

Rancang Bangun Aplikasi *E-Commerce* Berbasis *Android* Pada Toko Suryamart Menggunakan *Framework Flutter*

Adilla Syafira Putri^a, Ade Eviyanti^b, Hindarto^c

^aInformatika, Sains Dan Teknologi, Universitas Muhammadiyah Sidoarjo, 201080200081@umsida.ac.id

^bInformatika, Sains Dan Teknologi, Universitas Muhammadiyah Sidoarjo, adeeviyanti@umsida.ac.id

^cInformatika, Sains Dan Teknologi, Universitas Muhammadiyah Sidoarjo, hindarto@umsida.ac.id

Submitted: 12-06-2023, Reviewed: 26-07-2023, Accepted 03-07-2023
<https://doi.org/10.47233/jteksis.v5i3.851>

Abstract

Suryamart is one of the business entities of the Directorate of Business Development and Investment (DPBI) UMSIDA engaged in retail. Suryamart not only serves the needs of its internal buyers - employees and lecturers, but also the needs of the general public. In carrying out its business, the author found that many general people still feel unfamiliar with Suryamart stores. On the other hand, it was found that there were problems related to the obstruction of the ordering process because internal shoppers were too busy to come to the store to shop in person, so they were less interested in shopping. This research was conducted with the aim to help admins manage online orders, make it easier for shoppers to shop, and increase Suryamart's interaction with the wider general public. The author uses waterfall software development method to minimize errors and speed up the development process because it follows a sequential process starting from analysis, design, implementation, testing, and maintenance. The result of this study is an e-commerce system consisting of applications for users and admins. Based on the results of black box testing, it can be concluded that the functional features of the application can run well.

Keywords: Suryamart, Application, Waterfall

Abstrak

Suryamart merupakan salah satu badan usaha Direktorat Pengembangan Bisnis dan Investasi (DPBI) UMSIDA yang bergerak di bidang ritel. Suryamart tidak hanya melayani kebutuhan pembeli internalnya - karyawan dan dosen, tetapi juga kebutuhan masyarakat umum. Dalam pelaksanaan bisnisnya, penulis menemukan banyak masyarakat umum yang masih merasa asing dengan toko Suryamart. Di sisi lain, ditemukan adanya masalah terkait terhambatnya proses pemesanan karena pembeli internal terlalu sibuk untuk datang ke toko untuk berbelanja secara langsung, sehingga mereka kurang tertarik untuk berbelanja. Penelitian ini dilakukan dengan tujuan untuk membantu admin mengelola pemesanan secara *online*, memudahkan pembeli untuk berbelanja, serta meningkatkan interaksi Suryamart dengan masyarakat umum yang lebih luas. Penulis menggunakan metode pengembangan perangkat lunak *waterfall* untuk meminimalisir kesalahan dan mempercepat proses pengembangan karena mengikuti proses yang berurutan dimulai dari analisis, desain, implementasi, pengujian, dan pemeliharaan. Hasil dari penelitian ini adalah sebuah sistem *e-commerce* yang terdiri dari aplikasi untuk *user* dan admin. Berdasarkan hasil pengujian *black box*, dapat disimpulkan bahwa fitur fungsional aplikasi dapat berjalan dengan baik.

Keywords: Suryamart, Aplikasi, Waterfall.

This work is licensed under Creative Commons Attribution License 4.0 CC-BY International license



PENDAHULUAN

Era globalisasi baru dimulai dengan kontribusi digital yang meningkat terutama pada sektor perekonomian. Teknologi dan infrastruktur yang canggih memudahkan dan mempercepat ekonomi digital, mengubah penawaran dan permintaan pada sektor ekonomi, termasuk pemasaran, pembelian, distribusi produk, dan sistem pembayaran.[1]. Saat ini, jual-beli barang dapat dilakukan dengan mudah melalui sistem digital yang disebut *e-commerce*. Istilah *e-commerce* berasal dari Bahasa Inggris yaitu *electronic* dan *commerce* yang memiliki arti perdagangan secara elektronik[2]. Perdagangan elektronik dapat melibatkan transfer uang elektronik, pertukaran data elektronik, sistem manajemen inventaris otomatis, dan sistem pengumpulan data otomatis[3].

Kehadiran *e-commerce* tidak hanya meliberalisasi pasar impor dan ekspor. *E-commerce* tidak hanya berpengaruh pada kebiasaan konsumsi dan gaya hidup masyarakat, tetapi juga menciptakan peluang baru bagi para wirausahawan yang diharapkan dapat memperluas lapangan kerja di masa mendatang.

Penggunaan *e-commerce* diperkirakan akan terus meningkat seiring dengan perkembangan teknologi yang juga didukung oleh data statistik pengguna smartphone dan internet. Pada tahun 2020, 89% dari total penduduk Indonesia adalah pengguna smartphone, sedangkan pada 31 Juli 2022, menurut Internet World Stats, sekitar 76,3 persen penduduk Indonesia menggunakan Internet[4]. Menurut sebuah laporan, pasar *e-commerce* Indonesia akan menghasilkan

pendapatan sekitar 29 miliar dolar pada tahun 2020. Ini adalah peningkatan yang signifikan hampir 10 miliar dolar dibandingkan tahun 2019. Hal ini kemungkinan besar disebabkan oleh perubahan perilaku konsumen selama pandemi covid-19. Statista Digital Market Outlook memprediksi bahwa pendapatan *e-commerce* akan terus meningkat dan mencapai sekitar 56 miliar dolar pada tahun 2025[5].

