

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Landasan Teori**

Landasan Teori yakni bagian dari suatu karya ilmiah (seperti skripsi, tesis, atau disertasi) yang berisi kumpulan konsep, prinsip, dan teori yang relevan dan mendasari penelitian yang dilakukan. Landasan teori disusun berdasarkan teori-teori yang telah dikemukakan oleh para ahli sebelumnya dan berfungsi sebagai pijakan dalam menyusun kerangka berpikir serta menjawab permasalahan penelitian. Adapun sajian teori yang dapat mendasari penelitian ini.

##### **2.1.1 Pengertian Sistem**

Sistem adalah kumpulan elemen yang saling berinteraksi untuk mencapai tujuan tertentu. Menurut O'Brien dan Marakas (2020), sistem dapat didefinisikan sebagai suatu entitas yang terdiri dari komponen-komponen yang saling berhubungan dan berfungsi bersama untuk mencapai tujuan yang telah ditetapkan.

##### **2.1.2 Karakteristik Sistem**

Karakteristik sistem meliputi komponen, batasan, lingkungan, tujuan, dan interaksi. Setiap sistem memiliki batasan yang memisahkannya dari lingkungan luar, serta tujuan yang ingin dicapai melalui interaksi antar komponen (Laudon, 2021).

##### **2.1.3 Klasifikasi Sistem**

Sistem dapat diklasifikasikan Menjadi beberapa jenis, antara lain sistem fisik, sistem informasi, dan sistem sosial. Klasifikasi ini penting untuk memahami konteks dan aplikasi dari sistem yang sedang dianalisis (Stair & Reynolds, 2020).

#### **2.1.4 Informasi**

Informasi adalah data yang telah diolah dan disajikan dalam bentuk yang lebih bermakna. Menurut Turban et al. (2021), informasi memiliki nilai tambah yang dapat digunakan untuk pengambilan keputusan.

#### **2.1.5 Sistem Informasi**

Sistem informasi adalah sistem yang mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi. Menurut Turban et al. (2021), sistem informasi berperan penting dalam pengambilan keputusan dan pengelolaan organisasi.

#### **2.1.6 Komponen Sistem Informasi**

Komponen sistem informasi terdiri dari perangkat keras, perangkat lunak, data, prosedur, dan pengguna. Setiap komponen memiliki peran yang krusial dalam memastikan sistem informasi berfungsi dengan baik (Stair & Reynolds, 2020).

#### **2.1.7 Perancangan Sistem**

Perancangan sistem adalah proses merancang struktur dan komponen sistem untuk memenuhi kebutuhan pengguna. Menurut Sommerville (2020), perancangan sistem harus mempertimbangkan aspek fungsional dan non-fungsional.

#### **2.1.8 Perancangan *Website***

Perancangan *website* melibatkan proses pengembangan antarmuka pengguna, struktur navigasi, dan konten. Menurut W3C (2021), desain yang baik harus memperhatikan pengalaman pengguna (*UX*) dan aksesibilitas.

*Website* sendiri ialah kumpulan halaman informasi yang bisa diakses melalui internet menggunakan *browser* seperti *Google Chrome*, *Mozilla Firefox*, atau *Microsoft Edge*. Halaman-halaman ini saling terhubung satu sama lain dan biasanya berada di bawah satu nama domain, misalnya *www.google.com*.

Menurut Puspitasari & Suryanto (2022), *website* merupakan *platform* berbasis digital yang digunakan untuk menyampaikan informasi, promosi, serta interaksi antara pengguna dan penyedia layanan.

Secara umum, *website* terbagi menjadi dua jenis:

1. *Website* statis: Jenis *website* yang kontennya tidak berubah-ubah dan disajikan langsung kepada pengguna seperti apa adanya di *server*. Artinya, setiap halaman pada *website* statis dibuat secara manual menggunakan *HTML*, *CSS*, dan terkadang sedikit *JavaScript*, lalu disimpan di *server* tanpa proses pemrograman dinamis atau pengambilan data dari *database*. Cocok untuk menampilkan informasi umum seperti profil perusahaan atau dokumentasi sederhana.
2. *Website* dinamis: Jenis *website* yang kontennya dapat berubah secara otomatis berdasarkan interaksi pengguna, waktu, atau data dari *Database*. *Website* ini terhubung dengan sistem *back-end* (*server-side*) dan sering menggunakan bahasa pemrograman seperti *PHP*, *Python*, atau *JavaScript* (*Node.js*) serta *database* seperti *MySQL* atau *PostgreSQL*. Jenis ini cocok untuk *e-commerce*, portal berita, sistem informasi, katalog UMKM, dan sebagainya.

