

**RANCANG BANGUN APLIKASI MOBILE E-COMMERCE
UNTUK JUAL-BELI FURNITURE BERBASIS ANDROID
PADA TOKO WOODMAN.PROJECT**

TUGAS AKHIR



Disusun Oleh:

TAUFIQURRAHMAN

20200801031

**PROGRAM STUDI SISTEM INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS ESA UNGGUL
TAHUN 2024**

ABSTRACT

Digital transformation has driven increased mobile e-commerce application usage, particularly in the furniture industry. This research aims to design an Android-based mobile e-commerce application for Woodman.Project Store using Extreme Programming (XP) methodology. The primary focus is developing a digital solution to overcome conventional sales system limitations through an application with product search features, catalog management, and comprehensive ordering system. The application was developed using the Flutter framework, with orientation towards the Android operating system. The functional scope includes product management, search systems, order processing, and notification management. The research method adopts a systematic XP approach focusing on rapid and iterative coding. Research results demonstrate that the application successfully provides an interactive interface enabling customers to search and order furniture online. The application offers detailed product information, including photos, descriptions, materials, and dimensions, and provides price filter features to help consumers find products within their budget. The simple and intuitive design allows users without technological background to easily use the application.

Keywords: *E-commerce, Mobile Application, Android, Furniture, Extreme Programming*

ABSTRAK

Transformasi digital mendorong peningkatan penggunaan aplikasi mobile e-commerce, khususnya dalam industri furnitur. Penelitian ini bertujuan merancang aplikasi mobile e-commerce berbasis Android untuk Toko Woodman. Project menggunakan metodologi Extreme Programming (XP). Fokus utama penelitian adalah mengembangkan solusi digital yang mampu mengatasi keterbatasan sistem penjualan konvensional melalui aplikasi dengan fitur pencarian produk, manajemen katalog, dan sistem pemesanan yang komprehensif. Pengembangan aplikasi dilakukan dengan menggunakan framework Flutter, dengan orientasi pada sistem operasi Android. Lingkup fungsional aplikasi mencakup manajemen produk, sistem pencarian, pemrosesan pesanan, dan pengelolaan notifikasi. Metode penelitian mengadopsi pendekatan sistematis XP yang berfokus pada pengkodean cepat dan iteratif. Hasil penelitian menunjukkan bahwa aplikasi berhasil menyediakan antarmuka interaktif yang memudahkan pelanggan melakukan pencarian dan pemesanan furniture secara online. Aplikasi memberikan informasi detail produk, termasuk foto, deskripsi, material, dan ukuran, serta menyediakan fitur filter harga untuk membantu konsumen menemukan produk sesuai anggaran. Desain sederhana dan intuitif memungkinkan pengguna tanpa latar belakang teknologi dapat dengan mudah menggunakan aplikasi.

Kata Kunci: E-commerce, Mobile Aplikasi, Android, Furniture, Extreme Programming

DAFTAR ISI

<i>ABSTRACT</i>	ii
ABSTRAK	iii
DAFTAR ISI	iv
DAFTAR GAMBAR.....	viii
DAFTAR TABEL	x
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	2
1.3 Tujuan Tugas Akhir.....	2
1.4 Manfaat Tugas Akhir.....	3
1.5 Batasan Tugas Akhir	3
1.6 Kerangka Berfikir	4
1.7 Sistematika Penulisan Tugas Akhir.....	5
BAB II TINJAUAN PUSTAKA	6
2.1 Penelitian Terdahulu.....	6
2.1.1 Studi Literatur	8
2.2 Landasan Teori Rancang Bangun.....	10
2.3 Aplikasi	11
2.4 Flutter	12
2.5 Dart.....	14
2.6 Software Development	14
2.7 E-Commerce.....	15
2.8 Extreme Programming (XP).....	15
2.9 Teori Perancangan Basis Data.....	17

2.9.1	Entitiy Relationship Diagram (ERD).....	17
2.9.2	Transformasi ERD ke LRS	18
2.9.3	Logical Record Structure (LRS)	18
2.9.4	Normalisasi Database	19
2.10	Definisi <i>Unified Modelling Language (UML)</i>	19
2.10.1	Use Case Diagram	20
2.10.2	Activity Diagram	22
2.10.3	Sequence Diagram	24
2.10.4	Class Diagram.....	26
2.11	Aplikasi Pendukung	28
2.11.1	Android Software Development Kit (SDK)	28
2.11.2	JavaScript Object Notation (JSON)	29
2.11.3	PHP	29
2.11.4	MySQL (My Structure Query Language).....	30
2.11.5	XAMPP.....	30
2.11.6	Visual Studio Code	31
2.12	Teori Pengujian Sistem	31
2.12.1	Black Box Testing	32
BAB III METODE		33
3.1	Rencana Penelitian	33
3.2	Obyek Penelitian	34
3.3	Teknik Pengumpulan Data	35
3.3.1	Observasi	36
3.3.2	Wawancara.....	36
3.3.3	Studi Literatur	36
3.4	Metode Pengembangan Sistem	36

3.4.1	Planning	37
3.4.2	Design	38
3.4.3	Coding	39
3.4.4	Testing	41
3.5	Proses Bisnis yang Berjalan	42
3.6	Identifikasi Masalah Proses Bisnis yang Berjalan	44
3.7	Proses Bisnis Usulan	44
3.8	Analisis Kebutuhan Sistem	46
3.8.1	Kebutuhan Fungsional	46
3.8.2	Kebutuhan Non-Fungsional	47
BAB IV HASIL DAN PEMBAHASAN.....		49
4.1	Desain	49
4.1.1	Use Case Diagram	49
4.1.2	Activity Diagram	50
4.1.3	Sequence Diagram	58
4.1.4	Class Diagram.....	63
4.1.5	Perancangan Basis Data.....	65
4.2	Implementasi	67
4.2.1	Halaman Login User	67
4.2.2	Halaman Register.....	68
4.2.3	Halaman Lupa Password	68
4.2.4	Halaman Beranda.....	69
4.2.5	Halaman New Products	70
4.2.6	Halaman Detail Products	71
4.2.7	Halaman Shopping Chart.....	72
4.2.8	Halaman Product.....	73

4.2.9	Halaman Checkout.....	74
4.2.10	Halaman Place Order	75
4.2.11	Halaman Order Success	76
4.2.12	Halaman Payment Success	77
4.2.13	Halaman Profil	78
4.2.14	Halaman Order History	79
4.2.15	Halaman Logout	80
4.3	Pengujian Sistem	81
4.3.1	Black-Box Testing	82
4.3.2	System Usability Scalile (SUS)	85
BAB V PENUTUP		87
5.1	Kesimpulan.....	87
5.2	Saran.....	87
DAFTAR PUSTAKA.....		89

DAFTAR GAMBAR

Gambar 1. 1 Kerangka berfikir.....	4
Gambar 1. 2 Logo Flutter	12
Gambar 2. 1 Tahapan <i>Extreme Programming</i>	16
Gambar 2. 2 Logo PHP	29
Gambar 2. 3 Logo MySQL.....	30
Gambar 2. 4 Logo XAMPP	30
Gambar 2. 5 Logo VScode	31
Gambar 3. 1 Rencana Penelitian.....	34
Gambar 3. 2 Diagram Sistem Berjalan.....	43
Gambar 3. 3 Diagram Sistem Usulan	46
Gambar 4. 1 Usecase Diagram.....	49
Gambar 4. 2 <i>Activity Diagram Login</i>	51
Gambar 4. 3 <i>Activty Diagram Register</i>	52
Gambar 4. 4 <i>Activity Diagram Forgot Password</i>	53
Gambar 4. 5 <i>Activity Diagram Pemesanan</i>	54
Gambar 4. 6 <i>Activity Diagram Admin Login</i>	55
Gambar 4. 7 <i>Activity Diagram Admin Manage Products</i>	56
Gambar 4. 8 <i>Activity Diagram Admin Manage Order</i>	57
Gambar 4. 9 <i>Sequence Diagram Register</i>	58
Gambar 4. 10 <i>Seqeunce Diagram Login</i>	59
Gambar 4. 11 <i>Seqeunce Diagram Belanja</i>	60
Gambar 4. 12 <i>Seqeunce Diagram Pemesanan</i>	61
Gambar 4. 13 <i>Seqeunce Diagram Kelola Produk</i>	61

Gambar 4. 14 <i>Sequeunce Diagram</i> Kelola Pesanan	62
Gambar 4. 15 <i>Sequeunce Diagram</i> Kelola User	62
Gambar 4. 16 <i>Class Diagram</i> Aplikasi Woodman.....	64
Gambar 4. 17 <i>Entity Relationship Diagram (ERD)</i>	66
Gambar 4. 18 <i>Logical Record Structure (LRS)</i>	66
Gambar 4. 19 Halaman Login User.....	67
Gambar 4. 20 Halaman Pendaftaran.....	68
Gambar 4. 21 Halaman Lupa Password	69
Gambar 4. 22 Halaman Beranda	70
Gambar 4. 23 Halaman Produk Baru	71
Gambar 4. 24 Halaman Detail Produk	72
Gambar 4. 25 Halaman Shopping Chart.....	73
Gambar 4. 26 Halaman Produk	74
Gambar 4. 27 Halaman Cecheckout_Order	75
Gambar 4. 28 Halaman Place Order.....	76
Gambar 4. 29 Halaman OrderSukses	77
Gambar 4. 30 Halaman Pembayaran Berhasil.....	78
Gambar 4. 31 Halaman Profile	79
Gambar 4. 32 Halaman Order History	80
Gambar 4. 33 Halaman Logout	81

DAFTAR TABEL

Tabel 2. 1 Studi Literatur.....	8
Tabel 2. 2 Entity Relationship Diagram	17
Tabel 2. 3 Usecase Diagram	21
Tabel 2. 4 Activity Diagram	23
Tabel 2. 5 Sequence Diagram.....	24
Tabel 2. 6 Class Diagram	26
Tabel 4. 1 <i>Black-Box Testing Customer</i>	82
Tabel 4. 2 Hasil Pernyataan SUS	85
Tabel 4. 3 Skor Hasil Hitung Kuisisioner	86

BAB I

PENDAHULUAN

1.1 Latar Belakang

Transformasi digital yang pesat telah mendorong peningkatan signifikan dalam penggunaan perangkat mobile berbasis Android, khususnya dalam konteks e-commerce. Pergeseran preferensi masyarakat menuju transaksi digital mencerminkan kebutuhan akan sistem perdagangan yang lebih efisien, praktis, dan komprehensif. Dalam konteks ini, industri furnitur sebagai salah satu kebutuhan fundamental mengalami dinamika serupa, meskipun sistem perdagangan konvensional masih menghadapi berbagai keterbatasan operasional.

Toko Woodman.Project, sebagai pelaku usaha dalam industri furnitur, menghadapi tantangan strategis dalam mengoptimalkan penetrasi pasar di tengah persaingan yang semakin intensif. Sistem penjualan konvensional yang bertumpu pada interaksi fisik tidak hanya membatasi jangkauan pasar, tetapi juga menghambat aksesibilitas informasi produk yang komprehensif bagi calon konsumen. Dalam merespons tantangan tersebut, pengembangan aplikasi mobile e-commerce berbasis Android menjadi solusi strategis yang dapat mengintegrasikan kebutuhan akan efisiensi operasional dan pengalaman berbelanja yang optimal.

Dalam konteks pengembangan aplikasi, implementasi Flutter sebagai framework utama memberikan fleksibilitas lintas platform, meskipun fokus pengembangan tetap pada sistem operasi Android. Ruang lingkup fungsional aplikasi mencakup manajemen katalog produk, sistem pencarian, pemrosesan pemesanan, dan pengelolaan notifikasi. Adapun pembatasan implementasi meliputi eksklusi sistem transfer bank konvensional, dengan orientasi pada integrasi dompet digital sebagai metode pembayaran utama. Selain itu, sistem pengiriman dibatasi pada pemantauan status hingga tahap persiapan pengiriman.

Metodologi pengembangan aplikasi mengadopsi pendekatan sistematis, dimulai dari identifikasi permasalahan sistem konvensional, perumusan solusi

teknologi, analisis kebutuhan, perancangan sistem, implementasi, hingga evaluasi komprehensif untuk memastikan kesesuaian solusi dengan kebutuhan pengguna.

Sejalan dengan evolusi teknologi dan preferensi konsumen, pengembangan aplikasi mobile e-commerce untuk toko Woodman.Project merepresentasikan inovasi strategis dalam optimalisasi proses bisnis. Melalui implementasi **"RANCANG BANGUN APLIKASI MOBILE E-COMMERCE UNTUK JUAL-BELI FURNITURE BERBASIS ANDROID PADA TOKO WOODMAN.PROJECT"**, diharapkan dapat tercipta efisiensi operasional yang lebih tinggi serta perluasan jangkauan pasar yang signifikan.

1.2 Identifikasi Masalah

Berdasarkan latar belakang permasalahan diatas, maka permasalahan yang diangkat dalam bahasan penelitian ini adalah sebagai berikut:

1. Bagaimana meningkatkan aksesibilitas aplikasi mobile untuk berbelanja aksesoris rumah bagi masyarakat dengan kendala finansial atau keahlian teknologi?
2. Bagaimana mengatasi ketidakpercayaan konsumen terhadap produk online yang disebabkan oleh ketidakmampuan melihat produk secara langsung sebelum pembelian?
3. Apa strategi efektif untuk mengatasi tantangan navigasi dan penggunaan aplikasi, terutama terkait dengan tata letak yang kompleks?

1.3 Tujuan Tugas Akhir

Adapun tujuan tugas akhir yang ingin dicapai adalah sebagai berikut:

1. Melakukan analisis dan identifikasi kebutuhan serta preferensi pengguna terkait aksesoris rumah guna memastikan bahwa aplikasi dapat menampilkan produk sesuai dengan harapan pasar sasaran.
2. Merancang sistem keamanan transaksi yang kuat dan dapat dipercaya untuk meningkatkan rasa aman pengguna saat bertransaksi, termasuk penerapan metode pembayaran yang aman.

3. Meneliti potensi pasar aksesoris rumah dan merancang strategi untuk memperluas cakupan aplikasi, baik secara lokal maupun regional, guna mencapai pangsa pasar yang lebih luas.

1.4 Manfaat Tugas Akhir

Manfaat dari tugas akhir ini dapat menghasilkan beberapa aspek positif, antara lain:

1. Mempermudah akses bagi masyarakat dengan keterbatasan finansial atau keterampilan teknologi melalui antarmuka yang ramah pengguna namun menarik, sehingga aplikasi dapat dijangkau oleh beragam lapisan masyarakat.
2. Mengatasi ketidakpercayaan konsumen terhadap produk online dengan menyajikan informasi produk yang lengkap dan valid, termasuk gambar, deskripsi, serta ulasan dari pengguna lain, untuk memperkuat kepercayaan konsumen.
3. Menyediakan beragam kualitas, model, harga, dan ukuran barang untuk mencakup seluruh segmen masyarakat, sebagaimana yang dilakukan oleh IKEA dalam versi terjangkaunya.
4. Memberikan rekomendasi produk berdasarkan riwayat pembelian konsumen untuk memudahkan konsumen dalam menemukan barang yang sesuai untuk melengkapi pembelian sebelumnya.

1.5 Batasan Tugas Akhir

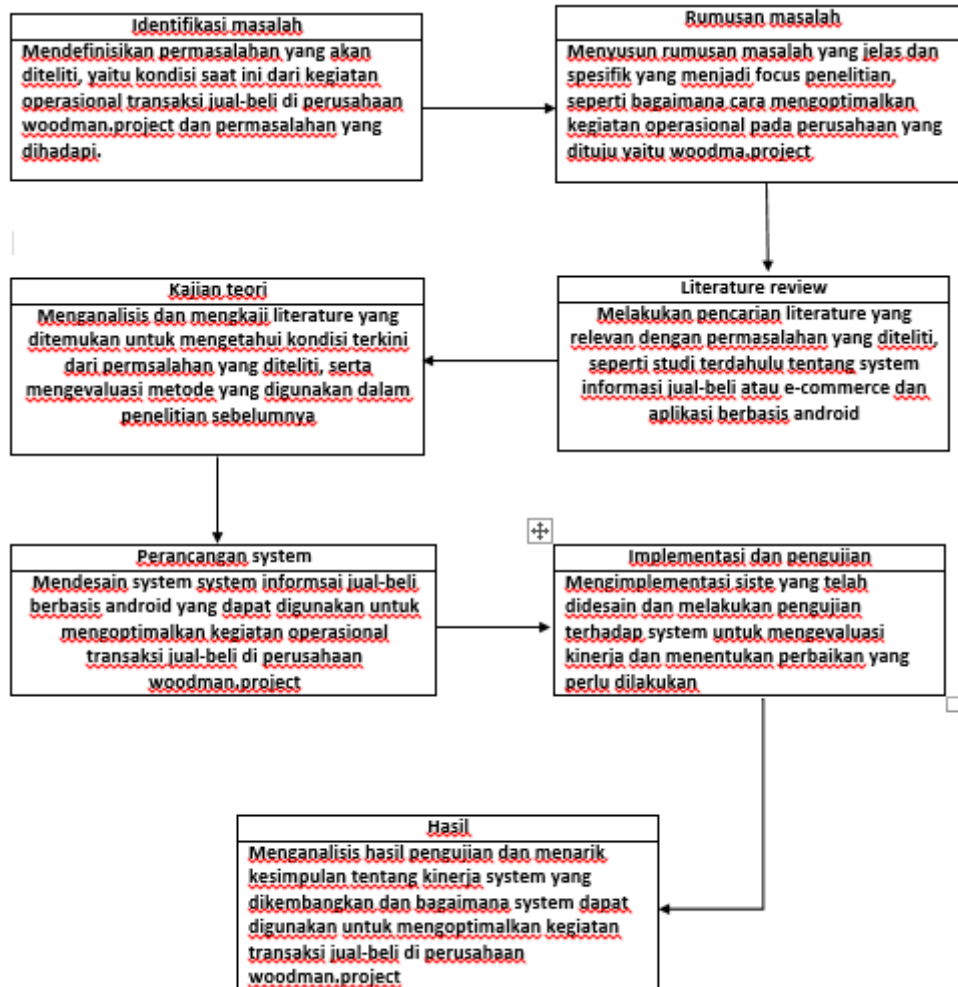
Beberapa batasan untuk penelitian mengenai rancang bangun aplikasi jual-beli aksesoris rumah berbasis mobile Android di toko Woodman. Project adalah sebagai berikut:

1. Fokus analisis penelitian terbatas pada aksesoris rumah saja, tanpa melibatkan barang atau produk lain dalam platform jual-beli aksesoris rumah.

2. Penelitian ini secara eksklusif menggunakan platform Android untuk mengakses aplikasi mobile yang dikembangkan.
3. Penelitian difokuskan pada toko Woodman.Project dengan wilayah penelitian terbatas pada kecamatan Panongan dan Cikupa.

1.6 Kerangka Berfikir

Kerangka berpikir merupakan struktur yang digunakan untuk menggambarkan bagaimana peneliti akan menyelesaikan masalah yang diteliti. Hal ini dimaksudkan untuk memandu pelaksanaan penelitian agar hasil yang diperoleh tidak menyimpang dari tujuan sebelumnya.



Gambar 1. 1 Kerangka berfikir

1.7 Sistematika Penulisan Tugas Akhir

Gambaran secara garis besar mengenai hal-hal yang akan dibahas dalam laporan penelitian dapat dilihat dalam sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang, identifikasi masalah, tujuan penelitian, batasan, penelitian, manfaat penelitian, dan sistematika penelitian.

BAB II DASAR TEORI

Bab ini menjelaskan tentang pembahasan mengenai dasar teori yang memperkuat kaitan dengan penyusunan penelitian yang dilakukan.

BAB III METODOLOGI PENELITIAN

Bab ini melakukan analisa alternatif mengenai sistem yang akan di aplikasikan menjadi mobile. Sehingga diharapkan aplikasi tersebut berfungsi seperti yang diinginkan dan akan dilakukan pengembangan prototipe aplikasi sistem operasi yang akan digunakan nantinya

BAB IV HASIL DAN PEMBAHASAN

Bab ini menjelaskan mengenai implementasi pengembangan aplikasi serta pengujian dan Analisa seluruh sistem terhadap hasil yang telah di ujikan.

BAB V KESIMPULAN DAN SARAN

Bab ini merupakan bab penutup yang berisi kesimpulan dari sistem yang telah dikembangkan serta saran untuk penelitian lanjutan..

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Dalam penelitian terdahulu ini, penelitian ini mencatat sejumlah penelitian terdahulu yang relevan dengan topik penelitian. Penelitian-penelitian tersebut melibatkan analisis dan evaluasi yang mendalam terhadap perkembangan terkini dalam bidang yang menjadi fokus penelitian ini dan sebagai pembanding, sebagai berikut:

1. Penelitian (Afriansyah et al., 2022)

Penelitian (Afriansyah et al., 2022) dengan judul penelitian “Rancang Bangun Website E-commerce di Toko Sean Shoes Menggunakan Metode Rapid Application Development”. Pada penelitian ini menggunakan model siklus metode RAD (Rapid application Development) untuk pengembangan perangkat lunak. Penelitian ini mempunyai tujuan untuk membuat sebuah sistem yang dapat mempermudah pengelola toko Sean Shoes dalam mempromosikan produknya, mempermudah transaksi antara pelanggan dan pemilik toko, meningkatkan penjualan dan daya saing sebagai wujud memajukan usaha di Toko Sean Shoes dengan menggunakan metode pengembangan RAD. Maka hasil yang dicapai oleh penulis adalah sebuah sistem penjualan online di toko Sean Shoes berbasis web. Berdasarkan hasil dapat disimpulkan sistem ini dapat mempermudah promosi produk yang terdapat di toko Sean Shoes serta memperluas jangkauan pemasaran dan menjadikan proses pembelian atau pemesanan barang bagi pelanggan menjadi lebih mudah dan fleksibel dari segi waktu dan tempat.

2. Penelitian (Samsuni & Erfiyani, 2018)

Penelitian (Samsuni & Erfiyani, 2018) dengan judul penelitian “RANCANG BANGUN APLIKASI E-COMMERCE PENJUALAN

PRODUK KECANTIKAN DAN FASHION PADA AC FASHION STYLE”. Pada penelitian ini Metode perancangan sistem e-commerce menggunakan pendekatan Waterfall dengan pendekatan berorientasi objek. Penelitian ini mempunyai tujuan untuk meminimalkan waktu proses penjualan dengan tujuan dapat meningkatkan volume penjualan sehingga pendapatan Toko AC Fashion Style dapat meningkat, serta mempermudah dalam mencari barang yang diinginkan dan dapat memberikan hasil pencarian dengan cepat. Penelitian ini berhasil merancang, membangun, dan menguji website penjualan produk berdasarkan metode User Centered Design (UCD). Dan dari data hasil analisis kebutuhan, perancangan sistem dan pengujian kepada sejumlah calon pengguna (responden) website ecommerce penjualan produk telah berhasil memenuhi kriteria sesuai dengan prinsip User Centered Design (UCD). Website dikatakan efektif dan efisien jika rata-rata 90%. Rata-rata pengerjaan task adalah 20 detik, error terendah yang terjadi terhadap suatu task yang sedang dikerjakan sebanyak 3 kali, dan penekanan tombol back di browser sebanyak 3 kali. Kemudian target tertinggi untuk matriks preferensi adalah 85% dan pengguna merasa puas terhadap situs penjualan online ini.

3. Penelitian (Y. B. Prasetya, 2024)

Penelitian (Y. B. Prasetya, 2024) dengan judul “Rancang Bangun Sistem Informasi E-commerce Untuk Toko Dykara Berbasis Website”. Pada penelitian ini penulis menggunakan metode RAD (rapid application development) untuk pengembangan perangkat lunak. Penelitian ini dibuat dengan tujuan mengembangkan toko pakaian e-commerce untuk memudahkan penjual dalam menjual pakaian dan pembeli dalam mencari informasi produk pakaian dengan lebih mudah dan proses transaksi yang lebih aman melalui wire transfer antar bank. Dari penelitian ini dapat disimpulkan bahwa sistem ini dapat membantu pembeli dalam proses pencarian properti dan dapat mempercepat transaksi karena sudah menggunakan payment gateway.