E-commerce telah banyak diterapkan di berbagai bidang industri. Pengembangan fitur didalam *e-commerce* dapat disesuaikan berdasarkan kebutuhan perusahaan dan permasalahan yang sedang dihadapi. Beberapa penelitian terdahulu yang telah menerapkan konsep *e-commerce* yaitu pada penelitian [6], konsep *e-commerce* diterapkan pada penjualan sayur tradisional yang menghubungkan pedagang sayur keliling dengan pembeli. Penelitian [7] menerapkan konsep *e-commerce* pada penjualan peralatan kebutuhan rumah tangga. Kemudian pada penelitian [8] konsep *e-commerce* diterapkan pada penjualan hasil pertanian jeruk yang melibatkan pengguna admin, petani, dan konsumen.

Dengan kata lain, banyak pelaku bisnis yang berusaha mengintegrasikan *e-commerce* ke dalam teknologi mereka agar tetap kompetitif dan menjaga bisnis tetap hidup [9]. Dengan menggunakan *e-commerce*, perusahaan dapat meningkatkan kepuasan pelanggan dengan membeli produk berkualitas dengan harga terbaik [10]. Selain itu, *e-commerce* dapat membantu memasarkan produk dan meraih jangkauan target pasar yang tinggi karena kecepatan penyebaran informasi melalui internet. Hal ini dapat menjadi pendukung bagi Suryamart untuk dikenal oleh pembeli eksternal. Pembeli eksternal Suryamart merupakan bagian luar civitas akademik UMSIDA atau masyarakat umum yang berbelanja di Suryamart. Keunggulan penggunaan *e-commerce* lainnya yaitu jika dibandingkan dengan penjualan tradisional, *e-commerce* menawarkan efisiensi dan efektivitas kepada pembeli yang sibuk dan tidak memiliki waktu untuk datang dan membeli produk yang mereka cari atau butuhkan. Dengan berbelanja melalui *e-commerce* pembeli akan menghemat waktu dan tenaga [11]. Hal tersebut berkaitan dengan kendala yang sedang dihadapi oleh Suryamart. Penulis menemukan bahwa pembeli internal Suryamart, yang merupakan civitas akademik UMSIDA, termasuk dosen dan pegawai tidak memiliki waktu luang untuk datang dan berbelanja karena kesibukan mereka.

Oleh karena itu, dalam penelitian ini penulis mengajukan sebuah solusi berupa aplikasi *e-commerce* berbasis *android* untuk pengembangan toko Suryamart. Dalam penelitian ini, penulis menggunakan metode pengembangan perangkat

lunak *waterfall* karena dapat meminimalisir kesalahan yang dapat terjadi selama proses pengembangan, sehingga tercipta sistem dengan kualitas yang baik.[12]

Aplikasi *E-commerce* ini merupakan solusi pengembangan lebih lanjut dari sistem informasi yang sudah ada. Aplikasi *e-commerce* yang akan diterapkan terdiri dari dua versi yaitu aplikasi untuk *user* dan admin. Pada aplikasi *e-commerce* admin terdapat fitur untuk menyimpan dan mengelola informasi seperti sistem terdahulu yakni, informasi produk mengenai berat, stok, harga barang, dan detail lainnya namun disajikan dalam bentuk *e-catalog*. Selain itu terdapat fitur untuk mengatur pengiriman, melihat detail pembelian, mencetak nota pembelian, dan mengatur pesanan yang masuk dari pembeli. Fitur *e-catalog* juga terdapat pada aplikasi *user*, sehingga *user* dapat melihat informasi katalog produk yang tersedia secara *realtime* dan dapat melakukan transaksi belanja dengan fleksibel. Aplikasi *user* juga dilengkapi dengan fitur *chatt* yang langsung mengarahkan *user* menuju kontak admin jika sedang membutuhkan informasi lebih lanjut.

Perbedaan penelitian ini dengan penelitian sebelumnya adalah pada penggunaan *framework* untuk pengembangan aplikasi. Penulis menggunakan *framework multi-platform flutter*. *Framework flutter* merupakan *template* pengembangan aplikasi yang memiliki performa tinggi dan tampilan yang menarik. Aplikasi yang dibangun menggunakan *framework flutter* dapat mudah dikembangkan melalui lintas *platform* seperti *android*, *ios*, dan *web*. *Flutter* juga bersifat dinamis yang berarti komponen pengkodean dapat dikustomisasi desainnya sehingga tampilan menjadi *user friendly* dan tidak kaku. Dalam struktur pengembangan penulis juga menggunakan aturan *state management* yang dapat meningkatkan performa penggunaan aplikasi. Hal ini telah dibuktikan pada penelitian [13] yang menyimpulkan bahwa penggunaan *state management* dalam pengembangan aplikasi menggunakan *framework flutter* dapat meningkatkan kinerja CPU, *memory*, dan waktu eksekusi kode.

