

LAPORAN TUGAS AKHIR

SISTEM INFORMASI E-COMMERCE BERBASIS WEB MENGUNAKAN PHP NATIVE DAN MYSQL

Mata Kuliah : Basis Data Terdistribusi
Dosen Pengampu : Dr. Dedy Hartama, S.T., M.Kom

Disusun Oleh

- 1 Muhammad Farel Setiawan Girsang (2501025)
- 2 Pandu Dyaksa (2501027)
- 3 Nauval Hakim (2501026)
- 4 Rafif Adhimukti. CH (2501029)
- 5 Dito Pasuka Deno (2501012)

Kelas : 25T01



**PROGRAM STUDI TEKNIK INFORMATIKA
STIKOM TUNAS BANGSA PEMATANGSIANTAR
2026**

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat, karunia, dan hidayah-Nya sehingga kami dapat menyelesaikan laporan proyek yang berjudul "**Sistem Informasi E-Commerce Berbasis Web Menggunakan PHP Native dan MySQL**" dengan baik dan tepat waktu.

Laporan ini disusun sebagai bentuk pertanggungjawaban atas pelaksanaan proyek yang telah kami kerjakan sebagai bagian dari pemenuhan tugas pada mata kuliah **Basis Data Terdistribusi**. Proyek ini berupa pengembangan sistem informasi e-commerce berbasis web yang dirancang untuk memfasilitasi proses transaksi jual beli secara daring, mulai dari pengelolaan produk, pemesanan, pembayaran, hingga pembuatan laporan penjualan. Sistem ini juga menyediakan fitur manajemen yang mendukung kebutuhan administrator serta antarmuka yang mudah digunakan oleh pengguna.

Dalam proses perancangan, pengembangan, pengujian, hingga penyusunan laporan ini, kami memperoleh banyak bantuan, dukungan, bimbingan, serta motivasi dari berbagai pihak. Oleh karena itu, pada kesempatan ini kami menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bapak **Dr. Deddy Hartama, S.T., M.Kom.** selaku dosen pengampu mata kuliah Basis Data Terdistribusi yang telah memberikan arahan, bimbingan, serta ilmu pengetahuan selama proses pembelajaran dan pengerjaan proyek ini.
2. Seluruh anggota kelompok yang telah bekerja sama, berdiskusi, dan berkontribusi dalam menyelesaikan proyek maupun penyusunan laporan ini.
3. Semua pihak yang telah memberikan dukungan dan bantuan, baik secara langsung maupun tidak langsung, sehingga proyek ini dapat diselesaikan dengan baik.

Kami menyadari bahwa laporan ini masih memiliki berbagai kekurangan dan jauh dari kata sempurna. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang bersifat membangun demi perbaikan dan pengembangan yang lebih baik di masa mendatang.

Akhir kata, kami berharap laporan ini dapat memberikan manfaat serta menambah wawasan bagi pembaca, khususnya dalam bidang pengembangan sistem informasi, basis data, dan teknologi e-commerce berbasis web.

Pematangsiantar, 16 Juni 2026

Penulis

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
DAFTAR GAMBAR.....	v
DAFTAR TABEL	vi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	2
1.3 Rumusan Masalah.....	2
1.4 Batasan Masalah	3
1.5 Tujuan Penelitian/Perancangan.....	3
1.6 Manfaat Penelitian.....	4
1.6.1 Manfaat Teoritis	4
1.6.2 Manfaat Praktis.....	4
BAB II LANDASAN TEORI.....	5
2.1 Pengertian Website	5
2.2 Pengertian E-Commerce.....	5
2.3 Pengertian Sistem Informasi.....	6
2.4 PHP (PHP: Hypertext Preprocessor)	6
2.5 MySQL.....	7
2.6 HTML (HyperText Markup Language)	8
2.7 CSS (Cascading Style Sheets)	8
2.8 JavaScript	8
2.9 Bootstrap.....	8
2.10 Database (Basis Data).....	9
2.11 UML (Unified Modeling Language)	9
2.12 Flowchart.....	9
2.13 Entity Relationship Diagram (ERD)	10
2.14 Studi Literatur.....	10
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	12
3.1 Analisis Sistem Berjalan.....	12
3.2 Analisis Kebutuhan Sistem.....	12
3.3 Kebutuhan Fungsional.....	12

3.4	Kebutuhan Non-Fungsional.....	13
3.5	Use Case Diagram	13
3.6	Activity Diagram	14
3.6.1	Activity Diagram Proses Login	14
3.6.2	Activity Diagram Proses Pemesanan.....	15
3.7	Sequence Diagram.....	16
3.7.1	Sequence Diagram Login	16
3.7.2	Sequence Diagram Checkout.....	16
3.8	Flowchart Sistem	17
3.8.1	Flowchart Sistem Keseluruhan.....	17
3.9	Perancangan Database	17
3.10	ERD (Entity Relationship Diagram)	18
3.11	Struktur Tabel Database	18
3.12	Relasi Antar Tabel.....	20
BAB IV IMPLEMENTASI DAN PEMBAHASAN.....		22
4.1	Implementasi Database.....	22
4.2	Implementasi Halaman Login	23
4.3	Implementasi Dashboard Admin	24
4.4	Implementasi CRUD Produk.....	25
4.5	Implementasi Manajemen Pesanan	26
4.6	Implementasi Pembayaran.....	27
4.7	Implementasi Registrasi User	27
4.8	Implementasi Halaman Produk	28
4.9	Implementasi Keranjang Belanja.....	29
4.10	Implementasi Checkout	29
4.11	Implementasi Riwayat Pesanan	31
4.12	Pengujian Sistem	32
4.13	Hasil Pengujian Black Box.....	32
4.14	Analisis Hasil Pengujian.....	34
4.15	Kelebihan dan Kekurangan Sistem.....	35
4.15.1	Kelebihan Sistem.....	35
4.15.2	Kekurangan Sistem.....	35
BAB V PENUTUP		36
5.1	Kesimpulan.....	36

5.2 Saran	36
DAFTAR PUSTAKA.....	38
LAMPIRAN	40

DAFTAR GAMBAR

<i>Gambar 3. 1</i> Diagram Admin dan User	13
<i>Gambar 3. 2</i> Diagram Proses Login	14
<i>Gambar 3. 3</i> Proses Pemesanan.....	15
<i>Gambar 3. 4</i> Login	16
<i>Gambar 3. 5</i> Checkout.....	16
<i>Gambar 3. 6</i> Flowchart Sistem.....	17
<i>Gambar 3. 7</i> ERD	18

DAFTAR TABEL

<i>Tabel 2. 1 Kategori e-commerce</i>	6
<i>Tabel 2. 2 Tipe data MySQL</i>	8
<i>Tabel 2. 3 Simbol Flowchart</i>	10
<i>Tabel 2. 4 Studi Literatur</i>	10
<i>Tabel 3. 1 Analisis Kebutuhan Fungsional</i>	13
<i>Tabel 3. 2 Analisis Kebutuhan Non-Fungsional</i>	13
<i>Tabel 3. 3 Tabel Users</i>	18
<i>Tabel 3. 4 Tabel `categories`</i>	19
<i>Tabel 3. 5 Tabel `products`</i>	19
<i>Tabel 3. 6 Tabel `cart`</i>	19
<i>Tabel 3. 7 Tabel `orders`</i>	19
<i>Tabel 3. 8 Tabel `order_items`</i>	20
<i>Tabel 3. 9 Tabel `payments`</i>	20
<i>Tabel 3. 10 Relasi antar Tabel</i>	20
<i>Tabel 4. 1 Status Pesanan</i>	26
<i>Tabel 4. 2 Pengujian Modul Autentikasi</i>	32
<i>Tabel 4. 3 Pengujian Modul Produk (Admin)</i>	33
<i>Tabel 4. 4 Pengujian Modul Keranjang & Checkout (User)</i>	34
<i>Tabel 4. 5 Pengujian Modul Admin (Pesanan & Pembayaran)</i>	34
<i>Tabel 4. 6 Rekapitulasi Hasil Pengujian</i>	34

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi dan komunikasi yang pesat pada era digital saat ini telah membawa perubahan signifikan dalam berbagai aspek kehidupan manusia, termasuk dalam dunia perdagangan. Perubahan pola konsumsi masyarakat yang semakin bergantung pada teknologi mendorong transformasi sistem perdagangan konvensional menuju perdagangan berbasis elektronik atau yang lebih dikenal dengan istilah *electronic commerce* (e-commerce). Berdasarkan data Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) tahun 2023, pengguna internet di Indonesia telah mencapai lebih dari 215 juta jiwa, dengan persentase pengguna yang melakukan transaksi belanja online terus meningkat setiap tahunnya.

E-commerce hadir sebagai solusi yang mempertemukan antara penjual dan pembeli dalam sebuah platform digital yang dapat diakses kapan saja dan di mana saja. Kemudahan akses, efisiensi waktu, transparansi harga, serta jangkauan pasar yang lebih luas menjadikan e-commerce sebagai alternatif bisnis yang sangat menjanjikan, baik bagi pelaku usaha skala besar maupun usaha kecil dan menengah (UKM). Hal ini sejalan dengan riset yang dipublikasikan oleh Statista (2023) yang memproyeksikan nilai transaksi e-commerce Indonesia akan menembus angka USD 82 miliar pada tahun 2025.

Namun demikian, tidak semua pelaku usaha mampu memanfaatkan peluang tersebut secara optimal, terutama bagi usaha-usaha yang masih mengandalkan sistem penjualan manual. Proses pencatatan transaksi secara manual rentan terhadap kesalahan, lambat, tidak efisien, dan menyulitkan pemilik usaha dalam memantau perkembangan bisnis secara real-time. Selain itu, keterbatasan jangkauan pasar menjadi hambatan utama bagi pelaku usaha konvensional untuk bersaing di era digital.

Berdasarkan permasalahan tersebut, diperlukan sebuah sistem informasi e-commerce berbasis web yang mampu mengintegrasikan seluruh proses bisnis secara digital, mulai dari pengelolaan produk, manajemen pesanan, hingga pemrosesan pembayaran. Sistem ini dirancang menggunakan teknologi PHP Native versi 7.4 sebagai bahasa pemrograman sisi server, MySQL sebagai sistem manajemen basis data relasional, serta Bootstrap sebagai framework CSS untuk memastikan tampilan antarmuka yang responsif dan ramah pengguna di berbagai perangkat.

Pemilihan teknologi PHP Native didasari pertimbangan bahwa PHP merupakan salah satu bahasa pemrograman web yang paling banyak digunakan secara global dengan ekosistem yang matang, dokumentasi yang lengkap, serta komunitas pengembang yang besar. Menurut W3Techs (2023), PHP digunakan oleh lebih dari 77% situs web di seluruh dunia yang menggunakan bahasa pemrograman sisi server. Penggunaan PHP Native tanpa framework berat memberikan fleksibilitas penuh dalam arsitektur sistem, efisiensi performa, dan kemudahan dalam pemeliharaan kode.

Dengan mempertimbangkan kondisi dan permasalahan yang telah diuraikan di atas, maka penulis mengambil judul "Sistem Informasi E-Commerce Berbasis Web Menggunakan PHP Native dan MySQL" sebagai topik proyek dalam laporan ini.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan, maka dapat diidentifikasi beberapa permasalahan yang menjadi dasar pengembangan sistem ini, antara lain:

1. **Keterbatasan sistem penjualan manual** yang tidak efisien dalam pencatatan transaksi, pengelolaan stok produk, dan pembuatan laporan penjualan.
2. **Jangkauan pasar yang terbatas** karena sistem penjualan konvensional hanya dapat melayani pelanggan yang hadir secara fisik di lokasi usaha.
3. **Kesulitan dalam monitoring bisnis secara real-time**, termasuk pemantauan stok produk, status pesanan, dan rekap pembayaran.
4. **Tidak adanya sistem manajemen pengguna** yang terstruktur, sehingga data pelanggan sulit dikelola dengan baik.
5. **Proses pembayaran yang tidak terverifikasi secara sistematis**, sehingga rentan terhadap kecurangan dan kesalahan validasi.
6. **Minimnya kemampuan analisis data penjualan** akibat tidak tersedianya sistem pelaporan yang terintegrasi.

1.3 Rumusan Masalah

Berdasarkan identifikasi masalah di atas, maka rumusan masalah dalam proyek ini adalah sebagai berikut:

1. Bagaimana merancang dan membangun sistem informasi e-commerce berbasis web yang dapat mendukung proses transaksi jual beli secara daring?
2. Bagaimana mengimplementasikan sistem autentikasi multi-level dengan perbedaan hak akses antara admin dan pengguna akhir?

3. Bagaimana merancang sistem manajemen produk yang mendukung pengelolaan kategori, stok, dan upload gambar produk?
4. Bagaimana membangun alur pemesanan dan pembayaran yang aman, terintegrasi, dan mudah diverifikasi oleh admin?
5. Bagaimana menghasilkan laporan penjualan yang informatif sebagai dasar pengambilan keputusan bisnis?

1.4 Batasan Masalah

Guna membatasi ruang lingkup permasalahan agar pembahasan lebih terfokus dan terarah, maka ditetapkan batasan-batasan masalah sebagai berikut:

1. Sistem dikembangkan menggunakan bahasa pemrograman PHP Native versi 7.4 dan MySQL sebagai basis data.
2. Sistem hanya mendukung dua level pengguna, yaitu **Admin** dan **User** (pelanggan).
3. Metode pembayaran yang diimplementasikan adalah **transfer bank manual** dengan konfirmasi upload bukti pembayaran.
4. Sistem tidak mengintegrasikan payment gateway pihak ketiga (seperti Midtrans, Xendit, atau PayPal).
5. Sistem tidak mencakup fitur pengiriman terintegrasi dengan ekspedisi pihak ketiga (hanya pencatatan alamat pengiriman).
6. Laporan penjualan yang dihasilkan terbatas pada rekap pesanan dan pembayaran.
7. Sistem dijalankan dalam lingkungan server lokal (localhost) menggunakan XAMPP sebagai development environment.
8. Desain antarmuka menggunakan framework Bootstrap 5.

1.5 Tujuan Penelitian/Perancangan

Tujuan yang ingin dicapai dalam perancangan dan pengembangan sistem informasi e-commerce ini adalah:

1. Merancang dan membangun sistem informasi e-commerce berbasis web yang fungsional dengan menggunakan PHP Native dan MySQL.
2. Mengimplementasikan sistem autentikasi yang aman dengan pembedaan hak akses untuk admin dan pengguna.
3. Membangun fitur manajemen produk lengkap meliputi CRUD produk, manajemen kategori, dan upload gambar.
4. Mengembangkan alur pemesanan (shopping cart hingga checkout) yang intuitif dan mudah digunakan.

5. Membangun sistem verifikasi pembayaran melalui upload bukti transfer yang dapat dikonfirmasi oleh admin.
6. Menghasilkan laporan penjualan yang dapat dijadikan dasar analisis kinerja bisnis.
7. Menghasilkan antarmuka yang responsif dan ramah pengguna menggunakan Bootstrap 5.

1.6 Manfaat Penelitian

1.6.1 Manfaat Teoritis

1. Memberikan kontribusi dalam pengembangan ilmu pengetahuan di bidang sistem informasi dan rekayasa perangkat lunak, khususnya dalam implementasi e-commerce berbasis web.
2. Menjadi referensi akademis bagi penelitian selanjutnya yang berhubungan dengan pengembangan sistem e-commerce.
3. Memperkaya wawasan terkait arsitektur MVC (Model-View-Controller) pada pengembangan aplikasi PHP Native.

1.6.2 Manfaat Praktis

1. **Bagi Pemilik Usaha:** Memudahkan pengelolaan produk, pemantauan pesanan, dan monitoring pembayaran secara digital melalui dashboard yang informatif.
2. **Bagi Pelanggan:** Memberikan kemudahan dalam berbelanja produk secara daring, melacak status pesanan, dan melakukan pembayaran kapan saja tanpa terbatas lokasi.
3. **Bagi Penulis:** Meningkatkan kompetensi teknis dalam pengembangan sistem informasi berbasis web serta memperdalam pemahaman tentang arsitektur aplikasi, manajemen basis data, dan keamanan sistem.
4. **Bagi Institusi Pendidikan:** Memberikan contoh implementasi nyata dari teori-teori yang diajarkan dalam mata kuliah pemrograman web dan rekayasa perangkat lunak.

BAB II

LANDASAN TEORI

2.1 Pengertian Website

Website atau situs web adalah kumpulan halaman web yang saling terhubung dan dapat diakses melalui jaringan internet menggunakan browser web. Menurut Hidayat (2010), website adalah keseluruhan halaman-halaman web yang terdapat dalam sebuah domain yang mengandung informasi. Sebuah website umumnya dibangun di atas sebuah domain dan terdiri dari beberapa halaman yang saling berkaitan menggunakan hyperlink.