Dalam proses pengembangannya, *website* terdiri dari dua sisi utama yaitu sebagai berikut:

1. *Front-end* (Tampilan Pengguna)

Bagian ini merupakan yang dilihat langsung oleh pengguna, seperti tata letak, warna, menu navigasi, gambar, dan formulir. *Front-end* biasanya dibangun dengan *HTML*, *CSS*, dan *JavaScript*.

2. *Back-end* (Sisi Server)

Bagian ini mengatur logika sistem, *Database*, dan proses-proses di balik layar seperti *login*, pengolahan data, dan penyimpanan. *Back-end* dibangun menggunakan bahasa pemrograman seperti *PHP*, *Python*, *Node.js*, atau *framework* seperti *Laravel* dan *CodeIgniter*.

### 2.1.9 UMKM

UMKM adalah singkatan dari Usaha Mikro, Kecil, dan Menengah, yaitu jenis usaha yang dijalankan oleh perorangan atau badan usaha kecil. Menurut Fitriani & Handayani (2023), UMKM merupakan unit usaha yang secara struktural lebih fleksibel dan adaptif terhadap perubahan ekonomi, namun memiliki keterbatasan dalam akses teknologi dan permodalan.

Pengertian UMKM mengacu pada Undang-Undang Republik Indonesia Nomor 20 Tahun 2008, yang membagi UMKM Menjadi tiga kategori berdasarkan aset dan omzet tahunan, yaitu:

1. Usaha Mikro: Memiliki aset maksimal Rp. 50.000.000,- dan omzet tahunan maksimal Rp. 300.000.000,-
2. Usaha Kecil: Memiliki aset lebih dari Rp.50.000.000,- hingga Rp. 500.000.000,- dan omzet lebih dari Rp. 300.000.000,- hingga Rp. 2.500.000.000,-

3. Usaha Menengah: Memiliki aset lebih dari Rp. 500.000.000 hingga Rp. 10.000.000.000,- dan omzet antara Rp. 2.500.000.000,- hingga Rp. 50.000.000.000,-

#### **2.1.10 Konsep Dasar Penjualan**

Konsep penjualan meliputi seluruh proses yang mencakup aktivitas menjual produk atau layanan, meliputi identifikasi kebutuhan pelanggan, penawaran, negosiasi, hingga penutupan transaksi. Model ini mencakup elemen penting seperti strategi harga, promosi, distribusi, serta interaksi langsung antara tenaga penjualan dan konsumen. Konsep ini merupakan bagian dari *sales process engineering*, yakni desain sistematis dari keseluruhan proses penjualan untuk mencapai efisiensi dan efektivitas yang lebih tinggi. Dalam konteks UMKM, konsep penjualan terhubung erat dengan penerapan metode penjualan berbasis digital maupun konvensional.

Implementasi konsep penjualan yang kuat terutama yang menggabungkan digital marketing dan strategi pemasaran tradisional memberikan keuntungan signifikan bagi UMKM, baik dari sisi peningkatan visibilitas, loyalitas pelanggan, maupun omzet. Penelitian pada UMKM *Eito Japanese Curry* menemukan bahwa penerapan *Total Quality Management (TQM)* dalam mengelola produk, layanan, dan sistem pembayaran mampu meningkatkan efisiensi operasional dan kualitas layanan, yang mendukung peningkatan penjualan secara konsisten (Hutasoit et al. 2024). Namun, tantangan utama tetap pada keterbatasan SDM yang memahami strategi digital serta adaptasi terhadap teknologi terbaru. Studi-studi mendorong pentingnya pelatihan digital marketing dan dukungan kebijakan untuk membantu UMKM memanfaatkan konsep penjualan modern secara optimal

### 2.1.11 *E-commerce*

*E-commerce* adalah transaksi komersial yang dilakukan secara elektronik. Menurut Laudon dan Traver (2021), *e-commerce* memungkinkan UMKM untuk menjangkau pasar yang lebih luas dengan biaya yang lebih rendah.

*E-commerce (electronic commerce)* merupakan aktivitas jual-beli barang dan jasa menggunakan media elektronik seperti *website*, aplikasi, atau *platform* digital. Ini mencakup transaksi *B2C (Business-to-Consumer)*, *B2B (Business-to-Business)*, *C2C (Consumer-to-Consumer)*, hingga *social commerce* dan *mobile commerce*.