METODE PENELITIAN

2.1. Alat dan Bahan Penelitian

Pada dasarnya, bahan penelitian yang menjadi subjek utama adalah sampel data *master* barang suryamart. Proses pelaksanaan penelitian menggunakan dua perangkat keras yaitu *smartphone android* dan laptop dengan spesifikasi RAM 12 GB, *Processor Intel Core i3*, dan SSD 512GB. Selain itu, penulis juga menggunakan perangkat lunak *Android Studio* dan *Framework Flutter* untuk melakukan proses pengkodean.

Android Studio adalah *tools Integrated Development Environment (IDE)* resmi hasil kolaborasi antara *Google* dan *Jetbrains* yang dibuat khusus untuk pengembangan aplikasi android. Didalamnya terdapat banyak fitur unggulan yang dapat mempermudah proses pengembangan aplikasi diantaranya, mengubah kode secara langsung untuk mengupdate tampilan di emulator dan perangkat fisik secara real time, integrasi *GitHub* untuk membantu membuat fitur aplikasi umum dan mengimpor kode sampel, kustomisasi *Android Virtual Device (Emulator)* untuk menjalankan dan men-debug aplikasi di *Android Studio* dan lain-lain[14].

Flutter adalah kerangka kerja *open source* yang dibuat oleh *Google* untuk membangun aplikasi *multi-platform* yang indah dan berkinerja tinggi. *Flutter* menawarkan tiga keunggulan dalam pengembangan sistem yakni kecepatannya dalam melakukan proses *runtime(fast)*, kemudahan memperbarui kode dengan fitur *hot reload(productive)*, dan memberikan kebebasan pengembang untuk membuat *custom design (flexible)*[15].

2.2. Metode Pengumpulan Data

Penulis melakukan pengumpulan data dengan teknik diantaranya:

A. Studi Literatur

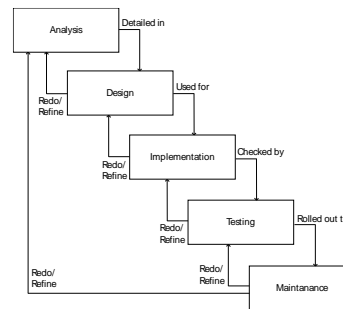
Studi literatur ini dilakukan dengan cara mengumpulkan referensi yang akan digunakan untuk landasan penelitian. Pengumpulan referensi ini dapat dilakukan dengan membaca buku, conference paper, jurnal ilmiah dalam maupun luar negeri, serta dengan melakukan penelusuran di internet.

B. Wawancara

Penulis mengumpulkan data yang akan digunakan sebagai bahan penelitian dengan mengajukan beberapa pertanyaan kepada pihak *Suryamart* terkait dengan proses pengembangan sistem.

2.2 Metode Pengembangan Sistem

Penelitian ini menggunakan pendekatan model pengembangan sistem *waterfall* yang terbagi dalam 5 proses yang dilakukan secara berurutan yaitu, Analisis, Desain, Implementasi, Testing dan Maintenance.



Gambar 2.1 Model *Waterfall*

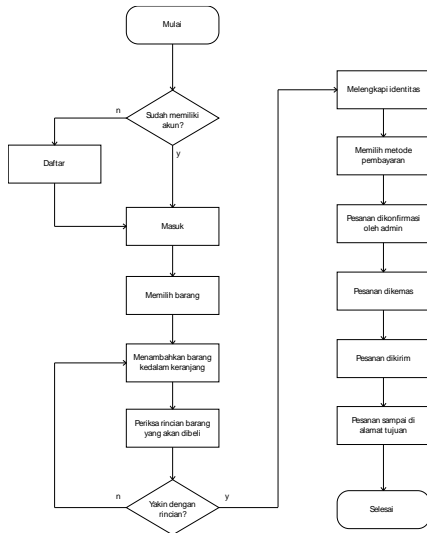
1. Pada tahap analisis, peneliti menganalisis sistem kerja toko *Suryamart* menyangkut user yang terlibat, alur dokumen, proses dan prosedur transaksi serta permasalahan yang sedang dihadapi. Kemudian dilakukan identifikasi terhadap kebutuhan fungsionalitas sistem yang akan dibangun.
2. Selanjutnya, pada tahap desain hasil dari analisis digambarkan melalui diagram sebagai representasi *blue print software* sebelum dilakukannya *coding*. Peneliti membagi desain menjadi 3 bagian yaitu, desain proses, desain basis data, dan desain antar muka.
3. Kemudian Desain yang telah dibuat diimplementasikan kedalam sistem menggunakan bahasa pemrograman *dart* dan *framework flutter*.
4. Peneliti melakukan pengujian yang terdiri dari *unit test* dan *widjet test*. Testing ini bertujuan untuk memverifikasi kebenaran sebuah logika aplikasi dalam berbagai kondisi dan memverifikasi bahwa *UI widjet* terlihat dan berperilaku seperti yang diharapkan.
5. Pada tahap maintenance, akan dilakukan beberapa modifikasi, perbaikan, dan penyempurnaan yang sesuai berdasarkan monitoring penggunaan.