Website dapat diklasifikasikan berdasarkan sifatnya menjadi dua jenis utama:

1. **Website Statis**, yaitu website yang kontennya tidak berubah dan bersifat tetap. Konten dibuat menggunakan HTML dan CSS murni tanpa pemrosesan sisi server.
2. **Website Dinamis**, yaitu website yang kontennya dapat berubah-ubah sesuai dengan interaksi pengguna atau data yang tersimpan dalam basis data. Website jenis ini menggunakan bahasa pemrograman server-side seperti PHP, Python, atau Node.js.

Sistem e-commerce yang dibangun dalam proyek ini merupakan website dinamis yang memanfaatkan PHP sebagai bahasa pemrograman server-side dan MySQL sebagai basis data untuk menyimpan dan mengelola seluruh data transaksi.

2.2 Pengertian E-Commerce

Electronic Commerce (E-Commerce) adalah proses pembelian, penjualan, atau pertukaran produk, layanan, dan informasi melalui jaringan komputer termasuk internet (Turban et al., 2015). Secara lebih spesifik, e-commerce merupakan aplikasi teknologi informasi yang memfasilitasi otomatisasi transaksi bisnis dan alur kerja antara penjual dan pembeli dalam lingkungan digital.

Menurut Kotler dan Armstrong (2018), e-commerce melibatkan penggunaan teknologi informasi dalam seluruh proses bisnis, mulai dari pemasaran produk, pemrosesan pesanan, pembayaran, pengelolaan inventori, hingga layanan purna jual.

E-commerce dapat dikategorikan berdasarkan hubungan antar pihak yang terlibat:

Tipe	Deskripsi	Contoh
B2C (Business-to-Consumer)	Transaksi antara perusahaan dan konsumen akhir	Tokopedia, Shopee, Lazada
B2B (Business-to-Business)	Transaksi antar perusahaan	Marketplace grosir, supplier
C2C (Consumer-to-Consumer)	Transaksi antar konsumen	OLX, Carousell
C2B (Consumer-to-Business)	Konsumen menawarkan produk/jasa ke perusahaan	Freelancer, Upwork
G2C (Government-to-Consumer)	Layanan pemerintah kepada	Sistem pajak online, e-KTP

Consumer)	masyarakat	
-----------	------------	--

Tabel 2. 1 Kategori e-commerce

Sistem yang dikembangkan dalam proyek ini termasuk dalam kategori B2C (Business-to-Consumer), di mana pihak pengelola (admin) berperan sebagai penjual dan pengguna terdaftar berperan sebagai konsumen.

2.3 Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Jogiyanto, 2005).

Komponen utama sistem informasi mencakup:

1. **Hardware:** Perangkat keras komputer dan jaringan yang digunakan.
2. **Software:** Perangkat lunak sistem operasi dan aplikasi.
3. **Data:** Fakta-fakta mentah yang diolah menjadi informasi.
4. **Prosedur:** Aturan dan tata cara penggunaan sistem.
5. **People:** Sumber daya manusia yang mengoperasikan sistem.

Sistem informasi e-commerce yang dibangun mengintegrasikan seluruh komponen tersebut untuk mendukung proses bisnis penjualan secara digital.

2.4 PHP (PHP: Hypertext Preprocessor)

PHP (PHP: Hypertext Preprocessor) adalah bahasa skrip open-source yang banyak digunakan untuk pengembangan web dan dapat disematkan langsung ke dalam HTML. PHP dieksekusi di sisi server (server-side scripting), sehingga kode PHP diproses oleh server sebelum hasilnya dikirim ke browser pengguna dalam bentuk HTML (PHP Manual, 2023).

PHP pertama kali dikembangkan oleh Rasmus Lerdorf pada tahun 1994 dan terus berkembang hingga saat ini. Beberapa karakteristik utama PHP yang menjadikannya pilihan populer untuk pengembangan web:

1. **Open Source** — Bebas digunakan, dimodifikasi, dan didistribusikan.
2. **Cross-Platform** — Dapat berjalan di berbagai sistem operasi (Windows, Linux, macOS).
3. **Mudah Dipelajari** — Sintaks PHP relatif sederhana dan mirip dengan bahasa C.
4. **Integrasi Database yang Kuat** — Mendukung berbagai DBMS termasuk MySQL, PostgreSQL, SQLite, dan lainnya.

5. **Komunitas Besar** — Memiliki ekosistem yang sangat besar dengan banyak library dan framework.

Dalam proyek ini digunakan PHP versi 7.4 yang menawarkan peningkatan performa signifikan dibandingkan versi sebelumnya, dukungan typed properties, arrow functions, dan berbagai optimasi lainnya.

```
// Contoh sintaks PHP 7.4
<?php
// Typed properties (fitur PHP 7.4)
class Product {
    public string $name;
    public float $price;
    public int $stock;

    public function __construct(string $name, float $price, int $stock) {
        $this->name = $name;
        $this->price = $price;
        $this->stock = $stock;
    }
}

// Arrow function (fitur PHP 7.4)
$products = [/* ... */];
$expensive = array_filter($products, fn($p) => $p->price > 100000);
?>
```

2.5 MySQL

MySQL adalah sistem manajemen basis data relasional (Relational Database Management System / RDBMS) yang bersifat open-source dan dikembangkan oleh Oracle Corporation. MySQL menggunakan bahasa SQL (Structured Query Language) sebagai antarmuka untuk mengakses dan mengelola data (MySQL Manual, 2023).

Kelebihan MySQL yang menjadikannya pilihan utama dalam pengembangan aplikasi web:

1. **Performa Tinggi** — Mampu menangani jutaan baris data dengan kecepatan tinggi.
2. **Skalabilitas** — Dapat dikonfigurasi dari aplikasi kecil hingga sistem enterprise.
3. **Keamanan** — Dilengkapi dengan sistem manajemen hak akses yang granular.
4. **Replikasi** — Mendukung replikasi data untuk ketersediaan tinggi (*high availability*).
5. **Kompatibilitas Luas** — Terintegrasi dengan baik dengan PHP dan berbagai bahasa pemrograman lainnya.

Tipe data utama MySQL yang digunakan dalam proyek ini:

Tipe Data	Deskripsi	Penggunaan
INT	Bilangan bulat	ID, stok, quantity
VARCHAR(n)	String karakter bervariasi	Nama, email, alamat
TEXT	Teks panjang	Deskripsi produk
DECIMAL(p,s)	Bilangan desimal presisi	Harga produk

TIMESTAMP	Tanggal dan waktu	created_at, updated_at
ENUM	Nilai dari set yang ditentukan	Status pesanan, role
TINYINT(1)	Boolean (0/1)	Status aktif/nonaktif

Tabel 2. 2 Tipe data MySQL

2.6 HTML (HyperText Markup Language)

HTML (HyperText Markup Language) adalah bahasa markup standar yang digunakan untuk membuat dan menyusun konten di halaman web. HTML mendefinisikan struktur dan semantik konten web menggunakan elemen-elemen yang direpresentasikan oleh tag (MDN Web Docs, 2023). Dalam proyek ini digunakan HTML5 yang merupakan versi terbaru dengan dukungan elemen semantik, multimedia, dan form yang lebih kaya.

2.7 CSS (Cascading Style Sheets)

CSS (Cascading Style Sheets) adalah bahasa style sheet yang digunakan untuk mendeskripsikan presentasi visual dokumen HTML. CSS memungkinkan pemisahan konten (HTML) dari presentasi visual (styling), sehingga kode menjadi lebih terstruktur dan mudah dipelihara. Dalam proyek ini digunakan CSS3 yang mendukung flexbox, grid layout, animasi, dan variabel CSS (MDN Web Docs, 2023).

2.8 JavaScript

JavaScript adalah bahasa pemrograman tingkat tinggi yang bersifat dinamis, berorientasi objek, dan multi-paradigma. JavaScript merupakan satu-satunya bahasa pemrograman yang dapat berjalan langsung di browser web (client-side) tanpa memerlukan kompilasi. JavaScript digunakan untuk menambahkan interaktivitas pada halaman web, seperti validasi form, manipulasi DOM, pengiriman data asinkron (AJAX), dan animasi (Flanagan, 2020).

Dalam proyek ini JavaScript dimanfaatkan untuk:

- Validasi form di sisi klien sebelum data dikirim ke server.
- Preview gambar produk sebelum diunggah.
- Konfirmasi dialog sebelum penghapusan data.
- Manipulasi antarmuka keranjang belanja secara dinamis.

2.9 Bootstrap

Bootstrap adalah framework CSS open-source yang paling populer untuk pengembangan web responsif dan mobile-first. Bootstrap menyediakan koleksi komponen UI yang siap pakai, seperti tombol, form, navbar, modal, tabel, grid system, dan lainnya (Bootstrap Docs, 2023). Dalam proyek ini digunakan Bootstrap versi 5 yang memperkenalkan fitur-fitur baru

seperti penghapusan dependensi jQuery, sistem utilitas yang lebih fleksibel, dan komponen yang diperbarui.

Keunggulan Bootstrap 5 dalam proyek ini:

1. **Grid System 12-Kolom** — Memudahkan pembuatan layout yang responsif.
2. **Responsive Breakpoints** — Tampilan otomatis menyesuaikan ukuran layar (mobile, tablet, desktop).
3. **Komponen Siap Pakai** — Mempercepat pengembangan antarmuka tanpa harus membangun dari nol.
4. **Tema yang Konsisten** — Memastikan konsistensi visual di seluruh halaman aplikasi.

2.10 Database (Basis Data)

Basis data (database) adalah kumpulan data yang terorganisir dan tersimpan secara sistematis dalam komputer, yang dapat diakses dan dikelola menggunakan perangkat lunak yang disebut DBMS (Database Management System). Menurut Connolly dan Begg (2015), basis data adalah kumpulan data yang saling terkait secara logis beserta deskripsinya, yang dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi.

Konsep basis data relasional yang diterapkan dalam proyek ini menggunakan model relasi antar tabel melalui primary key dan foreign key, memastikan integritas data (data integrity) dan menghindari redundansi data (data redundancy).

2.11 UML (Unified Modeling Language)

UML (Unified Modeling Language) adalah bahasa pemodelan standar yang digunakan untuk memvisualisasikan, menspesifikasikan, membangun, dan mendokumentasikan artefak dari sebuah sistem perangkat lunak (Booch et al., 2005). UML menyediakan berbagai jenis diagram yang digunakan pada tahapan analisis dan perancangan sistem.

Jenis diagram UML yang digunakan dalam proyek ini:

1. **Use Case Diagram** — Menggambarkan interaksi antara aktor (pengguna) dengan sistem.
2. **Activity Diagram** — Menggambarkan alur aktivitas atau proses bisnis dalam sistem.
3. **Sequence Diagram** — Menggambarkan interaksi antar objek dalam urutan waktu tertentu.
4. **Class Diagram** — Menggambarkan struktur kelas, atribut, metode, dan relasi antar kelas.

2.12 Flowchart

Flowchart adalah representasi grafis dari suatu proses atau algoritma yang menggunakan simbol-simbol standar untuk menggambarkan urutan langkah-langkah dalam proses tersebut

(Whitten et al., 2004). Flowchart memudahkan pemahaman alur kerja sistem sebelum implementasi dilakukan.

Simbol-simbol standar yang digunakan dalam flowchart:

Simbol	Nama	Fungsi
Oval	Terminal	Menandai awal/akhir proses
Persegi Panjang	Proses	Menggambarkan suatu tindakan/proses
Belah Ketupat	Decision	Menggambarkan percabangan kondisi
Jajar Genjang	I/O	Menggambarkan input atau output data
Panah	Flow Line	Menunjukkan arah aliran proses

Tabel 2. 3 Simbol Flowchart

2.13 Entity Relationship Diagram (ERD)

ERD (Entity Relationship Diagram) adalah model konseptual yang merepresentasikan struktur data dan hubungan antar entitas dalam sebuah basis data (Chen, 1976). ERD digunakan sebagai alat bantu dalam perancangan basis data sebelum diimplementasikan dalam DBMS tertentu.

Komponen utama ERD:

1. **Entitas (Entity):** Objek nyata atau abstrak yang memiliki data yang perlu disimpan.
2. **Atribut (Attribute):** Properti atau karakteristik dari suatu entitas.
3. **Relasi (Relationship):** Hubungan logis antara dua atau lebih entitas.
4. **Kardinalitas:** Jumlah instance entitas yang dapat berpartisipasi dalam relasi (1:1, 1:N, M:N).

2.14 Studi Literatur

Berikut adalah hasil kajian terhadap penelitian-penelitian sejenis yang menjadi referensi dalam pengembangan proyek ini:

No	Peneliti & Tahun	Judul	Teknologi	Keterkaitan
1	Prasetyo & Raharjo (2021)	Rancang Bangun Sistem E-Commerce Produk UMKM Berbasis Web	PHP, MySQL, Bootstrap	Arsitektur sistem e-commerce B2C
2	Sari & Wibowo (2022)	Implementasi Sistem Pembayaran Online pada Platform E-Commerce	PHP, MySQL, Midtrans	Alur checkout dan pembayaran
3	Handayani et al. (2020)	Pengembangan Aplikasi Toko Online dengan Metode Waterfall	PHP Native, MySQL	Penerapan metode Waterfall
4	Kurniawan & Dewi (2023)	Sistem Manajemen Inventori Terintegrasi E-Commerce	PHP Laravel, MySQL	Manajemen stok produk
5	Nugroho (2021)	Analisis dan Perancangan Sistem E-Commerce Multiseller	PHP, MySQL, API	Manajemen multi-pengguna

Tabel 2. 4 Studi Literatur

Berdasarkan studi literatur yang dilakukan, dapat disimpulkan bahwa sistem yang dikembangkan dalam proyek ini memiliki diferensiasi berupa implementasi arsitektur MVC dengan PHP Native (tanpa framework), sistem keamanan berlapis dengan middleware autentikasi, serta desain antarmuka yang modern menggunakan Bootstrap 5 dengan dukungan tema gelap (dark mode).

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem Berjalan

Sebelum sistem informasi e-commerce ini dibangun, proses penjualan yang berjalan masih dilakukan secara konvensional dengan karakteristik sebagai berikut:

Proses Penjualan Manual:

1. Calon pembeli datang langsung ke toko atau menghubungi melalui media sosial/WhatsApp.
2. Pengecekan ketersediaan produk dilakukan secara manual oleh pemilik.
3. Negosiasi harga dan konfirmasi pesanan dilakukan via chat atau telepon.
4. Pembayaran dilakukan tunai atau transfer bank tanpa sistem konfirmasi otomatis.
5. Pencatatan transaksi dilakukan secara manual dalam buku atau spreadsheet Excel.
6. Tidak ada mekanisme pelacakan status pesanan untuk pembeli.

Kelemahan Sistem Berjalan:

1. Rentan kesalahan pencatatan (human error).
2. Tidak efisien untuk volume transaksi yang tinggi.
3. Pembeli tidak dapat melihat produk secara lengkap beserta detail dan harganya.
4. Tidak ada riwayat pembelian yang terstruktur untuk pelanggan.
5. Sulit membuat laporan penjualan periodik.
6. Waktu layanan terbatas pada jam operasional toko.

