

Rancang Bangun Aplikasi Penjualan Barang Toserba Menggunakan Metode OOAD (Object Oriented Analysis Design) Berbasis Mobile Platform Pada CV FCH Kema Pole Indonesia

Fahmi Rizky Maulidy¹, Dedy Rahman Prehanto²

^{1,2}Jurusan Teknik Informatika/Sistem Informasi, Universitas Negeri Surabaya

¹fahmi.19115@mhs.unesa.ac.id

³dedyrahman@unesa.ac.id

Abstrak— Pemakai *smartphone* menjadi hal umum bagi masyarakat Indonesia. Hal ini terbukti dengan persentase penggunaan mencapai 66.36% dari total keseluruhan masyarakat Indonesia berdasarkan survey yang dilakukan oleh Kementerian Komunikasi dan Informatika. Dengan terjadinya COVID-19 banyak usaha yang mengalami penurunan sejak saat diberlakukan *social distancing* khususnya UMKM (Usaha Kecil dan Menengah). CV FCH Kema Pole merupakan UMKM (Usaha Kecil dan Menengah) yang berlokasi di Surabaya dengan menjalankan usaha penjualan TOSERBA (Toko Serba Ada). Mempertimbangkan pemakaian *smartphone* yang cukup besar di Indonesia menjadi kesempatan untuk menggunakan *smartphone* sebagai dasar transaksi sehingga memudahkan konsumen untuk memesan barang atau produk tanpa harus ke toko. Berhubung untuk pengembangan membutuhkan waktu dalam pembuatan aplikasi yang dapat berjalan di *smartphone*, sedangkan CV FCH Kema Pole membutuhkan solusi cepat untuk meningkatkan pemasukannya. Maka diperlukan perencanaan yang matang dimana pengerjaan dapat berjalan secara efektif agar CV FCH Kema Pole dapat segera menjalankan transaksi penjualan melalui *smartphone*. Penulis memiliki dua solusi untuk mempercepat realisasi tersebut. Pertama menerapkan teknologi mengadopsi teknologi yang dapat mensupport *multi-platform* mengingat *smartphone* memiliki beberapa jenis *operating system* seperti *Android* dan *IOS*. Pengembangan menggunakan *Flutter* dapat menjadi alternatif untuk permasalahan tersebut. Pasalnya, aplikasi yang dibuat dengan *Flutter* dapat berjalan secara *multi-platform*. Kedua menerapkan metode OOAD (*Object-Oriented Analysis and Design*) sebagai dasar arsitektur aplikasi. Dengan OOAD dapat mengatur arsitektur sistem agar dapat dikembangkan ke dalam OOP (*Object-Oriented Programming*) sehingga memudahkan programmer dalam mengimplementasikan sesuai dengan arsitektur rancangan sehingga dapat mengurangi waktu dalam pengerjaan serta meningkatkan produktivitas programmer.

Kata Kunci— Aplikasi *Mobile*, *Object Oriented Analysis Design*, *Payment Gateway*, *UMKM*, *E-Commerce*, *Flutter*, *Laravel*, *COVID-19*

I. PENDAHULUAN

Sejak pandemi COVID-19 berlangsung hampir semua sektor usaha terdampak seperti sektor pariwisata, sektor ritel, industri media, sektor jasa, dsb mengalami kelesuan ekonomi [1]. Kondisi ini berawal dari kebijakan untuk melakukan pembatasan sosial dalam rangka percepatan penanganan COVID-19 [2].

Kema Pole Indonesia merupakan UMKM (Usaha Kecil dan Menengah) yang berlokasi di Surabaya dengan menjalankan usaha penjualan TOSERBA (Toko Serba Ada) yang berlokasi di Surabaya. Penjualan dilakukan dengan interaksi secara langsung di toko. Kondisi tersebut menyebabkan persentase pengunjung yang datang berkurang secara drastis karena pandemi COVID-19. Jika Kema Pole Indonesia menerapkan aplikasi penjualan secara online pastinya akan membuka peluang baru tanpa harus belanja secara langsung ke toko. Pengembangan aplikasi yang diterapkan pada Kema Pole Indonesia bertujuan untuk mengembalikan bahkan menaikkan omset penjualan mereka serta dengan didukungnya alternatif pengiriman produk berupa kurir dari pihak toko Kema Pole Indonesia yang mencakup wilayah sekitar CV. FCH Kemapole Indonesia akan memudahkan konsumen. Harapan penulis dengan membangun aplikasi dapat membantu proses dan meningkatkan penjualan Kema Pole Indonesia.

Pada survey yang dilakukan oleh Kementerian Komunikasi dan Informatika didapatkan data bahwa masyarakat sudah banyak menggunakan *smartphone* dengan persentase penggunaan mencapai 66.36% dari total keseluruhan masyarakat Indonesia. Jika meninjau lebih dalam untuk perbandingan individu yang memiliki *smartphone* di setiap pulau, pulau Jawa memiliki persentase paling tinggi yaitu 86.6% dari keseluruhan di pulau yang ada di Indonesia [3]. Berdasarkan referensi tersebut jika ditelusuri dari segi segmentasi pengguna maka Kema Pole Indonesia memiliki prospek lebih tinggi dalam mendapatkan pelanggan ketika melakukan penjualan secara online khususnya dengan *smartphone* karena Kema Pole Indonesia berlokasi di Surabaya.

Perancangan sistem aplikasi menggunakan metode OOAD (*Object-Oriented Analysis and Design*) dapat mengatur arsitektur sistem agar dapat diterapkan ke pengembangan dengan menerapkan OOP (*Object-Oriented Programming*) sebagai dasarnya saja. Sayangnya, beberapa programmer masih menulis *spaghetti code* walaupun telah menerapkan *class* dan *packages* pada saat melakukan *coding* [4]. Untuk membangun *code* yang benar-benar OOP diperlukan penentuan delegasi dan pembagian tugas yang tepat supaya mengurangi tingkat pada *code* yang tumpang tindih. Tujuan dari pada penulis menerapkan metode OOAD supaya dapat memproyeksikan perencanaan sistem sesuai dengan kebutuhan sistem dan mengurangi kemungkinan terjadinya *overlapping* sehingga

mengurangi waktu pengerjaan dan meningkatkan produktivitas.