Secara teoritis, riset *e-commerce* sering menggunakan model seperti *Technology Acceptance Model (TAM)* dan *TOE Framework (Technology-Organization-Environment)* untuk menjelaskan penerimaan dan adopsi teknologi pada konsumen maupun pelaku usaha; misalnya faktor *perceived usefulness (PU)*, *perceived ease of use (PEU)*, *self-efficacy (SE)*, serta dukungan manajemen dan kesiapan internal organisasi

### 2.1.12 *Database*

*Database* adalah kumpulan data yang terorganisir dan dapat diakses dengan mudah. Menurut Elmasri dan Navathe (2020), sistem manajemen *Database (DBMS)* berfungsi untuk mengelola dan memanipulasi data.

*Database* yang digunakan dalam penelitian ini adalah *MySQL*, yang merupakan sistem manajemen basis data relasional. *MySQL* dikenal karena performanya yang baik dan kemudahan dalam pengelolaan data. Menurut Prasetyo (2020), *MySQL* sangat cocok untuk aplikasi *e-commerce* yang

memerlukan penyimpanan data yang terstruktur, seperti data produk, pengguna, dan transaksi.

Beberapa keunggulan *MySQL* antara lain:

1. Kecepatan dan Kinerja: *MySQL* mampu menangani sejumlah besar data dengan cepat, sehingga cocok untuk aplikasi yang memerlukan akses data yang cepat.
2. Kemudahan Penggunaan: *MySQL* memiliki antarmuka yang *user-friendly* dan dokumentasi yang lengkap, memudahkan pengembang dalam mengelola *Database* .
3. Keamanan: *MySQL* menyediakan berbagai fitur keamanan untuk melindungi data, termasuk kontrol akses dan enkripsi.

Dengan menggunakan *MySQL* sebagai *Database*, sistem yang dibangun diharapkan dapat menyimpan dan mengelola data dengan efisien, mendukung operasional *website* penjualan *online* yang dirancang.

### **2.1.13 XAMPP**

*XAMPP* adalah paket perangkat lunak yang menyediakan lingkungan pengembangan *web* yang lengkap, termasuk *server Apache*, *PHP*, dan *MySQL* . Menurut Sari dan Rahman (2023), penggunaan *XAMPP* dalam pengembangan aplikasi *web* memungkinkan pengembang untuk menguji dan mengembangkan aplikasi secara lokal sebelum diluncurkan ke server produksi. *MySQL* sebagai sistem manajemen basis data relasional memungkinkan penyimpanan dan pengelolaan data yang efisien, yang sangat penting dalam aplikasi *e-commerce* untuk mengelola informasi produk, transaksi, dan pengguna (Wahyu, 2021).



**Gambar 2.1 XAMPP**

#### **2.1.14 HTML**

*HTML (HyperText Markup Language)* merupakan bahasa markah yang digunakan untuk membangun struktur dan isi halaman *web*. Struktur dasar *HTML* terdiri atas elemen-elemen yang disusun membentuk *Document Object Model (DOM)* dengan berbagai atribut yang mengatur makna dan perilaku konten. Elemen semantik seperti `<header>`, `<main>`, `<article>`, `<section>`, `<footer>`, dan `<nav>` Menjadi fondasi penting dalam penyusunan dokumen karena tidak hanya membantu mesin pencari memahami struktur konten, tetapi juga meningkatkan aksesibilitas bagi teknologi bantu seperti pembaca *layer*.



**Gambar 2.2 HTML**

### 2.1.15 *PHP*

*PHP (Hypertext Preprocessor)* adalah bahasa pemrograman server-side yang digunakan untuk pengembangan *web*. Menurut Welling dan Thomson (2020), *PHP* memungkinkan pengembang untuk membuat aplikasi *web* dinamis.

Pada dasarnya, *PHP* berfungsi di server dan menangani proses data sebelum dikirimkan ke browser pengguna. Ini artinya, *PHP* termasuk dalam kategori *server-side scripting*, yang menjadikannya pilihan populer dalam pembuatan situs dinamis.

*PHP* pertama kali dikembangkan oleh Rasmus Lerdorf pada tahun 1994. Awalnya, ia hanya menggunakannya sebagai alat pribadi untuk melacak kunjungan ke situs pribadinya. Namun, dari situ, *PHP* berkembang menjadi sebuah bahasa pemrograman yang bisa digunakan oleh orang lain dan terus diperbarui hingga sekarang.