3.2 Analisis Kebutuhan Sistem

Berdasarkan analisis sistem berjalan, maka dirumuskan kebutuhan-kebutuhan sistem yang akan dibangun sebagai berikut:

3.3 Kebutuhan Fungsional

No	Kode	Deskripsi Kebutuhan	Aktor
1	F-01	Sistem menyediakan form registrasi akun pengguna baru	User
2	F-02	Sistem menyediakan form login dengan validasi email dan password	Admin, User
3	F-03	Sistem membedakan hak akses antara Admin dan User	Admin, User
4	F-04	Admin dapat menambah, mengubah, menghapus, dan melihat data produk	Admin
5	F-05	Admin dapat mengunggah gambar produk	Admin
6	F-06	Admin dapat mengelola kategori produk	Admin
7	F-07	Admin dapat melihat dan memperbarui status pesanan	Admin
8	F-08	Admin dapat memverifikasi dan mengkonfirmasi pembayaran	Admin

9	F-09	Admin dapat melihat laporan penjualan	Admin
10	F-10	Admin dapat mengelola data pengguna	Admin
11	F-11	User dapat melihat daftar produk dan detail produk	User
12	F-12	User dapat menambahkan produk ke keranjang belanja	User
13	F-13	User dapat mengubah jumlah item dan menghapus dari keranjang	User
14	F-14	User dapat melakukan proses checkout dengan mengisi alamat pengiriman	User
15	F-15	User dapat mengunggah bukti pembayaran	User
16	F-16	User dapat melihat riwayat pesanan beserta status terkini	User
17	F-17	User dapat mengubah data profil	User
18	F-18	Sistem menampilkan notifikasi flash message untuk setiap aksi penting	Admin, User

Tabel 3. 1 Analisis Kebutuhan Fungsional

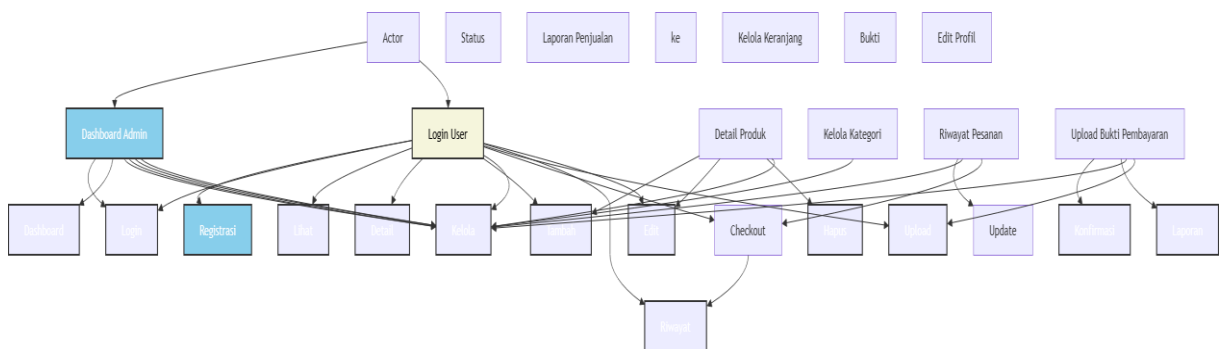
3.4 Kebutuhan Non-Fungsional

No	Kategori	Deskripsi
1	Keamanan	Sistem menggunakan hashing password dengan <code>password_hash()</code> (bcrypt), proteksi CSRF token pada setiap form, validasi input untuk mencegah SQL injection dan XSS
2	Performa	Halaman utama dimuat dalam waktu < 3 detik pada koneksi standar; query database dioptimalkan dengan indexing yang tepat
3	Kemudahan Penggunaan	Antarmuka intuitif, navigasi jelas, dan responsif di semua ukuran layar (mobile, tablet, desktop)
4	Ketersediaan	Sistem dapat berjalan 24/7 pada server hosting atau localhost
5	Maintainability	Kode ditulis dengan arsitektur MVC, komentar kode yang memadai, dan penamaan variabel yang deskriptif
6	Kompatibilitas	Mendukung browser modern: Chrome 90+, Firefox 88+, Edge 90+, Safari 14+
7	Skalabilitas	Struktur database dirancang untuk menampung data dalam jumlah besar tanpa degradasi performa signifikan

Tabel 3. 2 Analisis Kebutuhan Non-Fungsional

3.5 Use Case Diagram

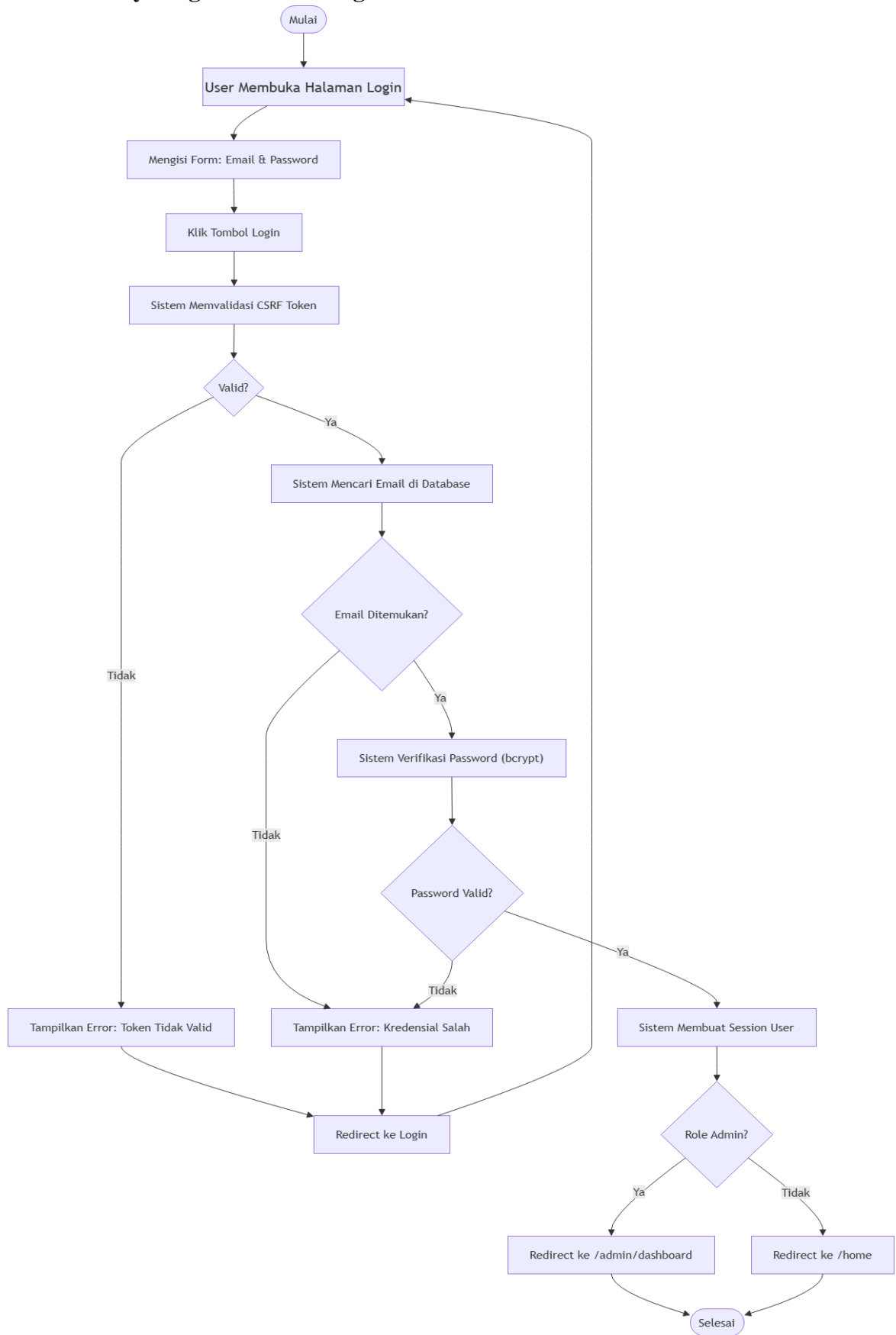
Use Case Diagram menggambarkan interaksi antara aktor dengan sistem. Sistem ini memiliki dua aktor utama, yaitu Admin dan User (Pelanggan), serta satu aktor pasif yaitu Guest (Pengunjung Belum Login).