Sebagai bahan pertimbangan untuk penerapan metode OOAD penulis melakukan perbandingan dengan beberapa penelitian sebelumnya. Pada penelitian yang dilakukan oleh Purwaningtiyas (2018) dengan judul *e-commerce* Penjualan Berbasis Metode OOAD memiliki permasalahan yang sama yaitu pada penerapan *e-commerce* untuk aplikasi penjualan, tetapi pada produk dan sistem berbeda yaitu penjualan garam [5]. Penelitian dilakukan untuk memperbaiki kinerja penjualan yang selama ini berjalan dengan melakukan planning dan modeling sesuai dengan kebutuhan PT Musi Utama Bercahaya Palembang yang merupakan studi kasus penelitian.

Terdapat penelitian lainnya milik Rahayu (2019) dan Nurlani dengan judul Sistem Informasi Kuesioner Evaluasi Kinerja Dosen Menggunakan Metode OOAD yang menawarkan penerapan teknologi menggunakan framework *Codeigniter* [6]. Pada pengembangan penelitian sudah ada standar formula yang digunakan untuk menghitung evaluasi kinerja dosen. Setiap entitas yang terhubung dengan aplikasi dijabarkan sedemikian rupa menggunakan *use case diagram*.

Jika ditinjau dari penelitian diatas sebagai referensi maka penulis dapat merumuskan terdapat tiga proses pada tahap pengembangan aplikasi menggunakan metode OOAD yaitu perancangan berdasarkan behavior object setiap entitas melalui UML (*Unified Modelling Language*) ditentukan dengan identifikasi skenario use-case untuk menciptakan model dari sistem yang dikembangkan, penjadwalan berdasarkan identifikasi struktur konsep model, dan unit testing yang dilakukan dengan uji coba *black-box*. Terdapat pembaruan penelitian yang dilakukan oleh penulis yaitu, perancangan OOAD dengan pola arsitektur MVVM (*Model View, View, Model*) yang dikhususkan untuk pola arsitektur *front-end*. Berbeda dengan dengan pola arsitektur yang biasa diterapkan menggunakan MVC (*Model, View, Controller*) yang dikhususkan untuk *back-end*.

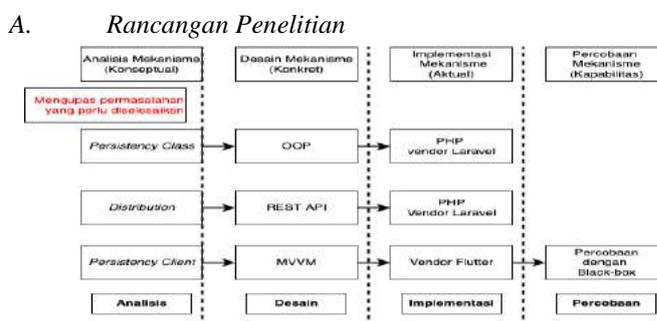
Secara garis besar penerapan aplikasi penjualan yang dikembangkan pada toko milik Kema Pole Indonesia menerapkan sistem *e-commerce* tetapi menerapkan penjualan produk berupa barang grosir maupun penjualan barang retail. Hal tersebut diharapkan menjadi keunggulan dari sistem aplikasi milik Kema Pole Indonesia dibandingkan dengan toko *online* yang beberapa diantaranya tidak menyediakan layanan pembelian produk secara grosir maupun retail. Admin Kema Pole Indonesia dapat mengatur satuan barang dan menentukan minimal pembelian produk yang dapat di beli oleh pelanggan. Untuk mendukung aplikasi dapat berjalan dengan baik pasti dibelakangnya terdapat teknologi yang mendukung jalannya aplikasi. Aplikasi mobile platform termasuk kedalam kategori *client-side* yang memerlukan dukungan *server-side* [7]. Pembangunan pada mobile platform dibuat menggunakan Flutter yang merupakan perangkat UI (*User Interface*) dengan keunggulan bisa digunakan untuk cross-platform seperti website, mobile, dan desktop [8]. Adopsi Flutter diterapkan agar aplikasi dapat berjalan dengan OS (*Operating System*) IOS dan Android. Untuk mengakses data ke server-side diperlukan *endpoint* untuk dapat mengakses data yang dibutuhkan. REST (*Representational State Transfer*) API (*Application Programming Interface*) dapat digunakan untuk mengambil

data dari database untuk perangkat lain yang diakses melalui internet [9]. Dalam implementasi REST API tentunya dibangun menggunakan bahasa pemrograman. Rencana pembangunan aplikasi REST API dibangun dengan *Laravel* yang merupakan framework pada bahasa pemrograman PHP (*Hypertext Preprocessor*) dan menerapkan MVC (*Model, View, Controller*). dengan mempertimbangkan kecocokan PHP dalam menerapkan OOP [10].

Jika dilihat lebih dalam pada proses transaksi merupakan area yang sangat sensitif karena menyangkut masalah finansial. Setelah dilakukan beberapa pertimbangan riset penerapan yang paling ideal pada kondisi Kema Pole Indonesia adalah dengan menerapkan teknologi *Payment Gateway* yang menjadi opsi standar pembayaran. Melalui *Payment Gateway*, aplikasi penjualan dapat melakukan akses transaksi online dengan beberapa opsi pembayaran melalui bank yang ditujukan untuk pelanggan dengan standar keamanan yang sudah tervalidasi [8].

Untuk memastikan aplikasi yang dikembangkan memiliki fitur-fitur yang sesuai dan dapat bekerja dengan baik, maka diperlukan pengujian untuk mengetahui kelayakan dari aplikasi tersebut. Menurut Nurudin, dkk.,(2019) pengujian kelayakan sebuah aplikasi sangat diperlukan dalam penilaian kualitas aplikasi yang ada, agar fungsi dalam aplikasi dapat bekerja sesuai dengan yang diharapkan pengguna [11]. Salah satu metode pengujian aplikasi yang banyak digunakan oleh para pengembang aplikasi adalah Black Box Testing. Hidayat dan Muttaqin (2020:3) menjelaskan bahwa *black-box testing* merupakan teknik pengujian kelayakan aplikasi yang berfokus pada spesifikasi fungsional dari perangkat lunak, Kema Pole Indonesia dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program sesuai tabel pengujian yang disediakan oleh pengembang [12]. Beberapa literasi yang telah diriset oleh penulis pada penelitian ini merujuk pada dampak penerapan OOAD dari segi pemberlakuan saat perancangan *main flow* dan pembangunan aplikasi penjualan Kema Pole Indonesia. Dengan demikian penulis menjadikannya sebagai poin bahan penelitian dengan judul “Rancang Bangun Aplikasi Penjualan Toserba Menggunakan Metode OOAD (*Object Oriented Analysis Design*) Berbasis Mobile Platform Pada CV FCH Kema Pole Indonesia”.