2.1.1 Studi Literatur

Dalam mengembangkan penelitian ini, pendekatan yang diambil adalah dengan dengan memanfaatkan kerangka kerja literature review sebagai alat evaluasi dan analisis yang mendalam untuk mengkategorikan dan menilai sumber data serta informasi umum yang relevan dengan konteks penelitian.

Tabel 2. 1 Studi Literatur

No	Peneliti & Judul	Summary	Metode
1.	(Putra & Putra, 2020), “Rancang Bangun Aplikasi E-Commerce Untuk Usaha Penjualan Helm”	Perancangan dan pembangunan sebuah website yang mendukung aplikasi e-commerce pada toko Edi Helm Bandar Lampung untuk memudahkan transaksi jual-beli dan dilakukan dengan menerapkan metode pengembangan perangkat lunak	<i>Extreme Programming.</i>
2.	(Putri et al., 2023), “Rancang Bangun Aplikasi E-Commerce Berbasis Android Pada Toko Suryamart Menggunakan Framework Flutter”	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.	<i>Waterfall</i>
3.	(Fitriyana & Muslihudin, 2022)	Dengan adanya mobile commerce berbasis android dengan cara online pemasaran produk hasil olahan tani dapat	<i>Prototyping</i>

	<p>“Mobile Commerce Pemasaran Produk Olahan Kelompok Wanita Tani (KWT) Flamboyan Berbasis Android Di Desa Kalirejo”</p>	<p>lebih memudahkan dalam memasarkan produk hasil tani lebih memperluas ke daerah-daerah luar desa. Dengan menggunakan M-commerce ini banyak fitur-fitur yang bisa digunakan supaya produk itu terlihat menarik di konsumen. Konsumen juga dalam membeli produk tidak perlu datang ketempat untuk melakukan pembelian dan dalam melakukan pembayaran bisa menggunakan credit card dan cash on delivery (COD). Didalam sistem M-commerce banyak menu halaman yang sudah tersedia untuk memudahkan penjual maupun konsumen untuk melakukan transaksi penjualan maupun pembelian. Dengan adanya aplikasi mobile android pemakai bisa mengakses kapan saja lewat gadgetnya.</p>	
4.	<p>(Ikhwan & Aslami, 2023)</p> <p>“Rancangan Aplikasi Mobile Apotik Online Untuk Membantu Masyarakat Di Masa Pandemi”</p>	<p>Aplikasi apotik online yang telah dirancang dan dibuat untuk mempermudah masyarakat membeli obat dan mengurangi aktifitas di luar rumah agar terhindar dari kerumunan pada masa pandemic ini serta aplikasi ini dapat menjadi salah satu</p>	<p><i>Prototype</i></p>

		alternatif para pengusaha apotik yang ingin menggunakan system dalam penjualan bisnisnya. Aplikasi ini mempermudah untuk mengetahui informasi obat dan detail obat yang diinginkan.	
5.	(Aditya et al., 2022) “Rancang Bangun Aplikasi Penjualan Berbasis Android (Studi Kasus: Pada Toko Murah Jaya Alumunium)”	Merancang dan membangun aplikasi penjualan dengan adroid pada Murah Jaya Alumunium dengan menggunakan metode Extrame Proqramming yang terdiri dari empat tahapan yaitu planning, design, coding, dan test. Dari metode tersebut jadilah sebuah aplikasi android yang bisa membantu proses penjualan lebih efektif dan efisien. Sedangkan untuk Membantu mengetahui produk secara lengkap pada Murah Jaya Alumunium bisa dengan aplikasi Murah Jaya Alumunium dan melakukan transaksi secara online tanpa perlu datang ke toko langsung.	<i>Extreme Programming</i>

2.2 Landasan Teori Rancang Bangun

Perancangan sistem merupakan proses sistematis dalam mentransformasikan hasil analisis ke dalam bahasa pemrograman, dengan tujuan mendeskripsikan secara terperinci implementasi setiap komponen sistem. Sementara itu,

pembangunan sistem mencakup aktivitas pengembangan sistem baru atau penyempurnaan sistem yang telah ada, baik secara parsial maupun menyeluruh.

Konsep rancang bangun memiliki korelasi yang erat dengan perancangan sistem, di mana keduanya membentuk kesatuan integral dalam proses pengembangan aplikasi. Perancangan sistem memiliki dua objektif utama: pertama, mengakomodasi kebutuhan pengguna sistem, dan kedua, menyediakan dokumentasi perancangan yang komprehensif dan terstruktur. Kedua aspek tersebut terutama berorientasi pada elaborasi detail perancangan sistem yang akan menjadi landasan dalam pengembangan program komputer.

Berdasarkan elaborasi tersebut, dapat disintesis bahwa rancang bangun sistem merupakan serangkaian aktivitas yang mencakup proses interpretasi hasil analisis dan implementasi atau optimalisasi sistem, baik dalam konteks pengembangan sistem baru maupun penyempurnaan sistem yang telah ada (Syadza et al., 2018).

2.3 Aplikasi

Aplikasi adalah program yang telah siap digunakan dan dapat dieksekusi oleh pengguna untuk menjalankan perintah-perintah tertentu. Tujuan utama dari pembuatan aplikasi adalah mendapatkan hasil yang lebih akurat sesuai dengan keperluan atau keinginan yang mendasari pembuatan aplikasi tersebut. (Geograf, 2023). Dengan kata lain, aplikasi merupakan alat terapan yang memungkinkan pengguna untuk melakukan tugas khusus dengan efisiensi dan ketepatan. Jadi, aplikasi adalah solusi pemrograman yang memfasilitasi pemrosesan data sesuai dengan kebutuhan pengguna.

Pengertian aplikasi juga mencakup aplikasi seluler yang semakin populer dalam beberapa tahun terakhir. Aplikasi seluler adalah aplikasi yang dirancang khusus untuk dijalankan pada perangkat seluler, seperti *smartphone* dan *tablet*. Dengan adanya aplikasi seluler, pengguna dapat dengan mudah mengakses berbagai layanan dan fitur yang disediakan oleh aplikasi tersebut, seperti media sosial, permainan, *e-commerce*, dan lain sebagainya.

2.4 Flutter

Flutter adalah teknologi sumber terbuka yang dikembangkan oleh Google untuk membuat aplikasi seluler lintas platform. Flutter memiliki kemampuan untuk membuat aplikasi yang indah, cepat, dan andal (Alfriansyah1 dkk., 2023). Flutter memiliki sifat dinamis, yang memungkinkan komponen kode dapat disesuaikan dengan desain untuk menciptakan tampilan yang ramah pengguna dan fleksibel. Dalam struktur pengembangan, penulis mengadopsi aturan manajemen state yang dapat meningkatkan performa aplikasi. Berdasarkan sumber yang dibaca, penggunaan manajemen state dalam pengembangan aplikasi dengan Flutter dapat meningkatkan kinerja CPU, memori, dan waktu eksekusi kode. (Evanemran, 2023).



Gambar 1. 2 Logo Flutter

Flutter adalah *framework* yang memungkinkan pengembangan aplikasi *mobile* lintas platform dengan mudah. Beberapa kelebihan dari Flutter meliputi:

1. Pengembangan aplikasi menjadi lebih efisien dan cepat berkat Flutter. Framework ini menyediakan user interface widgets yang telah dirancang oleh Google dengan kualitas kode yang tinggi. Dengan demikian, proses pengembangan aplikasi dapat dipercepat tanpa harus menulis kode dari awal.
2. Tampilan antarmuka pengguna (UI) yang menarik adalah salah satu keunggulan Flutter. Pengembang dapat memanfaatkan berbagai widget

yang telah disediakan untuk merancang tata letak, animasi, tema, jenis huruf, navigasi, dan tampilan gulir dengan lebih mudah dan efisien.

3. Meskipun Flutter adalah framework multiplatform, hasil aplikasi yang dihasilkan memiliki performa yang hampir setara dengan aplikasi native. Keunggulan ini disebabkan oleh penggunaan kombinasi kode dari mesin render canggih dan komponen aplikasi native.
4. Efisiensi Biaya: Flutter, sebagai framework untuk pembuatan aplikasi mobile multiplatform, memungkinkan pengembangan aplikasi menjadi lebih efisien. Pengembang tidak perlu mengembangkan aplikasi secara terpisah untuk setiap platform yang berbeda.

Selain memiliki kelebihan yang sangat menguntungkan, tentu saja setiap Framework memiliki kekurangan juga. Berikut adalah kekurangan yang dimiliki oleh Flutter diantaranya:

1. *Library* dari Pihak Ketiga Belum Banyak: Saat ini, ketersediaan *library* pihak ketiga pada Flutter masih terbatas dibandingkan dengan platform pengembangan native lainnya. Pengembang mungkin perlu mengatasi keterbatasan ini dengan mengandalkan solusi internal atau mengembangkan sendiri komponen yang tidak tersedia.
2. Ukuran Aplikasi yang Dihasilkan Berukuran Besar: Meskipun aplikasi yang dikembangkan dengan Flutter menunjukkan kinerja yang stabil, namun, mereka cenderung memiliki ukuran file yang besar.
3. Menggunakan Bahasa Pemrograman Dart yang Asing: Flutter menggunakan Bahasa pemrograman Dart, yang mungkin belum familiar bagi sebagian besar pengembang. Pengembang yang baru mengenal Dart perlu meluangkan waktu untuk mempelajari sintaks dan konsep Bahasa ini.

Meskipun demikian, penggunaan Flutter tetap memiliki manfaat yang signifikan, terutama dalam hal pengembang lintas platform dan desain tampilan yang menarik. (Coding Studio Team, 2023).

2.5 Dart

Dart adalah sebuah bahasa pemrograman serbaguna yang dapat digunakan untuk mengembangkan berbagai jenis aplikasi dan program, termasuk aplikasi web, aplikasi seluler, aplikasi desktop, dan server. Bahasa ini dirancang sebagai bahasa pemrograman klien yang dioptimalkan untuk pengembangan cepat aplikasi web dan seluler. Dart dapat dijalankan pada berbagai sistem operasi seperti Windows, Linux, Unix, dan MacOS, karena merupakan bahasa lintas platform atau platform independen (Nabila Fauziah, 2022)

Awalnya dikembangkan oleh Google, Dart kemudian diakui sebagai standar oleh Ecma. Dart saat ini digunakan secara luas dalam membangun aplikasi web, server, desktop, dan seluler. Bahasa ini mendukung berbagai paradigma pemrograman, termasuk imperatif, fungsional, reflektif, dan berorientasi objek. Dart mengadopsi konsep-konsep pemrograman berorientasi objek seperti kelas, pewarisan, abstraksi, enkapsulasi, dan polimorfisme.

Salah satu fitur kuat dari Dart adalah kemampuannya dalam mengelola sampah secara otomatis melalui fitur pengumpul sampah (*garbage collector*). Selain itu, Dart juga merupakan bahasa pemrograman open source yang dilisensikan di bawah *BDS (Berkeley Software Distribution License)*. Sintaksis Dart didasarkan pada sintaksis gaya-C yang sederhana.

2.6 Software Development

Pengembangan perangkat lunak (*Software Development*) adalah proses sistematis yang bertujuan untuk menghasilkan aplikasi berkualitas tinggi. Dalam pengembangan perangkat lunak, komunikasi antar tim developer sangat penting. Tujuan utamanya adalah memastikan hasil yang optimal dan berkualitas. Dengan kata lain, pengembangan adalah kegiatan programmer dalam menciptakan aplikasi yang memudahkan berbagai kegiatan dan memberikan hasil yang baik.

2.7 E-Commerce

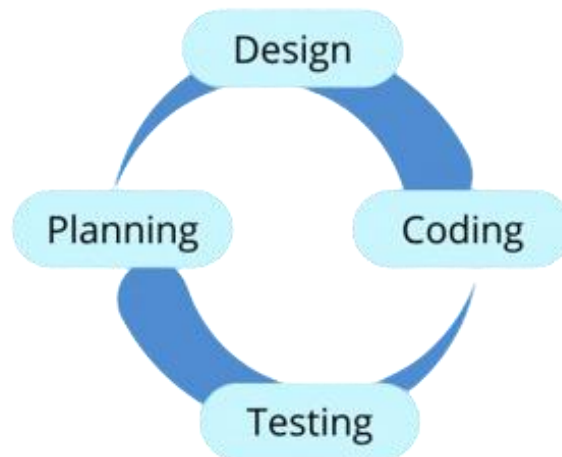
Internet Commerce atau Ecom atau E - Commerce atau Immerce yang pada dasarnya semua sebutan diatas mempunyai makna yang sama. Istilah-istilah tersebut berarti membeli atau menjual secara elektronik dan kegiatan ini dilakukan pada jaringan internet. (Ardiansyah et al., 2019)

Menurut (Jony, 2010:33). Pada jurnal (Ardiansyah et al., 2019) E-commerce atau dapat disebut perdagangan elektronik adalah penyebaran, pembelian, penjualan, pemasaran barang dan jasa melalui sistem elektronik seperti internet atau televisi, radio dan jaringan komputer lainnya.

E-Commerce dapat melibatkan transfer dana elektronik, pertukaran data elektronik, sistem manajemen inventori otomatis, dan sistem pengumpulan data otomatis. E-commerce akan merubah semua kegiatan marketing dan juga sekaligus memangkas biaya-biaya operasional untuk kegiatan trading (perdagangan). Industri teknologi informasi melihat kegiatan perdagangan elektronik ini sebagai aplikasi dan penerapan dari e-business yang berkaitan dengan transaksi komersial, seperti : transfer dana secara elektronik, supply chain management (SCM), e-pemasaran (e-marketing), atau pemasaran online (online marketing), pemrosesan transaksi online (online transaction processing), pertukaran data elektronik (electronic data interchange/EDI), dan lain-lain. (Ardiansyah et al., 2019)

2.8 Extreme Programming (XP)

Extreme Programming (XP) merupakan salah satu metode Agile yang cukup banyak digunakan, terutama pada proyek pengembangan aplikasi dalam skala kecil (Gamal Thabroni, 2022). Hal ini karena metode ini terhitung cukup sederhana dan ringkas namun tetap mengaplikasikan berbagai prinsip Agile yang dianggap *break through* dalam meningkatkan efisiensi serta efektivitas pengerjaan pengembangan perangkat lunak.



Gambar 2. 1 Tahapan *Extreme Programming*

Metode *Extreme Programming* (XP) memiliki beberapa tahapan dalam prosesnya, adapun tahapannya yaitu:

Planning

Pada tahap perencanaan ini penulis melakukan komunikasi dan observasi yang berkaitan dengan objek penelitian. Dalam perencanaan tersebut terdapat seperti pengumpulan data dan perancangan sistem yang akan dibangun menggunakan UML.

Design

Setelah melakukan tahapan perencanaan, tahap selanjutnya adalah tahapan design atau pemodelan. Pemodelan merupakan awal dari pembuatan sistem yang akan dibuat, dimana dapat dilihat proses-proses apa saja yang nantinya diperlukan dalam pembuatan suatu sistem.

Coding

Setelah melakukan tahapan design, tahapan selanjutnya adalah tahapan implementasi. Implementasi merupakan tahapan dimana setelah design selesai dibuat, maka akan dilakukan proses pengkodean menggunakan bahasa pemrograman Dart dan PHP.

Testing

Setelah tahapan pengkodean selesai, maka dilakukan tahapan pengujian sistem untuk mengetahui berbagai kesalahan yang timbul saat aplikasi sedang berjalan serta untuk memastikan apakah perangkat lunak yang dibangun telah sesuai dengan riwayat kebutuhan pengguna.

2.9 Teori Perancangan Basis Data

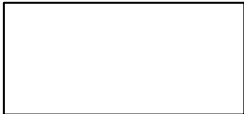
Database merupakan kumpulan data atau informasi yang tersimpan secara sistematis dalam komputer. Data ini dapat diproses atau dioperasikan oleh perangkat lunak untuk menghasilkan informasi baru bagi pengguna (Ishak dkk., 2022). Penggunaan database umumnya terjadi dalam aplikasi, baik itu website maupun aplikasi Android. Dalam mengoperasikan database, kita menggunakan bahasa SQL (Structured Query Language), yang memungkinkan kita untuk membuat data berelasi sesuai dengan kebutuhan aplikasi yang sedang dibangun.

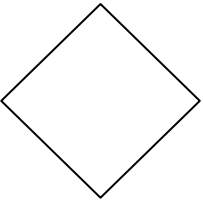
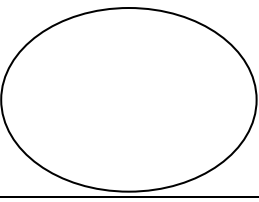

2.9.1 Entity Relationship Diagram (ERD)

Entity Relationship (ER) adalah suatu metode pemodelan basis data yang digunakan untuk menggambarkan skema konseptual suatu sistem. ER merupakan model data semantik yang digunakan dalam sistem basis data relasional dengan pendekatan top-down. Diagram yang digunakan untuk menggambarkan model ER disebut sebagai Entity-Relationship Diagram (ERD) atau diagram ER.

Entitas (entity) adalah objek yang dibedakan dan diidentifikasi secara unik, sedangkan relasi (relationship) menghubungkan antara entitas-entitas tersebut. Atribut (attribute) merupakan karakteristik yang membentuk setiap entitas dengan jumlah yang telah ditentukan (Pulungan dkk., t.t.).

Tabel 2. 2 Entity Relationship Diagram

No	Simbol	Nama	Keterangan
1.		Entitas	Digunakan untuk menggambarkan sebuah entitas

2.		Relasi	Digunakan untuk menggambarkan hubungan antar entitas
3.		Atribut	Sebagai property dari sebuah entitas
4.		Garis	Sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut

2.9.2 Transformasi ERD ke LRS

Transformasi ERD ke LRS adalah proses mengubah model ERD yang digunakan untuk mewakili hubungan antara entitas dalam basis data menjadi struktur data yang sesuai dengan Logical Record Structure (LRS). LRS adalah struktur data yang digunakan dalam pengorganisasian dan penyimpanan data dalam suatu sistem. Proses transformasi ini melibatkan pengelompokan entitas dan atribut dalam entitas dengan atribut yang lebih sedikit, penggabungan relasi ke entitas yang membutuhkan referensi, dan pembuatan LRS untuk setiap entitas dan relasi.

2.9.3 Logical Record Structure (LRS)

Logical Record Structure (LRS) adalah cara data diorganisir dan diwakili dalam suatu entitas atau catatan tunggal. LRS digunakan dalam pengorganisasian dan penyimpanan data dalam suatu sistem, baik itu dalam konteks basis data, sistem file, atau sistem informasi lainnya. LRS menggambarkan cara data diorganisir dan diwakili dalam suatu entitas atau catatan tunggal.

2.9.4 Normalisasi Database

Normalisasi database adalah suatu teknik untuk mengorganisasi data ke dalam tabel-tabel untuk memenuhi kebutuhan pengguna dalam suatu organisasi (Gamal Thabroni, 2023). Tahap normalisasi dimulai dari tahap paling ringan (1NF) hingga tahap paling ketat (5NF). Biasanya, normalisasi hanya mencapai tingkat 3NF atau BCNF karena sudah cukup memadai untuk menghasilkan tabel-tabel yang berkualitas baik. Sebuah tabel dikatakan baik (efisien) atau normal jika memenuhi tiga kriteria berikut:

1. Jika dilakukan dekomposisi tabel, dekomposisi tersebut harus memastikan keamanan dalam penggabungan kembali (Lossless Join Decomposition). Artinya, setelah tabel diuraikan menjadi tabel-tabel baru, tabel-tabel baru tersebut dapat digabungkan kembali untuk menghasilkan tabel asal dengan persis sama.
2. Ketergantungan fungsional harus tetap terjaga saat terjadi perubahan data (Dependency Preservation). Hal ini berarti bahwa perubahan data pada atribut tidak boleh mengakibatkan hilangnya ketergantungan fungsional antara atribut-atribut tersebut.
3. Tidak ada pelanggaran terhadap Bentuk Normal Boyce-Codd (BCNF). Jika kriteria BCNF tidak dapat terpenuhi, minimal tabel tersebut harus memenuhi Bentuk Normal Tahap Ketiga (3rd Normal Form / 3NF).

Normalisasi digunakan sebagai teknik analisis data dalam basis data untuk memastikan bahwa tabel-tabel yang saling terkait dalam database telah diatur dengan baik. Kondisi yang dianggap baik adalah ketika proses *insert*, *update*, *delete*, dan modifikasi pada satu atau beberapa atribut dalam sebuah tabel tidak mempengaruhi integritas data yang lain dalam hubungan relasinya.

2.10 Definisi *Unified Modelling Language (UML)*

UML (*Unified Modeling Language*) adalah bahasa visual yang digunakan untuk membuat diagram dan model yang mewakili sistem software (Revoupedia, 2024). UML dikembangkan untuk membantu pengembang sistem dan perangkat

lunak dalam menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak sistem perangkat lunak, serta untuk pemodelan bisnis dan sistem non-perangkat lunak lainnya.

UML digunakan oleh para pengembang perangkat lunak untuk menggambarkan struktur, perilaku, dan interaksi dari sebuah sistem. UML dapat digunakan untuk menggambarkan berbagai jenis sistem, termasuk sistem informasi, sistem bisnis, sistem komputer, dan sistem perangkat lunak.

UML terdiri dari berbagai jenis diagram seperti diagram kelas, diagram use case, diagram aktivitas, diagram sequence, dan lain-lain. Setiap jenis diagram digunakan untuk menggambarkan aspek yang berbeda dari sistem.

2.10.1 Use Case Diagram

Use case diagram adalah jenis diagram UML (Unified Modeling Language) yang menggambarkan fungsi, ruang lingkup, dan interaksi pengguna dengan sistem tersebut (Ariffud Muhammad, 2023). Use case merupakan sebuah teknik yang digunakan untuk menggambarkan interaksi antara satu atau lebih aktor dengan sistem informasi yang sedang dikembangkan.

Tujuan dari penggunaan use case adalah untuk memahami fungsi-fungsi yang ada dalam sistem informasi dan siapa saja yang berhak mengakses fungsi-fungsi tersebut. Dalam *Use case Diagram* terdapat dua elemen utama yaitu:

1. Actor

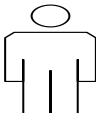


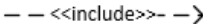


Merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.





2. Use Case

Merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Berikut adalah simbol-simbol yang ada pada *Use Case Diagram*:

Pada tabel dibawah bisa dilihat yaitu beberapa elemen yang terdapat pada *use case diagram* beserta kegunaannya.

Tabel 2. 3 Usecase Diagram

GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

	Sistem	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

2.10.2 Activity Diagram





Activity diagram adalah salah satu jenis diagram pada UML yang digunakan untuk menggambarkan aliran kerja atau aktivitas dalam suatu sistem. Diagram ini biasanya digunakan untuk memodelkan proses bisnis atau fungsionalitas dalam sistem perangkat lunak.


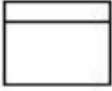
Activity diagram menunjukkan urutan aktivitas yang dilakukan dalam suatu proses, serta tindakan yang dilakukan dalam setiap aktivitas. Aktivitas dapat berupa tindakan konkret seperti mengirim email atau memproses data, atau aktivitas abstrak seperti mengambil keputusan atau menunggu persetujuan.

Activity diagram terdiri dari dua jenis elemen utama, yaitu aktivitas (activity) dan tindakan (action). Aktivitas dapat berupa setiap tindakan atau kegiatan dalam suatu proses, sedangkan tindakan adalah aktivitas yang tidak dapat dibagi lagi menjadi aktivitas yang lebih kecil.

Activity diagram juga dapat menggunakan berbagai jenis notasi, seperti garis panah, titik, dan bentuk-bentuk lainnya untuk menggambarkan hubungan antara aktivitas dan tindakan. Diagram ini dapat membantu para pengembang perangkat lunak untuk memahami secara visual alur kerja sistem dan membuat perubahan jika diperlukan.

Tabel 2. 4 Activity Diagram


SIMBOL	NAMA	KETERANGAN
	Status Awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan/Decision	Percabangan di mana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan/Join	Penggabungan di mana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.