Gambar 3. 1 Diagram Admin dan User

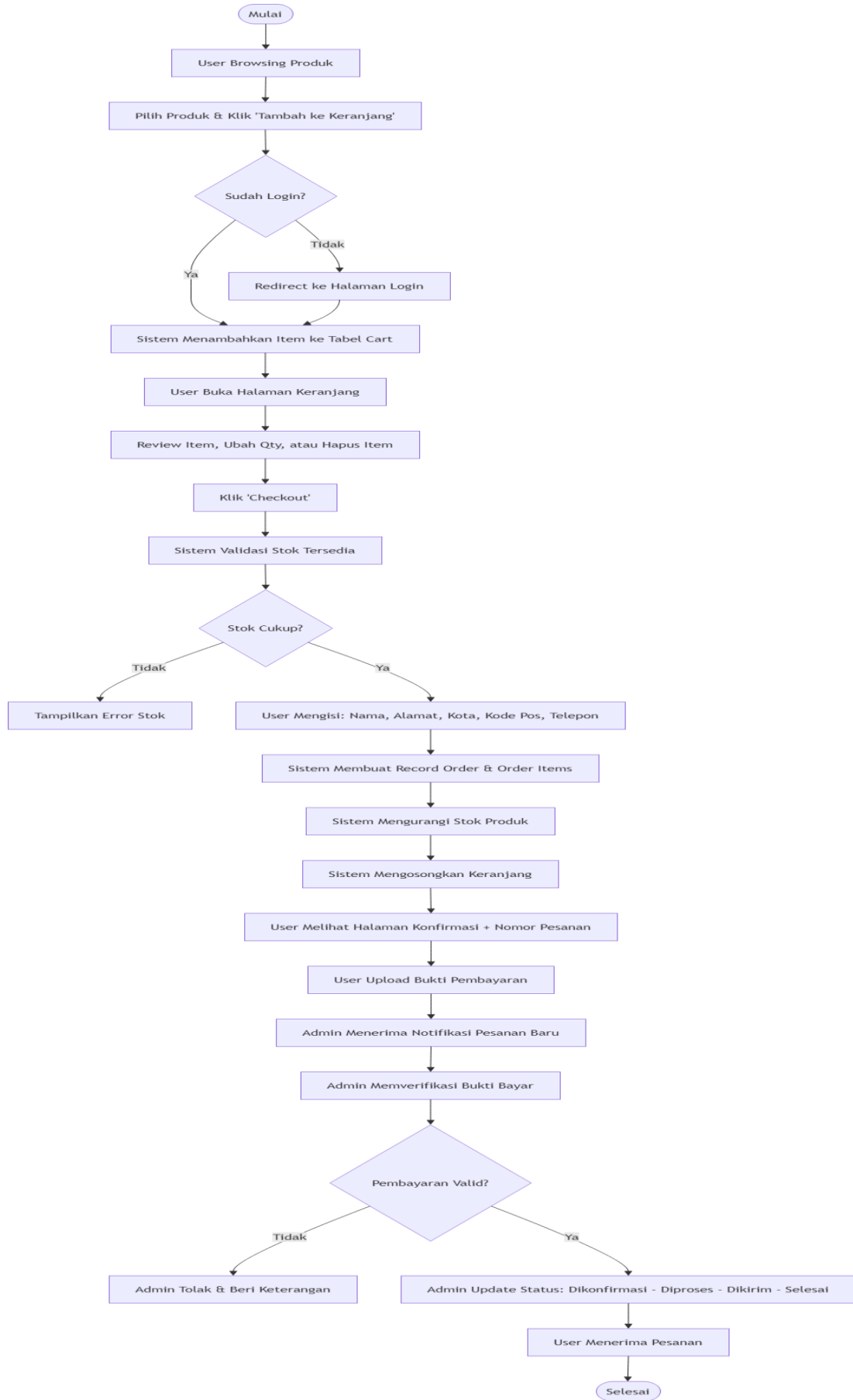
3.6 Activity Diagram

3.6.1 Activity Diagram Proses Login



Gambar 3. 2 Diagram Proses Login

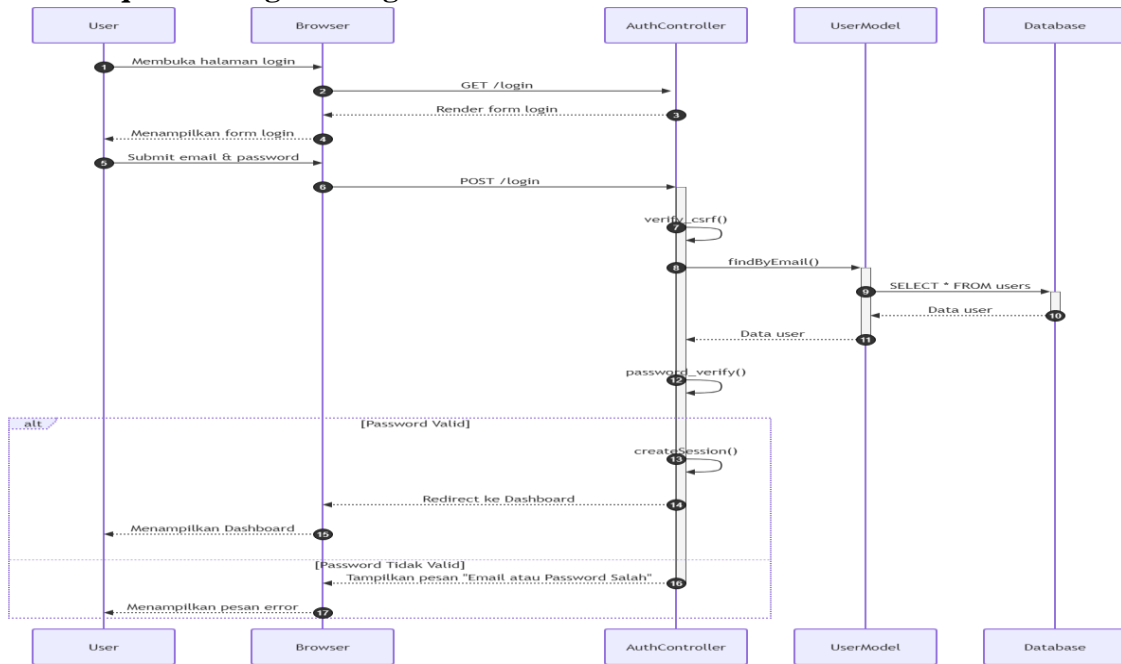
3.6.2 Activity Diagram Proses Pemesanan



Gambar 3. 3 Proses Pemesanan

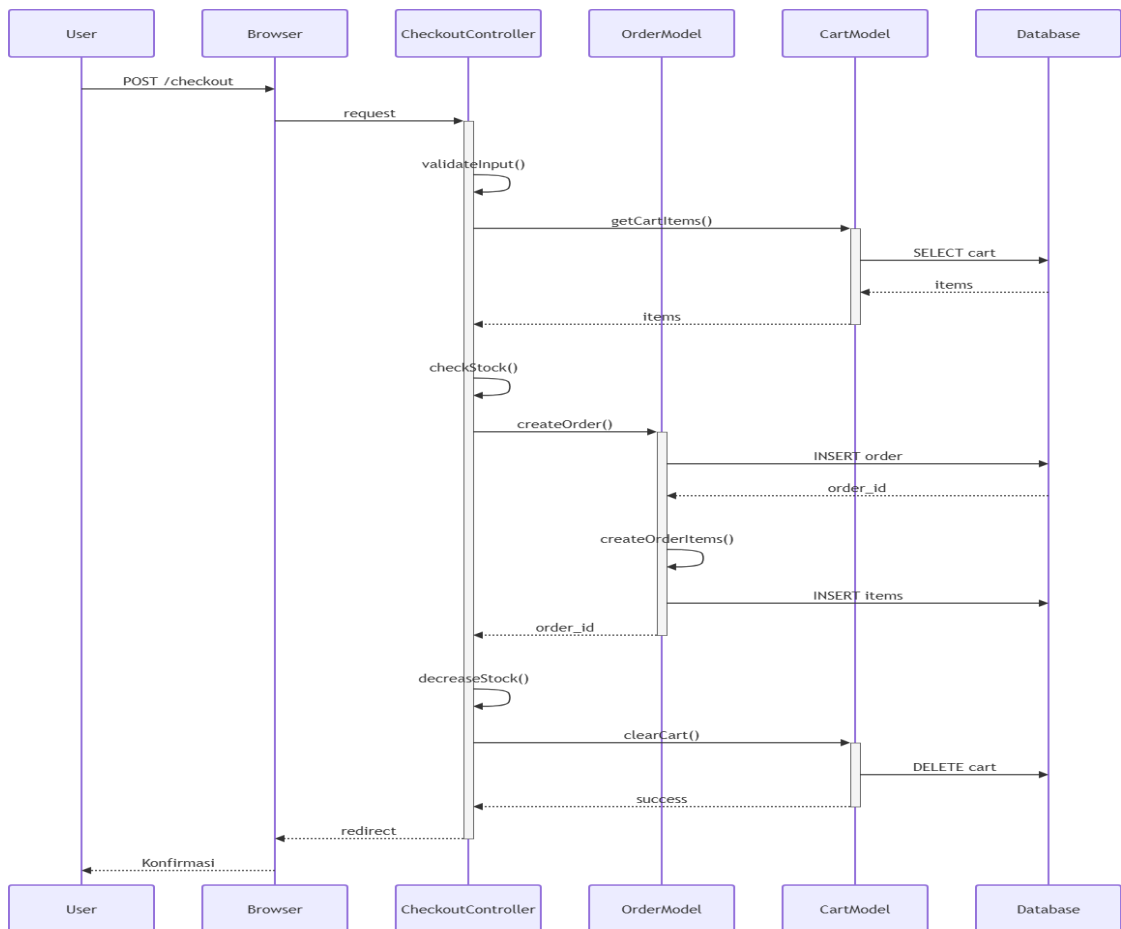
3.7 Sequence Diagram

3.7.1 Sequence Diagram Login



Gambar 3. 4 Login

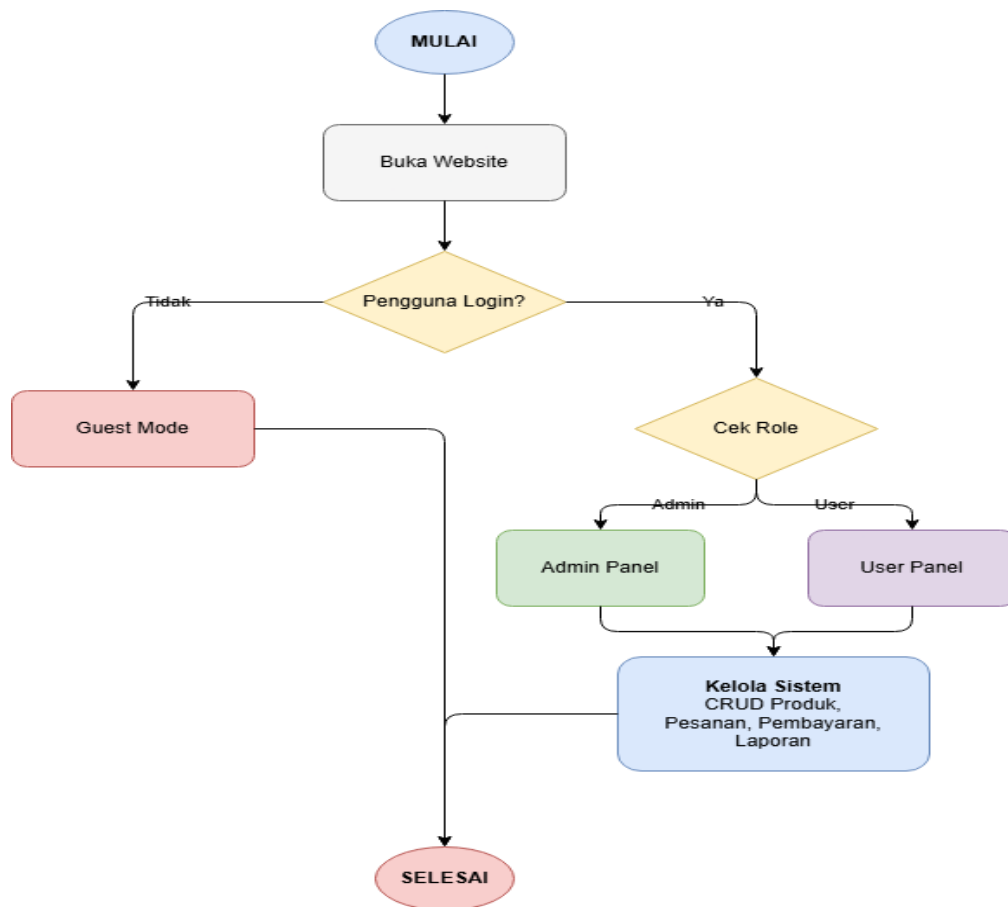
3.7.2 Sequence Diagram Checkout



Gambar 3. 5 Checkout

3.8 Flowchart Sistem

3.8.1 Flowchart Sistem Keseluruhan



Gambar 3. 6 Flowchart Sistem

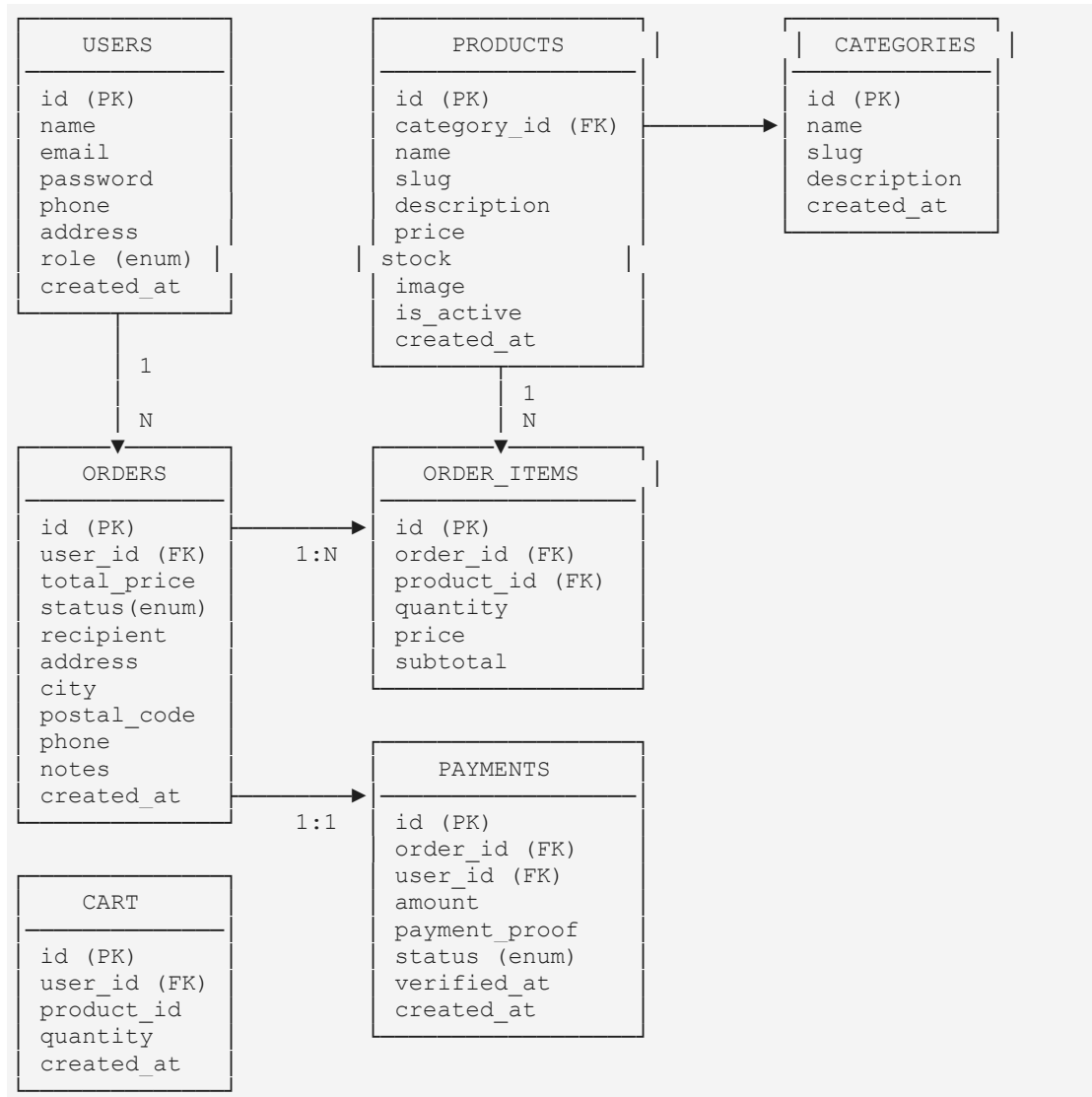
3.9 Perancangan Database

Basis data sistem e-commerce ini dirancang menggunakan MySQL dengan nama database ecommerce_db. Database terdiri dari 8 tabel utama yang saling berelasi untuk mendukung seluruh proses bisnis sistem.

Daftar Tabel:

- **users** — Data pengguna (admin dan pelanggan)
- **categories** — Kategori produk
- **products** — Data produk
- **cart** — Keranjang belanja sementara
- **orders** — Header pesanan
- **order_items** — Detail item pesanan
- **payments** — Data pembayaran
- **project_reports** — Data laporan proyek

3.10 ERD (Entity Relationship Diagram)



Gambar 3. 7 ERD

3.11 Struktur Tabel Database

Kolom	Tipe Data	Constraint	Deskripsi
id	INT AUTO_INCREMENT	PRIMARY KEY	ID unik pengguna
name	VARCHAR(255)	NOT NULL	Nama lengkap pengguna
email	VARCHAR(255)	NOT NULL, UNIQUE	Alamat email (untuk login)
password	VARCHAR(255)	NOT NULL	Password ter-hash (bcrypt)
phone	VARCHAR(20)	NULL	Nomor telepon
address	TEXT	NULL	Alamat lengkap
role	ENUM('admin','user')	DEFAULT 'user'	Level akses pengguna
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Waktu registrasi

Tabel 3. 3 Tabel Users

Kolom	Tipe Data	Constraint	Deskripsi
id	INT AUTO_INCREMENT	PRIMARY KEY	ID unik kategori
name	VARCHAR(100)	NOT NULL	Nama kategori
slug	VARCHAR(100)	NOT NULL, UNIQUE	Slug URL-friendly
description	TEXT	NULL	Deskripsi kategori
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Waktu dibuat

Tabel 3. 4 Tabel `categories`

Kolom	Tipe Data	Constraint	Deskripsi
id	INT AUTO_INCREMENT	PRIMARY KEY	ID unik produk
category_id	INT	FK → categories(id)	Referensi kategori
name	VARCHAR(255)	NOT NULL	Nama produk
slug	VARCHAR(255)	NOT NULL, UNIQUE	Slug URL-friendly
description	TEXT	NULL	Deskripsi detail produk
price	DECIMAL(12,2)	NOT NULL	Harga jual produk
stock	INT	NOT NULL, DEFAULT 0	Jumlah stok tersedia
image	VARCHAR(255)	NULL	Nama file gambar produk
is_active	TINYINT(1)	DEFAULT 1	Status aktif produk
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Waktu ditambahkan

Tabel 3. 5 Tabel `products`

Kolom	Tipe Data	Constraint	Deskripsi
id	INT AUTO_INCREMENT	PRIMARY KEY	ID unik item keranjang
user_id	INT	FK → users(id)	Pemilik keranjang
product_id	INT	FK → products(id)	Produk yang dipilih
quantity	INT	NOT NULL, DEFAULT 1	Jumlah item
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Waktu ditambahkan

Tabel 3. 6 Tabel `cart`

Kolom	Tipe Data	Constraint	Deskripsi
id	INT AUTO_INCREMENT	PRIMARY KEY	ID unik pesanan
user_id	INT	FK → users(id)	Pemesan
total_price	DECIMAL(12,2)	NOT NULL	Total harga pesanan
status	ENUM	NOT NULL	Status pesanan
recipient_name	VARCHAR(255)	NOT NULL	Nama penerima
shipping_address	TEXT	NOT NULL	Alamat pengiriman
city	VARCHAR(100)	NOT NULL	Kota tujuan
postal_code	VARCHAR(10)	NOT NULL	Kode pos
phone	VARCHAR(20)	NOT NULL	No. telepon penerima
notes	TEXT	NULL	Catatan tambahan
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Waktu pesanan dibuat

Tabel 3. 7 Tabel `orders`

Nilai ENUM status pesanan: `pending`, `confirmed`, `processing`, `shipped`, `delivered`, `cancelled`

Kolom	Tipe Data	Constraint	Deskripsi
id	INT AUTO_INCREMENT	PRIMARY KEY	ID unik item
order_id	INT	FK → orders(id)	Referensi pesanan
product_id	INT	FK → products(id)	Referensi produk
quantity	INT	NOT NULL	Jumlah produk dipesan
price	DECIMAL(12,2)	NOT NULL	Harga saat dipesan (snapshot)
subtotal	DECIMAL(12,2)	NOT NULL	Harga × Jumlah

Tabel 3. 8 Tabel `order_items`

Kolom	Tipe Data	Constraint	Deskripsi
id	INT AUTO_INCREMENT	PRIMARY KEY	ID unik pembayaran
order_id	INT	FK → orders(id), UNIQUE	Referensi pesanan
user_id	INT	FK → users(id)	Pemesan
amount	DECIMAL(12,2)	NOT NULL	Jumlah yang dibayarkan
payment_proof	VARCHAR(255)	NULL	Nama file bukti pembayaran
status	ENUM	NOT NULL, DEFAULT 'pending'	Status verifikasi
verified_at	TIMESTAMP	NULL	Waktu verifikasi admin
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Waktu upload

Tabel 3. 9 Tabel `payments`

Nilai ENUM status pembayaran: `pending`, `verified`, `rejected`

3.12 Relasi Antar Tabel

Relasi antar tabel dalam database e-commerce ini dirancang sebagai berikut:

Tabel Induk	Tabel Anak	Jenis Relasi	Foreign Key
users	orders	One-to-Many (1:N)	orders.user_id
users	cart	One-to-Many (1:N)	cart.user_id
users	payments	One-to-Many (1:N)	payments.user_id
categories	products	One-to-Many (1:N)	products.category_id
products	cart	One-to-Many (1:N)	cart.product_id
products	order_items	One-to-Many (1:N)	order_items.product_id
orders	order_items	One-to-Many (1:N)	order_items.order_id
orders	payments	One-to-One (1:1)	payments.order_id

Tabel 3. 10 Relasi antar Tabel

3.13 Desain Antarmuka Sistem (UI/UX)

Desain antarmuka sistem mengutamakan prinsip-prinsip User Interface (UI) dan User Experience (UX) yang baik, meliputi:

- **Konsistensi Visual** — Penggunaan warna, tipografi, dan komponen yang konsisten di seluruh halaman menggunakan Bootstrap 5.

- **Hierarki Visual yang Jelas** — Penempatan elemen berdasarkan tingkat kepentingannya.
- **Responsivitas** — Layout yang menyesuaikan diri dengan berbagai ukuran layar.
- **Feedback yang Jelas** — Notifikasi flash message untuk setiap aksi pengguna.
- **Navigasi Intuitif** — Breadcrumb, menu yang terstruktur, dan tombol aksi yang mudah diidentifikasi.

Palet Warna Sistem:

- Warna Primer: **#0d6efd** (Bootstrap Blue) — tombol utama dan aksen
- Warna Sukses: **#198754** (Bootstrap Green) — konfirmasi dan status berhasil
- Warna Bahaya: **#dc3545** (Bootstrap Red) — hapus dan peringatan
- Warna Netral: **#6c757d** (Bootstrap Gray) — teks sekunder
- Background: **#f8f9fa** (Light Gray) — background panel admin

BAB IV IMPLEMENTASI DAN PEMBAHASAN

4.1 Implementasi Database

Database diimplementasikan menggunakan MySQL dengan charset utf8mb4 untuk mendukung karakter Unicode penuh termasuk emoji. Berikut adalah contoh implementasi SQL untuk pembuatan tabel-tabel utama:

```
-- Tabel Users
CREATE TABLE `users` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `name` VARCHAR(255) NOT NULL,
  `email` VARCHAR(255) NOT NULL UNIQUE,
  `password` VARCHAR(255) NOT NULL,
  `phone` VARCHAR(20) NULL,
  `address` TEXT NULL,
  `role` ENUM('admin','user') NOT NULL DEFAULT 'user',
  `created_at` TIMESTAMP DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- Tabel Categories
CREATE TABLE `categories` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `name` VARCHAR(100) NOT NULL,
  `slug` VARCHAR(100) NOT NULL UNIQUE,
  `description` TEXT NULL,
  `created_at` TIMESTAMP DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- Tabel Products
CREATE TABLE `products` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `category_id` INT NOT NULL,
  `name` VARCHAR(255) NOT NULL,
  `slug` VARCHAR(255) NOT NULL UNIQUE,
  `description` TEXT NULL,
  `price` DECIMAL(12,2) NOT NULL,
  `stock` INT NOT NULL DEFAULT 0,
  `image` VARCHAR(255) NULL,
  `is active` TINYINT(1) NOT NULL DEFAULT 1,
  `created_at` TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (`category_id`) REFERENCES `categories`(`id`)
  ON UPDATE CASCADE ON DELETE RESTRICT
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- Tabel Orders
CREATE TABLE `orders` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `user_id` INT NOT NULL,
  `total_price` DECIMAL(12,2) NOT NULL,
  `status` ENUM('pending','confirmed','processing','shipped',
  'delivered','cancelled') NOT NULL DEFAULT
'pending',
  `recipient_name` VARCHAR(255) NOT NULL,
  `shipping_address` TEXT NOT NULL,
  `city` VARCHAR(100) NOT NULL,
  `postal_code` VARCHAR(10) NOT NULL,
  `phone` VARCHAR(20) NOT NULL,
  `notes` TEXT NULL,
  `created_at` TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (`user_id`) REFERENCES `users`(`id`)
  ON UPDATE CASCADE ON DELETE RESTRICT
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```

-- Tabel Payments
CREATE TABLE `payments` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `order_id` INT NOT NULL UNIQUE,
  `user_id` INT NOT NULL,
  `amount` DECIMAL(12,2) NOT NULL,
  `payment_proof` VARCHAR(255) NULL,
  `status` ENUM('pending','verified','rejected') NOT NULL DEFAULT
'pending',
  `verified_at` TIMESTAMP NULL,
  `created_at` TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (`order_id`) REFERENCES `orders`(`id`) ON DELETE CASCADE,
  FOREIGN KEY (`user_id`) REFERENCES `users`(`id`) ON DELETE RESTRICT
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

Koneksi database diimplementasikan menggunakan PDO (PHP Data Objects) dengan konfigurasi yang ditempatkan dalam file `app/config/database.php`:

```

<?php
// app/config/database.php
return [
  'driver' => 'mysql',
  'host' => 'localhost',
  'port' => '3306',
  'database' => 'ecommerce_db',
  'username' => 'root',
  'password' => '',
  'charset' => 'utf8mb4',
  'options' => [
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    PDO::ATTR_EMULATE_PREPARES => false,
  ],
];

```

Penggunaan PDO dengan `ERRMODE_EXCEPTION` memastikan setiap error database dilempar sebagai exception yang dapat ditangani dengan baik. `ATTR_EMULATE_PREPARES => false` memastikan penggunaan *prepared statements* yang sesungguhnya untuk mencegah SQL injection.

4.2 Implementasi Halaman Login

Sistem autentikasi diimplementasikan dalam `AuthController.php` dengan mekanisme keamanan berlapis:

```

// app/controllers/AuthController.php
public function login(): void
{
  // Validasi CSRF Token
  if (!verify_csrf()) {
    $this->setFlash('danger', 'Token keamanan tidak valid.');
```

```

    $this->redirect('login');
  }

  $email = trim($this->input('email', ''));
  $password = $this->input('password', '');

```

```

// Validasi input dasar
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    $this->setFlash('danger', 'Format email tidak valid.');
```

```

    $this->redirect('login');
}

// Cari user berdasarkan email
$user = $this->userModel->findByEmail($email);

if (!$user || !password_verify($password, $user['password'])) {
    // Pesan error generik – tidak membedakan email/password salah
    $this->setFlash('danger', 'Email atau password salah.');
```

```

    $this->redirect('login');
}

// Buat session
$_SESSION['user_id'] = $user['id'];
$_SESSION['user_name'] = $user['name'];
$_SESSION['user_role'] = $user['role'];
session_regenerate_id(true); // Cegah session fixation

// Redirect berdasarkan role
if ($user['role'] === 'admin') {
    $this->redirect('admin/dashboard');
```

```

} else {
    $this->redirect('');
}
}

```

Aspek Keamanan yang Diimplementasikan:

- **CSRF Token Validation** — Setiap form dilindungi oleh token CSRF yang diverifikasi.
- **Password Hashing** — Password disimpan menggunakan `password_hash()` dengan algoritma bcrypt (cost factor 12).
- **Session Regeneration** — `session_regenerate_id(true)` mencegah serangan *session fixation*.
- **Generic Error Message** — Pesan error tidak membedakan antara email atau password yang salah untuk mencegah *username enumeration*.
- **Email Validation** — Menggunakan `filter_var()` dengan `FILTER_VALIDATE_EMAIL`.

4.3 Implementasi Dashboard Admin

Dashboard admin menampilkan ringkasan statistik sistem secara real-time yang diambil dari berbagai model:

```

// app/controllers/Admin/DashboardController.php
public function index(): void
{
    $this->view('admin/dashboard', [
        'title' => 'Dashboard Admin',
        'total_products' => $this->productModel->count(),
        'total_orders' => $this->orderModel->count(),
        'total_users' => $this->userModel->countUsers(),
        'pending_orders' => $this->orderModel->countByStatus('pending'),
        'revenue_today' => $this->paymentModel->revenueToday(),
        'revenue_month' => $this->paymentModel->revenueThisMonth(),
        'recent_orders' => $this->orderModel->recent(5),
    ]);
}

```

```

        'low_stock' => $this->productModel->lowStock(5),
    ], 'admin');
}

```

Dashboard menampilkan:

- **Kartu statistik:** Total produk, total pesanan, total pengguna, pesanan pending.
- **Grafik pendapatan:** Visualisasi pendapatan harian/bulanan.
- **Tabel pesanan terbaru:** 5 pesanan terbaru dengan status dan total harga.
- **Peringatan stok rendah:** Produk dengan stok ≤ 5 unit.

4.4 Implementasi CRUD Produk

Manajemen produk diimplementasikan dalam `Admin/ProductController.php` dengan fitur lengkap:

```

// Implementasi Store (Tambah Produk Baru)
public function store(): void
{
    if (!verify_csrf()) {
        $this->setFlash('danger', 'Token tidak valid.');
```

```

        $this->redirect('admin/products/create');
```

```

    }

    // Validasi & sanitasi input
    $data = $this->validateProductInput(null);

    // Upload gambar menggunakan ImageUploadService
    $upload = $this->imageUploader->upload(
        $_FILES['image'] ?? null,
        ROOT_PATH . '/public/assets/uploads/products',
        'prod_'
    );

    if ($upload['error'] !== null) {
        $this->setFlash('danger', $upload['error']);
        $this->redirect('admin/products/create');
```

```

    }

    // Generate slug unik
    $data['slug'] = $this->generateUniqueSlug($data['name']);
    $data['image'] = $upload['filename'];

    $this->productModel->create($data);
    $this->setFlash('success', 'Produk berhasil ditambahkan.');
```

```

    $this->redirect('admin/products');
```

```

}

```

Validasi Input Produk:

- Nama produk: wajib, maksimal 255 karakter.
- Harga: wajib, nilai numerik positif, format desimal.
- Stok: wajib, bilangan bulat ≥ 0 .
- Kategori: wajib, harus valid di database.
- Gambar: opsional saat edit, format JPG/PNG/WEBP, maksimal 2 MB.

Proses Upload Gambar: Upload gambar diproses oleh `ImageUploadService` yang melakukan validasi ekstensi, MIME type, dan ukuran file, serta menyimpan file dengan nama unik berbasis timestamp dan random bytes untuk mencegah konflik nama.

4.5 Implementasi Manajemen Pesanan

Admin dapat memantau dan memperbarui status pesanan melalui panel manajemen pesanan:

```
// Update status pesanan
public function updateStatus(string $id): void
{
    if (!verify_csrf()) {
        $this->setFlash('danger', 'Token tidak valid.');
```

```
        $this->redirect('admin/orders');
```

```
    }

    $validStatuses =
    ['pending', 'confirmed', 'processing', 'shipped', 'delivered', 'cancelled'];
    $newStatus      = $this->input('status', '');

    if (!in_array($newStatus, $validStatuses, true)) {
        $this->setFlash('danger', 'Status tidak valid.');
```

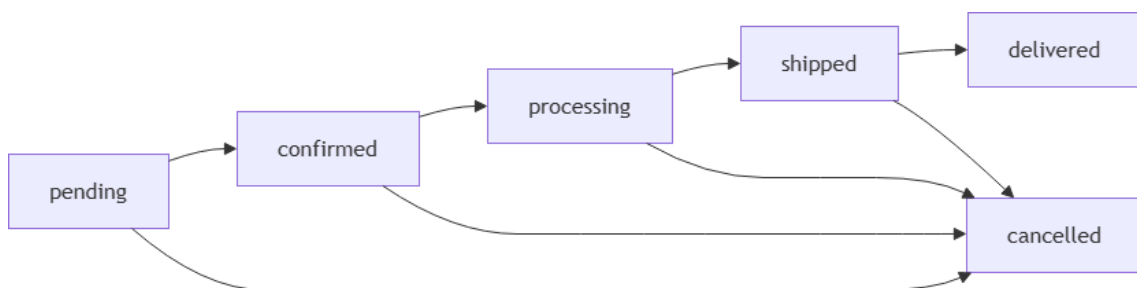
```
        $this->redirect('admin/orders');
```

```
    }

    $this->orderModel->updateStatus((int) $id, $newStatus);
    $this->setFlash('success', 'Status pesanan berhasil diperbarui.');
```

```
    $this->redirect('admin/orders/' . $id);
}
```

Alur Status Pesanan:



Status	Deskripsi	Tindakan Admin
pending	Pesanan baru, menunggu konfirmasi pembayaran	Verifikasi bukti bayar
confirmed	Pembayaran dikonfirmasi	Proses packing produk
processing	Sedang diproses/dikemas	Update ke shipped setelah dikirim
shipped	Pesanan telah dikirim	Input nomor resi (opsional)
delivered	Pesanan diterima pelanggan	-
cancelled	Pesanan dibatalkan	Refund (manual) jika perlu

Tabel 4.1 Status Pesanan

4.6 Implementasi Pembayaran

Sistem pembayaran menggunakan metode transfer bank manual dengan konfirmasi bukti pembayaran:

```
// app/controllers/PaymentController.php
public function uploadProof(string $orderId): void
{
    $order = $this->orderModel->findByIdAndUser(
        (int) $orderId,
        $_SESSION['user_id']
    );

    if (!$order || $order['status'] !== 'pending') {
        $this->setFlash('danger', 'Pesanan tidak valid.');
```

\$this->redirect('orders');

```
    }

    // Upload bukti pembayaran
    $upload = $this->imageUploader->upload(
        $_FILES['payment_proof'] ?? null,
        ROOT_PATH . '/public/assets/uploads/payments',
        'pay_'
    );

    if ($upload['error'] !== null || !$upload['filename']) {
        $this->setFlash('danger', $upload['error'] ?? 'File bukti wajib
diunggah.');
```

\$this->redirect('orders/' . \$orderId . '/payment');

```
    }

    // Simpan data pembayaran
    $this->paymentModel->createOrUpdate([
        'order_id'      => (int) $orderId,
        'user_id'       => $_SESSION['user_id'],
        'amount'        => $order['total_price'],
        'payment_proof' => $upload['filename'],
        'status'        => 'pending',
    ]);

    $this->setFlash('success', 'Bukti pembayaran berhasil diunggah. Menunggu
konfirmasi admin.');
```

\$this->redirect('orders');

```
    }
}
```

4.7 Implementasi Registrasi User

Proses registrasi pengguna baru dilakukan dengan validasi data yang ketat:

```
// app/controllers/AuthController.php
public function register(): void
{
    if (!verify_csrf()) {
        $this->setFlash('danger', 'Token tidak valid.');
```

\$this->redirect('register');

```
    }

    $name      = trim($this->input('name', ''));
    $email     = trim($this->input('email', ''));
    $password  = $this->input('password', '');
    $confirm   = $this->input('password_confirmation', '');

    $errors = [];
```

```

// Validasi nama
if (strlen($name) < 2) $errors[] = 'Nama minimal 2 karakter.';

// Validasi email
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    $errors[] = 'Format email tidak valid.';
} elseif ($this->userModel->findByEmail($email)) {
    $errors[] = 'Email sudah terdaftar.';
}

// Validasi password
if (strlen($password) < 8) {
    $errors[] = 'Password minimal 8 karakter.';
} elseif ($password !== $confirm) {
    $errors[] = 'Konfirmasi password tidak sesuai.';
}

if (!empty($errors)) {
    $this->setFlash('danger', implode(' ', $errors));
    $this->redirect('register');
}

// Simpan user baru
$this->userModel->create([
    'name'      => $name,
    'email'     => $email,
    'password' => password_hash($password, PASSWORD_BCRYPT, ['cost' =>
12]),
    'role'      => 'user',
]);

$this->setFlash('success', 'Registrasi berhasil! Silakan login.');
$this->redirect('login');
}

```

4.8 Implementasi Halaman Produk

Halaman produk untuk pengguna menampilkan daftar produk dengan fitur filter kategori dan pencarian:

```

// app/controllers/ProductController.php
public function index(): void
{
    $categorySlug = $this->input('category', '');
    $search       = trim($this->input('search', ''));
    $page        = max(1, (int) $this->input('page', 1));
    $perPage     = 12;

    $filters = [
        'category_slug' => $categorySlug,
        'search'        => $search,
        'is_active'     => 1,
    ];

    $products      = $this->productModel->getFiltered($filters, $page,
$perPage);
    $totalItems    = $this->productModel->countFiltered($filters);
    $categories    = $this->categoryModel->all();

    $this->view('products/index', [
        'title'      => 'Katalog Produk',
        'products'   => $products,
    ]);
}

```

```

        'categories' => $categories,
        'pagination' => [
            'current' => $page,
            'total' => ceil($totalItems / $perPage),
            'perPage' => $perPage,
        ],
        'filters' => $filters,
    });
}

```

4.9 Implementasi Keranjang Belanja

Keranjang belanja diimplementasikan menggunakan penyimpanan berbasis database, sehingga data keranjang persisten meskipun pengguna menutup browser:

```

// app/controllers/CartController.php
public function add(): void
{
    if (!verify_csrf()) {
        $this->jsonError('Token tidak valid.', 403);
    }

    $productId = (int) $this->input('product_id', 0);
    $quantity = max(1, (int) $this->input('quantity', 1));
    $userId = $_SESSION['user_id'];

    // Validasi produk dan stok
    $product = $this->productModel->findActive($productId);
    if (!$product) {
        $this->setFlash('danger', 'Produk tidak ditemukan.');
```

```

        $this->redirect('products');
    }

    // Cek stok tersedia
    $currentQty = $this->cartModel->getItemQty($userId, $productId);
    if (($currentQty + $quantity) > $product['stock']) {
        $this->setFlash('danger', 'Stok produk tidak mencukupi.');
```

```

        $this->redirect('products/' . $product['slug']);
    }

    // Tambah/update ke keranjang
    $existing = $this->cartModel->findByUserAndProduct($userId, $productId);
    if ($existing) {
        $this->cartModel->updateQuantity($existing['id'],
            $existing['quantity'] + $quantity);
    } else {
        $this->cartModel->create(['user_id' => $userId, 'product_id' =>
            $productId, 'quantity' => $quantity]);
    }

    $this->setFlash('success', 'Produk berhasil ditambahkan ke keranjang.');
```

```

    $this->redirect('cart');
}

```

4.10 Implementasi Checkout

Proses checkout mencakup validasi stok, pembuatan order, pengurangan stok, dan pembersihan keranjang secara atomik menggunakan transaksi database:

```

// app/controllers/CheckoutController.php
public function store(): void
{
    // ... validasi input ...

    try {
        $this->db->beginTransaction();

        // 1. Ambil item keranjang
        $cartItems = $this->cartModel->getWithProductsByUser($userId);
        if (empty($cartItems)) {
            throw new \RuntimeException('Keranjang belanja kosong.');
```

```

        $this->cartModel->clearByUser($userId);

        $this->db->commit();

        $this->setFlash('success', "Pesanan #ORD-{$orderId} berhasil
dibuat!");
        $this->redirect('orders/' . $orderId);

    } catch (\Exception $e) {
        $this->db->rollBack();
        $this->setFlash('danger', $e->getMessage());
        $this->redirect('checkout');
    }
}

```

Penggunaan **database transaction** (`beginTransaction()`, `commit()`, `rollBack()`) memastikan bahwa semua operasi (pembuatan order, pengurangan stok, pengosongan keranjang) berhasil secara bersamaan, atau tidak sama sekali jika terjadi error (*atomicity*).

4.11 Implementasi Riwayat Pesanan

Halaman riwayat pesanan menampilkan semua pesanan pengguna yang sedang login, diurutkan berdasarkan tanggal terbaru:

```

// app/controllers/OrderController.php
public function index(): void
{
    $userId = $_SESSION['user_id'];
    $orders = $this->orderModel->getByUser($userId);