II. METODOLOGI PENELITIAN



Gambar 1 Alur Rancangan Penelitian *sequence diagram*.

Dapat dilihat pada gambar 1 merupakan alur rancangan menggunakan *sequence diagram* yang dapat diklasifikasi menjadi menjadi 4 tahap seperti berikut ini.

1. Analisis Mekanisme

Analisis mekanisme secara garis besar penguraian terhadap bagian-bagian detail yang akan diimplementasikan ke dalam environment dan tidak menggantung ke spesifikasi implementasi. Berikut untuk detail analisis mekanisme::

- *Persistence Class* - objek yang digunakan untuk menyimpan berkas-berkas *back-end* seperti *services*, *helper*, *controler*.
- *Distribution* - penyaluran data dari *backend* sebagai penyedia data ke *front-end* sebagai UI pengguna. Elemen ini sebenarnya terdapat di dalam *persistence class* tetapi memiliki metodologi yang tidak terikat dengannya.
- *Persistence Client* - objek yang digunakan untuk menyimpan berkas-berkas *mobile*.

2. Desain Mekanisme

Desain mekanisme merupakan standar yang diterapkan dalam melakukan implementasi mekanisme. Tujuan dari pada penentuan standar dalam melakukan implementasi untuk identifikasi dan menentukan pola, struktur, dan objek secara menyeluruh untuk digunakan sebagai penentu parameter sekaligus pedoman dalam pembangunan aplikasi. Berikut untuk detail desain mekanisme:

- *OOP* - metode pemrograman yang digunakan oleh berbagai macam bahasa pemrograman.
- *REST API* - distribusi *architectural constraints* melalui protokol HTTP yang dapat terhubung dengan berbagai macam *platform*.
- *MVVM* - merupakan pola arsitektur yang diatur khusus untuk pengembangan *graphical user interface*.

3. Implementasi Mekanisme

Implementasi mekanisme mencakup apa saja yang digunakan ketika membangun aplikasi seperti implementasi bahasa pemrograman, vendor, dan teknik. Berikut untuk detail implementasi mekanisme:

- PHP Vendor *Laravel* - *framework* PHP yang dibuat oleh Taylor Otwell dengan pola desain MVC.
- PHP Vendor *Laravel* - *Laravel* juga menyediakan untuk REST API dengan metode *Get*, *Post*, *Put*, *Patch*, dan *Delete*.
- Vendor *Flutter* - *framework* yang digunakan untuk membangun *native app multi platform* yang dibuat oleh Google.

4. Percobaan Mekanisme

Percobaan akan berfokus pada *Persistence client* dengan menentukan skenario uji coba yang sesuai menggunakan *black-box*.

B. Subyek Penelitian

Terdapat 2 subjek pada penelitian ini yaitu :

1. Aplikasi-aplikasi *e-commerce* yang ada di Indonesia dapat digunakan sebagai figur tolak ukur kebutuhan aplikasi.
2. Pihak CV. FCH Kema Pole Indonesia sebagai penyesuaian kebutuhan aplikasi.
3. Masyarakat umum sebagai pihak pembeli pengguna dari sistem.

C. Tempat dan Waktu Penelitian

Penelitian dilaksanakan di toko CV. FCH Kema Pole Indonesia yang beralamat di Ds. Hendrosari, Kel. Hendrosari, Kec.

Menganti, Kab. Gresik, Prov. Jawa Timur. dengan estimasi waktu penelitian sekitar tiga bulan.

D. Analisis Kebutuhan

1. Kebutuhan Fungsional

Sistem aplikasi yang diterapkan pada Kema Pole Indonesia bertujuan utama untuk memudahkan pengguna yang ingin mencari suatu produk untuk dibeli dengan menggunakan sistem online. Dalam aplikasi ini juga terdapat beberapa menu yang memudahkan hal tersebut seperti menu daftar produk dilengkapi dengan fitur beberapa filter seperti kategori, jumlah *rating* atau ulasan, serta dilengkapi fitur sortir harga mulai dari yang termurah hingga yang termahal dan begitu pula sebaliknya. Terdapat juga menu detail produk yang berisikan deskripsi lengkap dari suatu produk meliputi sisa stok produk, harga normal produk, harga grosir produk, dan juga sisa stok dari produk tersebut. Dengan adanya fitur-fitur tersebut, diharapkan dapat memudahkan pengguna untuk mendapatkan produk yang benar-benar mereka inginkan.

2. Kebutuhan Non Fungsional

● Kebutuhan Sistem

Perangkat keras yang diperlukan dalam perancangan serta pembangunan sistem aplikasi ini meliputi : MacOS *device* untuk pengembangan ke IOS, *smartphone* Android, dan *smartphone* IOS.

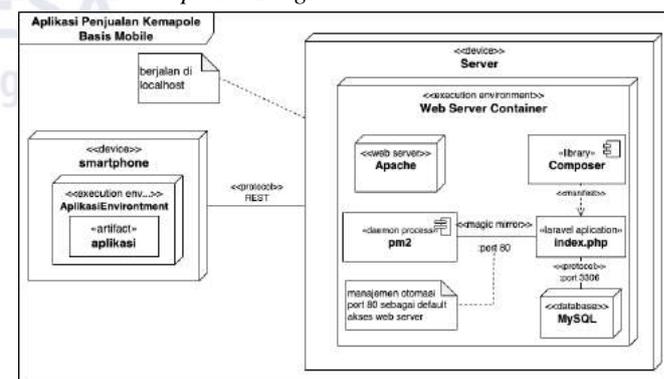
● Bahasa Pemrograman

Dalam pembuatan sistem aplikasi ini digunakan bahasa pemrograman *PHP* dengan *framework laravel* untuk *back-end* dan *Dart Programming Language* dengan *framework Flutter*.

● Batasan

Pada pembahasan penelitian penulis menjelaskan lebih spesifik ke arah OOAD dengan pola arsitektur MVVM dan belum diuraikan secara terperinci untuk rancangan dari sisi *back-end* yang menggunakan pola arsitektur MVC. Untuk *back-end* penulis menyisipkan interaksi REST API sebagai referensi penghubung dengan *class diagram* MVVM.