2.3 Desain Proses

Peneliti merancang sistem dengan menggunakan:

A. Flowchart

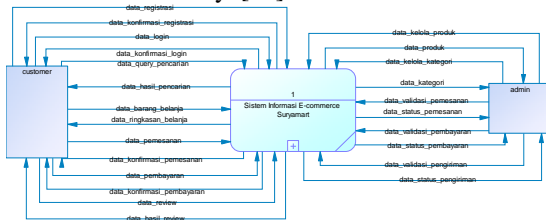
Flowchart adalah Diagram yang menggambarkan alur kerja sebuah program[16]. Berikut ini merupakan alur kerja sistem aplikasi *e-commerce*.



Gambar 2.2 Flowchart Sistem E-commerce

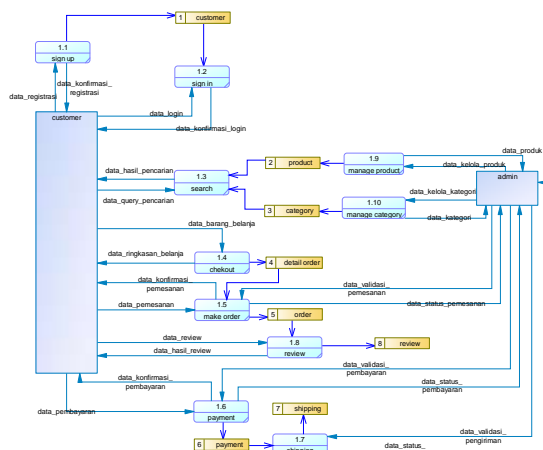
B. DFD

DFD adalah suatu langkah atau metode untuk membuat sebuah perancangan sistem yang mana berorientasi pada alur data yang bergerak ke sebuah sistem lainnya[17].



Gambar 2.3 DFD Context

Context Diagram (CD) atau biasa disebut dengan DFD level 0 merupakan level tertinggi dari Data Flow Diagram (DFD) yang menggambarkan seluruh input ke sistem atau output dari sistem.



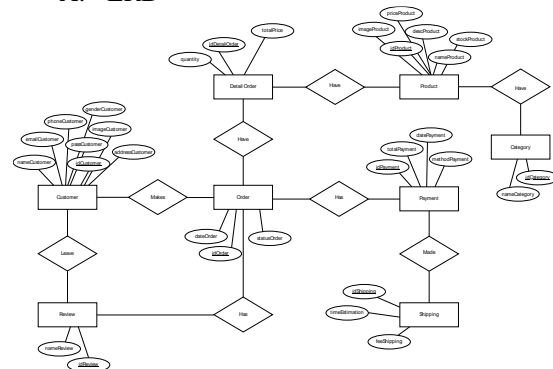
Gambar 2.4 DFD Level 1

DFD level 1 adalah *decompose* dari context diagram, dimana pada level 1 terdapat 10 proses, yaitu proses *sign-up*, *sign-in*, *search*, *checkout*, *make order*, *payment*, *shipping*, *review*, *manage product*, dan *manage category*.

2.4 Desain Basis Data

Basis data adalah basis data atau *database* adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis untuk memperoleh informasi dari basis data tersebut. Perancangan basis data akan dibagi menjadi 3 tahapan yaitu *Conceptual Design*, *Logical Design*, dan *Physical Design*. Tahap *conceptual design* fokus pada model data yang akan digunakan tanpa memikirkan logika-logika penyimpanan database dan pertimbangan fisik database tersebut. *Output* dari *conceptual design* adalah diagram ERD. Tahap kedua adalah menerjemahkan *Conceptual Design* menjadi *Logical Design*. Terdapat proses validasi model data apakah telah sesuai dengan logika-logika *database*. Tahap terakhir yaitu *physical design*. PDM merupakan model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data-data tersebut. Penggambaran rancangan PDM memperlihatkan struktur penyimpanan data yang benar pada basis data yang digunakan sesungguhnya.

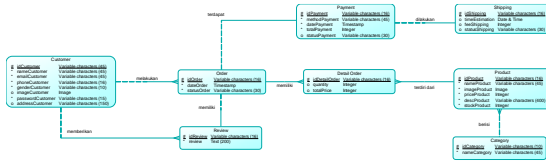
A. ERD



Gambar 2.5 ERD Sistem E-commerce

Dalam diagram diatas terdapat 8 entitas yang memiliki atribut dan saling berelasi. Atribut digambarkan dengan bentuk oval, entitas digambarkan dengan bentuk segi empat. Entitas saling terhubung dengan bentuk *diamond* yang menggambarkan sebuah relasi. Entitas *customer* memiliki relasi dengan *order* dan *review*. Entitas *review* memiliki relasi dengan *customer* dan *order*. Entitas *order* memiliki relasi dengan entitas *customer*, *product*, *payment*, dan *review*. Entitas *product* memiliki relasi dengan *category* dan *order*. Entitas *detail order* memiliki relasi dengan *order* dan *product* Entitas *shipping* memiliki relasi terhadap entitas *payment*.