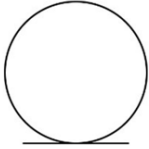
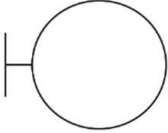
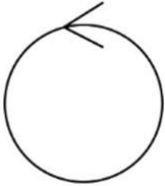



	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.10.3 Sequence Diagram

Sequence diagram atau diagram urutan adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci (Rony Setiawan, 2021). *Sequence Diagram* digunakan untuk menggambarkan kelakuan objek pada usecase dengan mendeskripsikan waktu hidup objek dan pesan yang dikirim dan diterima oleh objek (Afifah & Setyantoro, t.t.).

Tabel 2. 5 Sequence Diagram

NOTASI	NAMA	KETERANGAN
	Actor	Simbol yang digunakan sebagai peran yang sedang berinteraksi dengan sistem


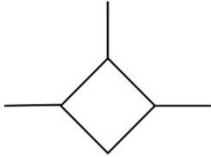
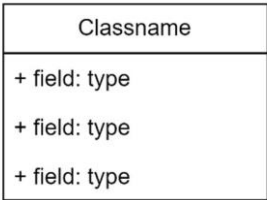
	Entity Class	Simbol yang digunakan untuk menggambarkan hubungan yang akan dilakukan.
	Boundary Class	Simbol yang digunakan untuk Kumpulan kelas yang menjadi interface atau interaksi antara satu atau lebih actor dengan sistem, seperti tampilan form entry dan form cetak.
	Control Class	Simbol yang digunakan sebagai penanggung jawab terhadap kelas-kelas atau objek-objek yang berisi logika.
	Life Line	Simbol yang merepresentasikan garis titik-titik yang terhubung dengan objekm sepanjang <i>life-line</i> terdapat <i>activation</i> .
	Recursive	Simbol yang digunakan sebagai pengiriman pesan untuk dirinya sendiri.
	Activation	Simbol yang digunakan untuk mewakili sebuah eksekusi operasi dari objek, Panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.


2.10.4 Class Diagram

Dalam rekayasa perangkat lunak, *Class diagram* dalam *Unified Modeling Language (UML)* adalah jenis diagram struktur statis yang menggambarkan struktur suatu sistem dengan menunjukkan kelas sistem, atributnya, operasi (atau metode), dan hubungan antar objek.

Class diagram dalam suatu proyek umumnya menggunakan konsep yang disebut *object-oriented*, sehingga membuatnya mudah untuk digunakan (Alifian Adam, 2022). Selain itu, class diagram membantu visualisasi struktur setiap kelas dalam sistem. Berikut, tabel *Class Diagram* berupa elemen dan kegunaannya:

Tabel 2. 6 Class Diagram

NOTASI	NAMA	KETERANGAN
	Generalization	Simbol yang digunakan untuk menggambarkan hubungan dimana objek berbagi perilaku.
	N-ary association	Simbol yang digunakan untuk menggambarkan hubungan dari tiga objek atau lebih.
	Class	Simbol yang digunakan untuk hubungan antara 2 class yang menunjukkan bahwa salah satu class mengimplementasikan sebuah metod dari kelas

		lain.
	Realization	Simbol yang digunakan untuk menggambarkan sebuah objek yang memiliki ketergantungan dengan objek lain.
	Dependency	Simbol yang digunakan untuk menggambarkan sebuah objek yang memiliki ketergantungan dengan objek lain.
	Association	Simbol yang digunakan untuk menggambarkan hubungan dari 2 <i>class</i>
	Aggregation	Simbol yang digunakan untuk menggambarkan hubungan dari 2 <i>class</i> , tetapi lebih spesifik dari <i>association</i> .
	Composition	Simbol yang digunakan untuk menggambarkan keterikatan antara <i>parent</i>

		dan <i>child</i> .
--	--	--------------------

2.11 Aplikasi Pendukung

Dalam proses pengembangan aplikasi yang sedang diteliti, berbagai alat pendukung digunakan. Contohnya, Software Development Kit (SDK) Android digunakan untuk mengembangkan aplikasi Android, JSON dan MySQL digunakan untuk mengelola data atau database, PHP berperan sebagai sisi server, dan Visual Studio Code menjadi alat utama dalam penulisan kode.

2.11.1 Android Software Development Kit (SDK)

Android SDK adalah tools API (Application Programming Interface) yang diperlukan untuk memulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java (Ardian Andre, 2021). Beberapa fitur Android yang penting adalah sebagai berikut:

1. Framework aplikasi yang mendukung penggantian komponen dan *reusable*.
2. DVM dioptimalkan untuk perangkat *mobile*.
3. Integrated browser berdasarkan engine open source WebKit.
4. Grafis yang dioptimalkan dan didukung oleh library grafis 2D, grafis 3D berdasarkan spesifikasi OpenGL ES 1.0.
5. SQLite untuk penyimpanan data.
6. Dukungan untuk audio, video dan gambar.
7. Bluetooth, Edge, 3G, Wifi.

2.11.2 JavaScript Object Notation (JSON)

JavaScript Object Notation (JSON) adalah merupakan sebuah format yang digunakan untuk menyimpan, membaca, serta menukar informasi dari web server sehingga dapat dibaca oleh para pengguna (Rony Setiawan, 2021).

JSON memiliki dua struktur utama. Pertama, ada objek yang terdiri dari pasangan nilai (value) dan kunci (key). Kedua, ada array yang berisi nilai-nilai yang berurutan. JSON juga dapat digunakan dalam berbagai bahasa pemrograman, termasuk PHP, Python, C++, Ruby, dan bahasa Dart yang akan digunakan dalam penelitian saat ini.

2.11.3 PHP



Gambar 2. 2 Logo PHP

PHP (Hypertext Preprocessor) adalah bahasa pemrograman yang digunakan secara luas untuk pengembangan aplikasi web. PHP adalah bahasa pemrograman *server-side* yang dapat digunakan untuk membuat dan mengembangkan suatu halaman web dinamis, mengakses database, mengelola formulir, mengirim email, dan masih banyak lagi. PHP biasanya digunakan bersama dengan HTML untuk membuat halaman web interaktif. PHP dapat dijalankan di berbagai platform, seperti Windows, Linux, macOS, dan dapat diintegrasikan dengan database seperti MySQL.

2.11.4 MySQL (My Structure Query Language)



Gambar 2. 3 Logo MySQL

MySQL adalah sistem manajemen basis data (DBMS) yang menggunakan perintah dasar SQL (Structured Query Language). DBMS ini banyak digunakan dalam pembuatan aplikasi berbasis website. MySQL mengelola dan menyimpan data dalam tabel-tabel di dalam basis data. Selain itu, MySQL memiliki banyak fitur untuk mengelola basis data, termasuk kemampuan untuk mengatur akses ke database dan hak pengguna (Yasin K, 2022). Jadi, MySQL adalah database server yang gratis dengan lisensi GNU General Public License (GPL), sehingga dapat digunakan untuk keperluan pribadi atau komersial tanpa membayar lisensi. MySQL termasuk dalam jenis Relational Database Management System (RDBMS), dan menggunakan bahasa SQL sebagai penghubung antara perangkat lunak aplikasi dengan database server.

2.11.5 XAMPP

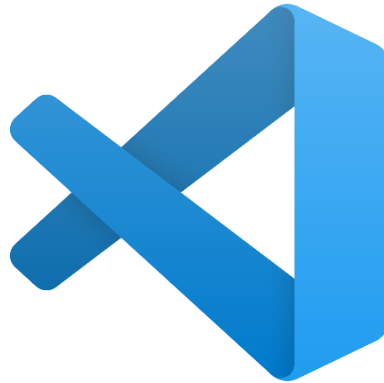


Gambar 2. 4 Logo XAMPP

XAMPP (X-platform, Apache, MySQL, PHP, Perl) merupakan perangkat lunak berbasis web server yang bersifat *open source* atau sumber terbuka yang

sudah mendukung di berbagai sistem operasi, seperti Windows, Linux, maupun macOS. XAMPP digunakan sebagai *localhost* server untuk memudahkan dalam proses pengeditan, desain, dan pengembangan aplikasi. Terdapat tiga komponen penyusun utama dari *software* ini yaitu htdocs, Control Panel, dan PhpMyAdmin.

2.11.6 Visual Studio Code



Gambar 2. 5 Logo VScode

Visual Studio Code merupakan sebuah perangkat lunak yang digunakan sebagai pengedit teks maupun kode yang dapat dijalankan di perangkat desktop berbasis Windows, Linux, dan macOS. Visual Studio Code adalah *software* editor yang *powerful*, tapi tetap ringan ketika digunakan (Ariffud Muhammad, 2022).

Visual Studio Code bisa digunakan untuk membuat dan mengedit *source code* berbagi bahasa pemrograman. Misalnya, seperti JavaScript, TypeScript, dan Dart. Bahkan, Visual Studio Code juga kompatibel dengan bahasa dan *runtime environment* lain, seperti PHP, Python, Java, C++, dan .NET. Hal ini karena ekosistemnya yang luas dan ketersediaan ekstensi yang melimpah.

2.12 Teori Pengujian Sistem

Pengujian perangkat lunak adalah proses sistematis yang melibatkan pemeriksaan dan verifikasi perangkat lunak untuk mengidentifikasi cacat, kesalahan, atau perbedaan dari persyaratan yang ditentukan. Proses ini bertujuan untuk memastikan bahwa perangkat lunak memenuhi tujuan yang saling dimaksudkan dan sesuai dengan standar kualitas yang ditetapkan.

2.12.1 Black Box Testing

Black-box testing atau dapat disebut juga *Behavioral Testing* adalah pengujian yang dilakukan untuk mengamati hasil *input* dan *output* dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak (Rony Setiawan, 2021). Oleh karena itu, *black-box testing* dapat dipahami sebagai pengujian sistem berdasarkan kebutuhan fungsional sederhana yang dilakukan untuk menguji sistem.

BAB III

METODE

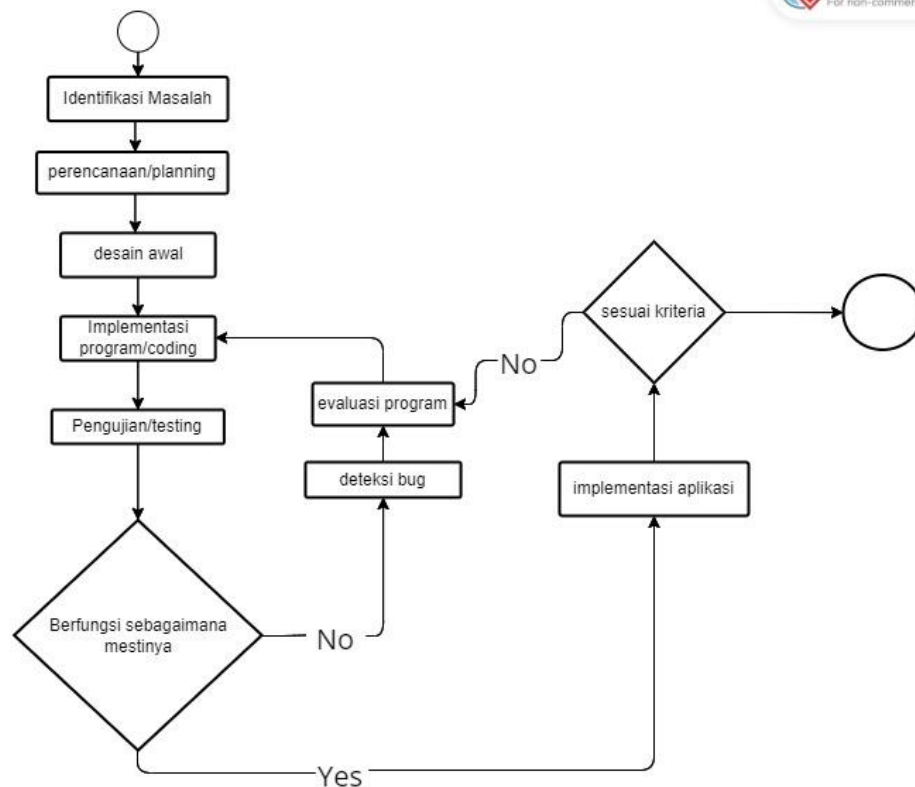
3.1 Rencana Penelitian

Tahapan penelitian dimulai dengan identifikasi masalah yang dihadapi oleh pengguna toko jual beli aksesoris rumah Woodman.project. Setelah itu, dilakukan perencanaan untuk membuat spesifikasi sistem dan rancangan arsitektur sistem e-commerce. Langkah selanjutnya adalah melakukan desain sistem yang mencakup desain antarmuka pengguna (UI), desain basis data, dan desain algoritma. Setelah desain selesai, dilakukan implementasi sistem, termasuk pembuatan kode program dan pengaturan basis data.

Proses berikutnya adalah implementasi desain pada program dan setelah itu melakukan pengujian sistem untuk memastikan bahwa semua fitur dan fungsionalitas bekerja sesuai dengan harapan. Jika terdapat bug atau masalah dalam sistem, maka dilakukan deteksi bug secepat mungkin. Setelah deteksi bug, dilakukan evaluasi dan perbaikan codingan untuk memastikan kualitas sistem.

Jika sistem berfungsi sebagaimana mestinya tanpa bug yang signifikan, maka dilakukan implementasi kepada konsumen. Proses implementasi melibatkan penyerahan produk atau layanan kepada konsumen untuk digunakan secara aktif. Setelah implementasi, dilakukan evaluasi kembali untuk memastikan bahwa sistem telah sesuai dengan kriteria yang diinginkan oleh konsumen.

Jika hasil evaluasi sesuai dengan kriteria, maka dapat dianggap bahwa penelitian telah berhasil. Namun, jika terdapat perbaikan atau perubahan yang diperlukan, maka dilakukan langkah-langkah yang diperlukan untuk memastikan sistem dapat memberikan manfaat yang optimal dan memuaskan bagi konsumen.



Gambar 3. 1 Rencana Penelitian

3.2 Obyek Penelitian

Obyek penelitian dari penelitian yang dilakukan adalah sistem aplikasi e-commerce yang akan dikembangkan untuk menjual berbagai aksesoris rumah melalui platform Woodman.project. Aplikasi ini bertujuan untuk memudahkan pengelolaan inventaris, manajemen transaksi penjualan, dan pengiriman produk kepada pelanggan. Obyek penelitian ini akan dikembangkan berbasis Android agar dapat diakses dengan mudah oleh para pengguna.

Dalam pengembangan obyek penelitian ini, akan digunakan framework Flutter untuk mempercepat proses pengembangan aplikasi e-commerce. Flutter, yang dikembangkan oleh Google, menyediakan alat yang efektif untuk membangun antarmuka pengguna yang responsif dan menarik. Penggunaan framework Flutter diharapkan dapat mempercepat pengembangan aplikasi e-commerce

Woodman.project, memberikan pengalaman belanja online yang optimal, dan memudahkan pemeliharaan aplikasi di masa mendatang.

Obyek penelitian ini diharapkan dapat membantu Woodman.project dalam menjalankan kegiatan bisnisnya dengan lebih efektif dan efisien, serta memberikan pengalaman berbelanja yang memuaskan bagi para pelanggan yang mencari aksesoris rumah yang unik dan berkualitas.

3.3 Teknik Pengumpulan Data

Pada penelitian ini pengumpulan data yang akan digunakan terdiri dari dua jenis, yaitu data primer dan data sekunder.

1. Pada data primer, informasi pokok adalah kategori data yang diperoleh langsung dari sumber aslinya tanpa melalui proses interpretasi atau analisis oleh pihak lain. Data ini berasal dari observasi langsung, dialog, wawancara, survei, atau pengumpulan informasi secara langsung dari individu atau kelompok yang terkait dengan tujuan pengumpulan data atau penelitian. Contoh dari data primer mencakup respon langsung dari responden dalam kuesioner atau wawancara, hasil pengukuran langsung, atau observasi yang langsung dilakukan oleh peneliti. Keunggulan data primer terletak pada tingkat ketepatan dan relevansinya yang tinggi karena informasinya diperoleh langsung dari sumbernya.
2. Pada data sekunder, informasi sekunder merupakan jenis data yang telah terhimpun oleh pihak lain atau telah ada sebelumnya untuk keperluan lain. Data ini tidak diperoleh secara langsung oleh peneliti, tetapi diambil dari sumber-sumber seperti literatur, jurnal, basis data, laporan resmi, atau dokumentasi lainnya. Peneliti menggunakan data sekunder untuk mendukung atau melengkapi penelitian mereka. Kelebihan dari data sekunder meliputi efisiensi waktu dan biaya, serta kemampuan untuk memperoleh wawasan tambahan dari penelitian atau pengumpulan data yang telah terjadi sebelumnya. Namun, kelemahan data sekunder dapat mencakup kemungkinan keterbatasan atau ketidaksesuaian dengan kebutuhan penelitian saat ini.

3.3.1 Observasi

Pada tahap ini melakukan pengamatan data yang dilakukan secara langsung pada took woodman project. Pengamatan ini bertujuan untuk mengetahui data dan memahami proses transaksi jual-beli yang sedang berjalan saat ini, sehingga kita dapat mengetahui kebutuhan yang harus dikembangkan nantinya.

3.3.2 Wawancara

Pada tahap ini dilakukan pengumpulan data dengan cara melakukan wawancara kepada pemimpin dari toko woodman project, pihak administrasi dan customer terkait permasalahan yang terjadi pada proses kegiatan jual-beli yang sedang berjalan. Hasil wawancara akan berguna untuk menyelesaikan permasalahan pada proses kegiatan transaksi yang terjadi pada toko woodman project.

3.3.3 Studi Literatur

Pada tahap ini dilakukan pengumpulan data pustaka dengan cara membaca, mencatat, dan mengolah data dari penelitian terdahulu atau yang sudah ada. Hasil studi literatur berguna untuk mendapatkan ide dari penyelesaian dan perbandingan evaluasi masalah pada penelitian ini.

3.4 Metode Pengembangan Sistem

Metode pengembangan sistem yang akan dilakukan pada penelitian ini menggunakan metode extreme programming. Dengan menggunakan metode extreme programming ini dapat melakukan pembuatan aplikasi sesuai dengan keinginan yang akan merubah proses sistem yang berjalan saat ini pada took furniture woodman project.

Pada penelitian ini aplikasi dikembangkan dengan menggunakan metodologi extreme programming (XP) sebagai metode dalam pengembangan sistem. Extreme programming (XP) adalah metodologi dalam pengembangan agile software development methodologies yang berfokus pada pengkodean (coding) yang

menjadi aktivitas utama dalam semua tahapan pada siklus pengembangan perangkat lunak. Metode XP dapat diterapkan dengan waktu pembangunan aplikasi yang tidak lama dan sesuai dengan penggunaan pengembangan perangkat lunak. XP menawarkan tahapan dalam waktu yang singkat dan berulang untuk bagian-bagian yang berbeda sesuai dengan fokus yang akan dicapai. Dalam XP terdapat iterasi yang bisa dilakukan berulang kali sesuai dengan kebutuhan (Gunawan et al., 2020).

3.4.1 Planning

Pada tahap ini, proses komunikasi atau yang sering disebut sebagai analisis kebutuhan dilakukan dengan melibatkan berbagai aktor yang akan menggunakan sistem ini, yaitu pemilik toko furniture, admin marketplace, dan pelanggan. Langkah awal dimulai dengan mengidentifikasi peran masing-masing aktor dalam sistem. Misalnya, pemilik toko bertugas mengunggah produk furniture, admin bertanggung jawab mengelola platform marketplace, dan pelanggan melakukan pencarian serta pembelian furniture. Selanjutnya, pengumpulan data dan informasi dilakukan menggunakan berbagai teknik, seperti observasi, wawancara, dan studi literatur.

Observasi dilakukan dengan melihat bagaimana proses transaksi furniture berjalan di toko-toko offline saat ini. Wawancara dilakukan dengan pemilik toko untuk mengetahui tantangan yang sering mereka hadapi, seperti kesulitan menjangkau pelanggan yang lebih luas. Pelanggan juga diwawancarai untuk memahami kebutuhan mereka, seperti fitur pencarian yang mudah atau opsi pembayaran yang beragam. Studi literatur dilakukan untuk meninjau aplikasi sejenis dan menentukan fitur yang relevan. Setelah data terkumpul, kebutuhan sistem mulai ditentukan, yang meliputi kebutuhan fungsional dan kebutuhan non-fungsional. Contoh kebutuhan fungsional adalah fitur pencarian produk berdasarkan kategori dan harga, sistem keranjang belanja, dan integrasi pembayaran online. Sementara itu, kebutuhan non-fungsional meliputi tampilan aplikasi yang responsif dan kecepatan pemrosesan data yang optimal.

Tahap pengumpulan data ini sangat penting untuk memastikan bahwa aplikasi marketplace furniture yang akan dibuat benar-benar memenuhi kebutuhan

para pengguna, baik dari sisi penjual maupun pembeli. Proses ini juga menjadi dasar dalam menentukan prioritas fitur yang akan dikembangkan pada tahap-tahap selanjutnya. Dengan demikian, aplikasi yang dihasilkan dapat memberikan solusi yang relevan dan efektif dalam memfasilitasi transaksi furniture secara online.

3.4.2 Design

Setelah tahap analisis kebutuhan selesai, langkah berikutnya adalah proses desain sistem. Pada tahap ini, desain awal sistem dibuat untuk memberikan gambaran jelas mengenai struktur dan alur kerja aplikasi marketplace furniture yang akan dibangun. Desain difokuskan pada kemudahan penggunaan dan fleksibilitas sistem, sesuai dengan kebutuhan pengguna yang telah diidentifikasi sebelumnya. Desain dimulai dengan membuat sketsa antarmuka pengguna (mockup) untuk setiap fitur utama, seperti halaman pencarian produk, detail produk, keranjang belanja, dan checkout. Misalnya, halaman pencarian produk dirancang agar pengguna dapat dengan mudah menggunakan filter kategori, harga, dan lokasi.

Tampilan ini dirancang untuk responsif sehingga nyaman digunakan di perangkat mobile maupun desktop. Selanjutnya, dibuat diagram alur proses (flowchart) untuk menggambarkan bagaimana sistem bekerja. Contohnya, alur untuk fitur checkout dimulai dari pengguna menambahkan produk ke keranjang belanja, memverifikasi pesanan, memilih metode pembayaran, hingga mendapatkan konfirmasi pembayaran yang sukses. Untuk memastikan struktur data yang mendukung, dirancang juga database yang mencakup tabel-tabel utama seperti:

Tabel Pengguna: Berisi informasi pelanggan dan penjual.

Tabel Produk: Berisi data produk furniture, seperti nama, deskripsi, kategori, harga, dan stok.

Tabel Transaksi: Berisi riwayat pembelian dan status pembayaran.

Tabel Ulasan: Berisi ulasan dan rating dari pelanggan.

Kemudian, desain ini dilanjutkan dengan membuat arsitektur sistem menggunakan pola Model-View-Controller (MVC). Model digunakan untuk mengelola data seperti produk dan transaksi. View menangani antarmuka pengguna, sementara Controller bertanggung jawab menghubungkan model dan view untuk menjalankan logika aplikasi.

Untuk memastikan setiap komponen sistem dapat bekerja sama dengan baik, dilakukan design prototyping. Prototipe ini diuji dalam skala kecil untuk mengevaluasi fungsionalitas dasar, seperti pencarian produk atau proses checkout. Feedback dari pengujian ini digunakan untuk memperbaiki dan menyempurnakan desain sebelum masuk ke tahap pengembangan. Proses desain ini dilakukan secara iteratif, dengan melibatkan tim pengembang dan pelanggan. Pelanggan memberikan masukan terhadap sketsa antarmuka dan prototipe, sehingga desain yang dihasilkan tidak hanya sesuai kebutuhan, tetapi juga ramah pengguna. Dengan desain yang matang, aplikasi marketplace furniture siap untuk memasuki tahap pengembangan lebih lanjut.

3.4.3 Coding

Setelah desain sistem selesai dan disepakati oleh tim dan pelanggan, langkah berikutnya adalah proses coding untuk membangun aplikasi marketplace furniture. Pada tahap ini, pengembangan dilakukan berdasarkan prinsip XP, seperti *Pair Programming*, *Test-Driven Development (TDD)*, dan *Continuous Integration*.

a. Persiapan Pengembangan

Pengembang memulai dengan mengatur lingkungan kerja.