E. Development Diagram



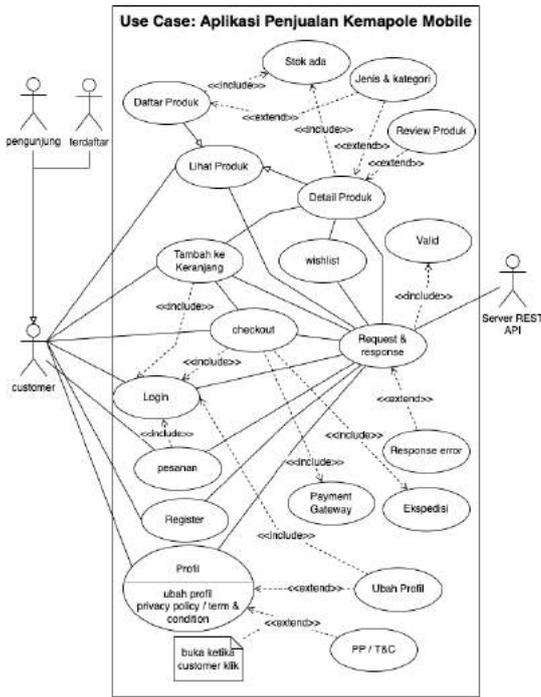
Gambar 2 Development Diagram Aplikasi.

Pada gambar 2 merupakan Development Diagram memvisualisasikan kebutuhan sistem aplikasi dari sisi teknologi. Terdapat dua *device* yang saling terhubung yaitu *smartphone* dan server. keduanya saling terhubung melalui protokol REST. Pada *smartphone device* terdapat execute environment yang menjalankan aplikasi *artifaki* yang menjalankan aplikasi *mobile*. Pada sisi server berjalan pada *Apache* web server yang memuat *Laravel* sebagai penyedia

REST API. Port tuner Laravel diubah menjadi port 80 melalui PM2 daemon process sebagai port default web server.

Pada Gambar 4 merupakan *sequence diagram* pada tab menu yang terdapat tab produk, keranjang, dan juga personal tab.

F. Use Case Diagram



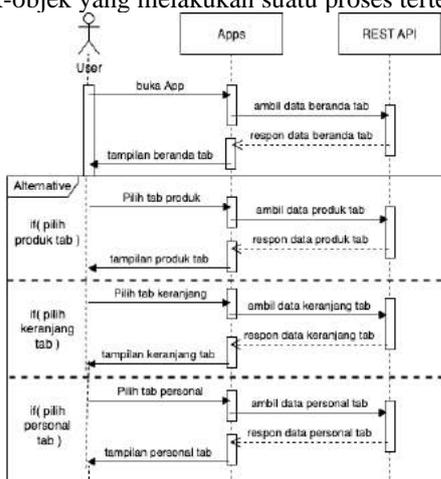
Gambar 3 Use Case Diagram Aplikasi.

Terlihat gambar 3 yang merupakan *Use Case* yang dibuat menyesuaikan kebutuhan CV. FCH Kema Pole Indonesia meliputi informasi produk, *wishlist*, hingga transaksi menggunakan Payment Gateway untuk transaksi pembayaran.

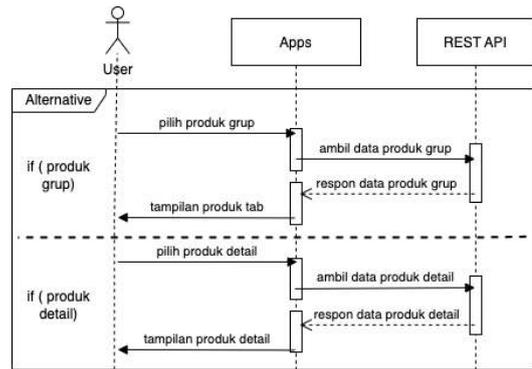
G. Sequence Diagram

Sequence diagram merupakan gambaran interaksi antar objek dan mengindikasikan hubungan antara masing-masing objek. *Sequence diagram* juga menampilkan serangkaian interaksi oleh objek-objek yang melakukan suatu proses tertentu.

Pada Gambar 6 merupakan *sequence diagram* pada tab produk yang terdapat filter produk dan dapat menampilkan produk detail.

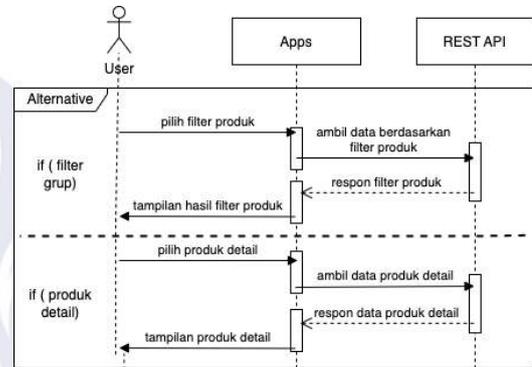


Gambar 4 Sequence Diagram Interaksi Tab.



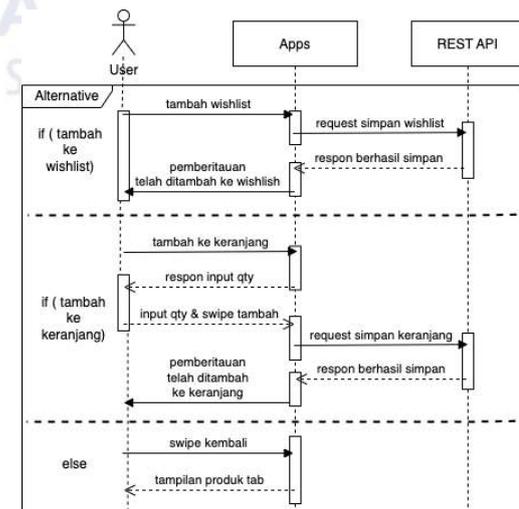
Gambar 5 Sequence Diagram Beranda Tab.

Pada gambar 5 merupakan *sequence diagram* pada beranda tab yang memuat daftar produk berdasarkan grup. Setiap produk dapat dilihat secara mendetail.