B. CDM



Gambar 2.6 CDM Sistem E-commerce

Keterangan lebih detail entitas masing-masing sebagai berikut:

1. Entitas Customer

Jenis relasi terhadap entitas *order* adalah *one to many* yang artinya setiap *customer* dapat melakukan banyak pemesanan dan dalam tiap satu pemesanan hanya terdapat satu *customer*. Jenis relasi terhadap entitas *review* adalah *one to one* yang artinya setiap kali transaksi *customer* dapat memberikan satu kali *review* dan dalam *review* hanya terdapat satu *customer*.

2. Entitas Product

Jenis relasi terhadap entitas *category* adalah *many to one* yang artinya tiap *product* hanya memiliki satu *category* dan tiap *category* dapat memiliki banyak *product*. Jenis relasi terhadap entitas *DetailOrder* adalah *one to many* yang artinya tiap *product* dapat dimiliki oleh banyak *idDetailOrder* dan tiap *idDetailOrder* memiliki satu *idProduct*.

3. Entitas Category

Jenis relasi terhadap entitas *product* adalah *one to many* yang artinya setiap *idCategory* terdapat banyak *idProduct* dan tiap satu *product* hanya memiliki satu *idCategory*.

4. Entitas Order

Jenis relasi terhadap entitas *customer* adalah *many to one* yang artinya tiap pemesanan hanya memiliki satu *idCustomer* dan tiap *customer* dapat melakukan banyak pemesanan. Jenis relasi terhadap entitas *DetailOrder* adalah *one to many* yang artinya setiap *order* memiliki minimal 1 atau banyak *idDetailOrder* dan setiap *idDetailOrder* hanya memiliki 1 *idOrder*. Jenis relasi terhadap entitas *Payment* adalah *one to one* yang berarti setiap satu *idOrder* hanya memiliki satu *id payment* dan tiap satu *idPayment* hanya terdapat satu *id order*. Jenis relasi terhadap entitas *Review* adalah *one to one* yang artinya satu *idOrder* dapat memiliki minimal 0 *review* dan maksimal 1 *review* dan tiap satu *idReview* hanya terdapat satu *idOrder*.

5. Entitas Detail Order

Jenis relasi terhadap entitas *Order* adalah *many to one* yang artinya tiap *idDetailOrder* hanya memiliki satu *idOrder* dan tiap satu *idOrder* dapat memiliki banyak *idDetailOrder*. Jenis relasi terhadap entitas *product* adalah *many to*

one yang artinya setiap *idDetailOrder* terdiri dari satu *idProduct* dan tiap satu *idProduct* dapat memiliki banyak *idDetailOrder*.

6. Entitas Payment

Jenis relasi terhadap entitas *Order* adalah *one to one* yang artinya setiap *id payment* hanya terdapat satu *id order* dan setiap *id order* terdapat satu *id payment*. Jenis relasi terhadap entitas *shipping* adalah *one to one* yang artinya tiap *id payment* hanya memiliki satu *id shipping* dan satu *id shipping* hanya memiliki satu *id payment*.

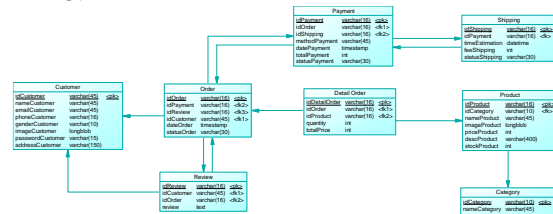
7. Entitas Review

Jenis relasi terhadap entitas *Customer* adalah *one to many* yang artinya setiap satu *id review* memiliki satu *customer* dan setiap satu *id customer* dapat memiliki banyak *id review*. Jenis relasi terhadap entitas *Order* adalah *one to one* yang artinya setiap *id review* hanya memiliki satu *id order* dan setiap *id order* hanya memiliki satu *id review*.

8. Entitas Shipping

Jenis relasi terhadap entitas *payment* adalah *one to one* yang artinya tiap *id shipping* harus terdapat satu *id payment* dan setiap *id payment* hanya terdapat satu *id shipping*.

C. PDM



Gambar 2.7 PDM Sistem E-commerce

Berdasarkan diagram diatas, terdapat 8 tabel dengan rincian sebagai berikut :

Tabel 2.1 Tabel Customer

No.	Field Name	Data Type	Field Size
1	<i>idCustomer</i>	Varchar	45
2	<i>nameCustomer</i>	Varchar	45
3	<i>emailCustomer</i>	Varchar	45
4	<i>phoneCustomer</i>	Varchar	16
5	<i>genderCustomer</i>	Varchar	10
6	<i>imageCustomer</i>	Longblob	-
7	<i>passwordCustomer</i>	Varchar	15
8	<i>addressCustomer</i>	Varchar	150