1. Alat pengembangan seperti IDE (Integrated Development Environment), kontrol versi menggunakan Git, dan sistem manajemen proyek seperti Notion diatur untuk mendukung kolaborasi tim.
2. Framework dan library yang relevan, seperti Flutter.

b. Test-Driven Development (TDD)

Setiap fitur dimulai dengan menulis unit test terlebih dahulu sebelum kode fungsionalitas ditulis. Contoh *Unit Test* untuk Keranjang Belanja:

1. "Ketika pengguna menambahkan produk ke keranjang, jumlah produk di keranjang harus bertambah."
2. "Jika stok produk habis, pengguna tidak bisa menambahkan produk ke keranjang."

Setelah tes dibuat, pengembang mulai menulis kode untuk memenuhi tes tersebut. Proses ini memastikan bahwa setiap kode yang ditulis sudah sesuai dengan kebutuhan.

c. Refactoring

Selama proses pengembangan, kode yang sudah selesai terus diperbaiki (refactoring) untuk meningkatkan kualitas tanpa mengubah fungsionalitas. Contoh:

1. Mengoptimalkan fungsi pencarian produk agar lebih cepat dengan menggunakan indeks database.
2. Merapikan struktur kode sehingga lebih mudah dibaca dan dipelihara.

d. Kolaborasi dengan Pelanggan

Setelah fitur dasar selesai, aplikasi diuji bersama pelanggan. Feedback mereka digunakan untuk memperbaiki atau menambahkan fitur. Misalnya: Jika pelanggan merasa filter pencarian kurang intuitif, tim menyesuaikan antarmuka agar lebih mudah digunakan.

e. Iterasi Berulang

Pengembangan dilakukan dalam iterasi pendek. Contohnya:

1. Iterasi pertama: Menyelesaikan fitur dasar seperti login, pencarian produk, dan keranjang belanja.
2. Iterasi kedua: Mengembangkan fitur tambahan seperti ulasan dan sistem pembayaran.

3. Iterasi ketiga: Menyempurnakan performa aplikasi dan memperbaiki bug berdasarkan hasil pengujian.

Proses coding ini dilakukan secara kolaboratif dan berulang untuk memastikan aplikasi marketplace furniture dibangun dengan kualitas tinggi dan sesuai dengan kebutuhan pengguna. Hasil akhirnya adalah aplikasi yang stabil, mudah digunakan, dan siap diluncurkan ke pasar.

3.4.4 Testing

Setelah proses coding selesai, tahap testing dilakukan untuk memastikan aplikasi marketplace furniture berfungsi dengan baik:

1. Unit Testing

Setiap fungsi, seperti pencarian produk dan checkout, diuji untuk memastikan hasil sesuai. Misalnya, sistem harus menampilkan produk yang relevan saat kata kunci dimasukkan.

2. Integration Testing

Pengujian dilakukan untuk memastikan modul seperti keranjang belanja dan pembayaran dapat bekerja bersama tanpa masalah.

3. System Testing

Aplikasi diuji secara keseluruhan, termasuk responsivitas pada berbagai perangkat dan kemampuan menangani banyak pengguna.

4. User Acceptance Testing (UAT)

Pelanggan mencoba fitur utama seperti pencarian dan transaksi. Masukan mereka digunakan untuk menyempurnakan aplikasi.

5. Regression Testing

Setiap kali ada perubahan, pengujian dilakukan kembali untuk memastikan fitur yang sudah ada tetap berfungsi dengan baik.

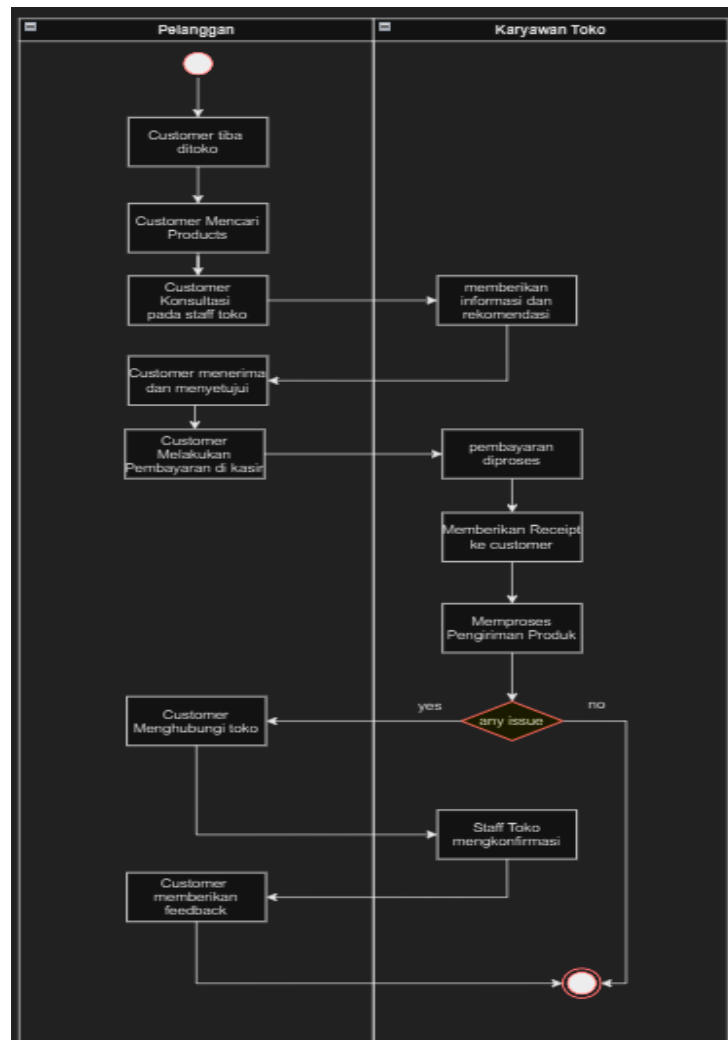
Dengan kombinasi manual dan automated testing, aplikasi dipastikan stabil dan siap digunakan.

3.5 Proses Bisnis yang Berjalan

Woodman Project adalah sebuah toko furniture yang menyediakan berbagai jenis produk seperti kursi, meja, lemari, dan dekorasi rumah lainnya. Sebelum adanya sistem digital, seluruh kegiatan operasional di toko ini dilakukan secara manual. Pelanggan yang ingin membeli furniture datang langsung ke toko. Mereka akan melihat-lihat berbagai produk yang dipajang di dalam toko, dan mencari tahu lebih lanjut tentang jenis material, desain, dan harga produk tersebut. Setiap produk diatur dan dipajang berdasarkan kategori, namun pelanggan harus berjalan ke setiap bagian toko untuk menemukan item yang mereka butuhkan. Setelah menemukan produk yang diinginkan, pelanggan akan berbicara dengan seorang sales atau pegawai toko. Pegawai tersebut akan memberikan informasi lebih lanjut mengenai spesifikasi produk, memberikan saran jika dibutuhkan, dan menjelaskan cara pemesanan atau cara membeli barang jika barang yang diinginkan tidak tersedia di tempat.

Jika pelanggan setuju dengan harga, transaksi kemudian dilanjutkan ke bagian kasir. Pembayaran dilakukan secara langsung, biasanya menggunakan uang tunai atau transfer bank. Jika ada promosi atau diskon, pegawai toko akan menghitung ulang harga dan memberikan tagihan final. Setelah pembayaran selesai, pelanggan akan menerima tanda terima dan barang yang dibeli akan dikemas atau diantarkan ke pelanggan. Proses pengiriman produk dilakukan oleh toko, dengan pegawai atau jasa pengiriman yang mengatur pengantaran barang ke alamat pelanggan. Biasanya, proses pengiriman hanya bisa dilakukan setelah pembayaran selesai dan barang sudah siap dikirim. Jika ada masalah atau kerusakan pada produk yang diterima, pelanggan harus menghubungi toko dan mengatur proses retur atau penggantian produk, yang biasanya memerlukan komunikasi langsung via telepon atau kunjungan ulang ke toko. Di sisi lain, Woodman Project harus mengelola stok barang secara manual. Setiap produk dicatat dalam buku atau spreadsheet, dan pegawai toko harus memantau inventaris secara rutin untuk memastikan produk tidak kehabisan stok. Begitu pula dengan pemasaran yang lebih bersifat tradisional, seperti iklan di media cetak atau brosur yang dibagikan langsung ke pelanggan.

Dengan sistem yang sepenuhnya manual ini, bisnis berjalan dengan cara yang lebih terbatas. Pelanggan hanya bisa datang langsung ke toko untuk melihat barang, dan transaksi hanya bisa terjadi pada jam buka toko. Proses ini mengurangi fleksibilitas dan efisiensi, serta sulit untuk menjangkau pasar yang lebih luas tanpa menggunakan metode promosi yang lebih modern atau platform digital. Seiring berjalannya waktu, dengan bertambahnya pelanggan dan produk yang dijual, Woodman Project merasakan kesulitan untuk mengelola operasional dan memenuhi kebutuhan pelanggan dengan lebih cepat dan efisien. Inilah yang menjadi alasan utama untuk beralih ke sistem digital atau marketplace yang lebih terorganisir dan memudahkan transaksi baik bagi penjual maupun pembeli.



Gambar 3. 2 Diagram Sistem Berjalan

3.6 Identifikasi Masalah Proses Bisnis yang Berjalan

Woodman Project, sebuah toko furniture terkemuka, menghadapi beberapa tantangan dalam menjalankan bisnisnya sebelum adanya sistem digital. Seluruh kegiatan operasional di toko ini dilakukan secara manual, mulai dari pemasaran, hingga proses transaksi jual-beli. Bagi pelanggan, mencari produk yang diinginkan seringkali menjadi pengalaman yang rumit. Mereka harus berjalan ke setiap bagian toko untuk menemukan item yang mereka cari, dan hanya dapat memperoleh informasi mengenai material, desain, dan harga dari perbincangan langsung dengan pegawai toko. Proses transaksi pun terbatas pada jam buka toko, di mana pelanggan harus datang secara langsung untuk melakukan pembayaran. Pembayaran biasanya dilakukan dengan uang tunai atau transfer bank, dan jika ada promosi atau diskon, pegawai toko harus menghitung ulang harga. Kegiatan pemasaran toko juga masih sangat terbatas, hanya mengandalkan iklan di media cetak atau brosur yang dibagikan langsung kepada pelanggan. Kondisi ini tentu membatasi kemampuan Woodman Project untuk menjangkau pasar yang lebih luas.

Dengan sistem yang sepenuhnya manual, Woodman Project menghadapi tantangan dalam hal fleksibilitas, efisiensi, dan jangkauan pasar. Kebutuhan pelanggan yang semakin meningkat juga tidak dapat dipenuhi secara optimal. Oleh karena itu, Woodman Project merasa perlu untuk beralih ke sistem digital atau marketplace yang lebih terorganisir dan memudahkan transaksi bagi penjual maupun pembeli.

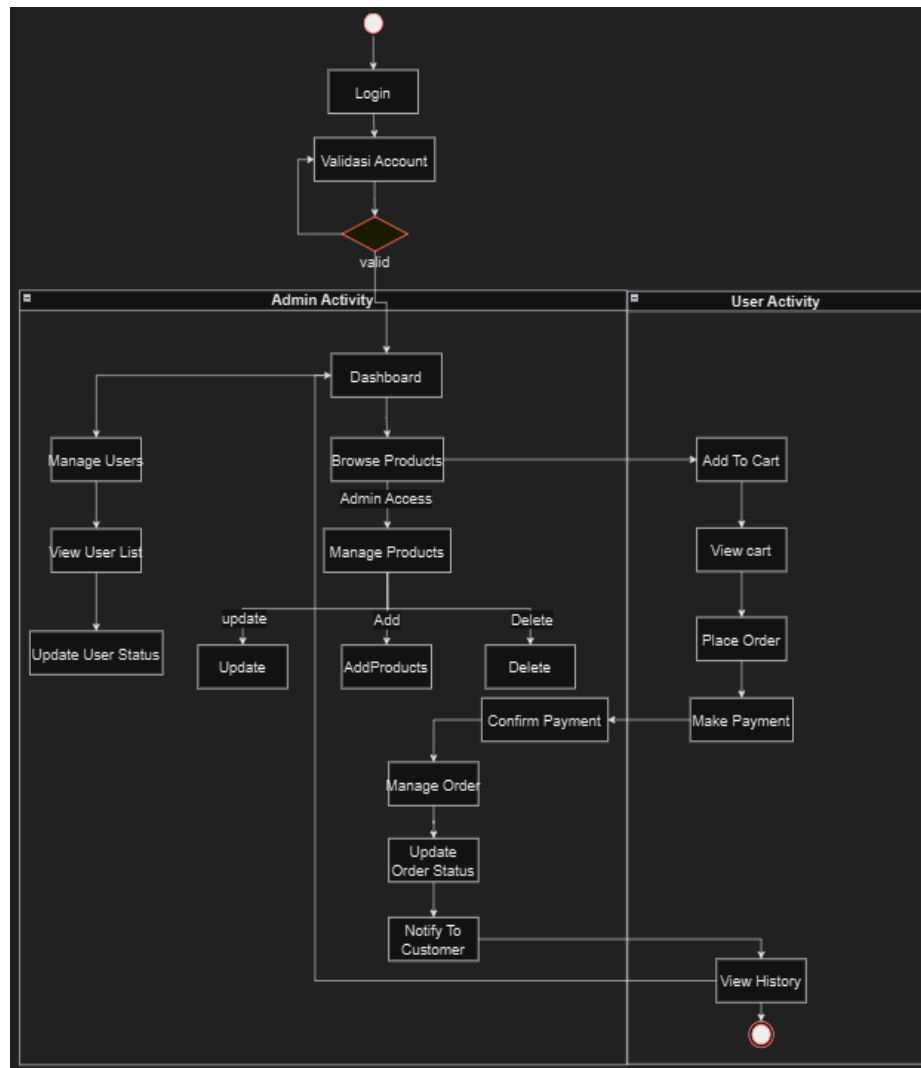
3.7 Proses Bisnis Usulan

Untuk mengatasi berbagai tantangan yang dihadapi Woodman Project dengan sistem manual, maka diusulkan sebuah solusi aplikasi mobile yang dapat memberikan jangkauan luas dan fleksibilitas belanja dari mana saja bagi pelanggan. Aplikasi mobile Woodman Project akan menawarkan pengalaman berbelanja yang lebih efisien dan menyenangkan bagi pelanggan. Melalui aplikasi ini, pelanggan dapat dengan mudah menjelajahi katalog produk furniture dan dekorasi yang tersedia, tanpa harus datang langsung ke toko. Aplikasi ini akan menyajikan informasi produk secara lengkap, seperti gambar, deskripsi, material, desain, dan

harga. Pelanggan dapat dengan cepat mencari dan menemukan item yang mereka butuhkan hanya dengan beberapa klik di smartphone mereka. Fitur pencarian dan filter akan membantu pelanggan menyaring produk berdasarkan kategori, harga, atau preferensi lainnya. Selain itu, aplikasi ini akan memungkinkan pelanggan untuk melakukan transaksi pembelian secara online. Pelanggan dapat memasukkan produk ke dalam keranjang belanja, memilih metode pembayaran yang diinginkan (seperti kartu kredit, dompet digital, atau pembayaran di tempat), dan menyelesaikan proses pembelian dengan mudah.

Untuk meningkatkan efisiensi pengiriman, aplikasi akan terintegrasi dengan sistem logistik Woodman Project. Pelanggan dapat memilih opsi pengiriman yang sesuai, seperti pengantaran langsung ke rumah atau pengambilan di toko. Riwayat pembelian juga akan tersimpan dalam akun pengguna, memudahkan pelanggan untuk melakukan pembelian ulang atau melacak status pengiriman. Dari sisi pengelolaan operasional, aplikasi ini akan dilengkapi dengan fitur manajemen stok, pemesanan, dan pelaporan yang terintegrasi. Pegawai toko dapat dengan mudah memantau persediaan, membuat pesanan baru, dan menganalisis tren penjualan. Hal ini akan membantu Woodman Project dalam memastikan ketersediaan produk dan meningkatkan efisiensi operasional. Untuk meningkatkan jangkauan pemasaran, aplikasi mobile Woodman Project akan dilengkapi dengan fitur-fitur seperti promosi. Pelanggan dapat dengan mudah berbagi produk, menerima notifikasi tentang penawaran menarik, serta diskon.

Dengan aplikasi mobile yang terintegrasi ini, Woodman Project akan dapat memberikan pengalaman berbelanja yang lebih nyaman, fleksibel, dan efisien bagi pelanggan. Selain itu, sistem digital yang lebih terorganisir ini juga akan membantu toko dalam mengelola operasional secara lebih efektif, serta memperluas jangkauan pemasaran dan penjualan.



Gambar 3. 3 Diagram Sistem Usulan

3.8 Analisis Kebutuhan Sistem

Sebelum masuk ke tahap perancangan, pada tahap ini dilakukan analisa kebutuhan sistem yang nantinya akan digunakan pada tahapan perancangan untuk UML, ERD, UI.

3.8.1 Kebutuhan Fungsional

Analisa kebutuhan fungsional merupakan proses-proses yang di perlukan oleh system serta informasi yang harus di hasilkan dan diolah oleh system.

Berikut ini adalah kebutuhan fungsional dari perancangan aplikasi e-commerce pada woodman project furniture.

1. Sistem menyediakan fitur login register dan *forgot password* guna authorisasi user admin dan customer.
2. *Customer* dapat menampilkan daftar menu yang tersedia dan juga dapat memfilter sesuai dengan kategori atau *by price*.
3. *Customer* dapat melakukan pesanan barang yang sudah di konfirmasi oleh admin.
4. Admin dapat memperbarui, merubah serta mengkonfirmasi pembelian oleh *customer*.
5. *Customer* dapat mengelola profilnya.
6. Sistem dapat menyimpan riwayat transaksi *Customer*
7. Sistem dapat menyimpan produk yang telah di masukan ke *cart* oleh *customer*.
8. Customer dapat mengubah kata sandi akun mereka.
9. Admin dapat melihat riwayat pembelian pelanggan.
10. Admin dapat mengelola produk.
11. Admin mengkonfirmasi transaksi dan memperbarui status transaksi.

3.8.2 Kebutuhan Non-Fungsional

Analisi kebutuhan non-fungsional sangat penting dalam menentukan spesifikasi kebutuhan system karna focus pada aspek-aspek yang tidak terkait langsung dengan fungsionalitas aplikasi, namun sangat mempengaruhi performa, kemanan serta pengalaman pengguna, berikut beberapa komponen yang termasuk kedalam analisis non-fungsional.

1. Analisis kebutuhan perangkat lunak

Analisis ini di butuhkan untuk mengetahui spesifikasi minimum yang dibutuhkan untuk membangun sebuah aplikasi, berikut software pendukung:

- a. Android studio emulator
 - b. VScode
 - c. Flutter sdk
 - d. Mysql
 - e. Laravel
 - f. Browser chrome dan sejenisnya
2. Analisi kebutuhan perangkat keras

Kebutuhan spesifikasi minimum perangkat keras mobile yang dibutuhkan oleh pengguna untuk menjalankan system ini adalah sebagai berikut:

- a. Processor smartphone android/Ios minimum Snapdragon.
- b. Ram 2 GB
- c. ROM 16 GB
- d. System Operation Marshadow 6.0

BAB IV

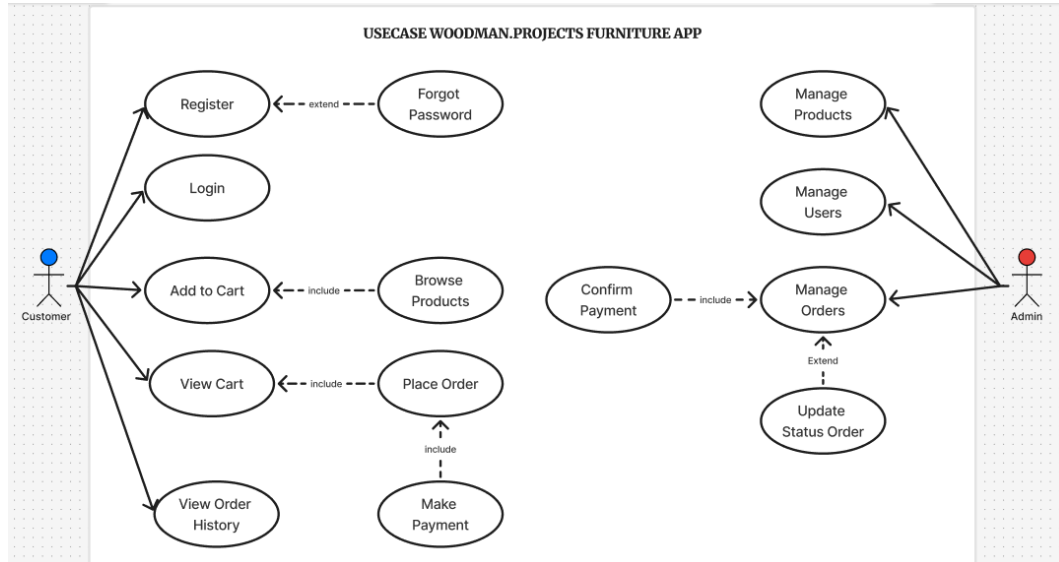
HASIL DAN PEMBAHASAN

4.1 Desain

Penelitian ini menggunakan Unified Modeling Language (UML) untuk menggambarkan rancangan sistem perangkat lunak untuk aplikasi mobile Woodman.project. UML digunakan untuk menggambarkan use case diagram, activity diagram, class diagram, dan sequence diagram.

4.1.1 Use Case Diagram

Use Case Diagram adalah suatu model yang sangat fungsional dalam sebuah sistem yang menggunakan *Actor* dan *Use Case* itu sendiri. Sedangkan *Use Case* sendiri adalah layanan atau fungsi-fungsi yang tersedia pada sistem untuk penggunaannya.



Gambar 4. 1 Usecase Diagram

Use case diagram tersebut menggambarkan interaksi antara pengguna (aktor) dengan sistem aplikasi furniture Woodman.Project. Diagram ini mengidentifikasi dua aktor utama: Customer dan Admin, masing-masing dengan fungsi dan hak akses yang berbeda dalam sistem.

Dari sisi Customer, sistem menyediakan beberapa fungsionalitas dasar yang mencakup manajemen akun dan proses pembelian. Untuk manajemen akun, customer dapat melakukan “Register” sebagai pengguna baru dan “Login” ke dalam sistem. Terdapat juga fitur “Forgot Password” yang terintegrasi dengan proses Register untuk memulihkan akses akun.

Dalam konteks pembelian, customer dapat “Browse Products” untuk melihat katalog furniture yang tersedia. Selanjutnya, customer dapat “Add to Cart” untuk produk yang diminati dan “View Cart” untuk mengelola keranjang belanja mereka. Proses pembelian dilanjutkan dengan “Place Order” yang terintegrasi dengan “Make Payment” untuk menyelesaikan transaksi. Setelah pembayaran, customer perlu melakukan “Confirm Payment” untuk verifikasi. Customer juga dapat memantau “View Order History” untuk melihat riwayat pembelian mereka.

Dari perspektif Admin, sistem menyediakan fungsi-fungsi pengelolaan yang lebih komprehensif. Admin memiliki akses untuk “Manage Products” yang memungkinkan pengelolaan katalog produk, “Manage Users” untuk administrasi pengguna, dan “Manage Orders” untuk mengelola pesanan yang masuk. Sebagai bagian dari manajemen pesanan, admin dapat melakukan “Update Status Order” untuk memperbarui status pesanan customer.

Diagram ini menunjukkan bahwa sistem dirancang dengan pemisahan yang jelas antara fungsi customer dan admin, memastikan pengelolaan yang efektif dan pengalaman pengguna yang terstruktur. Hubungan *include* dan *extend* antar use case menggambarkan dependensi dan perluasan fungsionalitas yang mendukung alur kerja sistem secara keseluruhan.