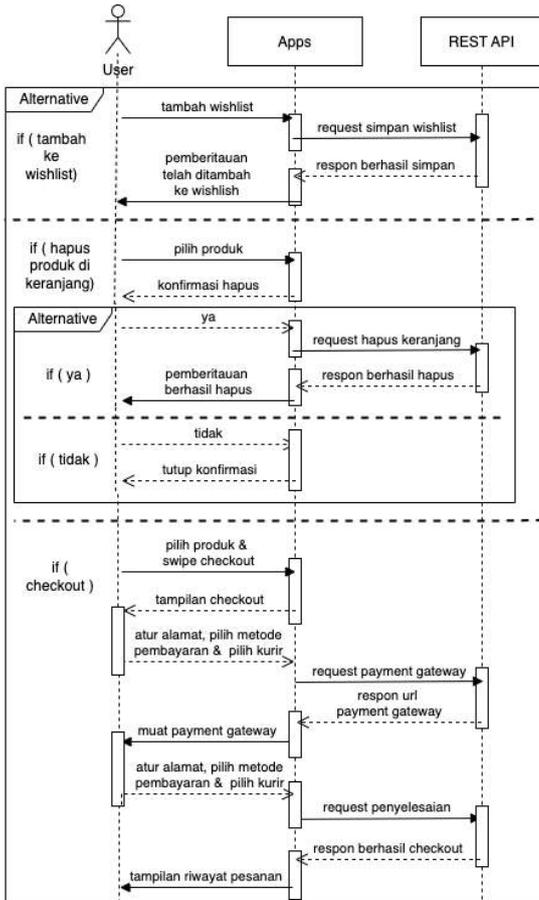
Gambar 6 Sequence Diagram Produk Tab.

Pada Gambar 7 merupakan *sequence diagram* pada tab produk detail yang terdapat interaksi pada tab *wishlist*, keranjang dan kembali ke tab sebelumnya.



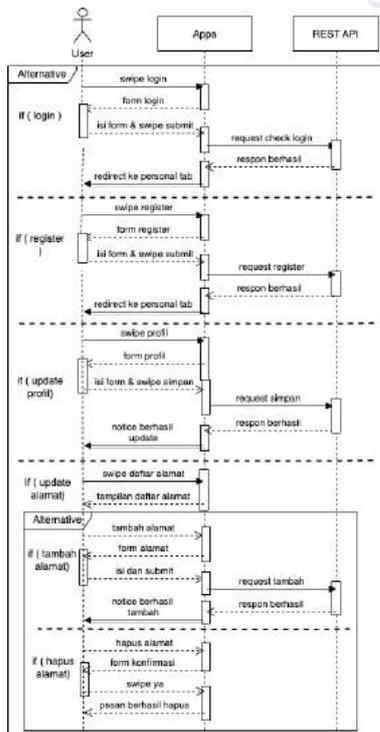
Gambar 7 Sequence Diagram Produk Detail

Pada Gambar 7 merupakan *sequence diagram* pada tab produk detail yang terdapat interaksi pada tab *wishlist*, keranjang dan kembali ke tab sebelumnya.



Gambar 8 Sequence Diagram Keranjang Tab.

Pada Gambar 8 merupakan *sequence diagram* pada tab keranjang yang terdapat interaksi pada tab *wishlist*, menu hapus produk dari keranjang, dan tab *checkout*.

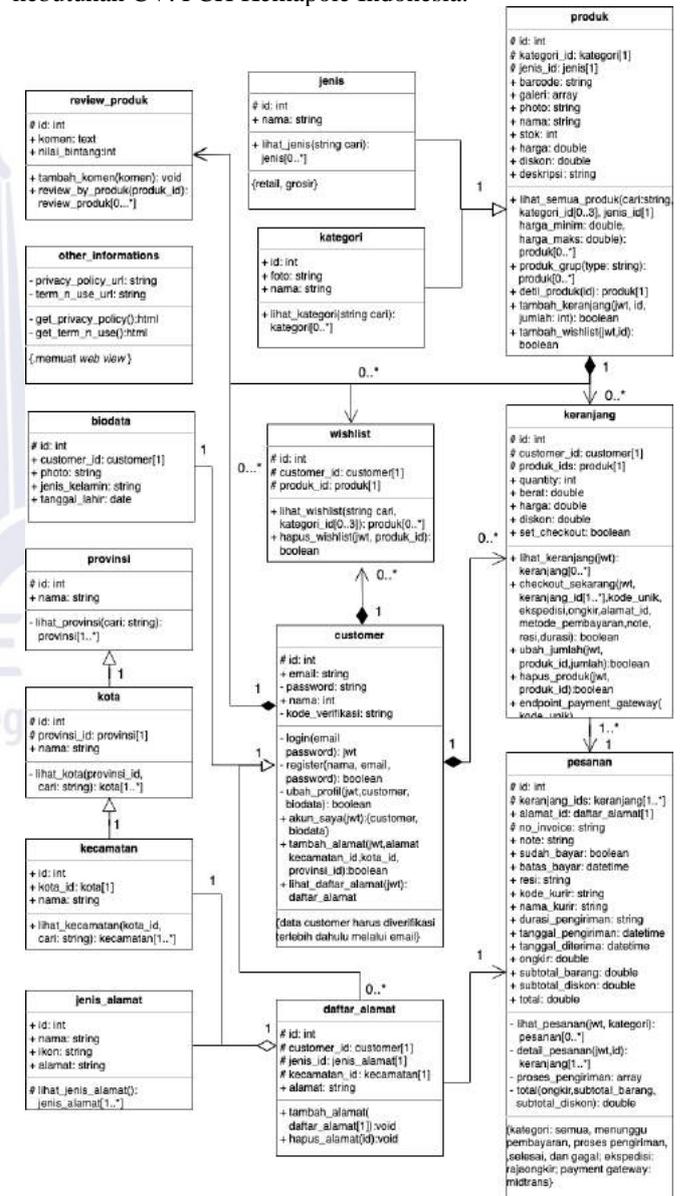


Gambar 9 Sequence Diagram Personal Tab.

Pada Gambar 9 merupakan *sequence diagram* pada tab personal yang terdapat interaksi pada tab menu *login*, *register*, *update profil*, dan menu *create update delete* alamat.