Primary key : *idCustomer*

Foreign key : -

Tabel 2.2 Tabel Product

No.	Field Name	Data Type	Field Size
1	<i>idProduct</i>	Varchar	16
2	<i>idCategory</i>	Varchar	10
3	<i>nameProduct</i>	Varchar	45
4	<i>imageProduct</i>	Longblob	-
5	<i>priceProduct</i>	Int	-
6	<i>descProduct</i>	Varchar	400
7	<i>stockProduct</i>	Int	-

Primary key : *idProduct*
 Foreign key : *idCategory*

Tabel 2.3 Tabel *Category*

No.	Field Name	Data Type	Field Size
1	<i>idCategory</i>	Varchar	10
2	<i>nameCategory</i>	Varchar	45

Primary key : *idCategory*
 Foreign key : -

Tabel 2.4 Tabel *Order*

No.	Field Name	Data Type	Field Size
1	<i>idOrder</i>	Varchar	16
2	<i>idPayment</i>	Varchar	16
3	<i>idReview</i>	Varchar	16
4	<i>idCustomer</i>	Varchar	45
5	<i>dateOrder</i>	Timestamp	-
6	<i>statusOrder</i>	Varchar	30

Primary key : *idOrder*
 Foreign key : *idCustomer, idPayment, idReview*

Tabel 2.5 Tabel *Detail Order*

No.	Field Name	Data Type	Field Size
1	<i>idOrder</i>	Varchar	16
2	<i>idProduct</i>	Varchar	16
3	<i>quantity</i>	Int	-
4	<i>totalPrice</i>	Int	-

Primary key : *idOrder*
 Foreign key : *idProduct*

Tabel 2.6 Tabel *Payment*

No.	Field Name	Data Type	Field Size
1	<i>idPayment</i>	Varchar	16
2	<i>idOrder</i>	Varchar	16
3	<i>methodPayment</i>	Varchar	45
4	<i>datePayment</i>	Timestamp	-
5	<i>totalPayment</i>	Int	-

Primary key : *idPayment*
 Foreign key : *idOrder*

Tabel 2.7 Tabel *Review*

No.	Field Name	Data Type	Field Size
1	<i>idReview</i>	Varchar	16
2	<i>idCustomer</i>	Varchar	45
3	<i>idOrder</i>	Varchar	16
4	<i>review</i>	Text	-

Primary key : *idReview*
 Foreign key : *idCustomer, idOrder*

Tabel 2.8 Tabel *Shipping*

No.	Field Name	Data Type	Field Size
1	<i>idShipping</i>	Varchar	16
2	<i>idPayment</i>	Varchar	16
3	<i>timeEstimation</i>	Date & Time	-
4	<i>feeShipping</i>	Int	-

Primary key : *idShipping*
 Foreign key : *idPayment*

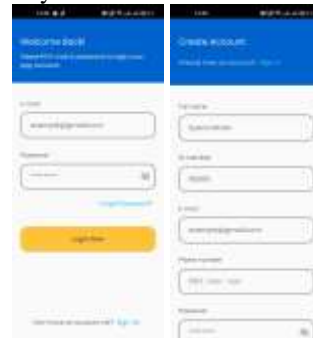
HASIL DAN PEMBAHASAN

3.1 Implementasi Sistem User

1. Halaman Login dan Register

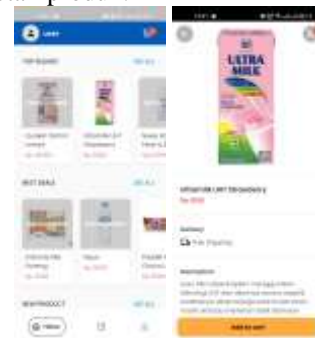
User dapat mengakses sistem setelah melakukan login, jika belum memiliki akun maka diarahkan untuk melakukan

registrasi. Proses registrasi menandakan *user* otomatis terdaftar sebagai member Suryamart.



Gambar 3.1 Tampilan *Sign-in* dan *Sign-up*

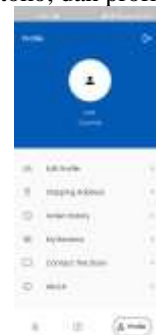
- Halaman Home dan Detail Produk
User yang berhasil masuk akan melihat halaman beranda. Ketika salah satu produk di klik maka akan menampilkan detail produk.



Gambar 3.2 Tampilan menu *home* dan detail produk

- Halaman Profil

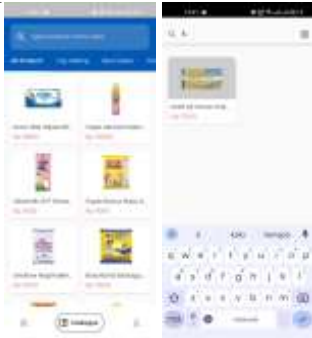
Pada menu *profile*, tersedia beberapa *sub menu* yang dapat dikontrol oleh *user*, diantaranya menambah, menghapus dan mengubah alamat pengiriman, mengubah data diri, melacak status pesanan, melihat riwayat ulasan yang telah diberikan, fitur chat admin toko, dan profil singkat toko.