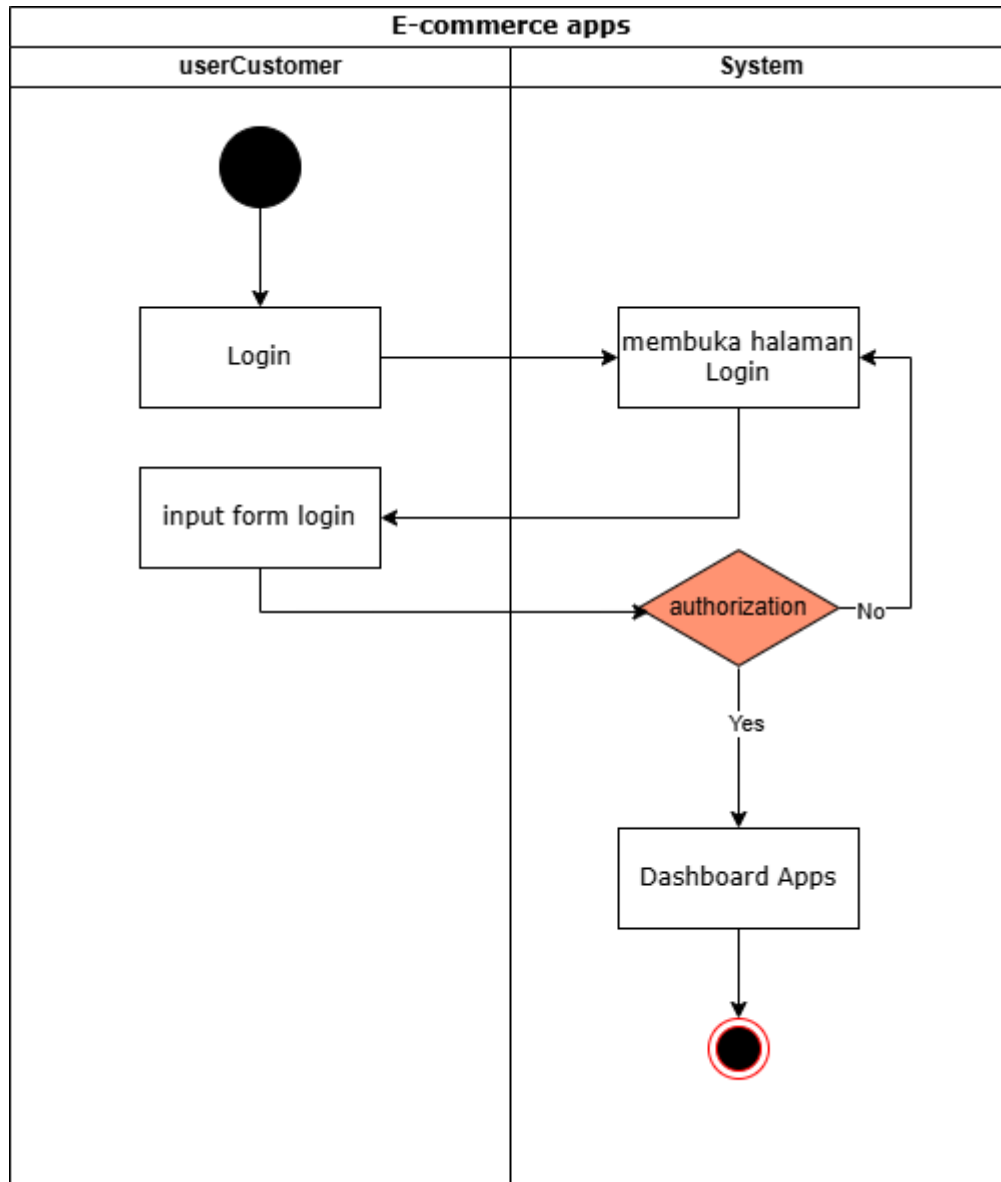
4.1.2 Activity Diagram

Activity Diagram merupakan aktifitas sistem dalam bentuk kumpulan aksi-aksi, bagaimana masing-masing aksi tersebut dimulai, keputusan yang mungkin terjadi hingga berakhirnya aksi (Putra dkk., 2022).

Activity Diagram juga dapat menggambarkan proses lebih dari satu aksi dalam waktu bersamaan. Berikut adalah *Activity Diagram* secara keseluruhan pada aplikasi yang sedang dirancang.

a. Activity Diagram Login

Berikut adalah *Activity Diagram Login* untuk *customer*:

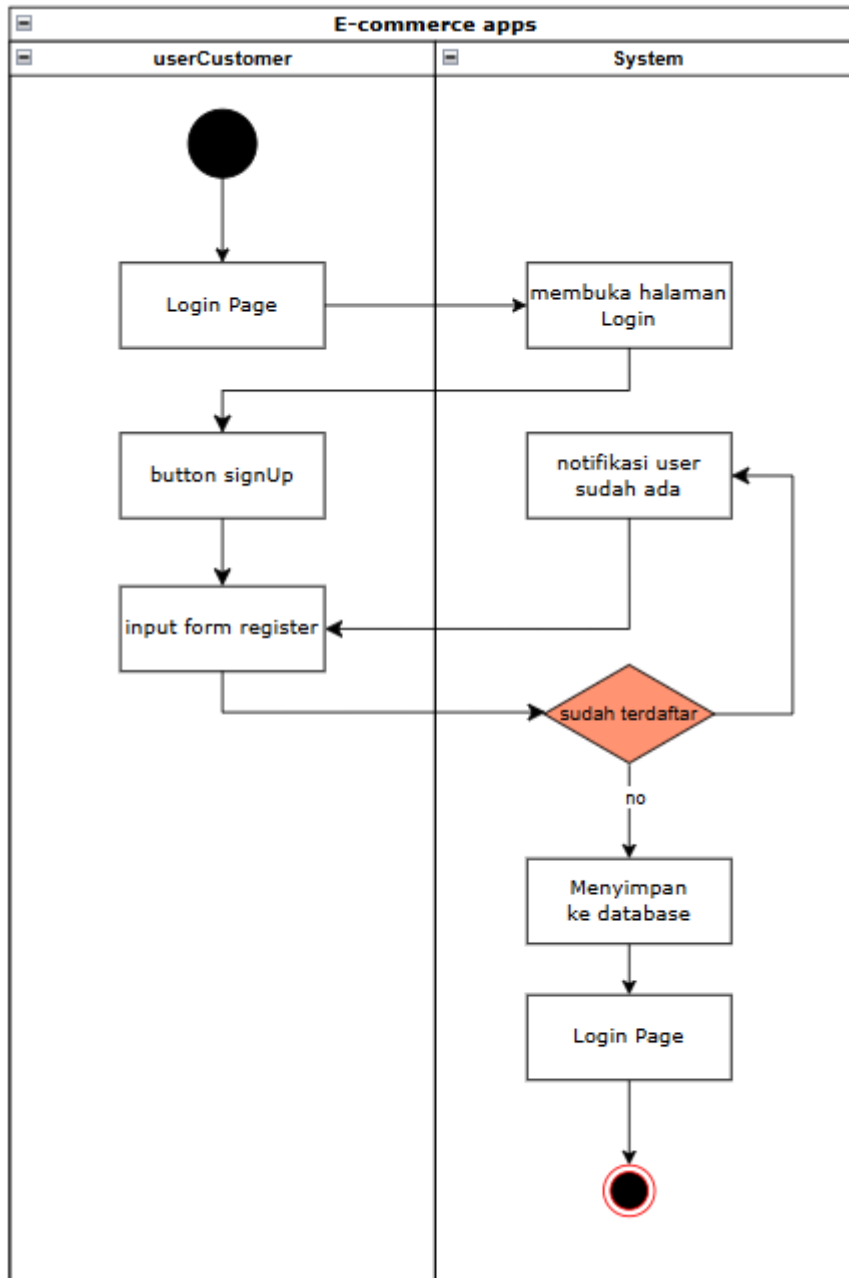


Gambar 4. 2 Activity Diagram Login

Dari gambar diagram aktivitas di atas dapat kita simpulkan alurnya customer menuju ke halaman login kemudian system meresponse dan menampilkan halaman login, setelah berada di halaman login customer menginputkan username dan password pada form login, dan system akan mengauthentikasi data user dan password jika belum terdaftar di database maka akan kembali ke halaman login namun jika username dan password sesuai akan menuju ke halaman dashboard.

b. Activity Diagram Register

Berikut adalah *Activity Diagram Register* untuk *customer*:



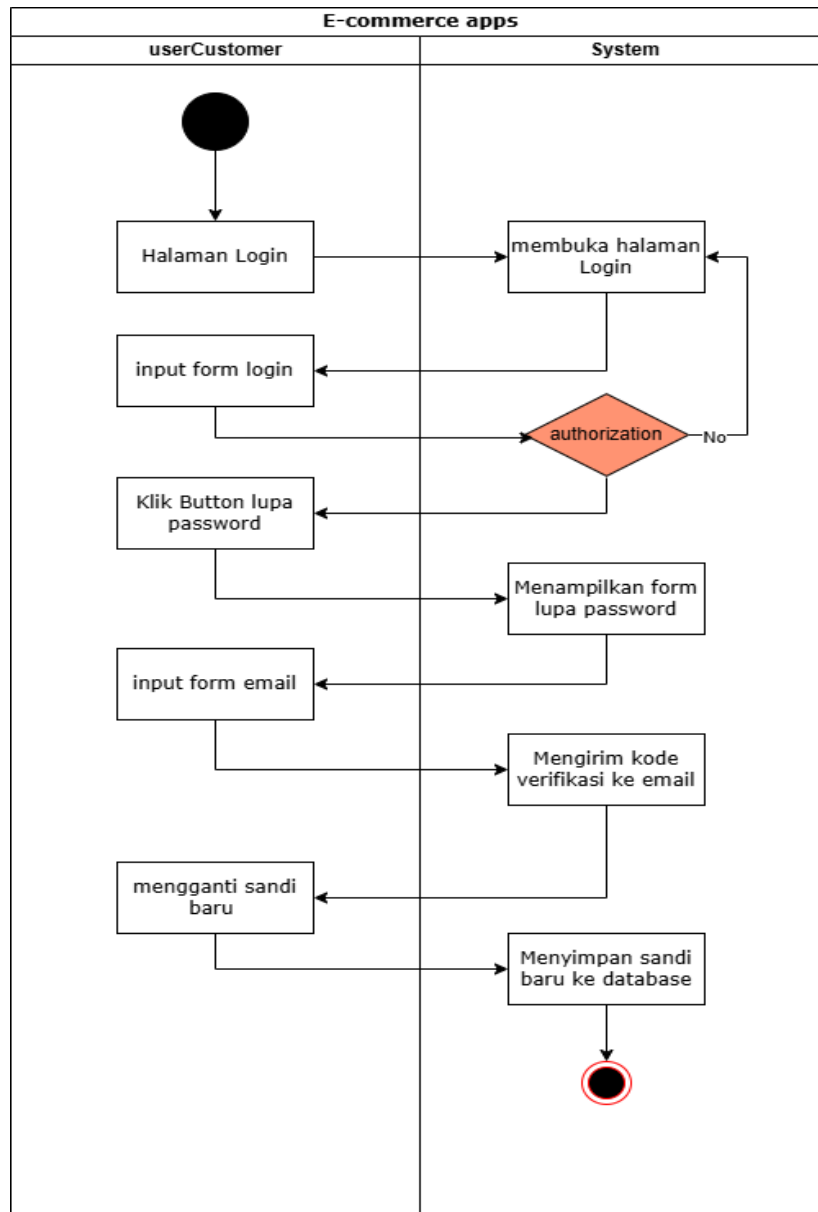
Gambar 4. 3 Activty Diagram Register

Dari diagram aktivitas diatas dapat dijelaskan alur pendaftaran user, user menuju ke halaman login kemudian mengklik button signUp, setelah itu system akan mendriect ke halaman pendaftaran dan user menginputkan data username gmail password untuk di daftarkan kedalam database, jika data sudah pernah di

daftarkan system akan menampilkan nitifikasi bahwa data yg di input sudah ada, namun jika belum pernah terdaftar system akan menyimpan ke dalam database.

c. Activity Diagram Forgot Password

Berikut adalah *Activity Diagram Forgot Password* untuk *customer*:



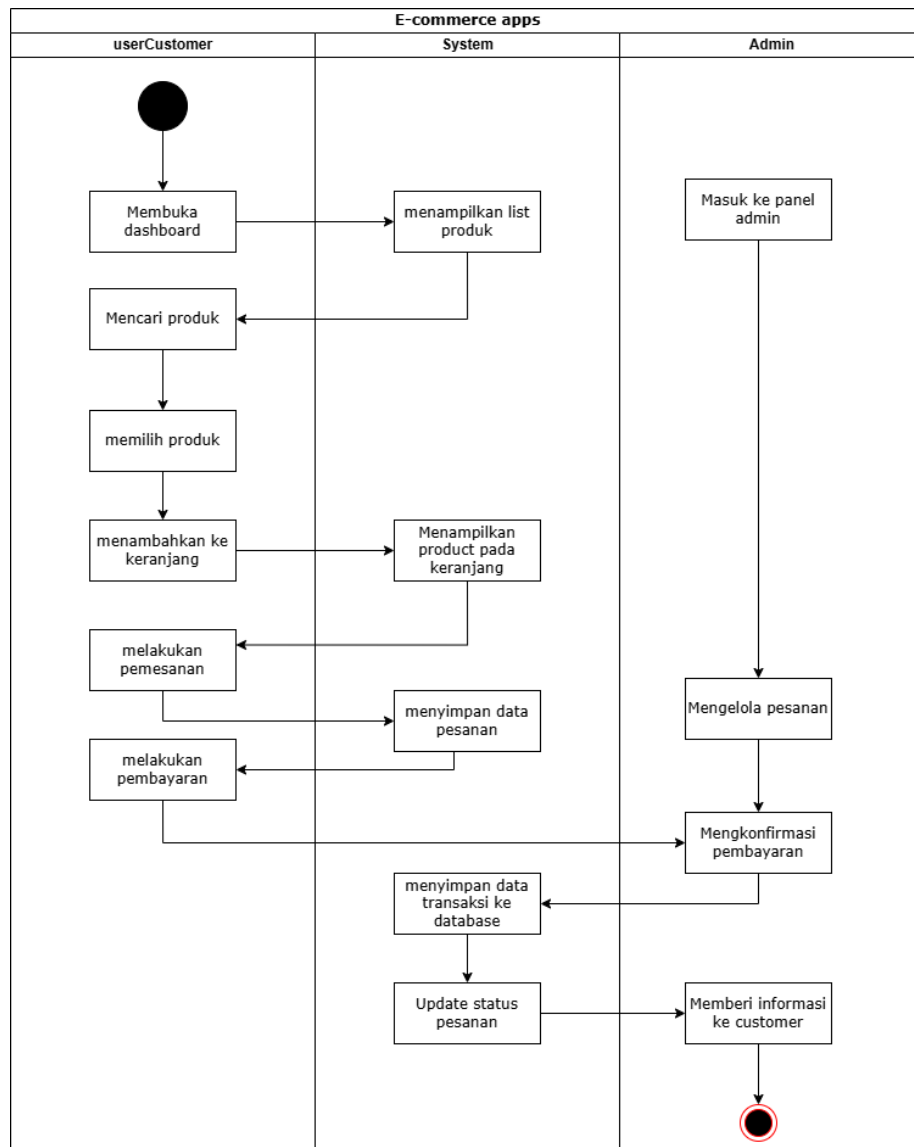
Gambar 4. 4 Activity Diagram Forgot Password

Dari diagram aktivitas diatas dapat disimpulkan, user menekan button lupa password, kemudian system akan menampilkan halaman lupa password setelah itu user menginput data gmail yang terdaftar, kemudian system akan memberikan kode

verifikasi ke gmail yang tadi sudah di input, user mendaftarkan password baru ke database, lalu system mengganti password lama dengan yang baru kemudian disimpan ke dalam database.

d. Activity Diagram Pesanan

Berikut adalah *Activity Diagram* Pesanan:



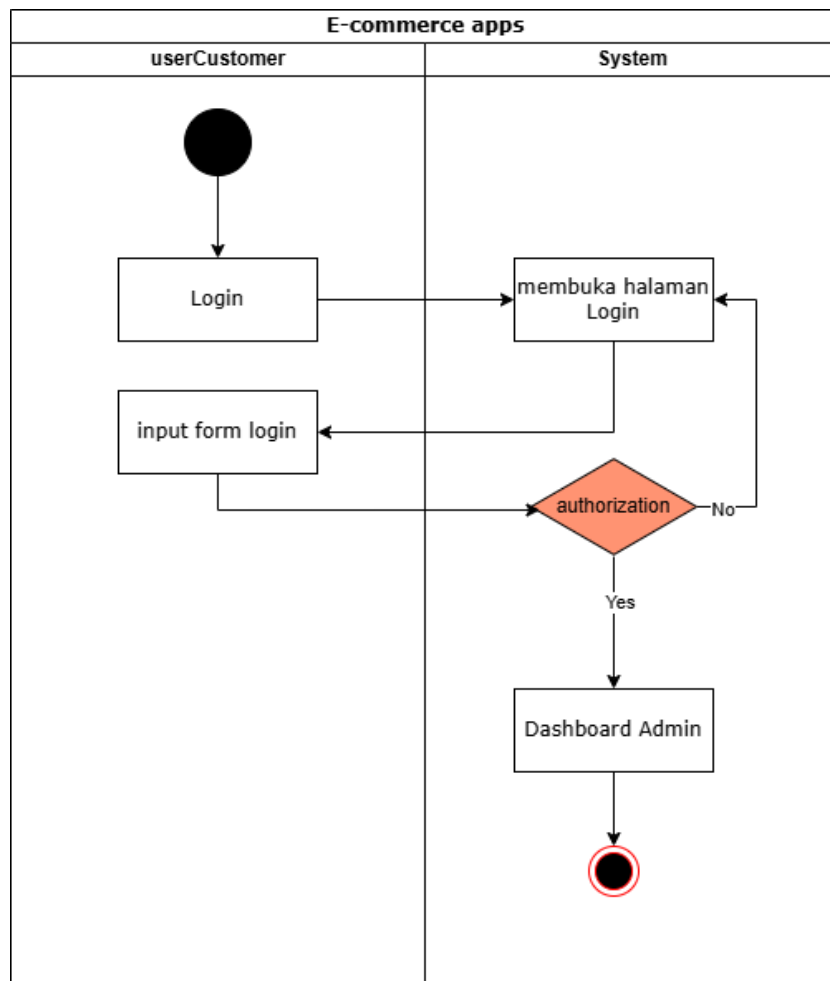
Gambar 4. 5 Activity Diagram Pemesanan

Dari diagram aktivitas diatas dapat disimpulkan alur pemesanan pada aplikasi e-commerce ini, user dimulai dari halaman dashboard system menampilkan list produk, user mencari produk yang di perlukan kemudian memilih beberapa produk,

user menambahkan ke keranjang, kemudian system menampilkan product yang sudah ditambahkan ke keranjang, user melakukan pesanan lalu system menyimpan data pesanan, user melakukan pembayaran kemudian admin mengelola pesanan lalu mengkonfirmasi pembayaran, system menyimpan data transaksi ke database dan mengupdate status pesanan, admin memberikan informasi ke pelanggan.

e. Activity Diagram Admin Login

Berikut adalah *Activity Diagram Login Admin*:



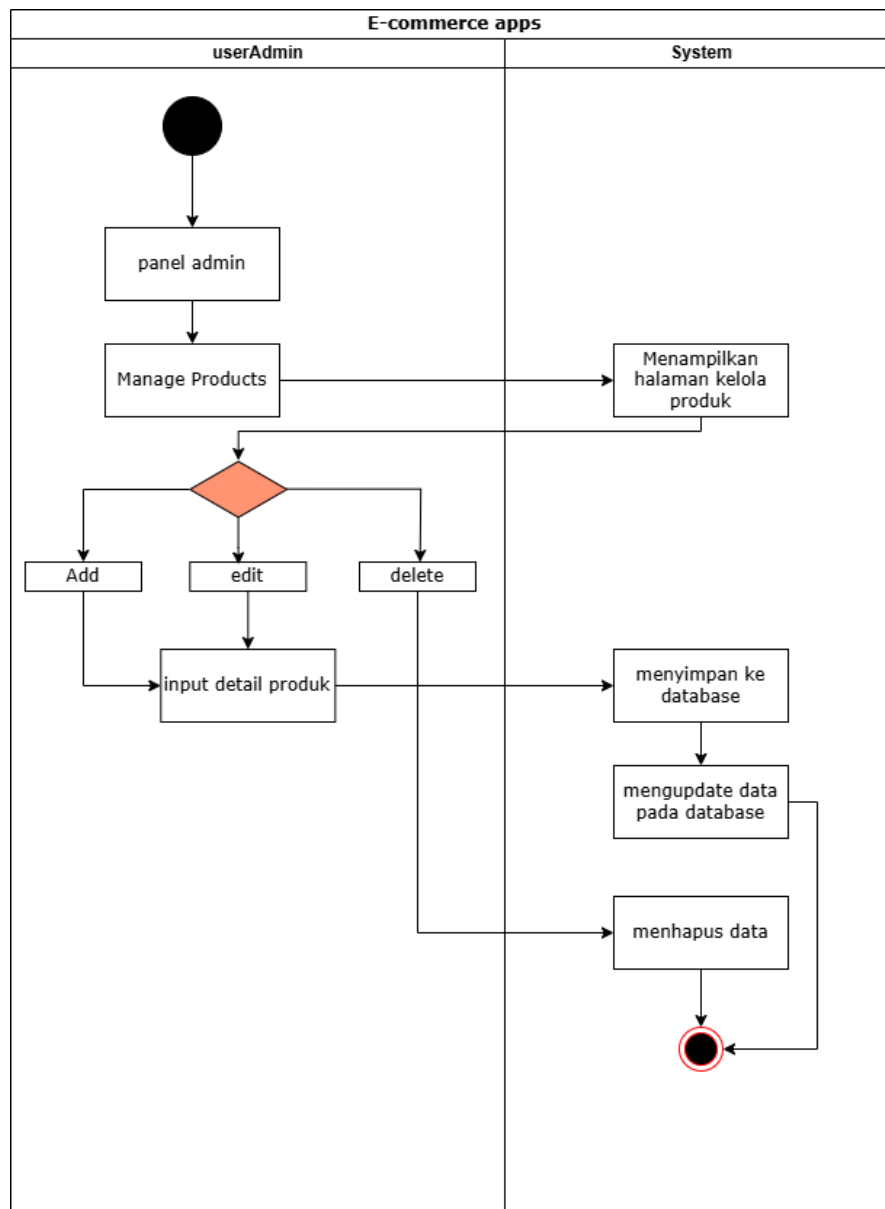
Gambar 4. 6 Activity Diagram Admin Login

Pada diagram aktivitas diatas adapt disimpulkan alur login untuk admin, admin melakukan input data user admin emlalui halaman login panel admin login, setelah itu system akan meresponse authorization jika data yang diamsukkan salah

maka akan kemabli menampilkan halaman login namum jika data yang id inputkan sudah benar akan menuju ke halaman dashboard panel admin.

f. Activity Diagram Admin Manage Product

Berikut adalah Activity Diagram Admin Manage Product:



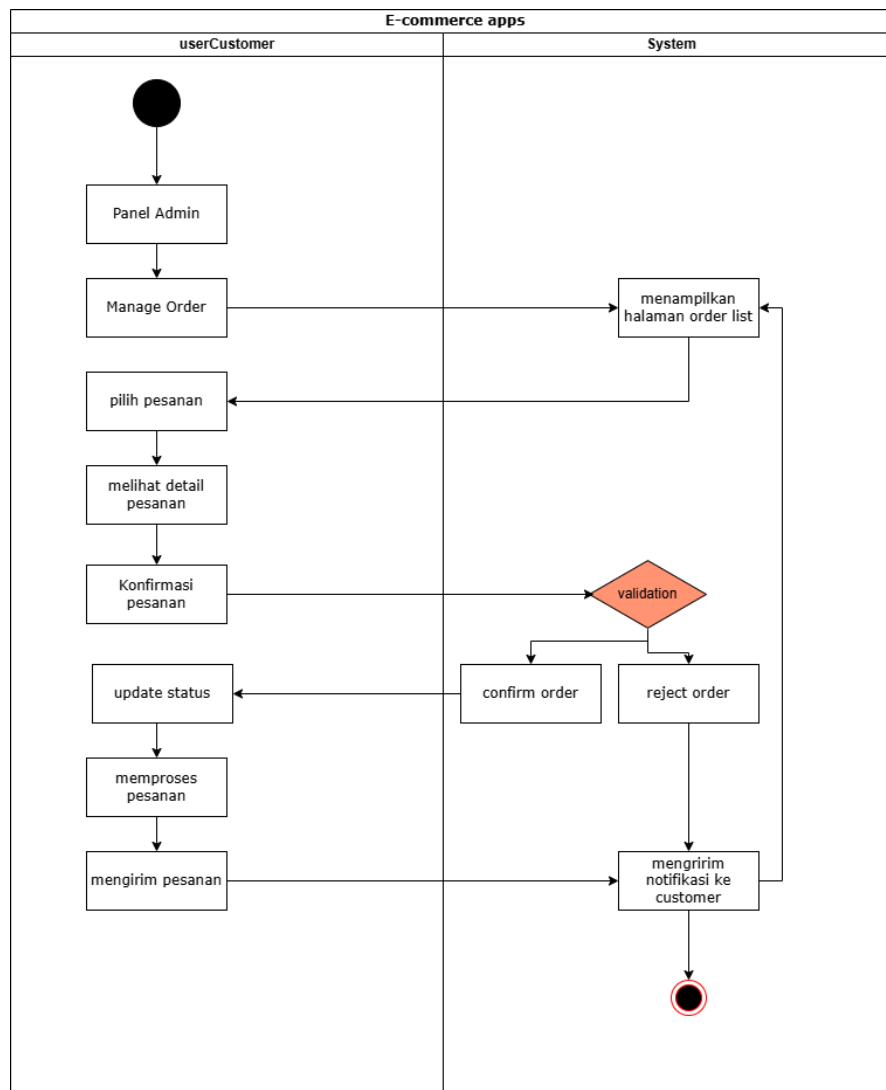
Gambar 4. 7 Activity Diagram Admin Manage Products

Pada activity diagram diatas dapat disimpulkan alur pengelolaan product yang di lakukan oleh admin, mulai dari panel admin system menampilkan halaman manage product, admin menambahkan produk dengan mengisi form produk termasuk detail

dan gambar serta harga produk kemudia system pemnyimpan ke database, admin melakukan edit product, system menampilkan list product kemudian admin menginput data yang ingin diubah setelah itu system mengapdate data pada database, admin melakukan delete system menghapus data pada database.

g. Activity Diagram Admin Manage Order

Berikut adalah *Activity Diagram Admin Manage Order*:



Gambar 4. 8 Activity Diagram Admin Manage Order

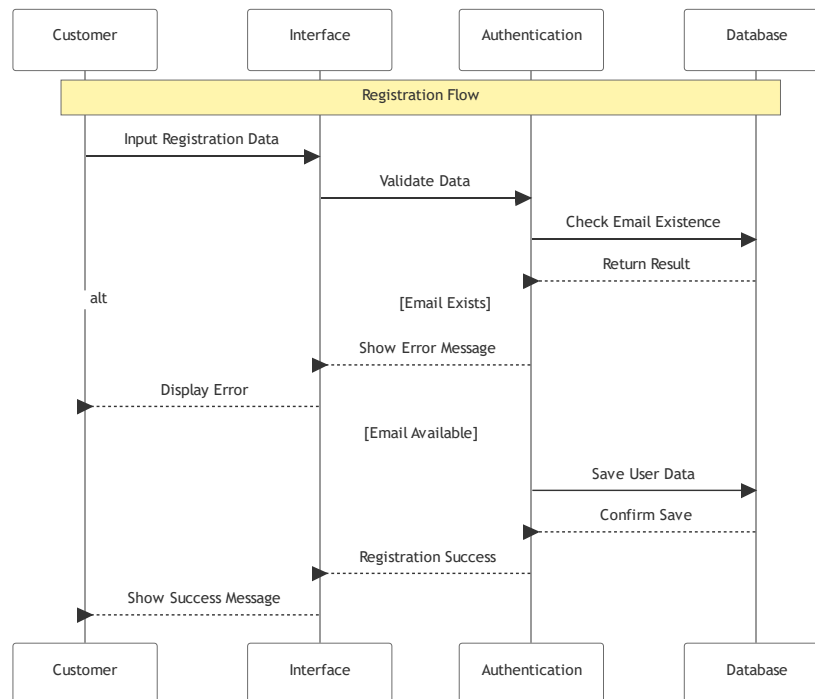
Admin memulai dengan meninjau dan memvalidasi pesanan yang masuk. Mereka dapat melihat daftar pesanan dan memilih pesanan tertentu untuk diperiksa lebih lanjut sebelum memvalidasinya. Setelah validasi, admin dapat mengonfirmasi

atau menolak pesanan. Pesanan yang dikonfirmasi akan melanjutkan ke tahap konfirmasi, di mana admin mengubah statusnya menjadi "Confirmed" dan memberitahu pelanggan. Jika pesanan ditolak, statusnya akan menjadi "Rejected" dan alasan penolakan akan dikirimkan kepada pelanggan. Pesanan yang dikonfirmasi dapat diupdate statusnya sesuai tahapan proses, dengan notifikasi yang dikirim setiap kali ada perubahan status agar pelanggan dapat memantau pesannya.

4.1.3 Sequence Diagram

Sequence Diagram adalah diagram yang dirancang untuk mengetahui alur asal hubungan antar objek. Berikut adalah *sequence diagram* dari aplikasi yang sedang dirancang:

a. Register

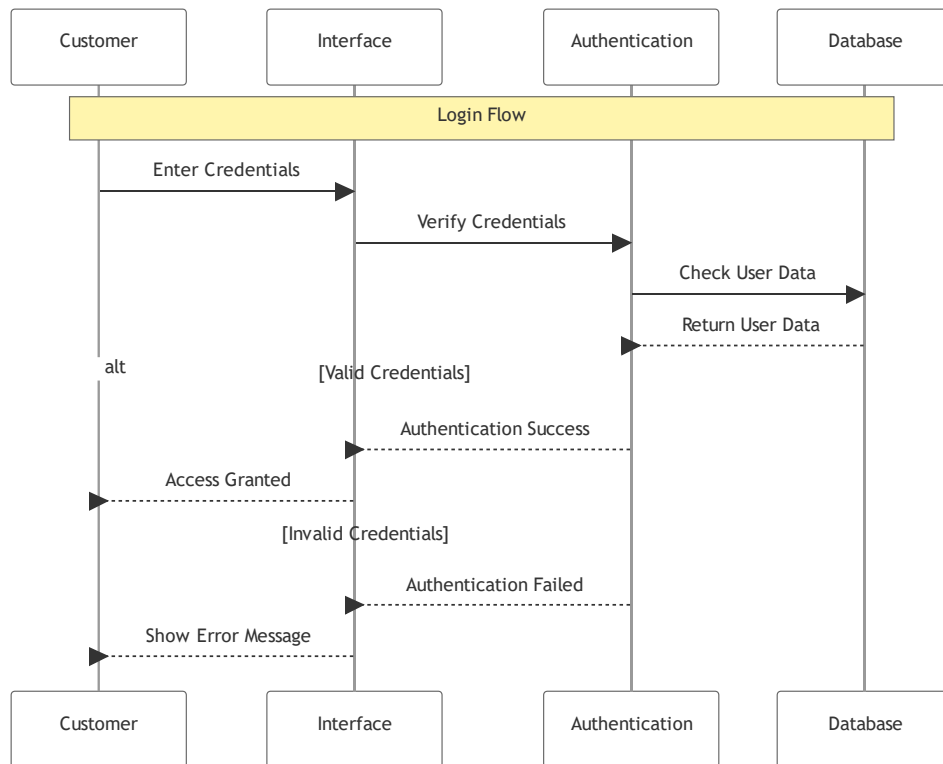


Gambar 4. 9 Sequence Diagram Register

Proses dimulai ketika customer memasukkan data registrasi melalui antarmuka aplikasi. Sistem kemudian melakukan validasi data, khususnya memeriksa keberadaan email dalam database untuk menghindari duplikasi akun.

Jika email telah terdaftar, sistem akan menampilkan pesan error. Jika email tersedia, data pengguna baru akan disimpan dalam database dan sistem menampilkan pesan sukses kepada customer.

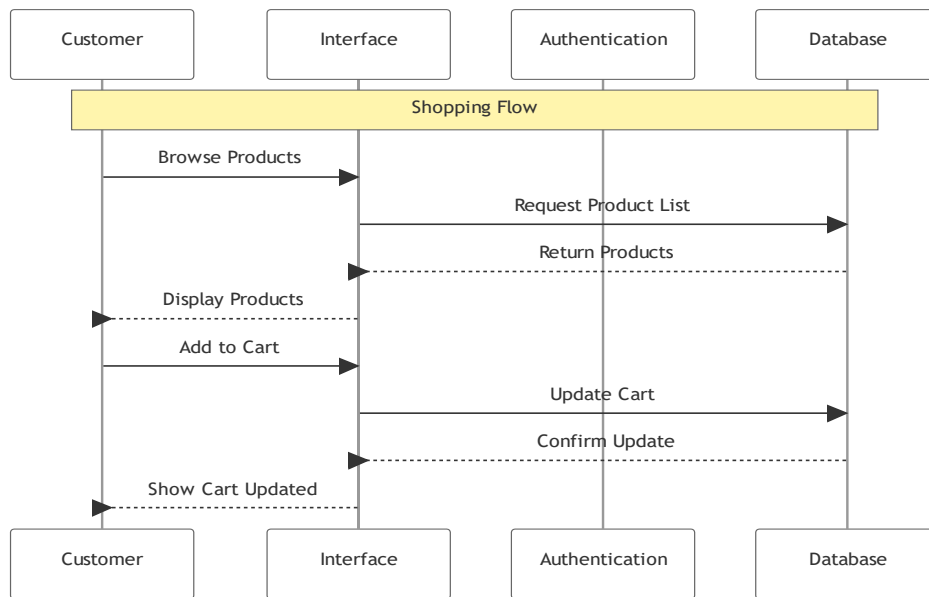
b. Login



Gambar 4. 10 Sequeunce Diagram Login

Customer memasukkan kredensial (email dan password) melalui antarmuka. Sistem melakukan verifikasi dengan memeriksa data di database. Jika kredensial valid, sistem memberikan akses dan mengarahkan customer ke halaman utama. Jika tidak valid, sistem menampilkan pesan error dan meminta customer untuk mencoba kembali.

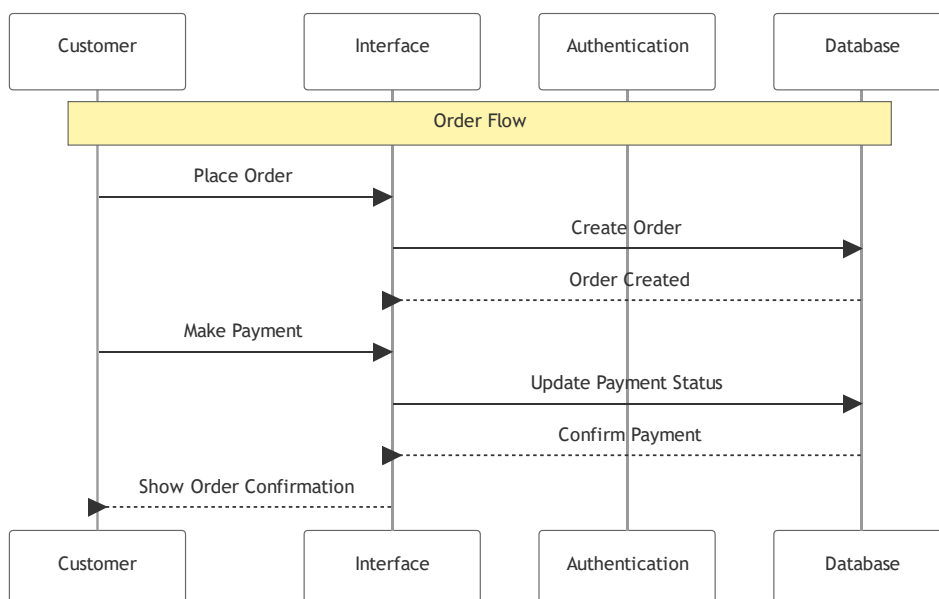
c. Belanja



Gambar 4. 11 Sequeunce Diagram Belanja

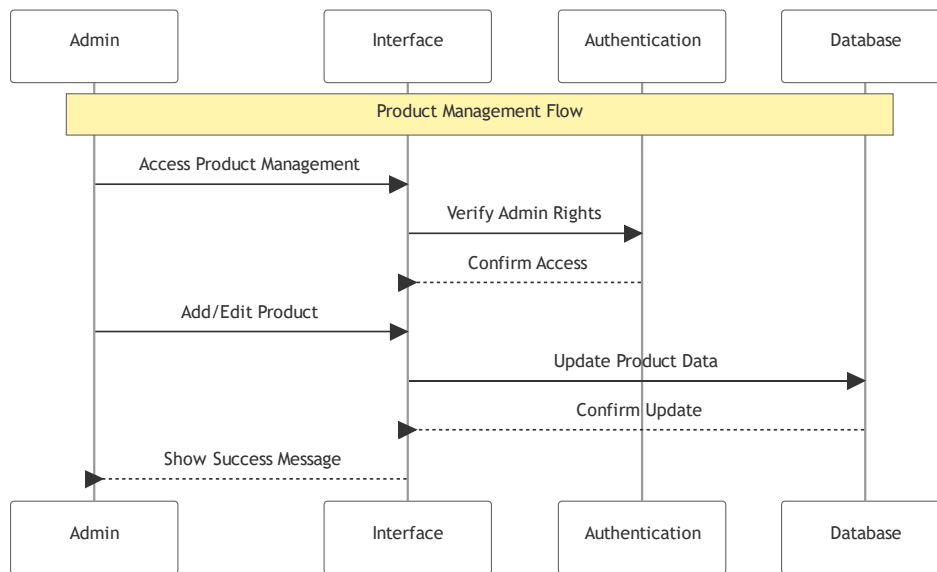
Setelah login berhasil, customer dapat menjelajahi katalog produk yang ditampilkan sistem dari database. Ketika customer memilih untuk menambahkan produk ke keranjang, sistem memperbarui data keranjang dalam database dan menampilkan konfirmasi kepada customer. Customer dapat melihat dan mengelola isi keranjang sebelum melanjutkan ke proses pemesanan.

d. Pemesanan



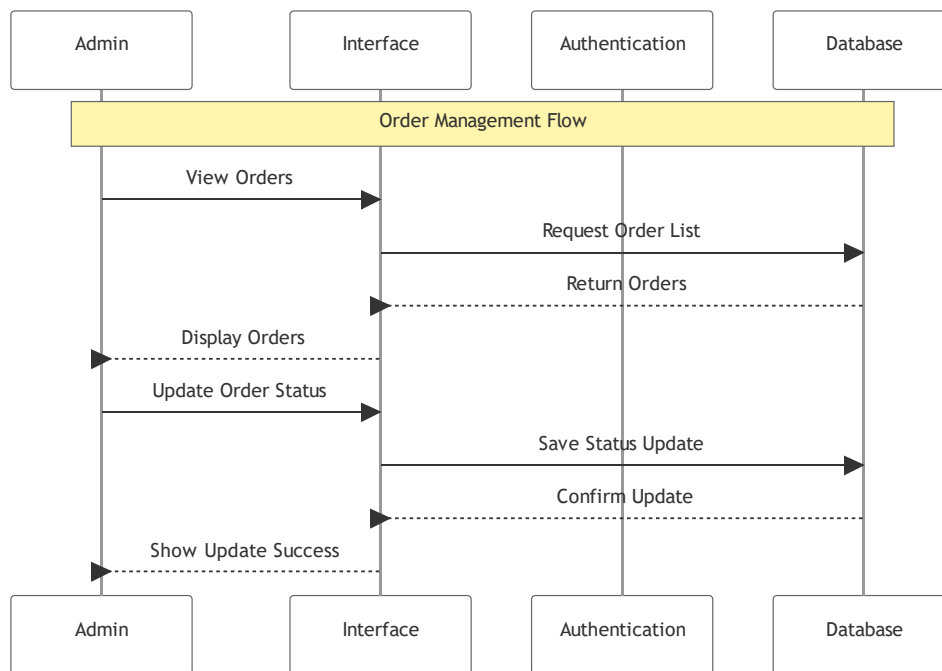
Gambar 4. 12 *Sequeunce Diagram Pemesanan***e. Kelola Produk**

Ketika customer memutuskan untuk melakukan pemesanan, sistem mencatat order baru dalam database. Customer kemudian melakukan pembayaran, dan sistem memperbarui status pembayaran. Setelah pembayaran dikonfirmasi, sistem mengirimkan konfirmasi order kepada customer.

**Gambar 4. 13 *Sequeunce Diagram Kelola Produk***

Admin mengakses panel manajemen produk setelah verifikasi hak akses. Admin dapat menambah, mengedit, atau menghapus produk. Setiap perubahan data produk langsung diperbarui dalam database dan sistem memberikan konfirmasi keberhasilan operasi.

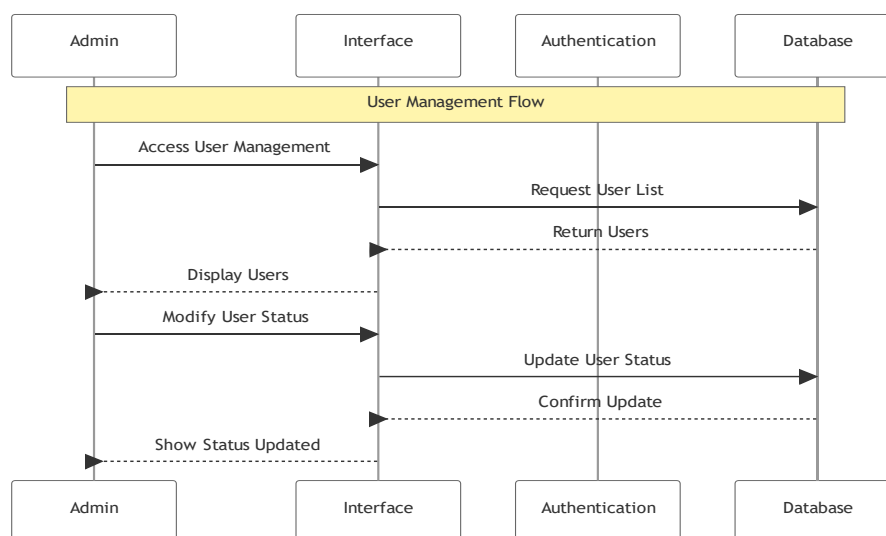
f. Kelola Pesanan



Gambar 4. 14 *Sequeunce Diagram* Kelola Pesanan

Admin dapat melihat daftar pesanan yang masuk melalui sistem. Ketika admin memperbarui status pesanan, sistem menyimpan perubahan dalam database dan mengirimkan notifikasi kepada customer terkait. Sistem juga memberikan konfirmasi pembaruan status kepada admin.

g. Kelola User

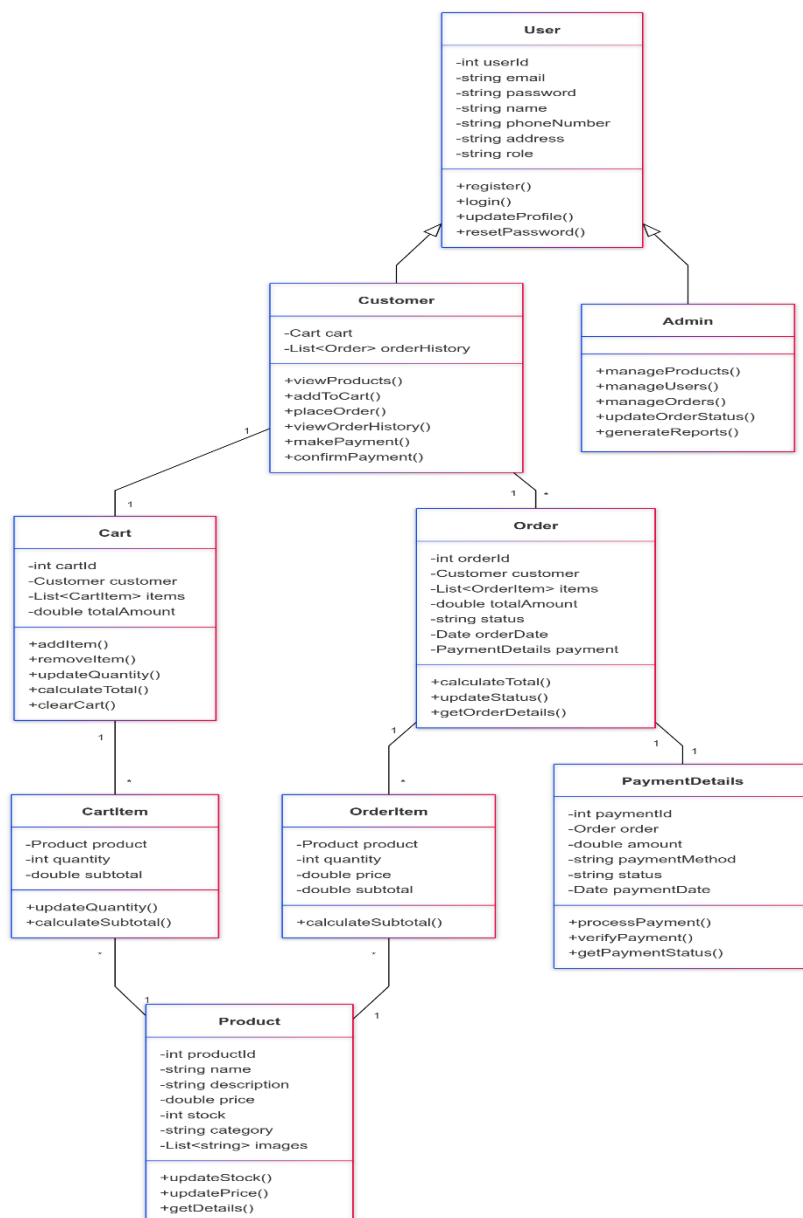


Gambar 4. 15 *Sequeunce Diagram* Kelola User

Admin dapat mengakses daftar pengguna sistem dan melihat informasi detail mereka. Ketika admin melakukan modifikasi status pengguna, sistem memperbarui data dalam database dan memberikan konfirmasi perubahan. Admin dapat mengelola hak akses dan status aktif pengguna dalam sistem.

4.1.4 Class Diagram

Diagram Class berfungsi untuk mendeskripsikan struktur sistem asal segi pendefinisian kelas-kelas yang akan dirancang untuk menciptakan sistem. Berikut adalah *class diagram* dari aplikasi yang sedang dirancang:



Gambar 4. 16 *Class Diagram* Aplikasi Woodman

Class diagram di atas menggambarkan struktur sistem e-commerce Woodman.Project dengan detail sebagai berikut:

a. User (Class Diagram)

Menyimpan informasi dasar pengguna seperti userId, email, password, dan data pribadi Menangani fungsionalitas umum seperti registrasi dan login Berfungsi sebagai kelas induk untuk Customer dan Admin.