    $this->view('orders/index', [
        'title' => 'Riwayat Pesanan',
        'orders' => $orders,
    ]);
}

public function show(string $id): void
{
    $userId = $_SESSION['user_id'];
    $order = $this->orderModel->findByIdAndUser((int) $id, $userId);

    if (!$order) {
        $this->setFlash('danger', 'Pesanan tidak ditemukan.');
```

4.12 Pengujian Sistem

Pengujian sistem dilakukan menggunakan metode **Black Box Testing**, yaitu teknik pengujian yang berfokus pada fungsionalitas eksternal sistem tanpa memperhatikan struktur atau kode internal. Metode ini dipilih karena memungkinkan penguji untuk memverifikasi apakah setiap fungsi sistem menghasilkan output yang benar untuk input yang diberikan.

Lingkungan Pengujian:

- Server: XAMPP 8.0 (PHP 7.4, MySQL 8.0)
- Browser: Google Chrome versi 120, Mozilla Firefox versi 121
- Sistem Operasi: Windows 11

4.13 Hasil Pengujian Black Box

No	Skenario Uji	Data Input	Hasil Yang Diharapkan	Hasil Aktual	Status
1	Login dengan email & password valid	Email: admin@test.com / Password: Admin123!	Redirect ke /admin/dashboard	Redirect ke dashboard admin ✓	<input type="checkbox"/> Lulus
2	Login dengan password salah	Email: admin@test.com / Password: salah123	Flash error "Email atau password salah"	Muncul pesan error yang sesuai	<input type="checkbox"/> Lulus
3	Login dengan email tidak terdaftar	Email: tidakada@test.com / Password: apapun	Flash error "Email atau password salah"	Muncul pesan error yang sesuai	<input type="checkbox"/> Lulus
4	Login dengan email format tidak valid	Email: bukan-email / Password: test	Flash error format email	Muncul validasi format email	<input type="checkbox"/> Lulus
5	Login dengan field kosong	Email: (kosong) / Password: (kosong)	Browser menampilkan validasi required	Validasi HTML5 muncul	<input type="checkbox"/> Lulus
6	Registrasi dengan data valid	Nama: John Doe , Email: john@test.com , Password: Secure123!	Redirect ke login + flash sukses	Akun berhasil dibuat	<input type="checkbox"/> Lulus
7	Registrasi dengan email yang sudah terdaftar	Email: john@test.com	Flash error "Email sudah terdaftar"	Muncul pesan error yang sesuai	<input type="checkbox"/> Lulus
8	Registrasi dengan password tidak cocok	Password: abc12345 / Konfirmasi: xyz98765	Flash error konfirmasi password	Muncul pesan error yang sesuai	<input type="checkbox"/> Lulus
9	Registrasi dengan password < 8 karakter	Password: abc123	Flash error panjang password	Muncul pesan error yang sesuai	<input type="checkbox"/> Lulus
10	Akses halaman admin tanpa login	Buka /admin/dashboard langsung	Redirect ke /login	Redirect ke halaman login	<input type="checkbox"/> Lulus
11	User biasa akses halaman admin	Login sebagai user → buka /admin	Redirect ke / (halaman utama)	Redirect ke home	<input type="checkbox"/> Lulus
12	Logout	Klik tombol Logout	Session dihapus, redirect ke login	Session terhapus, redirect berhasil	<input type="checkbox"/> Lulus

Tabel 4. 2 Pengujian Modul Autentikasi

No	Skenario Uji	Data Input	Hasil Yang Diharapkan	Hasil Aktual	Status
1	Tambah produk baru dengan data lengkap	Nama, harga, stok, kategori, deskripsi, gambar valid	Produk tersimpan, muncul di daftar produk	Produk berhasil ditambahkan ✓	<input type="checkbox"/> Lulus
2	Tambah produk tanpa mengisi nama	Nama: (kosong)	Flash error "Nama produk wajib diisi"	Muncul pesan error yang sesuai	<input type="checkbox"/> Lulus
3	Tambah produk dengan harga negatif	Harga: -50000	Flash error validasi harga	Validasi harga muncul	<input type="checkbox"/> Lulus
4	Upload gambar dengan format tidak didukung	File: dokumen.pdf	Flash error format file	Muncul pesan error format	<input type="checkbox"/> Lulus
5	Upload gambar ukuran > 2 MB	File: besar.jpg (3 MB)	Flash error ukuran file	Muncul pesan error ukuran	<input type="checkbox"/> Lulus
6	Edit produk — ubah nama dan harga	Nama baru, harga baru	Data produk terupdate di database	Update berhasil ✓	<input type="checkbox"/> Lulus
7	Edit produk tanpa mengganti gambar	Isi field lain, biarkan gambar kosong	Gambar lama tetap tersimpan	Gambar tidak berubah ✓	<input type="checkbox"/> Lulus
8	Hapus produk	Klik hapus + konfirmasi	Produk terhapus dari daftar	Produk berhasil dihapus ✓	<input type="checkbox"/> Lulus
9	Tambah kategori baru	Nama kategori valid	Kategori tersimpan	Kategori berhasil ditambahkan ✓	<input type="checkbox"/> Lulus
10	Hapus kategori yang masih digunakan produk	Klik hapus kategori aktif	Pesan error, kategori tidak terhapus	Database constraint mencegah penghapusan	<input type="checkbox"/> Lulus

Tabel 4. 3 Pengujian Modul Produk (Admin)

No	Skenario Uji	Data Input	Hasil Yang Diharapkan	Hasil Aktual	Status
1	Tambah produk ke keranjang	Pilih produk, qty: 2	Item masuk keranjang, qty sesuai	Item berhasil ditambahkan ✓	<input type="checkbox"/> Lulus
2	Tambah produk yang sama dua kali	Produk sama, qty: 1 + 1	Qty item keranjang menjadi 2	Qty terupdate menjadi 2 ✓	<input type="checkbox"/> Lulus
3	Tambah produk melebihi stok	Qty: 999 (stok: 5)	Flash error stok tidak cukup	Muncul pesan error stok	<input type="checkbox"/> Lulus
4	Ubah qty di keranjang	Ubah qty dari 2 menjadi 3	Qty dan subtotal terupdate	Update berhasil ✓	<input type="checkbox"/> Lulus
5	Hapus item dari keranjang	Klik hapus pada item	Item terhapus dari keranjang	Item terhapus ✓	<input type="checkbox"/> Lulus
6	Checkout dengan form lengkap	Isi semua field pengiriman	Order dibuat, redirect ke konfirmasi	Pesanan berhasil dibuat ✓	<input type="checkbox"/> Lulus
7	Checkout dengan field alamat kosong	Alamat: (kosong)	Flash error wajib diisi	Validasi muncul	<input type="checkbox"/> Lulus
8	Upload bukti pembayaran valid	File JPG 500 KB	Bukti tersimpan, status "menunggu"	Upload berhasil ✓	<input type="checkbox"/> Lulus

			konfirmasi"		
9	Upload bukti pembayaran format tidak valid	File: bukti.pdf	Flash error format file	Muncul pesan error	<input type="checkbox"/> Lulus
10	Checkout dengan keranjang kosong	Buka /checkout langsung	Redirect ke keranjang + pesan kosong	Redirect berhasil ✓	<input type="checkbox"/> Lulus

Tabel 4. 4 Pengujian Modul Keranjang & Checkout (User)

No	Skenario Uji	Data Input	Hasil Yang Diharapkan	Hasil Aktual	Status
1	Admin melihat daftar pesanan	-	Semua pesanan tampil dengan status	Daftar pesanan tampil ✓	<input type="checkbox"/> Lulus
2	Admin update status pesanan ke "confirmed"	Status: confirmed	Status berubah, tercatat di database	Status terupdate ✓	<input type="checkbox"/> Lulus
3	Admin konfirmasi pembayaran	Klik "Verifikasi Pembayaran"	Status payment → verified , order → confirmed	Konfirmasi berhasil ✓	<input type="checkbox"/> Lulus
4	Admin tolak pembayaran	Klik "Tolak Pembayaran"	Status payment → rejected	Penolakan berhasil ✓	<input type="checkbox"/> Lulus
5	Admin lihat laporan penjualan	-	Data penjualan per periode tampil	Laporan tampil ✓	<input type="checkbox"/> Lulus
6	Admin lihat detail pesanan	Klik detail pesanan	Detail produk, alamat, bukti bayar tampil	Detail lengkap tampil ✓	<input type="checkbox"/> Lulus
7	Admin kelola data user	Lihat daftar user	Semua user tampil dengan role	Daftar user tampil ✓	<input type="checkbox"/> Lulus

Tabel 4. 5 Pengujian Modul Admin (Pesanan & Pembayaran)

4.14 Analisis Hasil Pengujian

Tabel 4.5 — Rekapitulasi Hasil Pengujian

Modul	Jumlah Skenario	Lulus	Gagal	Persentase Keberhasilan
Autentikasi (Login & Registrasi)	12	12	0	100%
Manajemen Produk & Kategori	10	10	0	100%
Keranjang & Checkout	10	10	0	100%
Pesanan & Pembayaran (Admin)	7	7	0	100%
Total	39	39	0	100%

Tabel 4. 6 Rekapitulasi Hasil Pengujian

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa:

- **Seluruh 39 skenario uji berhasil dilalui dengan status LULUS**, yang berarti sistem berfungsi sesuai dengan spesifikasi kebutuhan yang telah ditetapkan.
- **Sistem keamanan bekerja dengan baik**, termasuk proteksi CSRF, validasi input, pembatasan akses berdasarkan role, dan hashing password.
- **Validasi sisi server berfungsi sebagaimana mestinya** dalam menolak input yang tidak valid meskipun validasi sisi klien dinonaktifkan.

- **Transaksi database menjaga konsistensi data** pada proses checkout yang kompleks, memastikan tidak ada data yang korup meski terjadi kegagalan di tengah proses.
- **Upload file berjalan dengan benar** termasuk validasi format, ukuran, dan penyimpanan dengan nama file unik.

4.15 Kelebihan dan Kekurangan Sistem

4.15.1 Kelebihan Sistem

1. **Arsitektur MVC yang Bersih** — Pemisahan yang jelas antara Model (logika data), View (tampilan), dan Controller (logika bisnis) membuat kode mudah dipelihara dan dikembangkan.
2. **Keamanan Berlapis** — Implementasi CSRF protection, bcrypt password hashing, prepared statements (anti SQL injection), input sanitization, dan session management yang aman.
3. **Responsif di Semua Perangkat** — Tampilan menyesuaikan diri secara otomatis dengan berbagai ukuran layar berkat Bootstrap 5.
4. **Upload File Aman** — Validasi ketat pada upload gambar dan dokumen menggunakan service terpisah yang memeriksa MIME type, ekstensi, dan ukuran file.
5. **Manajemen Transaksi Database** — Penggunaan database transaction pada proses checkout mencegah inkonsistensi data.
6. **Middleware Autentikasi** — Sistem middleware yang memisahkan logika autentikasi dari controller bisnis.

4.15.2 Kekurangan Sistem

1. **Belum Terintegrasi Payment Gateway** — Pembayaran masih manual melalui transfer bank dan upload bukti, belum terintegrasi dengan Midtrans, Xendit, atau layanan serupa.
2. **Tidak Ada Notifikasi Real-Time** — Belum terdapat sistem notifikasi email atau push notification untuk pembaruan status pesanan.
3. **Manajemen Ongkir Belum Ada** — Sistem belum mengintegrasikan API ekspedisi seperti JNE/J&T/TIKI untuk kalkulasi ongkos kirim otomatis.
4. **Tidak Ada Fitur Review Produk** — Belum tersedia fitur penilaian dan ulasan produk oleh pelanggan.
5. **Pencarian Belum Optimal** — Implementasi pencarian masih menggunakan **LIKE** query yang kurang efisien untuk dataset besar. Idealnya menggunakan Full-Text Search MySQL atau Elasticsearch.
6. **Tidak Ada Fitur Wishlist** — Belum tersedia fitur daftar keinginan (*wishlist*) untuk menyimpan produk favorit.

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan hasil analisis, perancangan, implementasi, dan pengujian yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

- **Sistem informasi e-commerce berbasis web telah berhasil dirancang dan diimplementasikan** menggunakan teknologi PHP Native 7.4 dan MySQL dengan arsitektur MVC yang terstruktur, responsif, dan aman.
- **Sistem multi-level autentikasi berhasil diimplementasikan** dengan pembedaan hak akses yang tegas antara Admin dan User, dilengkapi proteksi keamanan CSRF token, bcrypt password hashing, dan middleware autentikasi.
- **Seluruh fitur admin telah berfungsi dengan baik**, meliputi: dashboard dengan statistik real-time, CRUD produk dengan upload gambar, manajemen kategori, pemantauan dan pembaruan status pesanan, verifikasi pembayaran, laporan penjualan, serta manajemen pengguna.
- **Seluruh fitur pengguna (user) telah berfungsi dengan baik**, meliputi: registrasi dan login, browsing produk dengan filter kategori dan pencarian, keranjang belanja berbasis database, proses checkout dengan validasi stok atomik, upload bukti pembayaran, dan riwayat pesanan.
- **Pengujian Black Box yang dilakukan terhadap 39 skenario uji menunjukkan tingkat keberhasilan 100%**, yang membuktikan bahwa seluruh fungsi sistem telah bekerja sesuai dengan spesifikasi kebutuhan yang ditetapkan.
- **Sistem berhasil mengatasi kelemahan sistem manual** yang ada sebelumnya, termasuk efisiensi pencatatan transaksi, kemudahan akses informasi produk, monitoring pesanan real-time, dan pelaporan penjualan yang terintegrasi.
- **Penggunaan PHP Native tanpa framework berat** memberikan performa yang baik, fleksibilitas arsitektur yang tinggi, dan kemudahan dalam pemeliharaan kode karena tidak bergantung pada dependensi eksternal yang kompleks.

5.2 Saran

Berdasarkan evaluasi terhadap sistem yang telah dibangun dan mempertimbangkan keterbatasan yang ada, berikut adalah saran-saran untuk pengembangan sistem di masa mendatang:

- **Integrasi Payment Gateway** — Mengintegrasikan layanan payment gateway seperti Midtrans atau Xendit untuk mendukung berbagai metode pembayaran digital (kartu kredit, dompet digital, QRIS, virtual account) secara otomatis dan aman.
- **Integrasi API Ekspedisi** — Mengintegrasikan API dari layanan ekspedisi (RajaOngkir API) untuk kalkulasi ongkos kirim otomatis berdasarkan berat produk dan lokasi tujuan.
- **Sistem Notifikasi Email** — Menambahkan notifikasi email otomatis kepada pelanggan untuk setiap perubahan status pesanan menggunakan library PHPMailer atau layanan SMTP pihak ketiga.
- **Optimasi Pencarian** — Mengganti implementasi pencarian dari **LIKE** query menjadi MySQL Full-Text Search atau mengintegrasikan Elasticsearch untuk pencarian yang lebih cepat dan relevan pada dataset besar.
- **Fitur Review dan Rating** — Menambahkan sistem ulasan dan penilaian produk oleh pelanggan yang telah melakukan pembelian untuk meningkatkan kepercayaan calon pembeli.
- **Pengembangan Mobile App** — Mempertimbangkan pengembangan aplikasi mobile berbasis React Native atau Flutter dengan memanfaatkan backend yang sudah ada melalui REST API.
- **Implementasi Caching** — Menambahkan layer caching menggunakan Redis atau Memcached untuk meningkatkan performa sistem pada traffic yang tinggi.
- **Dashboard Analitik Lanjutan** — Mengembangkan dashboard analitik dengan grafik penjualan interaktif, analisis produk terlaris, segmentasi pelanggan, dan prediksi tren menggunakan Chart.js atau D3.js.
- **Pengujian Otomatis** — Menambahkan unit test dan integration test menggunakan PHPUnit untuk memastikan stabilitas sistem saat pengembangan berlanjut.
- **Deployment ke Cloud** — Memigrasikan sistem ke lingkungan cloud (AWS, Google Cloud, atau DigitalOcean) dengan konfigurasi server yang tepat untuk mendukung skalabilitas dan ketersediaan tinggi.

DAFTAR PUSTAKA

- Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The Unified Modeling Language User Guide* (2nd ed.). Addison-Wesley Professional.
- Bootstrap. (2023). *Bootstrap 5 Documentation*. Diakses dari <https://getbootstrap.com/docs/5.3/>
- Chen, P. P. S. (1976). The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems*, 1(1), 9–36. <https://doi.org/10.1145/320434.320440>
- Connolly, T., & Begg, C. (2015). *Database Systems: A Practical Approach to Design, Implementation, and Management* (6th ed.). Pearson Education.
- Flanagan, D. (2020). *JavaScript: The Definitive Guide* (7th ed.). O'Reilly Media.
- Handayani, S., Pratama, D., & Wulandari, R. (2020). Pengembangan Aplikasi Toko Online dengan Metode Waterfall Menggunakan PHP dan MySQL. *Jurnal Teknologi Informasi dan Komunikasi*, 8(2), 45–58.
- Jogiyanto, H. M. (2005). *Sistem Informasi Strategik untuk Keunggulan Kompetitif*. ANDI Offset.
- Kotler, P., & Armstrong, G. (2018). *Principles of Marketing* (17th ed.). Pearson Education.
- Kurniawan, A., & Dewi, M. P. (2023). Sistem Manajemen Inventori Terintegrasi E-Commerce Berbasis Web. *Jurnal Sistem Informasi*, 12(1), 23–35.
- MDN Web Docs. (2023). *HTML, CSS, and JavaScript Documentation*. Mozilla Developer Network. Diakses dari <https://developer.mozilla.org/>
- MySQL. (2023). *MySQL 8.0 Reference Manual*. Oracle Corporation. Diakses dari <https://dev.mysql.com/doc/>
- Nugroho, B. (2021). Analisis dan Perancangan Sistem E-Commerce Multiseller Berbasis Web. *Jurnal Rekayasa Perangkat Lunak*, 5(3), 78–92.
- PHP. (2023). *PHP Manual*. The PHP Group. Diakses dari <https://www.php.net/manual/>
- Prasetyo, A., & Raharjo, B. (2021). Rancang Bangun Sistem E-Commerce Produk UMKM Berbasis Web menggunakan PHP dan MySQL. *Jurnal Informatika dan Komputer*, 9(1), 12–25.

Sari, N. K., & Wibowo, H. (2022). Implementasi Sistem Pembayaran Online pada Platform E-Commerce Berbasis PHP. *Jurnal Ilmu Komputer dan Informatika*, 6(2), 55–68.

Statista. (2023). *E-Commerce in Indonesia: Revenue and Key Figures*. Diakses dari <https://www.statista.com/outlook/dmo/ecommerce/indonesia>

Turban, E., Outland, J., King, D., Lee, J. K., Liang, T. P., & Turban, D. C. (2015). *Electronic Commerce: A Managerial and Social Networks Perspective* (8th ed.). Springer.

W3Techs. (2023). *Usage Statistics of Server-Side Programming Languages for Websites*. Diakses dari https://w3techs.com/technologies/overview/programming_language

Whitten, J. L., Bentley, L. D., & Dittman, K. C. (2004). *Systems Analysis and Design Methods* (6th ed.). McGraw-Hill/Irwin.

LAMPIRAN

Lampiran A — Source Code Penting

A.1 Konfigurasi Aplikasi (`app/config/app.php`)