Gambar 3.3 Tampilan menu *profile*

- Halaman Pencarian dan Katalog Produk
User dapat menjelajahi atau mencari produk di setiap kategori. Pada fungsi pencarian, *user* dapat memasukkan kata kunci untuk nama produk, karena

pencarian dilakukan berdasarkan nama produk.



Gambar 3.4 Tampilan menu *catalogue* dan pencarian

5. Halaman Keranjang

Pada halaman keranjang belanja, *user* dapat mengubah produk yang ada di keranjang belanja, antara lain menambah produk baru, mengubah jumlah produk dan mengeluarkan produk dari keranjang belanja.



Gambar 3.5 Tampilan halaman keranjang

6. Halaman Checkout

Pada halaman ini *user* diminta untuk memeriksa detail pesanan sebelum melakukan pemesanan. Untuk *user* baru, biasanya belum menambahkan alamat pengiriman, maka akan ada pengecekan sebelum proses cekout apakah *user* telah menambahkan alamat pengiriman. Jika belum, proses *checkout* tidak dapat dilakukan, jika sudah maka *user* akan diarahkan menuju halaman *checkout*.



Gambar 3.6 Tampilan halaman *checkout*

7. Halaman Riwayat Pembelian

Setelah menyelesaikan pemesanan, *user* memiliki opsi untuk membatalkan pesanan sebelum status pesanan "SHIPPED" atau sudah dikemas dan dalam proses pengiriman.



Gambar 3.7 Tampilan riwayat belanja

3.2 Implementasi Sistem Admin

1. Halaman Utama

Admin memiliki 5 fitur utama yaitu kelola produk, kelola kategori produk, kelola pesanan, meninjau daftar member dan *review*.



Gambar 3.8 Tampilan halaman utama admin

2. Halaman Kelola Produk

Pada halaman ini admin dapat menambahkan produk baru, menghapus, dan mengubah produk.



Gambar 3.9 Tampilan halaman kelola produk

Jika admin memilih salah satu produk maka akan menampilkan halaman detail produk yang baru saja ditambahkan.



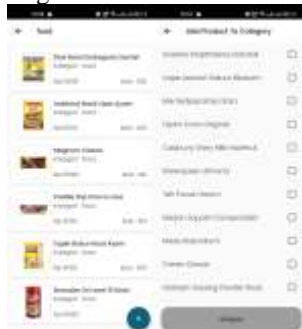
Gambar 3.10 Tampilan detail produk

- Halaman Kelola Kategori Produk
 Pada halaman ini admin dapat menambahkan kategori baru, menghapus, dan mengubah kategori. Jika kategori dihapus, produk dengan kategori tersebut akan diperbarui secara otomatis tanpa kategori.



Gambar 3.11 Tampilan halaman kelola kategori produk

- Halaman Kelola Produk Dalam Kategori
 Ketika admin mengklik salah satu kategori pada halaman kategori, daftar produk dari kategori tersebut akan ditampilkan, admin dapat menambahkan produk yang belum memiliki kategori ke dalam kategori tersebut atau menghapus produk dari kategori.



Gambar 3.12 Tampilan halaman menambahkan produk baru ke dalam kategori

- Tampilan Daftar Member
 Admin dapat meninjau member yang telah terdaftar sebagai pengguna aplikasi (*user*) pada halaman ini.



- Tampilan Daftar Review Member
 Hasil review dari semua member akan ditampilkan di halaman ini.



Gambar 3.14 Tampilan daftar review member

- Halaman Kelola Pemesanan dan Detail Pemesanan
 Pada halaman ini admin dapat melihat daftar pesanan *user*. Pesanan yang memiliki status "PACKED", memiliki arti bahwa toko harus mengemas produk yang dipesan, yang muncul di halaman detail pesanan atau faktur. Setelah pesanan siap, admin dapat mengubah status menjadi "SHIPPED".



Gambar 3.15 Tampilan halaman kelola pemesanan

- Halaman Invoice Pemesanan
 Saat menggunakan aplikasi untuk pertama kalinya, admin diminta memberikan izin akses ke sistem untuk membuka file faktur. Setelah akses diberikan, faktur dapat diunduh, didistribusikan, atau dicetak.



Gambar 3.15 Tampilan halaman invoice

SIMPULAN

Penelitian ini menghasilkan sebuah sistem *e-commerce* yang terdiri dari dua aplikasi yaitu aplikasi untuk user dan admin. Berdasarkan hasil pengujian *black box*, fungsi-fungsi pada aplikasi dapat menghasilkan output sesuai dengan yang diharapkan dan sistem berjalan sesuai dengan rancangan dan fungsinya.

Seiring berjalannya waktu, semakin banyak kebutuhan yang harus dipenuhi. Untuk penelitian lebih lanjut, penulis menyarankan untuk menambahkan fitur-fitur yang dapat mendukung efisiensi penggunaan, seperti notifikasi, metode pembayaran dan pengiriman tambahan, penyimpanan item favorit dan pembaruan kapasitas dan keamanan penyimpanan data.