1. Mengelola aktivitas spesifik pelanggan
2. Memiliki relasi dengan Cart dan Order
3. Menangani fungsi pembelian dan pembayaran

b. Admin (turunan User)

1. Mengelola fungsi administratif sistem
2. Mengontrol produk, pengguna, dan pesanan
3. Memiliki akses ke fungsi pelaporan

c. Product:

1. Menyimpan informasi detail produk
2. Mengelola stok dan harga
3. Terhubung dengan CartItem dan OrderItem

d. Cart dan CartItem:

1. Mengelola keranjang belanja
2. Menghitung total belanja
3. Memungkinkan modifikasi item

e. Order dan OrderItem:

1. Mencatat detail pesanan
2. Mengelola status pesanan
3. Terhubung dengan sistem pembayaran

f. PaymentDetails:

1. Menangani informasi pembayaran
2. Memproses dan memverifikasi pembayaran
3. Melacak status pembayaran

Hubungan antar kelas:

1. Inheritance: Customer dan Admin mewarisi User
2. Composition: Cart dimiliki oleh Customer
3. Association: Order terhubung dengan Customer dan PaymentDetails
4. Aggregation: Cart dan Order memiliki koleksi Item

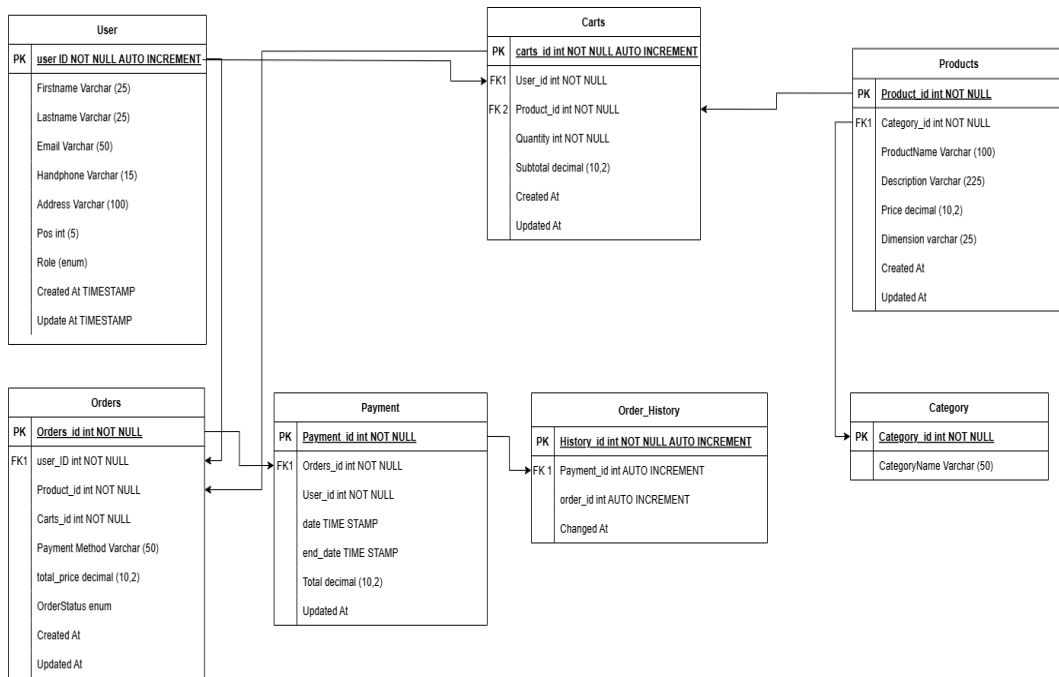
Diagram ini memberikan panduan komprehensif untuk pengembangan sistem, menunjukkan struktur data dan hubungan antar komponen dengan jelas.

4.1.5 Perancangan Basis Data

Perancangan basis data untuk aplikasi forum diskusi yang bertujuan meningkatkan pembelajaran teknologi informasi di Fullstuck University dengan menggunakan *Entity Relationship Diagram* (ERD), mengubah ERD menjadi *Logical Record Structure* (LRS), dan menampilkan LRS.

a. Entity Relationship Diagram (ERD)

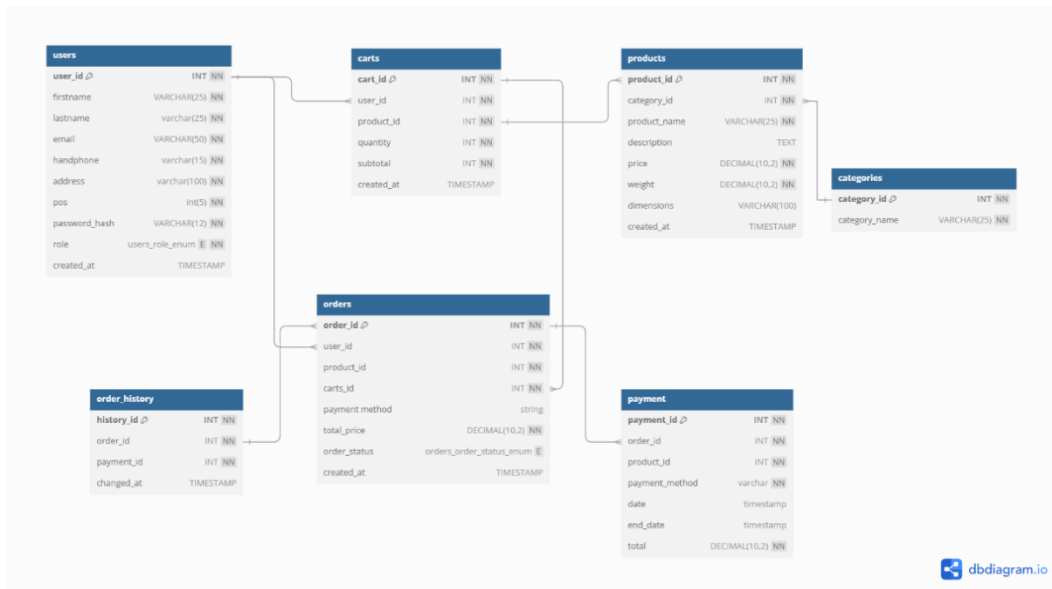
ERD adalah teknik yang menggambarkan hubungan antara entitas dan relasi dalam suatu model. Fokus utama diagram ER adalah menyorot entitas data dan hubungan antara entitas tersebut. Gambaran ERD untuk aplikasi yang sedang dirancang sebagai berikut:



Gambar 4. 17 Entity Relationship Diagram (ERD)

b. Logical Record Structure (LRS)

Logical Record Structure (LRS) pada aplikasi yang sedang dirancang adalah sebagai berikut:



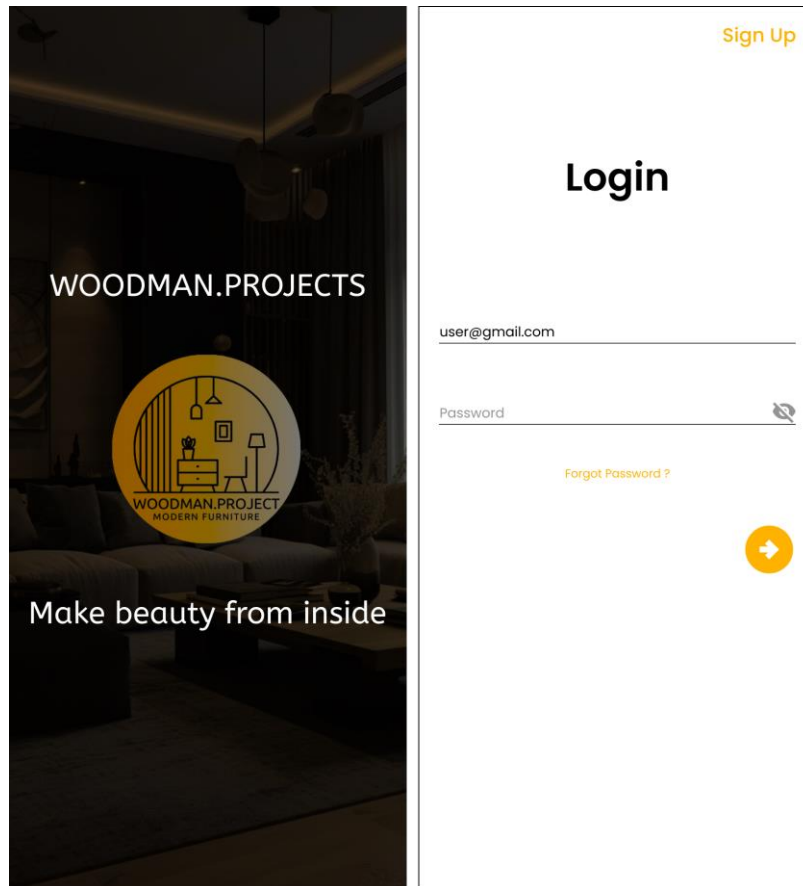
Gambar 4. 18 Logical Record Structure (LRS)

4.2 Implementasi

Setelah melalui tahapan perancangan, selanjutnya adalah tahapan implementasi. Implementasi dilakukan dengan menuliskan sebuah *code* untuk menciptakan aplikasi sesuai dengan tahapan-tahapan sebelumnya. Berikut di bawah ini hasil implementasi dari proses pemesanan sampai dengan selesai:

4.2.1 Halaman Login User

Pada halaman Login User diharuskan memasukkan Email dan Password yang telah terdaftar pada sistem untuk masuk ke halaman utama.



Gambar 4. 19 Halaman Login User

4.2.2 Halaman Register

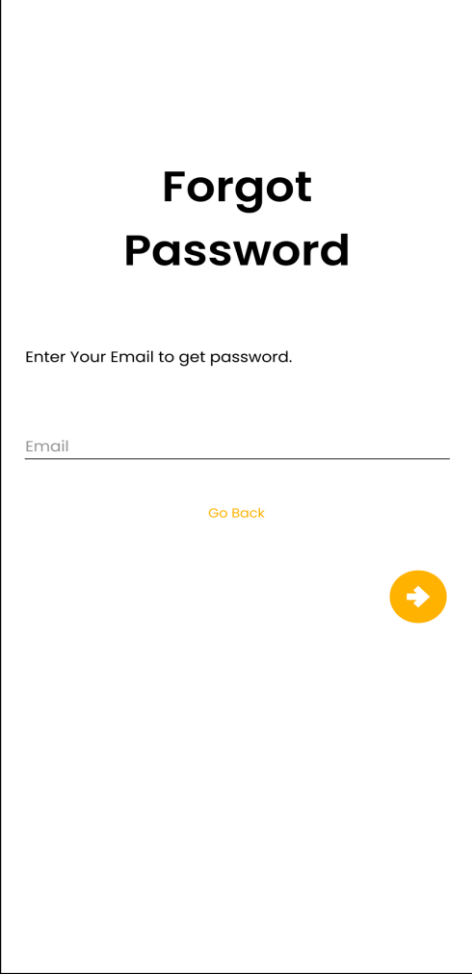
Sebelum melakukan Login user diharuskan memiliki akun untuk masuk kedalam system, pada halaman pendaftaran terdapat form data untuk yang akan diisi oleh user dan disimpan kedalam database guna memvalidasi akun sebelum login.

A screenshot of a 'Sign Up' registration form. The form is titled 'Sign Up' in a large, bold, black font. Below the title, there are several input fields: 'Email', 'Phone', 'First Name', and 'Last Name'. Each field has a horizontal line for text entry. Below the 'First Name' and 'Last Name' fields, there are two password fields, each with a horizontal line and a small eye icon to the right. The first password field contains a series of asterisks. The second password field contains the text 'afrizalprayogi02020212@'. At the bottom right of the form, there is a yellow circular button with a white right-pointing arrow.

Gambar 4. 20 Halaman Pendaftaran

4.2.3 Halaman Lupa Password

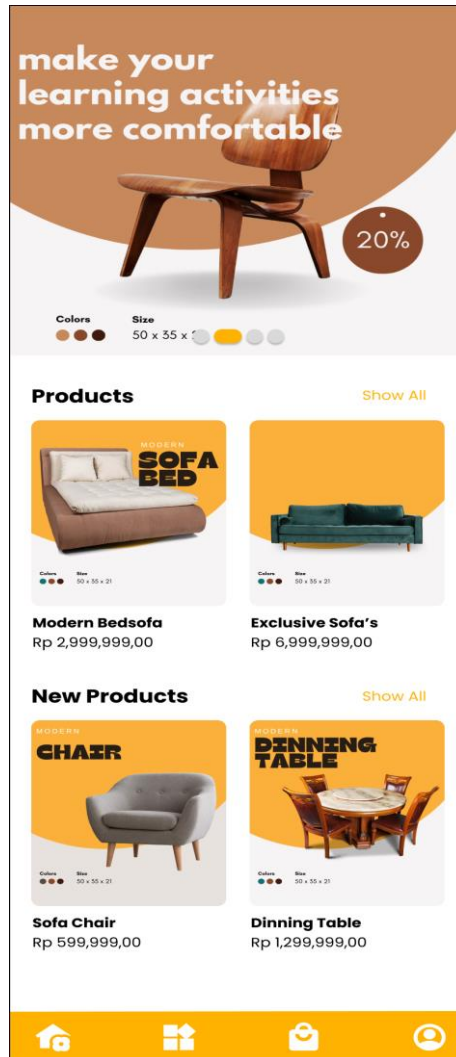
Setelah user memiliki akun terkadang sering terjadi human eror seperti kehilangan akun atau lupa kata sandi, maka dari itu halaman untuk mengembalikan akun yang bermasalah karna lupa katasandi diperlukan, prosesnya memerlukan gmail yang sudah di daftarkan kemudian system akan mengirimkan pembaruan password ke gmail.

A mobile app screen for the 'Forgot Password' feature. At the top, the title 'Forgot Password' is centered in a large, bold, black font. Below the title, a smaller line of text reads 'Enter Your Email to get password.' Underneath this is a text input field with the placeholder text 'Email'. To the right of the input field is a yellow circular button with a white right-pointing arrow. Above the button, the text 'Go Back' is displayed in a small, orange font.

Gambar 4. 21 Halaman Lupa Password

4.2.4 Halaman Beranda

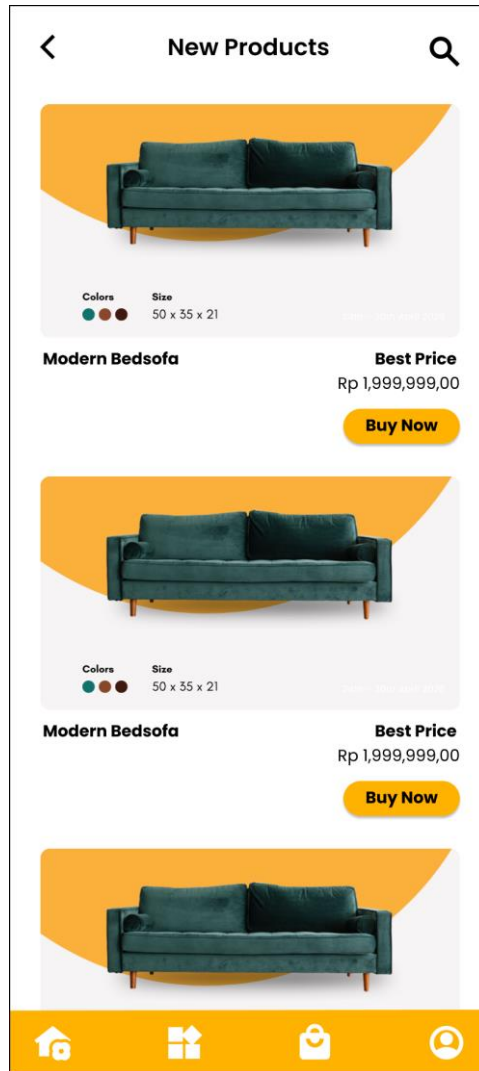
Halaman beranda disini akan menampilkan produk – produk yang di jual oleh woodman.project terdapat beberapa display dan beberapa fitur diantaranya adalah produk terbaru dan produk, selain itu ada beberapa fitur untuk menavigasi ke halaman lainnya seperti menu cart dan profil pengguna.



Gambar 4. 22 Halaman Beranda

4.2.5 Halaman New Products

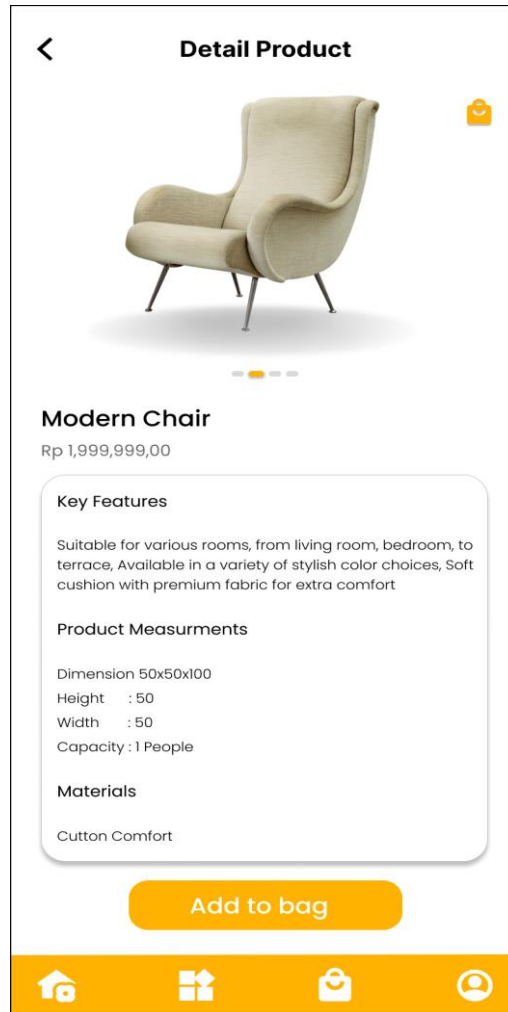
Halaman produk baru berguna untuk menampilkan produk terupdate yang baru dirilis oleh Woodman.project, terdapat informasi harga dan button buy yang akan menavigasi ke halaman order.



Gambar 4. 23 Halaman Produk Baru

4.2.6 Halaman Detail Products

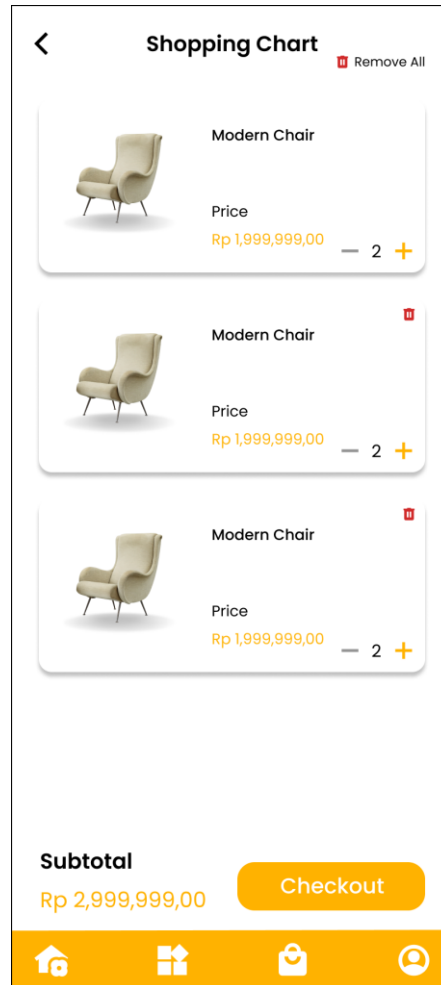
Halaman detail produk memberikan informasi bagi user yang diantaranya adalah informasi detail terkait tentang produk seperti kelebihan produk, berat dan ukuran produk serta material, halaman ini memiliki button Add to bag untuk menyimpan kedalam keranjang.



Gambar 4. 24 Halaman Detail Produk

4.2.7 Halaman Shopping Chart

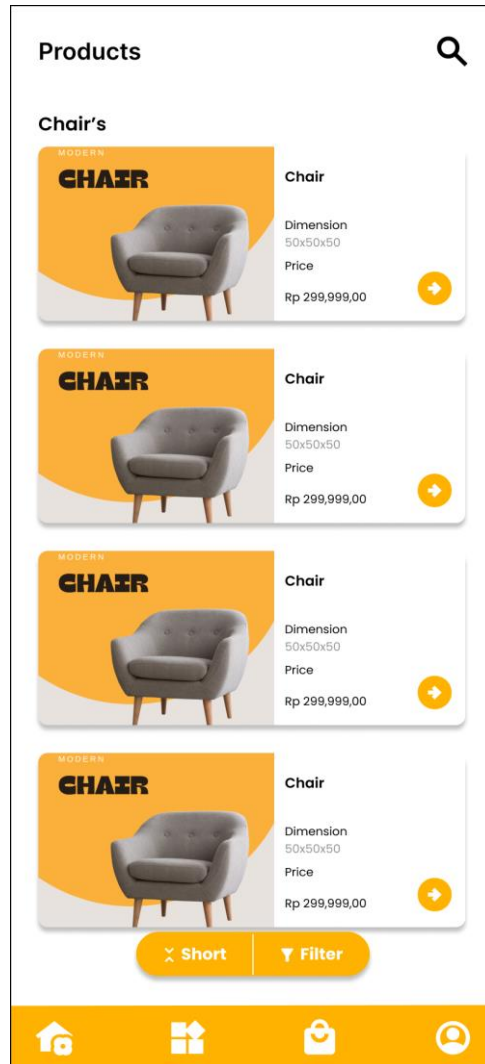
Halaman Shopping Chart memungkinkan pengguna untuk menampung beberapa barang yang sudah dipilih oleh pengguna, menambah serta mengurangi jumlah pembelian, menghapus item yang dipilih dan menampilkan harga, kemudian dapat di pesan/Checkout secara bersamaan.



Gambar 4. 25 Halaman Shopping Chart

4.2.8 Halaman Product

Halaman Produk menampilkan produk berikut beberapa informasi tambahan seperti harga, nama barang dan volume/dimensi, pada halaman ini terdapat fitur memfilter berdasarkan harga tertinggi – harga terendah dan memperkecil pilihan berdasarkan kategori produk guna memudahkan user dalam proses pemilihan produk.



Gambar 4. 26 Halaman Produk

4.2.9 Halaman Checkout

Halaman Checkout berisi order summary yang sudah akan terisi otomatis oleh system berdasarkan barang yang di pesan melalui halaman chart, kemudian ada Form untuk mengisi alamat guna memberikan informasi ke admin bahwa produk akan dikirim ketujuan sesuai dengan yang di deskripsi.

<

Checkout

Order Summary

Products

Modern Chair

Details :

Count

x2

Color

White

Price

Rp 1,499,999,00

Sub Total

Rp 2,999,999,00

Delivery Fee

Rp 0,00

Order Total

Rp 2,999,999,00

Shipping Address

Your Delivery Address

Your name :

Phone :

Address :

Country :

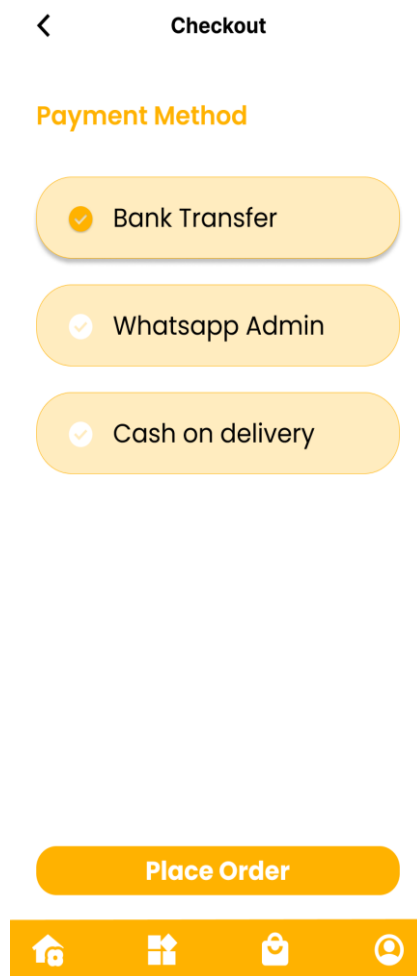
Pos Number :

Next

Gambar 4. 27 Halaman Cecheckout_Order

4.2.10 Halaman Place Order

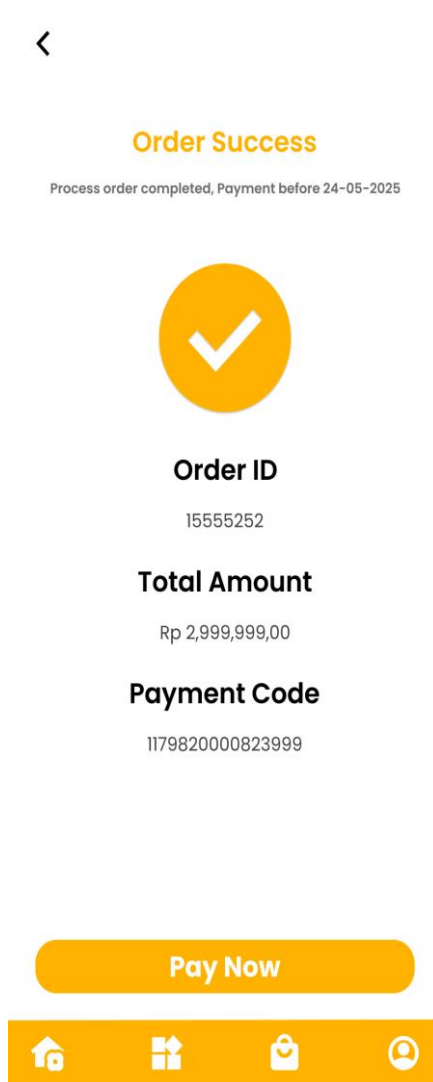
Pada halaman ini user memilih metode pembayaran ada beberapa metode pembayaran diantaranya pembayaran melalui Bank Transfer, pembayaran melalui menghubungi kontak admin, kemudian pembayaran cash on delivery.