H. REST API Class Diagram

REST API digunakan supaya dari sisi *user client (mobile apps)* dapat menerima, mengolah, dan memanipulasi data dari sisi server dengan batas-batas yang diatur penyedia. Untuk aplikasi mobile CV. FCH Kemapole Indonesia, penulis telah menentukan batas-batas *endpoint* berdasarkan gambar 4 yang menggambarkan *class diagram* aplikasi. *Class diagram* menentukan *attribute*, *attribute type*, dan *operator* beserta parameternya ditentukan kedalam *perspective* sebagai substansi *domain model* dan saling terasosiasi satu sama lain dengan menentukan jenis *relation* yang terhubung. *REST API class diagram* yang dibuat oleh penulis menyesuaikan kebutuhan CV. FCH Kemapole Indonesia.



Gambar 10 REST API Class Diagram Aplikasi.

I. MVVM Class Diagram

Berbeda dengan REST API class diagram pada *MVVM class diagram* tidak memiliki *protected attribute* karena pada

pembangunan aplikasi mobile menggunakan Flutter dengan menggunakan bahasa Dart yang supported pada public attribute dan private attribute. Berikut class diagram yang disesuaikan dengan kebutuhan CV. FCH Kemapole Indonesia:

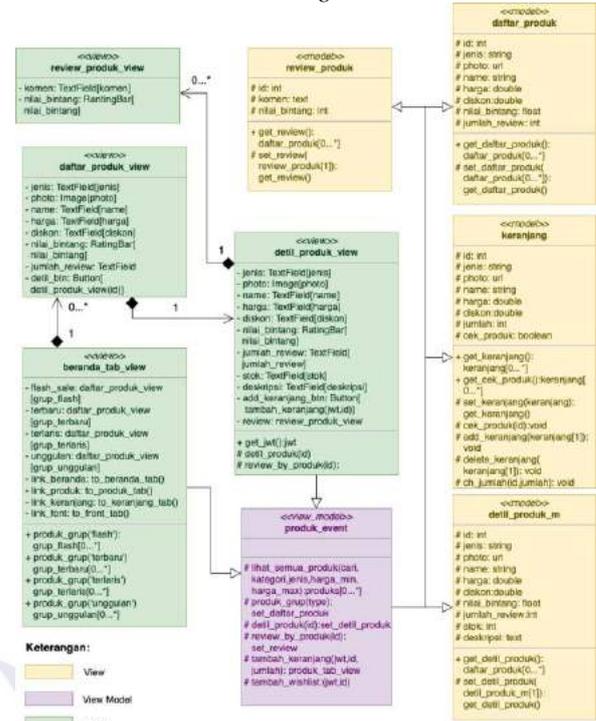
1. Personal Tab Class Diagram



Gambar 11 Personal Tab Class Diagram Aplikasi.

Pada gambar 11 merupakan class diagram untuk personal tab yang secara garis besar memiliki perspective yang berhubungan transaksi produk.

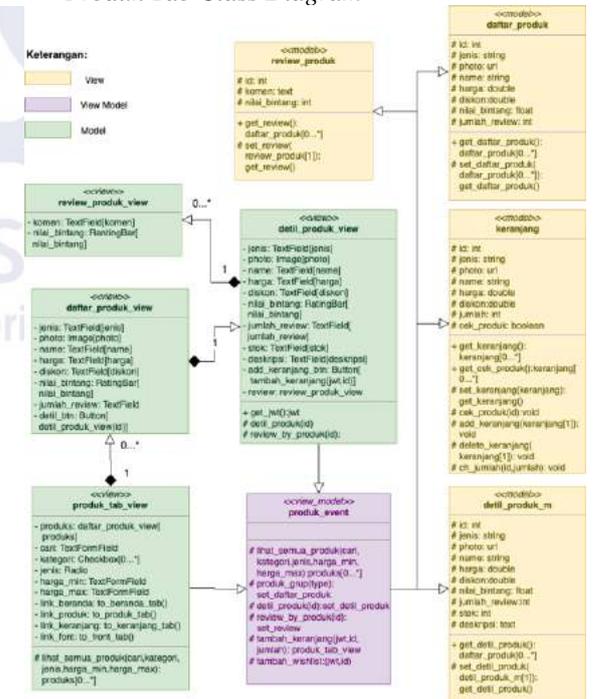
2. Beranda Tab Class Diagram



Gambar 12 Beranda Tab Class Diagram Aplikasi.

Pada gambar 12 menjelaskan beranda tab meliputi tampilan UI (User Interface) terkait klasifikasi ringkasan beberapa produk yang langsung terhubung dengan produk detail dan keranjang.

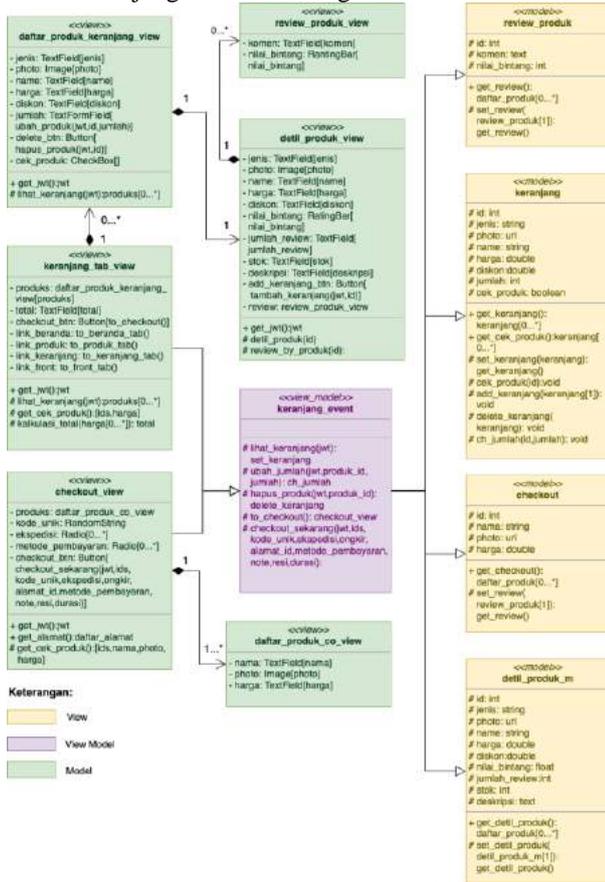
3. Produk Tab Class Diagram



Gambar 13 Produk Tab Class Diagram Aplikasi.

Pada gambar 13 menjelaskan class diagram untuk produk tab yang secara garis besar memiliki perspective yang berhubungan dengan produk dan transaksi ke keranjang.

4. Keranjang Tab Class Diagram



Gambar 14 Keranjang Tab Class Diagram Aplikasi.