DAFTAR PUSTAKA

- [1] A. L. Kusumatriana *et al.*, *Statistik E-Commerce 2021*. Badan Pusat Statistik, 2021.
- [2] A. Prasetyo *et al.*, *Konsep Dasar E-Commerce*. Yayasan Kita Menulis, 2021. [Online]. Available: <https://books.google.com.my/books?id=xt8hEAAAQBAJ>
- [3] H. Harmayani, D. Marpaung, A. Hamzah, N. Mulyani, and J. Hutahaean, *E-Commerce: Suatu Pengantar Bisnis Digital*. Yayasan Kita Menulis, 2020. [Online]. Available: <https://books.google.com.my/books?id=DLjwDwAAQBAJ>
- [4] "ASIA INTERNET USE, POPULATION STATISTICS DATA AND FACEBOOK DATA - MID-YEAR 2022," 2022. <https://www.internetworldstats.com/stats3.htm> (accessed Nov. 13, 2022).
- [5] "Retail e-commerce revenue in Indonesia from 2017 to 2025 (in billion U.S. dollars)," *Statista Research Department*, 2021. <https://www.statista.com/forecasts/280925/e-commerce-revenue-forecast-in-indonesia> (accessed Nov. 13, 2022).
- [6] D. T. Absari, Liliana, and Bambang Prijambodo, "E-Greengrocers: A Mobile Information System for Supporting Business in The Traditional Market," *Inf. J. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 7, no. 2, pp. 120–124, 2022, doi: 10.25139/inform.v7i2.4674.
- [7] R. Hisham and Heru Wijayanto Aripardono, "Perancangan Dan Pengembangan Web Marketplace Kebutuhan Rumah Tangga Menggunakan Model Wdlc Dengan Metode Scrum," *J. Ilm. Betrik*, vol. 13, no. 1, pp. 75–86, 2022, doi: 10.36050/betrik.v13i1.424.
- [8] Y. Bustomi and S. Rahayu, "Pemanfaatan Teknologi Informasi untuk Penjualan Hasil Pertanian Jeruk di Kabupaten Garut Menggunakan Pendekatan Extreme Programming," *JRST (Jurnal Ris. Sains dan Teknol.*, vol. 7, no. 1, p. 51, 2023, doi: 10.30595/jrst.v7i1.15501.
- [9] J. Martin and A. R. Tanaamah, "Perancangan Dan Implementasi Sistem Informasi Penjualan Berbasis Desktop Website Menggunakan Framework Bootstrap Dengan Metode Rapid Application Development, Studi Kasus Toko Peralatan Bayi 'Eeng Baby Shop,'" *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 1, p. 57, 2018, doi: 10.25126/jtiik.201851547.
- [10] A. M. Almaududi Ausat, E. S. Astuti, and W. Wilopo, "Analisis Faktor yang Berpengaruh pada Adopsi E-Commerce dan Dampaknya Bagi Kinerja UKM di Kabupaten Subang," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 2, p. 333, 2022, doi: 10.25126/jtiik.2022925422.
- [11] E. Elmayati and R. Kurniasari, "Perancangan Aplikasi Dashboard Sri (System Of Rice Intensification) Untuk Hasil Pertanian Padi Di Kecamatan Sumber Harta Berbasis Web," *J. Ilm. Betrik*, vol. 10, no. 02, pp. 55–62, 2019, doi: 10.36050/betrik.v10i02.14.
- [12] D. Murdiani and M. Sobirin, "PERBANDINGAN METODOLOGI WATERFALL DAN RAD (RAPID APPLICATION DEVELOPMENT) DALAM PENGEMBANGAN SISTEM INFORMASI," *JINTEKS (Jurnal Inform. Teknol. dan Sains)*, vol. 4, no. 4, pp. 302–306, 2022.
- [13] J. Mantik, R. Rama Prayoga, G. Munawar, R. Jumiyani, and A. Syalsabila, "Performance Analysis of BLoC and Provider State Management Library on Flutter," *J. Mantik*, vol. 5, no. 3, pp. 1591–1597, 2021.
- [14] "Mengenal Android Studio," 2023. <https://developer.android.com/studio/intro?hl=id>
- [15] "Build apps for any screen," 2022. <https://flutter.dev/>
- [16] A. Ramadhanu, R. Husna Arsyah, N. S. Wahyuni Nengsi, and N. Nurhaliza, "Rancang Bangun Aplikasi Mobile Edukasi Mengenai Parenting Skills Bagi Orang Tua Berbasis Android Dengan Menggunakan Bahasa Pemrograman Java Android," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 1, no. 1, pp. 58–66, 2019, doi: 10.47233/jteksis.v1i1.23.
- [17] A. S. P. Aris Susanto, "Rancang Bangun Aplikasi E-Commerce Penjualan Helm Menggunakan Metode Simple Additive Weighting (Saw) (Studi Kasus: Gallery Helm Jogja)," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 4, no. 1, pp. 20–34, 2022, [Online]. Available: <http://jurnal.unidha.ac.id/index.php/jteksis/article/view/346%0Ahttp://jurnal.unidha.ac.id/index.php/jteksis/article/download/346/199>