Gambar 4. 28 Halaman Place Order

4.2.11 Halaman Order Success

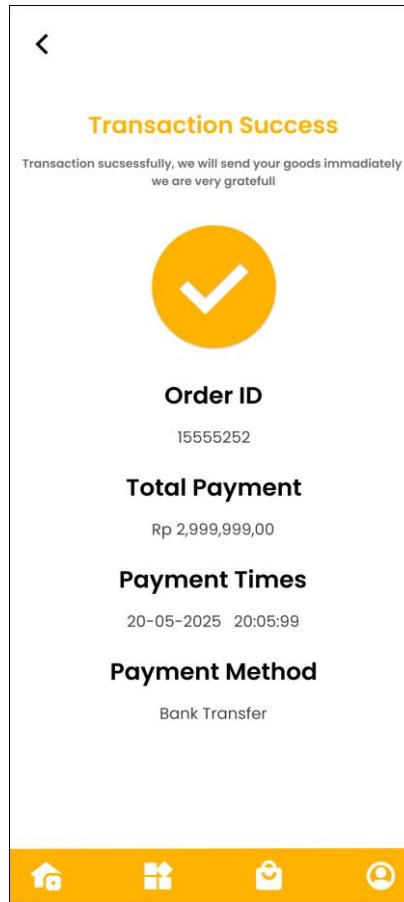
Pada halaman ini memberikan informasi kepada user bahwa produk yang di pesan sudah di validasi oleh admin system, kemudian halaman ini mengeluarkan payment code guna proses lebih lanjut.



Gambar 4. 29 Halaman OrderSukses

4.2.12 Halaman Payment Success

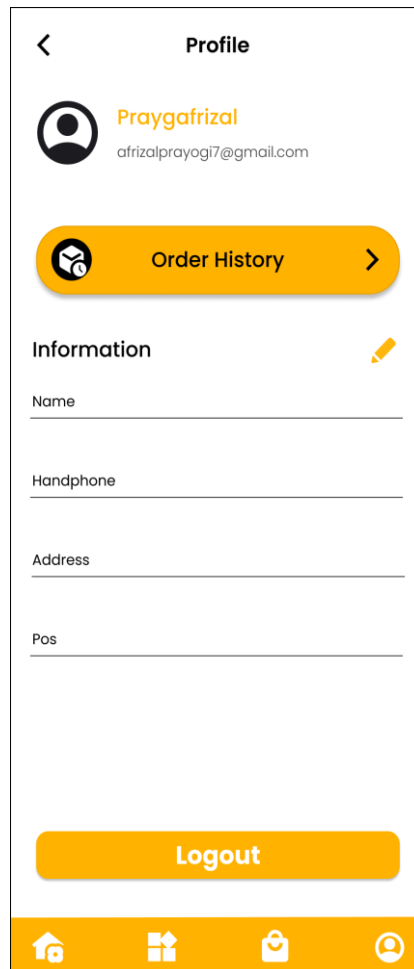
Pada halaman ini proses order dan pembayaran sudah divalidasi oleh admin, memeberikan informasi kepada user waktu dan tanggal pembayaran, total pembayaran dan waktu pembayaran.



Gambar 4. 30 Halaman Pembayaran Berhasil

4.2.13 Halaman Profil

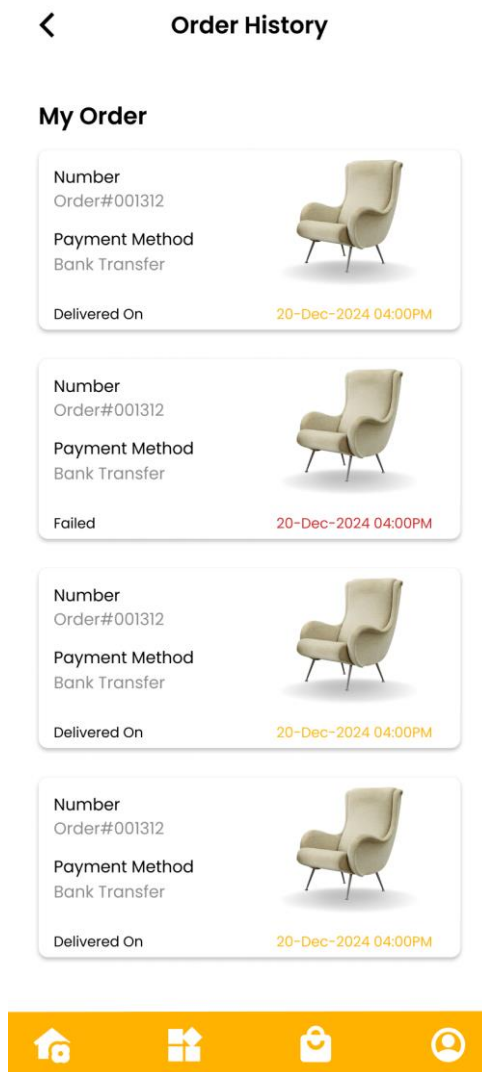
Halaman profil memberikan informasi seputar user, terdapat button history pada halaman ini yang akan menavigasi ke halaman history.



Gambar 4. 31 Halaman Profile

4.2.14 Halaman Order History

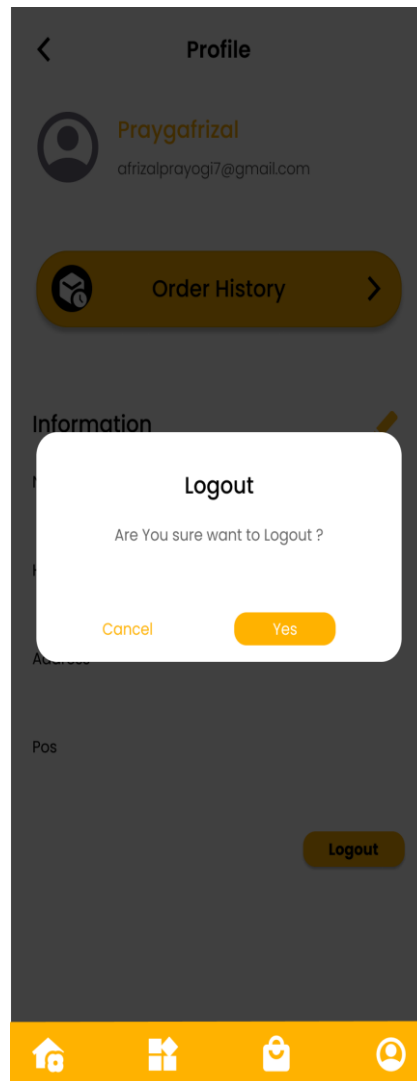
Halaman order history menampilkan riwayat pembelian serta informasi status, harga barang yang dibeli, barang yang dibeli serta pemmetode pembayaran yang dilakukan. Menampung semua riwayat pembelian berikut informasinya.



Gambar 4. 32 Halaman Order History

4.2.15 Halaman Logout

Halaman logout berisi button logout yang ketika di klik akan memunculkan notifikasi logout memungkinkan pengguna keluar dari system.



Gambar 4. 33 Halaman Logout

4.3 Pengujian Sistem

Testing atau uji coba merupakan tahapan dari metodologi Agile. Setelah menyelesaikan tahap perencanaan, perancangan, dan pengkodean, langkah selanjutnya adalah melakukan pengujian menggunakan pendekatan *black-box* atau *white-box*. Tujuan utama dari tahap ini adalah untuk memverifikasi apakah setiap fitur berfungsi seperti yang diharapkan.

Pengujian sistem merupakan proses yang dilakukan untuk mengukur kinerja, keandalan, dan kesesuaian suatu sistem atau perangkat dengan persyaratan dan spesifikasi yang telah ditetapkan sebelumnya. Tujuan dari pengujian sistem adalah

untuk memastikan bahwa sistem atau perangkat tersebut dapat berfungsi dengan baik dan memenuhi kebutuhan pengguna.


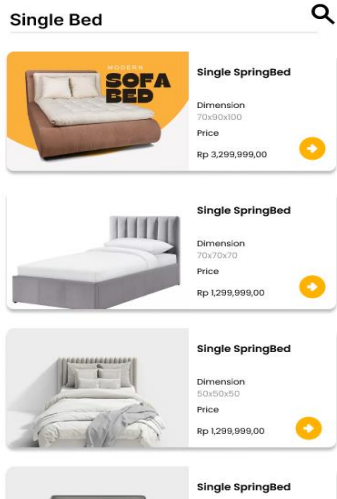
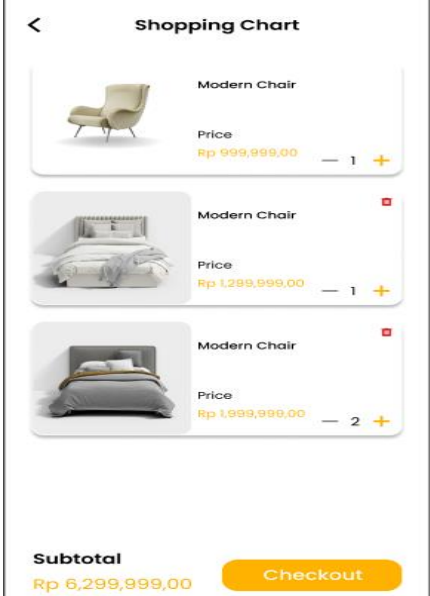
4.3.1 Black-Box Testing




Black box testing adalah metode pengujian perangkat lunak yang berfokus pada fungsionalitasnya. Pengujian ini dilakukan tanpa melihat kode program atau struktur internal perangkat lunak.

a. Black Box Testing Pengguna Pelanggan

Tabel 4. 1 *Black-Box Testing Customer*

No	Skenario Pengujian	Hasil yang di harapkan	Hasil Pengujian
1	Registrasi Akun	Akun Berhasil dibuat dengan response system menampilkan notifikasi “Berhasil Daftar”	Valid 
2	Login Akun	Akun Berhasil Login dengan response system menampilkan Notifikasi “Login Berhasil”	Valid 
3	Ubah Kata Sandi	Kata sandi berhasil diubah dengan response system menampilkan Notifikasi	Valid 

		“berhasil ubah sandi”	
4	Edit Profile	Profil berhasil di ubah dan system menampilkan notifikasi “Berhasil Mengubah Profil”	<p>Valid</p> 
5	List Menu Products	Menampilkan daftar menu products yang tersedia saat ini	<p>Valid</p> 
6	Shopping Chart	Menampilkan List Products yang sudah di pilih dan siap disorder, menampilkan jumlah produk yang di order berikut total biaya	<p>Valid</p> 

7	Pembayaran	Menampilkan halaman pembayaran, pelanggan dapat mengisi form alamat kemudian memilih metode pembayaran kemudian admin akan mengkonfirmasi pembayaran serta mengupdate status pembayaran	<p>Valid</p> <p><</p> <p>Transaction Success</p> <p>Transaction successfully, we will send your goods immediately we are very grateful</p>  <p>Order ID 15555252</p> <p>Total Payment Rp 2,999,999,00</p> <p>Payment Times 20-05-2025 20:05:99</p> <p>Payment Method Bank Transfer</p>
8	History Order	Menampilkan Rincian pemesanan serta status pesanan product yang telah dipesan dan sedang di pesan	<p>Valid</p> <p>History</p> <p>Progress Finish</p> <div>  <p>Single Springbed</p> <p>Date 06/07/2024</p> <p>Rp 1,999,999,00 1 Items</p> <p>Lihat →</p> </div> <div>  <p>Modern Chair</p> <p>Date 06/07/2024</p> <p>Rp 1,999,999,00 1 Items</p> <p>Lihat →</p> </div>

b. Black Box Testing Admin

4.3.2 System Usability Scalile (SUS)

System Usability Scale (SUS) adalah alat yang digunakan untuk mengukur persepsi pengguna terhadap kegunaan suatu sistem atau produk. Alat ini terdiri dari 10 pernyataan yang dinilai oleh pengguna untuk mengevaluasi seberapa mudah sistem tersebut digunakan, kompleksitasnya, serta kepuasan pengguna dalam penggunaan sistem tersebut.

Tabel 4. 2 Hasil Pernyataan SUS

No	Quesioner
1	Saya merasa aplikasi ini mudah digunakan.
2	Fungsi-fungsi di dalam aplikasi ini terintegrasi dengan baik.
3	Saya merasa fitur aplikasi ini tidak terlalu rumit untuk dipahami.
4	Saya merasa percaya diri ketika menggunakan aplikasi ini.
5	Saya merasa membutuhkan bantuan atau panduan untuk menggunakan aplikasi ini.
6	Informasi yang ditampilkan di aplikasi ini mudah ditemukan dan dimengerti.
7	Aplikasi ini terlihat kompleks dan membingungkan.
8	Saya merasa aplikasi ini memiliki fitur-fitur yang sesuai dengan kebutuhan saya.
9	Pengalaman berbelanja atau mencari produk di aplikasi ini berjalan lancar.
10	Saya akan merekomendasikan aplikasi ini kepada orang lain karena kemudahannya.

Dari hasil penyebaran Kuisisioner dengan menggunakan Google Form [Usability Scale \(google.com\)](https://www.google.com) diperoleh data 20 responden. Responden terdiri dari pengguna sistem, yaitu pelanggan, barista dan admin. Hasil penilaian SUS sebagai berikut :

Tabel 4. 3 Skor Hasil Hitung Kuisisioner

NO	RESPONDEN	SKOR HASIL HITUNG										JUMLAH SKOR	SKOR (Jumlah x 2,5)
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10		
1	Resp 1	4	3	4	2	5	3	3	5	5	5	39	97,5
2	Resp 2	3	3	3	3	3	5	4	4	3	3	34	85
3	Resp 3	5	5	5	5	5	4	4	5	1	1	40	100
4	Resp 4	4	4	4	4	4	4	4	4	4	2	38	95
5	Resp 5	3	2	4	4	5	5	3	5	3	5	39	97,5
6	Resp 6	5	5	5	5	5	1	2	1	5	5	39	97,5
7	Resp 7	4	4	4	4	1	4	5	4	5	4	39	97,5
8	Resp 8	4	1	5	2	4	1	5	5	5	5	37	92,5
9	Resp 9	4	2		4	4	5	4	5	4	5	37	92,5
10	Resp 10	4	4	5	3	3	3	3	3	3	3	34	85
11	Resp 11	4	2		5	1	2	5	5	5	5	34	85
12	Resp 12	5	2	4	3	4	2	4	5	4	5	38	95
13	Resp 13	5	2	3	4	3	2	3	4	3	4	33	82,5
14	Resp 14	5	2	5	2	4	4	4	4	4	4	38	95
15	Resp 15	5	4	4	4	4	1	4	4	4	4	38	95
16	Resp 16	5	5	3	2	3	3	5	5	5	3	39	97,5
17	Resp 17	3	5	3	5	3	5	3	3	4	5	39	97,5
18	Resp 18	3	1	1	5	5	5	5	5	5	5	40	100
19	Resp 19	3	2	2	2	5	5	5	5	5	5	39	97,5
20	Resp 20	4	5	3	3	3	3	3	5	5	5	39	97,5
TOTAL													1882,5
SKOR RATA - RATA													94,125

Hasil dari perhitungan System Usability Scale (SUS) yang telah dihitung sesuai dengan rumus yang telah ditentukan didapatkan skor rata – rata sebesar 94,125 dari semua responden.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan dari proses penelitian yang telah dilakukan terkait RANCANG BANGUN APLIKASI MOBILE E-COMMERCE UNTUK JUAL-BELI FURNITURE BERBASIS ANDROID PADA TOKO WOODMAN.PROJECT (Study Kasus: Woodman Project Furniture). Dapat diperoleh beberapa Kesimpulan yaitu:

1. Perancangan aplikasi E-Commerce woodman projects dapat memudahkan pelanggan melakukan pencarian barang serta pemesanan secara online memungkinkan pelanggan memesan dari mana saja dan tanpa batasan waktu, pada aplikasi ini menyediakan fitur filterasi agar memudahkan pelanggan menentukan harga sesuai dengan anggaran mereka.
2. Aplikasi ini menyediakan informasi yang mendetail, Foto, deskripsi produk dengan material, ukuran, dan kelebihan produk sehingga konsumen dapat memahami produk dengan jelas.
3. Dengan antarmuka design yang interaktif dan sederhana memungkinkan pelanggan tanpa pengalaman teknologi dapat memahami cara belanja dengan cepat.

5.2 Saran

Saran dari peneliti terkait pembahasan APLIKASI MOBILE E-COMMERCE UNTUK JUAL-BELI FURNITURE BERBASIS ANDROID PADA TOKO WOODMAN.PROJECT pada penelitian ini yaitu :

1. Saran dari penelitian ini adalah untuk mempermudah pembayaran dan keamanan bagi pelanggan aplikasi ini harus di update system pembayaran nya menggunakan pihak ke -3 seperti Midtrans dan lainnya.

2. Agar memudahkan pihak admin menghitung penjualan disarankan untuk menambah fitur penjualan dengan filter kurun waktu.

DAFTAR PUSTAKA

- Admin_dti. (2023, Januari 17). *Apa itu Autentikasi?* jakarta.telkomuniversity.ac.id.
<https://jakarta.telkomuniversity.ac.id/apa-itu-autentikasi/>
- Afifah, V., & Setyantoro, D. (t.t.). *Rancangan Sistem Pemilihan dan Penetapan Harga dalam Proses Pengadaan Barang dan Jasa Logistik Berbasis Web*.
- Alfriansyah¹, A., Mayada², I., & Fauzi³, M. (2023). Perancangan Sistem Booking Jadwal Pernikahan Berbasis Mobile Apps Menggunakan Flutter Microservice. Dalam *Scientia Sacra: Jurnal Sains* (Vol. 3, Nomor 2).
<http://pijarpemikiran.com/index.php/Scientia>
- Alifian Adam. (2022, Februari 3). *Class Diagram Adalah: Pengertian, Manfaat, Komponen dan Contohnya*. accurate.id. <https://accurate.id/teknologi/class-diagram-adalah/>
- Ardian Andre. (2021). Perancangan Aplikasi Pengolah Data Siswa Berbasis Android (Studi Kasus : Mis Nurul Huda Labuhan Batu Selatan). *Journal of Computer Science and Information Systems (JCoInS)*, 2, 113–123.
- Ariffud Muhammad. (2022, Februari 22). *Mengenal Visual Code Studio dan Fitur-Fitur Pentingnya*. niagahoster.co.id.
https://www.niagahoster.co.id/blog/visual-code-studio/#Apa_itu_Visual_Code_Studio
- Ariffud Muhammad. (2023, Mei 24). *Use Case Diagram: Pengertian, Contoh, Simbol, Cara Membuatnya*. www.niagahoster.co.id.
<https://www.niagahoster.co.id/blog/use-case-diagram-adalah/>
- Arifin, N. Y., Kom, S., Kom, M., Tyas, S. S., Sulistiani, H., Kom, M., Alim Hardiansyah, S. T., Kom, M., Suri, G. P., & Kom, M. (2022). *Analisa Perancangan Sistem Informasi*. Cendikia Mulia Mandiri.

- Coding Studio Team. (2023, Februari 28). *Flutter Adalah: Pengertian, Kelebihan, dan Kekurangan*. codingstudio.id. <https://codingstudio.id/blog/flutter-adalah-kelebihan-dan-kekurangan/>
- Dasar, P. G. S., & Pesona Dasar, J. (2022). PENINGKATAN HASIL BELAJAR MAHASISWA MELALUI FORUM DISKUSI. *Universitas Syiah Kuala*, 10(1), 32–38.
- Evanemran. (2023, Mei 29). *Flutter State Management: A Comprehensive Guide*. codingwithhevan.com. <https://codingwithhevan.com/flutter-state-management-a-comprehensive-guide/>
- Febriyanti, K., & Irwanto, D. (2023). Perancangan Sistem Informasi Akademik Untuk Mengetahui Perkembangan Nilai Siswa Berbasis Mobile Pada Sekolah RA. Ar-Rahmah. *BINER: Jurnal Ilmu Komputer, Teknik dan Multimedia*, 1(3), 629–649.
- Gamal Thabroni. (2022, Februari 22). *Extreme Programming (XP) – Definisi, Nilai, Tahapan*. serupa.id. <https://serupa.id/extreme-programming-xp-definisi-nilai-tahapan/>
- Gamal Thabroni. (2023, November 20). *Normalisasi Database: Definisi, Proses (Tahapan Lengkap) & Tujuan*. serupa.id. <https://serupa.id/normalisasi-database/>
- Geograf. (2023, Desember 2). *Pengertian Aplikasi Menurut Kbbi: Definisi dan Penjelasan Lengkap Menurut Ahli*. geograf.id. <https://geograf.id/jelaskan/pengertian-aplikasi-menurut-kbbi/>
- Ishak, K. M., Yusman, N. I., & Nurmeilana, A. (2022). *Rancang Bangun Sistem Informasi Pengarsipan Surat Masuk Keluar Berbasis Website di Desa Gudang Tanjungsari*. 1(2), 120–125. <https://doi.org/10.32627>
- Lucky. (2024, Januari 5). *Pengertian Komunitas: Konsep dan Peran Penting dalam Masyarakat*. wikipedia.co.id. <https://wikipedia.co.id/pengertian-komunitas/>
- Nabila Fauziah. (2022, Februari 21). *Bahasa Pemrograman Dart – Pengertian, Sejarah, Fitur, dan Contoh Aplikasi*. academia.alterra.id. <https://academy.alterra.id/blog/bahasa-pemrograman-dart/>

- Praniffa, A. C., Syahri, A., Sandes, F., Fariha, U., Giansyah, Q. A., & Hamzah, M. (2023). Pengujian Sistem Informasi Parkir Berbasis Web Pada UIN SUSKA RIAU Menggunakan White Box dan Black Box Testing. *Jurnal Testing dan Implementasi Sistem Informasi*, 1(1), 1–16. <https://www.journal.al-matani.com/index.php/jtisi/article/view/321>
- Pulungan, S. M., Febrianti, R., Lestari, T., Gurning, N., & Fitriana, N. (t.t.). *Analisis Teknik Entity-Relationship Diagram Dalam Perancangan Database*. 01(2), 143–147. <https://doi.org/10.47233/jemb.v2i1.533>
- Putra, P. B. A. A., Widiatry, W., Pranatawijaya, V. H., & Sari, N. N. K. (2022). Implementasi aplikasi android untuk sistem pendaftaran dan antrian pada poli covid rsud doris sylvanus. *Jurnal Teknologi Informasi: Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika*, 16(1), 81–91.
- Revoupedia. (2024). *UML*. revou.co. <https://revou.co/kosakata/uml>
- Rony Setiawan. (2021a, Agustus 21). *Apa Itu Sequence Diagram dan Contohnya*. dicoding.com. <https://www.dicoding.com/blog/apa-itu-sequence-diagram/>
- Rony Setiawan. (2021b, November 17). *Black Box Testing Untuk Menguji Perangkat Lunak*. dicoding.com. <https://www.dicoding.com/blog/black-box-testing/>
- Rony Setiawan. (2021c, November 20). *White Box Testing untuk Menguji Perangkat Lunak*. dicoding.com. <https://www.dicoding.com/blog/white-box-testing/>
- Rony Setiawan. (2021d, Desember 22). *Apa itu JSON? Simak Perbedaanannya dengan XML*. dicoding.com. <https://www.dicoding.com/blog/apa-itu-json/>
- Santoso, S., Surjawan, D. J., & Handoyo, E. D. (2020). Pengembangan Sistem Informasi Tukar Barang Untuk Pemanfaatan Barang Tidak Terpakai dengan Flutter Framework. *Jurnal Teknik Informatika dan Sistem Informasi*, 6(3). <https://doi.org/10.28932/jutisi.v6i3.3071>
- Yasin K. (2022, April 24). *Pengertian MySQL, Fungsi, dan Cara Kerjanya (Lengkap)*. niagahoster.co.id. https://www.niagahoster.co.id/blog/mysql-adalah/#Pengertian_MySQL