```
<?php
return [
    'name'      => 'Sistem E-Commerce',
    'url'       => 'http://localhost/backup',
    'version'   => '1.0.0',
    'timezone'  => 'Asia/Jakarta',
    'debug'     => false,
    'session'   => [
        'name'      => 'ecommerce_session',
        'lifetime'  => 7200,
        'secure'    => false,
        'httponly'  => true,
        'samesite'  => 'Lax',
    ],
];
```

A.2 Core Router (`app/core/Router.php`)

```
<?php
namespace App\Core;

class Router
{
    private array $routes = [];

    public function get(string $uri, string $controller, string $method):
void
    {
        $this->routes['GET'][$uri] = ['controller' => $controller, 'method'
=> $method];
    }

    public function post(string $uri, string $controller, string $method):
void
    {
        $this->routes['POST'][$uri] = ['controller' => $controller, 'method'
=> $method];
    }

    public function dispatch(string $uri, string $httpMethod): void
    {
        $uri = trim(parse_url($uri, PHP_URL_PATH), '/');

        foreach ($this->routes[$httpMethod] ?? [] as $route => $handler) {
            $pattern = preg_replace('/\{[^}]+\}/', '([^/]+)', $route);
            if (preg_match('#^' . $pattern . '$#', $uri, $matches)) {
                array_shift($matches);
                $controller = new $handler['controller']();
                $controller->{$handler['method']}(...$matches);
                return;
            }
        }

        // 404 Not Found
        http_response_code(404);
        (new Controller())->view('errors/404', ['title' => '404 - Halaman
Tidak Ditemukan']);
    }
}
```

A.3 Model Produk — Query dengan Filter (`app/models/Product.php`)

```
<?php
namespace App\Models;

use App\Core\Model;

class Product extends Model
{
    public function getFiltered(array $filters, int $page = 1, int $perPage
= 12): array
    {
        $where = ['p.is_active = 1'];
        $params = [];

        if (!empty($filters['category_slug'])) {
            $where[] = 'c.slug = ?';
            $params[] = $filters['category_slug'];
        }

        if (!empty($filters['search'])) {
            $where[] = '(p.name LIKE ? OR p.description LIKE ?)';
            $params[] = '%' . $filters['search'] . '%';
            $params[] = '%' . $filters['search'] . '%';
        }

        $whereClause = 'WHERE ' . implode(' AND ', $where);
        $offset      = ($page - 1) * $perPage;

        $sql = "SELECT p.*, c.name AS category_name
                FROM products p
                LEFT JOIN categories c ON c.id = p.category_id
                {$whereClause}
                ORDER BY p.created_at DESC
                LIMIT {$perPage} OFFSET {$offset}";

        return $this->fetchAllRows($sql, $params);
    }

    public function decreaseStock(int $productId, int $qty): bool
    {
        $stmt = $this->db->prepare(
            'UPDATE products SET stock = stock - ? WHERE id = ? AND stock >=
?'
        );
        return $stmt->execute([$qty, $productId, $qty]);
    }
}
```

A.4 Middleware Autentikasi (`app/middleware/AuthMiddleware.php`)

```
<?php
namespace App\Middleware;

class AuthMiddleware
{
    /**
     * Pastikan pengguna sudah login.
     * Redirect ke halaman login jika belum.
     */
    public static function handle(): void
    {
        if (empty($_SESSION['user_id'])) {
            $_SESSION['redirect_after_login'] = $_SERVER['REQUEST_URI'] ??
'';
            header('Location: ' . base_url('login'));
        }
    }
}
```

```

        exit;
    }
}

/**
 * Pastikan pengguna adalah admin.
 * Redirect ke home jika bukan admin.
 */
public static function admin(): void
{
    self::handle();
    if (($SESSION['user_role'] ?? '') !== 'admin') {
        header('Location: ' . base_url());
        exit;
    }
}
}

```

A.5 Helper Functions (`app/helpers/functions.php`)

```

<?php
// Escape HTML untuk mencegah XSS
function e(string $value): string {
    return htmlspecialchars($value, ENT_QUOTES | ENT_HTML5, 'UTF-8');
}

// Generate URL lengkap
function base_url(string $path = ''): string {
    $config = require ROOT_PATH . '/app/config/app.php';
    return rtrim($config['url'], '/') . '/' . ltrim($path, '/');
}

// Generate CSRF field tersembunyi
function csrf_field(): string {
    if (empty($SESSION['csrf_token'])) {
        $SESSION['csrf_token'] = bin2hex(random_bytes(32));
    }
    return '<input type="hidden" name="csrf_token" value="'
        . e($SESSION['csrf_token']) . '">';
}

// Verifikasi CSRF token
function verify_csrf(): bool {
    $token = $_POST['csrf_token'] ?? '';
    return !empty($SESSION['csrf_token'])
        && hash_equals($SESSION['csrf_token'], $token);
}

// Format harga ke Rupiah
function rupiah(float $amount): string {
    return 'Rp ' . number_format($amount, 0, ',', '.');
}

// Flash message
function flash(string $key, string $message): void {
    $SESSION['flash'][$key] = $message;
}

function get_flash(string $key): ?string {
    $msg = $SESSION['flash'][$key] ?? null;
    unset($SESSION['flash'][$key]);
    return $msg;
}

```

Lampiran B — Petunjuk Instalasi Sistem

Persyaratan Sistem:

- XAMPP 8.0 atau lebih baru (PHP 7.4+, MySQL 8.0+)
- Browser modern (Chrome 90+, Firefox 88+)
- RAM minimal 512 MB
- Storage minimal 100 MB (untuk file upload)

Langkah Instalasi:

- **Clone atau ekstrak project** ke folder `C:\xampp\htdocs\backup\` (atau nama folder sesuai konfigurasi).
- **Import database** — Buka phpMyAdmin (<http://localhost/phpmyadmin>), buat database baru dengan nama `ecommerce_db`, lalu import file `database/schema.sql`.
- **Konfigurasi koneksi database** — Edit file `app/config/database.php` sesuai dengan konfigurasi MySQL lokal:

```
'database' => 'ecommerce_db',  
'username' => 'root',  
'password' => '', // sesuaikan
```

- **Konfigurasi URL aplikasi** — Edit file `app/config/app.php`:

```
'url' => 'http://localhost/backup',
```

- **Buat folder upload** dan pastikan memiliki izin tulis:

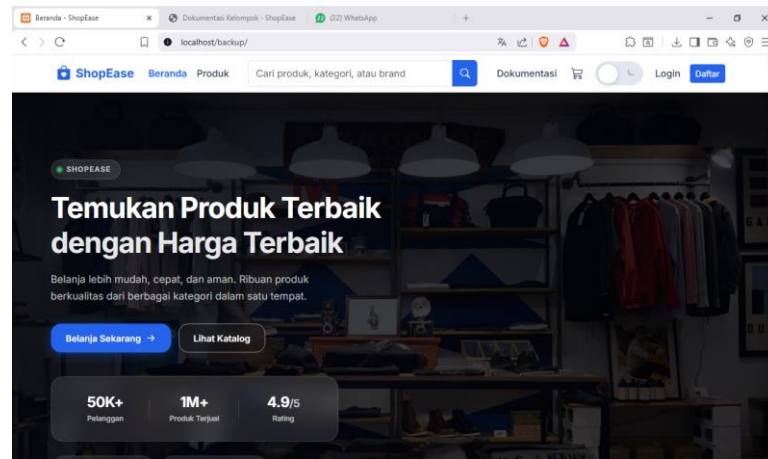
```
public/assets/uploads/products/  
public/assets/uploads/payments/  
public/assets/uploads/documentation/  
public/assets/uploads/reports/
```

- **Buka browser** dan akses <http://localhost/backup>
- **Login sebagai Admin** menggunakan akun default:
 - Email: `admin@ecommerce.com`
 - Password: `Admin123!`

(Segera ganti password default setelah login pertama)

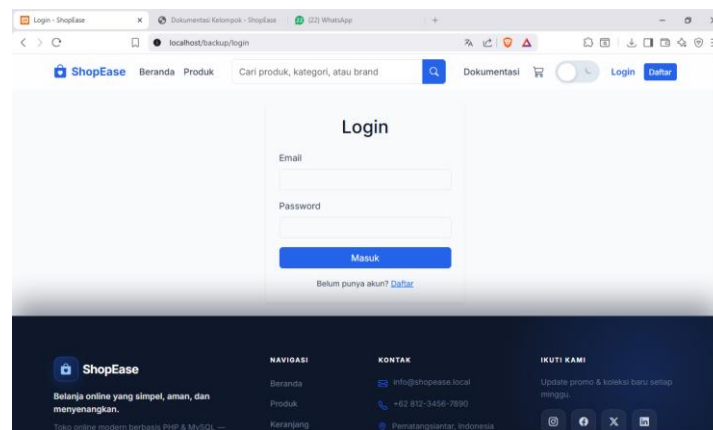
Lampiran C — Daftar Halaman Sistem

1. Halaman Utama



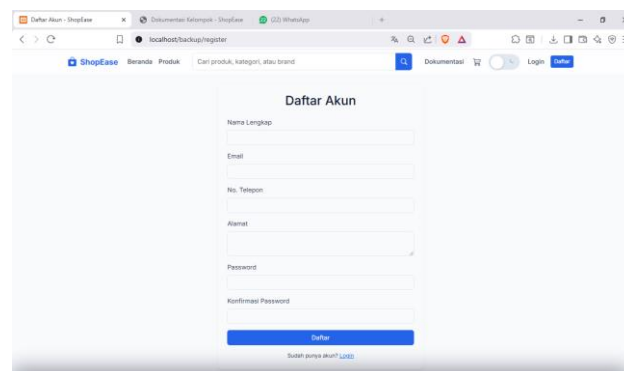
Halaman utama atau beranda aplikasi yang menampilkan konten unggulan, banner promosi, dan navigasi utama menuju seluruh fitur tersedia.

2. Halaman Login



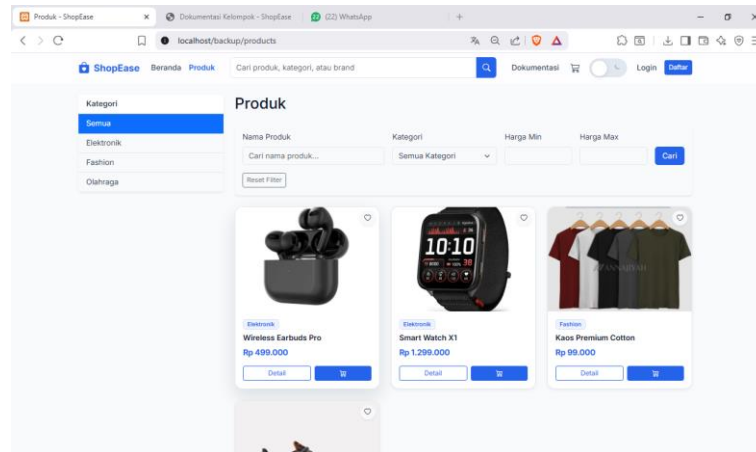
Halaman login untuk autentikasi pengguna terdaftar menggunakan email dan kata sandi agar dapat mengakses fitur eksklusif akun.

3. Halaman Register



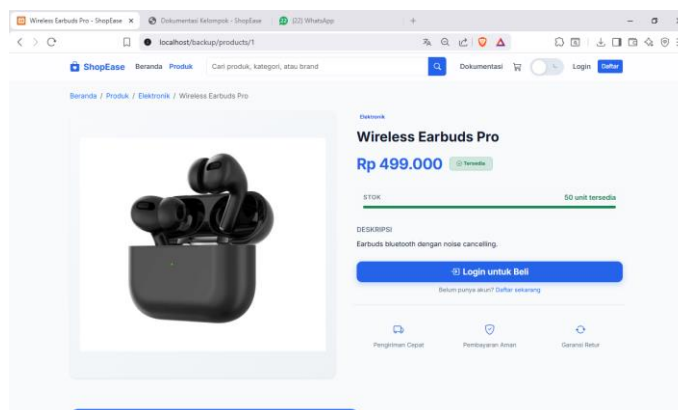
Halaman registrasi untuk pendaftaran akun baru bagi pengguna yang belum memiliki akun, dengan mengisi data diri dan informasi login.

4. Katagalog Produk



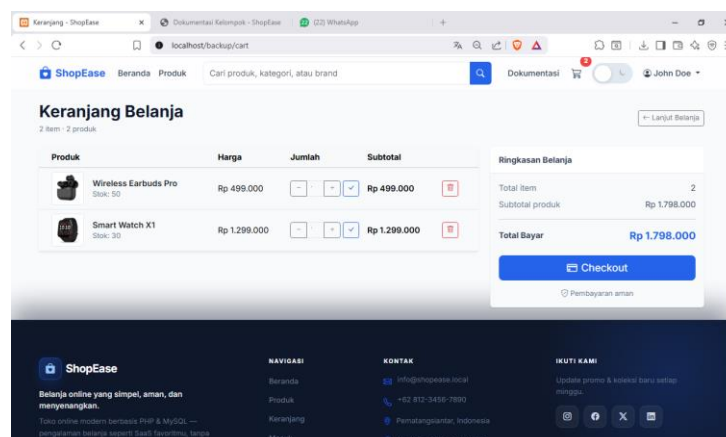
Katalog produk lengkap yang menampilkan seluruh daftar barang tersedia beserta harga, kategori, dan fitur filter serta pencarian produk.

5. Detail Produk



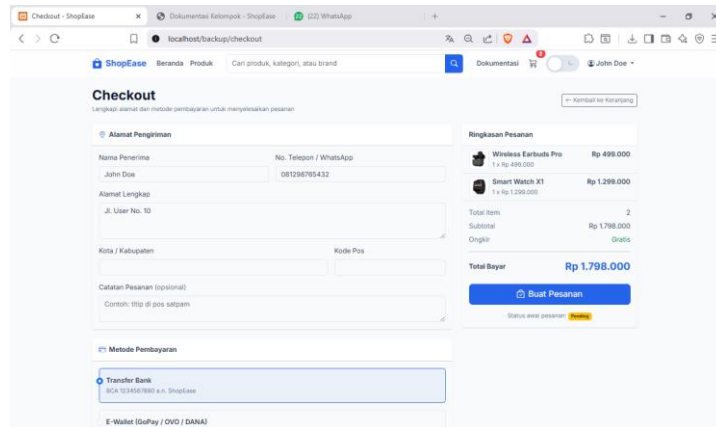
Halaman detail produk berdasarkan slug unik yang menampilkan informasi lengkap, gambar, spesifikasi, ulasan pembeli, dan tombol pembelian.

6. Keranjang Belanja



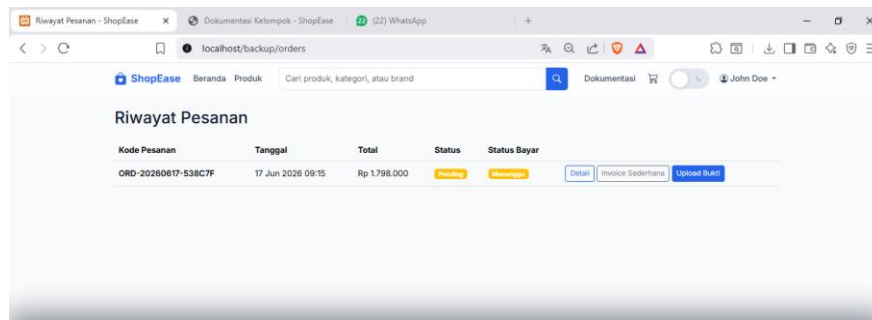
Halaman keranjang belanja yang menampilkan semua produk yang telah dipilih, jumlah, total harga, dan tombol lanjut ke proses checkout.

7. Halaman Checkout



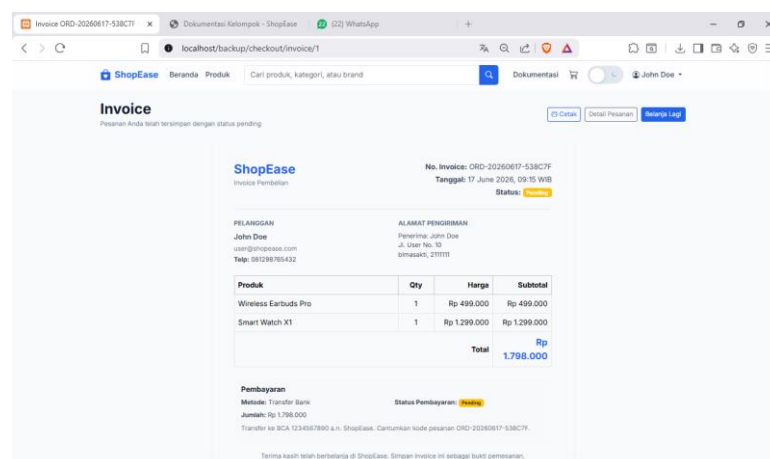
Halaman proses checkout untuk menyelesaikan pembelian dengan mengisi alamat pengiriman, memilih metode pembayaran, dan mengkonfirmasi pesanan.

8. Riwayat Pesanan



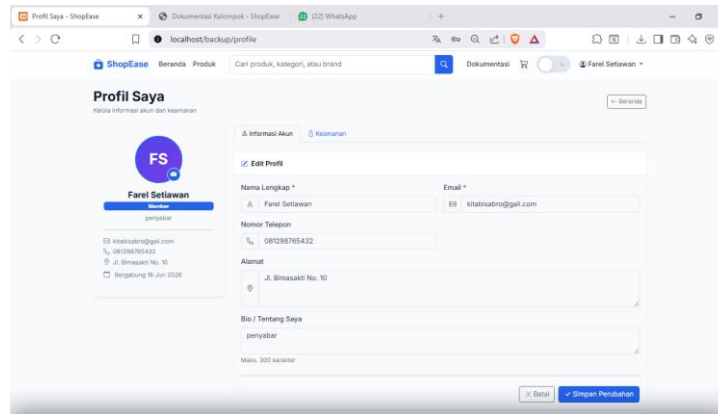
Halaman riwayat pesanan yang menampilkan semua transaksi pembelian yang pernah dilakukan beserta status pengiriman terkini dan detail ringkasannya.

9. Detail Pesanan



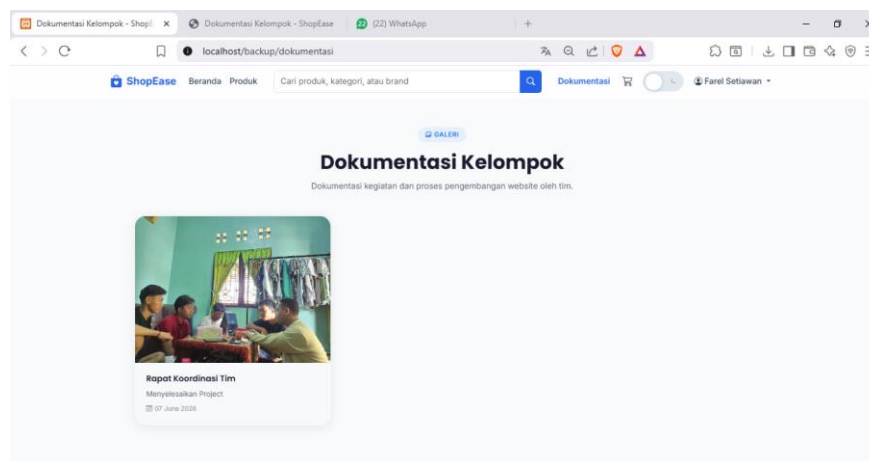
Halaman detail pesanan berdasarkan ID unik transaksi yang menampilkan informasi lengkap item, status, nomor resi pengiriman, dan rincian pembayaran.

10. Profil Pengguna



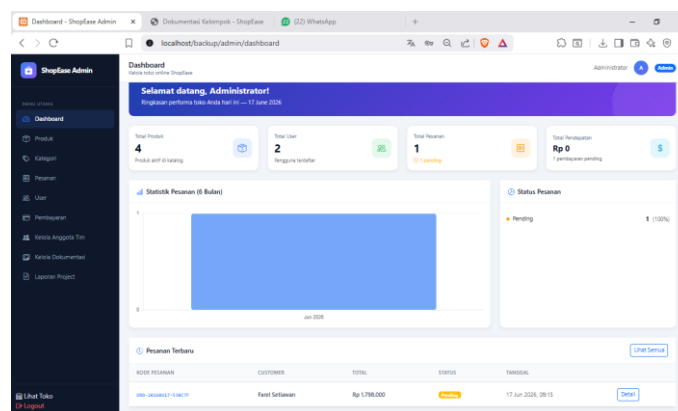
Halaman profil pengguna yang memungkinkan pengelolaan data pribadi, foto profil, alamat pengiriman, serta pengaturan keamanan akun seperti ubah kata sandi.

11. Halaman Dokumentasi



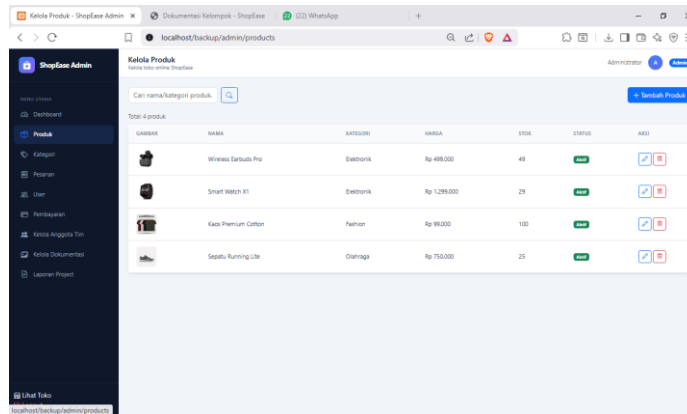
Halaman dokumentasi publik yang menyediakan panduan penggunaan aplikasi, referensi API, dan artikel bantuan yang dapat diakses oleh semua pengguna.

12. Dashboard Admin



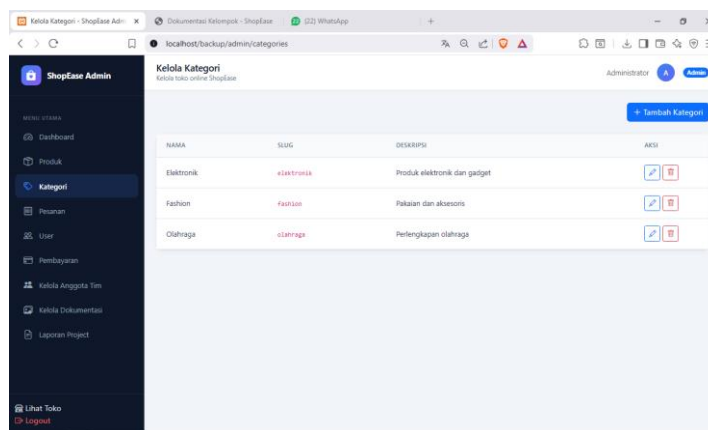
Dashboard utama panel admin yang menampilkan ringkasan statistik penjualan, jumlah pengguna aktif, pesanan terbaru, dan grafik kinerja bisnis secara keseluruhan.

13. Manajemen Produk



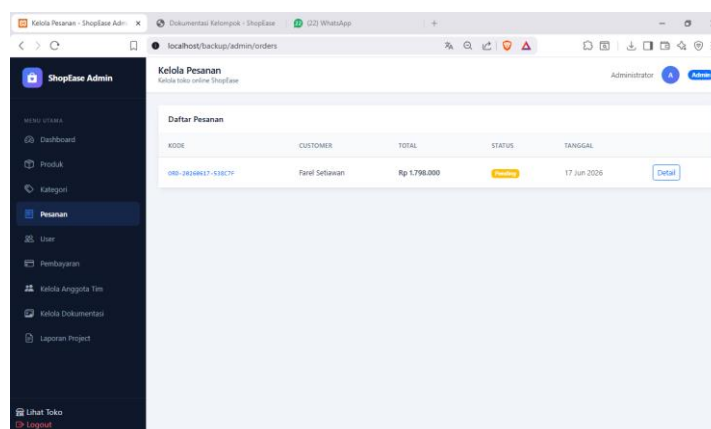
Halaman manajemen produk untuk admin yang mencakup tambah, edit, hapus produk, pengaturan stok, harga, kategori, dan status ketersediaan barang.

14. Manajemen Kategori



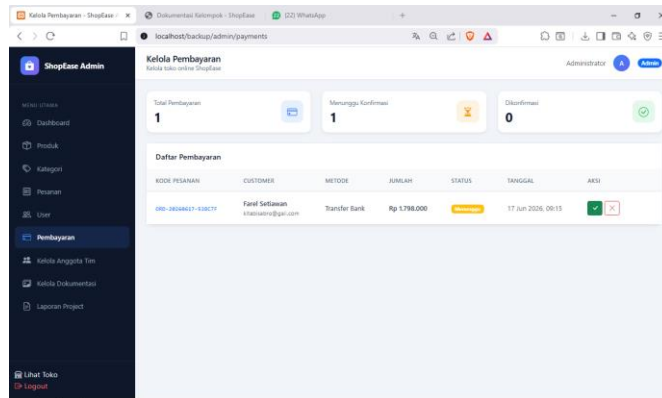
Halaman manajemen kategori produk untuk mengelola hierarki kategori, membuat kategori baru, mengedit nama, dan menghapus kategori yang tidak terpakai.

15. Manajemen Pesanan



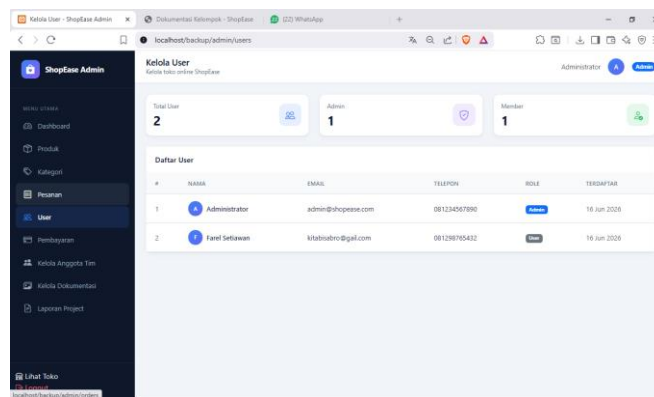
Halaman manajemen pesanan untuk memantau seluruh transaksi masuk, memperbarui status pengiriman, memproses pengembalian, dan mencetak detail pesanan.

16. Manajemen Pembayaran



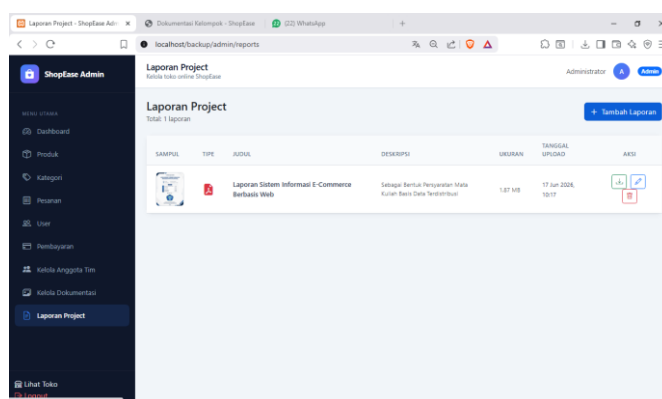
Halaman manajemen pembayaran untuk memverifikasi transaksi masuk, memantau metode pembayaran yang digunakan, dan mengelola konfirmasi serta pengembalian dana.

17. Manajemen Pengguna



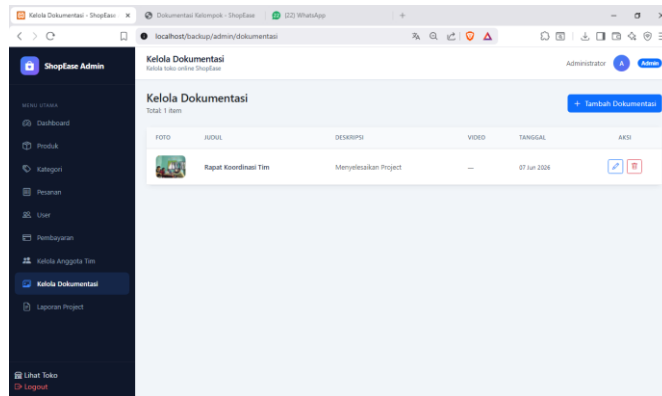
Halaman manajemen pengguna untuk melihat daftar akun terdaftar, mengelola peran dan hak akses, memblokir akun mencurigakan, serta melihat aktivitas pengguna.

18. Laporan



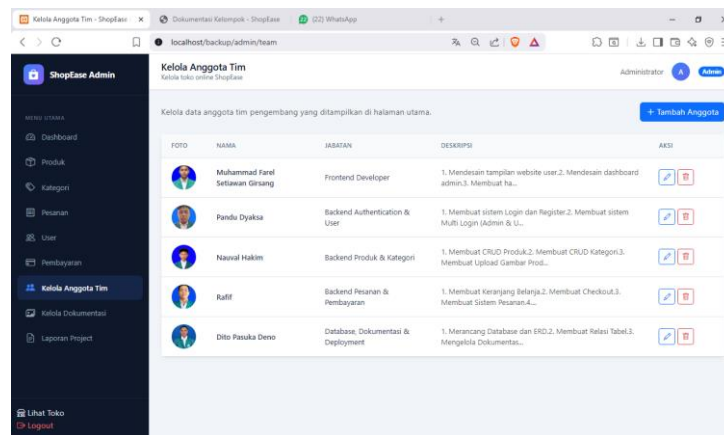
Halaman laporan berfungsi untuk menyediakan laporan dari tugas, tetapi halaman laporan ini bisa berfungsi sebagai laporan penjualan yang menyediakan analisis data transaksi, grafik pendapatan harian atau bulanan, ekspor laporan, dan ringkasan performa toko secara berkala.

19. Dokumentasi



Halaman pengelolaan dokumentasi tugas kelompok yang memungkinkan admin mengunggah, menyusun, dan menampilkan foto-foto kegiatan atau hasil kerja tim sebagai arsip visual aktivitas kelompok.

20. Kelola Anggota Tim



Halaman pengelolaan anggota tim internal yang memungkinkan admin mengatur daftar staf, peran masing-masing anggota, dan hak akses dalam sistem administrasi.

Untuk lebih singkat jelasnya bisa dilihat di table berikut

No	URL	Deskripsi	Akses
1	/	Halaman utama / beranda	Public
2	/login	Halaman login	Guest
3	/register	Halaman registrasi	Guest
4	/products	Katalog produk	Public
5	/products/{slug}	Detail produk	Public
6	/cart	Keranjang belanja	User
7	/checkout	Proses checkout	User
8	/orders	Riwayat pesanan	User
9	/orders/{id}	Detail pesanan	User
10	/profile	Profil pengguna	User
11	/dokumentasi	Halaman dokumentasi	Public
12	/admin/dashboard	Dashboard admin	Admin
13	/admin/products	Manajemen produk	Admin
14	/admin/categories	Manajemen kategori	Admin
15	/admin/orders	Manajemen pesanan	Admin
16	/admin/payments	Manajemen pembayaran	Admin

17	/admin/users	Manajemen pengguna	Admin
18	/admin/reports	Laporan penjualan	Admin
19	/admin/dokumentasi	Kelola dokumentasi	Admin
20	/admin/team	Kelola anggota tim	Admin

Laporan ini disusun sebagai dokumentasi resmi proyek Sistem Informasi E-Commerce Berbasis Web.