Pada gambar 14 menjelaskan class diagram untuk keranjang tab yang secara garis besar memiliki perspective yang berhubungan transaksi produk.

III. HASIL DAN PEMBAHASAN

A. Hasil Akhir Aplikasi

Berdasarkan hasil perancangan sistem yang telah dibahas sebelumnya, maka proses pengembangan sistem aplikasi ini menghasilkan hasil pengembangan sebagai berikut :

1. Tampilan Home

Tampilan home pada aplikasi ini merupakan tampilan paling awal ketika pengguna membuka dan mengakses aplikasi yang telah terpasang pada smartphone pengguna.



Gambar 15 Tampilan Home.

Pada Gambar 15 di atas merupakan tampilan paling awal ketika pengguna membuka aplikasi yang telah terpasang pada smartphone mereka. Pada tampilan ini menampilkan banner yang berisikan promo yang disediakan pihak toko pada bulan atau momen tertentu. Selain itu, terdapat tampilan produk yang sedang dilakukan flash sale.

2. Tampilan Daftar Produk

Tampilan daftar produk merupakan tampilan yang menyajikan daftar produk sesuai yang dicari pengguna pada kolom "Cari Produk".

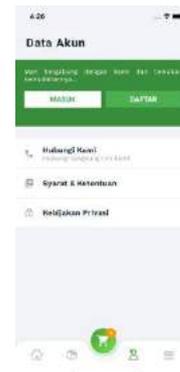


Gambar 16 Tampilan Daftar Produk.

Pada Gambar 16 menunjukkan tampilan daftar produk yang menyajikan beberapa produk yang sesuai dengan pencarian dari pengguna. Pada 1 tampilan layar, menampilkan 4 produk dilengkapi dengan gambar, judul, harga, dan juga rating dari produk tersebut.

3. Tampilan Tab Akun Sebelum Login

Ketika pengguna ingin memulai menjalankan transaksi pembelian seperti menambah produk ke dalam wishlist, keranjang, maupun melakukan transaksi pembayaran, maka pengguna diharuskan untuk melakukan proses login terlebih dahulu.



Gambar 18 Tampilan Daftar Produk.

Gambar 18 di atas merupakan tampilan tab akun ketika pengguna belum melakukan proses login pada aplikasi.

4. Tampilan Login dan Register

Pada Gambar 19 merupakan form untuk pengguna melakukan proses login. Kolom yang disediakan terdapat email dan password bagi pengguna yang telah memiliki akun terdaftar pada aplikasi. Pengguna juga dapat melakukan proses login dengan menggunakan fitur "Masuk Dengan Google" dan juga

“Masuk Dengan Apple” yang nantinya data akun akan tetap tercatat pada database aplikasi. pengguna juga dapat melakukan pendaftaran dengan menekan tombol register.



Gambar 19 Tampilan Login.

5. Tampilan Tab Akun Setelah Login



Gambar 20 Tampilan Tab Akun Setelah Login.

Pada Gambar 20 menampilkan tampilan tab akun bagi pengguna yang telah melakukan proses login. Berisikan status transaksi yang telah dilakukan, daftar alamat, wishlist, serta tab logout.

6. Tampilan Detail Produk

Tampilan detail produk dapat diakses secara keseluruhan ketika pengguna telah melakukan proses login, proses yang dapat dilakukan selanjutnya meliputi : memasukkan produk ke dalam keranjang, menambahkan produk ke dalam wishlist, dan juga melakukan pembelian secara langsung.



Gambar 21 Tampilan Detail Produk.

Pada Gambar 21 merupakan tampilan dari detail produk yang sedang dilihat oleh pengguna. Pada halaman ini menyajikan detail produk berupa deskripsi penjelasan tentang produk tersebut, stok produk yang tersedia, berat bersih dari produk,

serta minimal pembelian produk untuk mendapatkan harga retail.

7. Tampilan Keranjang



Gambar 22 Tampilan Detail Keranjang.

Pada Gambar 22 menampilkan tampilan pada tab keranjang yang berisikan produk-produk yang sebelumnya telah pengguna tambahkan ke keranjang mereka. Pengguna dapat melakukan aktivitas seperti menambah produk ke dalam wishlist, mengubah jumlah kuantitas produk yang ingin dibeli, maupun menghapus produk dari keranjang. Pengguna juga dapat memilih ulang produk yang ingin dibeli dengan mencentang masing-masing produk yang ingin dibeli.

8. Tampilan Checkout



Gambar 23 Tampilan Checkout.

Pada Gambar 23 merupakan tampilan checkout ketika pengguna ingin melakukan pembelian produk yang telah dipilih sebelumnya. Pengguna dapat menambahkan catatan tambahan dan juga pengisian alamat tujuan pengiriman. Pada “Opsi Pengiriman” pengguna dapat memilih jasa ekspedisi pengiriman sesuai selera. Pengguna juga dapat memilih opsi pembayaran dengan sistem COD (Cash On Delivery) yang kemudian mengisi alamat penagihan yang tersedia.

9. Tampilan Rincian Pesanan



Gambar 24 Tampilan Rincian Pesanan.

Pada Gambar 24 merupakan tampilan beberapa metode pembayaran yang dapat dipilih oleh pengguna sesuai yang diinginkan. Terdapat opsi pembayaran melalui beberapa bank, credit/debit card, maupun melalui gopay.



Gambar 26 Tampilan Wishlist.

Pada Gambar 26 menampilkan rincian pesanan dari produk yang sebelumnya telah dipesan. Menampilkan status pembayaran, daftar pesanan meliputi produk, dan juga alamat tujuan pengiriman.

B. Hasil Uji Coba

pada Hasil Uji Coba dilakukan dengan metode *Black-box* dengan menentukan skenario klasifikasi yang memuat empat aspek yaitu: *Functionality*, *Reliability*, *Usability* dan *efficiency*. Untuk menentukan apakah hasil dari pengujian layak atau tidak penulis melibatkan 13 responden yang terdiri dari pihak CV. FCH Kemapole Indonesia dan konsumennya untuk melakukan uji coba yang dituangkan dalam kuesioner. Dalam menentukan hasil dari responden penulis menggunakan persentase kelayakan dengan rumus seperti berikut:

$$\frac{\text{Nilai Aktual}}{\text{Nilai Ideal}} \times 100$$

10. Tampilan Riwayat Pesanan



Gambar 25 Tampilan Riwayat Pesanan.

