing* serta pengujian validasi dengan menggunakan metode *blackbox testing*. Pada tahap ini diambil beberapa sampel fungsional yang akan diuji.

Pengujian unit memiliki tujuan untuk menguji unit seperti kode program, objek, atau kelas. Pengujian unit ini dijalankan dengan salah satu metode pada *Whitebox Testing* yaitu basis *path testing* terhadap tiga sampel uji. Hasil pengujian unit yaitu 100% valid untuk semua skenario uji pada tiga sampel. Tiga sampel tersebut adalah fungsi untuk melihat katalog produk, menambahkan ke keranjang dan menambah produk.

Pengujian integrasi dilakukan dengan 1 fungsional sampel yaitu menambah produk ke dalam keranjang. Hasilnya adalah valid untuk semua skenario uji dengan jumlah 2 skenario yaitu ketika *product quantity* bernilai 0 dan *product quantity* bernilai diatas 0. Pengujian validasi dilakukan dengan menguji respon sistem terhadap *behavior* aktor. Hasil pengujian validasi berdasarkan skenario pada fitur menambahkan produk. Pengujian yang dilakukan antara lain kasus uji berhasil

menambah produk, kasus uji *field* kosong saat menambah produk, kasus uji gambar produk tidak diunggah, kasus uji meng-*input* harga dibawah Rp. 100 dan kasus uji ketika harga dan stok di-*input* dengan nilai 0. Hasil pengujian valid untuk 5 skenario validasi.

5. PENUTUP

5.1 Kesimpulan

Kesimpulan yang dapat peneliti peroleh dari penelitian yang sudah dikerjakan adalah sebagai berikut:

1. Hasil dari identifikasi kebutuhan dalam ini yaitu diperoleh 22 kebutuhan fungsional dengan 3 aktor yang berperan dalam sistem yaitu *guest*, pihak toko, dan pengguna jasa. Dari identifikasi kebutuhan yang telah diperoleh tersebut dihasilkan permodelan kebutuhan berupa *use case diagram* dan *use case scenario*.
2. Hasil dari perancangan berdasarkan identifikasi kebutuhan yang sudah dijabarkan yaitu diperoleh perancangan data dengan 6 *collection*, perancangan 3 sampel *sequence diagram*, perancangan diagram *class* yang memiliki total 50 *class* dengan rincian 7 kelas diantaranya merupakan kelas BLOC (*business logic / controller*), 7 kelas diantaranya merupakan kelas *state* data, 22 kelas diantaranya merupakan kelas *event* (fungsi), 1 kelas diantaranya merupakan kelas *main*, dan 8 kelas diantaranya merupakan kelas tampilan antar muka (*view*). Seluruh objek kelas pada diagram *class* tersebut saling berasosiasi satu sama lain sesuai dengan garis penghubungnya. Perancangan algoritma yang berupa *pseudocode* dan perancangan antarmuka sistem yang berupa *wireframe*.
3. Implementasi dari penelitian ini menghasilkan sistem dengan fitur – fitur utama diantaranya *layanan print*, pembelian ATK secara *online*, pembayaran dan fitur lainnya sesuai dengan hasil identifikasi kebutuhan yang telah didefinisikan dan sistem yang dibangun menggunakan dengan bahasa pemrograman *flutter*.
4. Hasil pengujian sistem yaitu pada *unit testing* mencapai status valid pada 3 prosedur uji, *integration testing* mencapai status valid pada 1 kasus uji dan validation

testing yang mencapai status valid pada 1 sampel fungsional dengan 5 skenario uji.

5.2 Saran

Penelitian untuk mengembangkan sistem ini diharapkan dapat melakukan pembayaran dengan berbagai macam metode pembayaran yang mana tidak terbatas pada uang elektronik saja. Sistem juga dapat dikembangkan lebih *advance* dengan menambahkan fitur *maps* untuk layanan *online delivery*. Sistem juga dapat dikembangkan dengan mempertimbangkan untuk melakukan riset mendalam terkait *user experience* terlebih dahulu.

DAFTAR REFERENSI

- Andrean, K., Armanto, H. & Pickerling, C., 2022. Sistem Tempat Parkir Terintegrasi yang Dilengkapi dengan Aplikasi Mobile dan Mikrokontroler. *Journal of Information System, Graphics, Hospitality and Technology*, pp. 22-29.
- Firebase, 2022. *Firestore Documentation Guides*. [Online] Available at: <https://firebase.google.com/products/realtime-database?hl=id> [Accessed 2022].
- Flutter, 2022. *Flutter architectural overview*. [Online] Available at: <https://docs.flutter.dev/resources/architectural-overview>
- Gulati, V. P. & Srivastava, S., 2007. The Empowered Internet Payment Gateway.