Pada Gambar 25 merupakan tampilan beberapa metode pembayaran yang dapat dipilih oleh pengguna sesuai yang diinginkan. Terdapat opsi pembayaran melalui beberapa bank, credit/debit card, maupun melalui gopay.

11. Tampilan Riwayat Pesanan

Setelah nilai diperoleh, maka selanjutnya dilakukan proses konversi berdasarkan klasifikasi yang dapat dilihat pada tabel I berikut ini.

TABEL I
 HASIL KUESIONER UJI COBA.

Nilai (%)	Kategori Kelayakan
<21	Sangat tidak layak
21 - 40	Tidak layak
41 - 60	Cukup
61 - 80	Layak
81 - 100	Sangat Layak

Hasil dari pada uji coba yang didapatkan dari koresponden dapat dilihat pada tabel II. Berdasarkan hasil total dari perhitungan kelayakan secara keseluruhan dapat dimasukkan dalam kriteria sangat layak.

TABEL II
 HASIL KUESIONER UJI COBA.

Aspek	Nilai Aktual	Nilai Ideal	Nilai Aktual (%)	Kriteria
Functionality	61	75	93.84%	Sangat layak

Aspek	Nilai Aktual	Nilai Ideal	Nilai Aktual (%)	Kriteria
Reliability	53	75	70.66%	Layak
Usability	114	130	87.69%	Sangat layak
Efficiency	63	75	96.92%	Sangat layak
Total	291	355	81.97%	Sangat layak

IV. KESIMPULAN

Berdasarkan penelitian yang dilakukan oleh penulis sesuai yang diuraikan diatas dapat ditarik kesimpulan seperti berikut ini.

1. Identifikasi kebutuhan aplikasi dapat diperoleh dari penyesuaian aplikasi-aplikasi *E-commerce* yang ada di Indonesia dengan kebutuhan pihak CV. FCH Kema Pole Indonesia diiringi dengan *feedback* konsumen.
2. Metode OOAD dapat menjadi alternatif ketika membuat desain aplikasi mobile dengan menerapkan pola arsitektur MVVM.
3. Pengujian aplikasi dilakukan dengan menggunakan Black-box untuk setiap fungsi pada *aplikasi mobile CV. FCH Kema Pole Indonesia* dengan output nilai kelayakan secara menyeluruh termasuk dalam kriteria sangat layak.

V. SARAN

Adapun saran dari penulis untuk pengembangan penelitian kedepan seperti berikut ini.

1. Pada penelitian ini penulis masih menerapkan pengujian secara manual dengan menggunakan Black-box. Untuk pengembangan aplikasi jarak panjang disarankan melakukan pengujian secara otomatis agar tidak mengulang uji coba yang telah dilakukan.
2. Penelitian yang dilakukan oleh penulis tidak termasuk penerapan *UI (User Interface)* secara *Best Practices*. disarankan menerapkan metode yang cukup matang seperti *UCD (User Centered Design)*.

REFERENSI

- [1] A. D. Indriyanti, I. G. L. E. Putra, D. R. Prehanto, I. K. D. Nuryana, and A. Wiyono, "Development Of Mapping Area Software For Dismissal People Affected By Covid-19," 2020 Third International Conference on Vocational Education and Electrical Engineering (ICVEE), 2020.
- [2] H. K. Dauri, "BENTUK TANGGUNGJAWAB NEGARA MENGHADAPI COVID-19 DALAM PERSEPTIF OTONOMI DAERAH (Telaah Peraturan Pemerintah Nomor 21 Tahun 2020 tentang Pembatasan Sosial Berskala Besar)," SUPREMASI: Jurnal Pemikiran, Penelitian Ilmu-ilmu Sosial, Hukum dan Pengajarannya, vol. 15, no. 2, 2020.
- [3] Survey Penggunaan TIK 2017 Serta Implikasinya Terhadap Aspek Sosial Budaya Masyarakat", Pusat Penelitian dan Pengembangan Aplikasi Informatika dan Informasi dan Komunikasi Publik, Jakarta, 2017.
- [4] R. S. Wazlawick, Object-oriented analysis and design for information systems: modeling with UML, OCL, and IFML. Waltham (Mass.): Morgan Kaufmann, 2014.
- [5] Fitri Purwaningtias, "E-COMMERCE PENJUALAN BERBASIS METODE OOAD (Studi Kasus : PT Musi Utama Bercahaya Palembang)," Jurnal Cendikia, vol. XV, no. 2018, pp. 1-5, Apr. 2018.
- [6] S. Rahayu and L. Nurlani, "Sistem Informasi Kuisisioner Evaluasi Kinerja Dosen Menggunakan Metode OOAD," JTERA (Jurnal Teknologi Rekayasa), vol. 4, no. 1, p. 69, 2019.
- [7] B. Liu, "Research on Server In System Based on Mobile Client", Procedia Computer Science, vol. 154, pp. 750-753, 2019. Available: 10.1016/j.procs.2019.06.117.

[8] H. Septiansyah and D. Prehanto, "Rancang Bangun Sistem Pemesanan Menu Café "KopiRide" menggunakan Metode SAW dan QR-Code berbasis Android", Journal of Emerging Information Systems and Business Intelligence, vol. 02, no. 03, 2021, pp. 38-46, 2021.

[9] B. Kowal, T. Schreier, J. Dauer and T. Helikar, "Programmatic access to logical models in the Cell Collective modeling environment via a REST API", Biosystems, vol. 139, pp. 12-16, 2016. Available: 10.1016/j.biosystems.2015.11.005.

[10] D. Ferdiansyah and D. Prehanto, "RANCANG BANGUN APLIKASI SAHABATANI (PENJUALAN KOMODITAS TANI) BERBASIS WEB MENGGUNAKAN FRAMEWORK LARAVEL", Jurnal Manajemen Informatika, vol. 12, no. 1, pp. 1-8, 2021.

[11] M. Nurudin, W. Jayanti, R. Saputra, M. Saputra and Y. Yulianti, "Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Teknik Boundary Value Analysis", Jurnal Informatika Universitas Pamulang, vol. 4, no. 4, p. 143, 2019. Available: 10.32493/informatika.v4i4.3841.

[12] T. Hidayat and M. Muttaqin, "Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis", Jurnal Teknik Informatika UNIS, vol. 6, no. 1, pp. 25-29, 2018.