**Gambar 2.3 *PHP***

*PHP* sering digunakan untuk membuat *website* yang interaktif dan responsif. Dengan *PHP*, pengembang bisa membuat halaman *web* yang dapat berubah tergantung pada *input* dari pengguna, seperti sistem *login*, *form*, hingga *e-commerce*. Tak heran jika *PHP* tetap menjadi bahasa pemrograman favorit banyak pengembang *web*.

### 2.1.15 *JavaScript*

*JavaScript* adalah bahasa pemrograman yang digunakan untuk membuat halaman *web* interaktif. Menurut Flanagan (2020), *JavaScript* memungkinkan pengembang untuk membuat elemen-elemen interaktif pada halaman *web*, seperti validasi *form*, animasi, dan pengelolaan *event*.

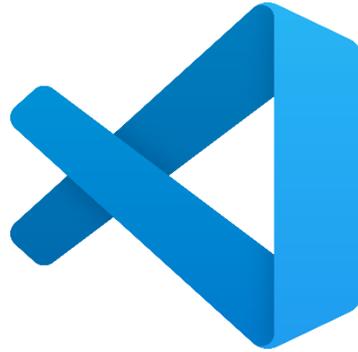


**Gambar 2.4 *JavaScript***

Kombinasi antara *PHP* dan *JavaScript* memungkinkan pengembangan aplikasi *web* yang responsif dan interaktif, di mana pengguna dapat berinteraksi dengan *website* secara *real-time* tanpa perlu memuat ulang halaman. Hal ini sangat penting dalam konteks *e-commerce*, di mana pengalaman pengguna yang baik dapat meningkatkan tingkat konversi penjualan.

### 2.1.16 *Visual Studio Code*

*Visual Studio Code* adalah editor kode sumber (*source code editor*) yang ringan namun sangat kuat, dikembangkan oleh *Microsoft*. Aplikasi ini dirancang untuk digunakan oleh *programmer* dan pengembang perangkat lunak untuk menulis, mengedit, men-*debug*, dan mengelola kode dalam berbagai bahasa pemrograman.



**Gambar 2.1** *Visual Studio Code*

*VS Code* menawarkan berbagai fitur yang mendukung pengembangan aplikasi *web*, seperti *debugging*, kontrol versi, dan integrasi dengan berbagai ekstensi. Penggunaan *VS Code* dalam pengembangan *website e-commerce* dapat meningkatkan produktivitas dan efisiensi pengembang. (Setiawan, 2022).

#### **2.1.17** *Framework CodeIgniter*

*CodeIgniter* adalah salah satu *framework PHP* yang bersifat *open-source* dan digunakan untuk membangun aplikasi *web* dinamis dengan konsep *Model-View-Controller (MVC)*. *Framework* ini dikenal karena performanya yang ringan, dokumentasi lengkap, dan kemudahan dalam pengembangan sistem berbasis *web*.

Menurut Nugroho dan Hidayat (2021) dalam Jurnal Teknologi dan Sistem Komputer, *CodeIgniter* merupakan *framework* yang cocok untuk pengembangan aplikasi skala kecil hingga menengah karena struktur folder yang sederhana dan mudah dipahami oleh pemula.



**Gambar 2. 2 CodeIgniter**

## **2.2 UML (Unified Modelling Language)**

*Unified Modeling Language (UML)* adalah bahasa pemodelan standar yang dikembangkan untuk membantu perancangan dan pendokumentasian sistem perangkat lunak yang kompleks. *UML* berfungsi sebagai alat komunikasi visual untuk mempermudah pemahaman struktur dan perilaku sistem, baik bagi pengembang perangkat lunak maupun pemangku kepentingan *non-teknis*. *UML* mencakup berbagai jenis diagram, seperti *structural diagrams* (*Class diagram*, *component diagram*, dan *deployment diagram*) dan *behavioral diagrams* (*use case diagram*, *Activity diagram*, dan *Sequence diagram*). Setiap diagram memberikan perspektif yang berbeda terhadap sistem, mulai dari representasi hubungan antar entitas hingga aliran interaksi antar komponen (Koç et al., 2021).

### **2.2.1 Diagram Konteks**

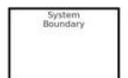
Diagram konteks merupakan representasi tingkat tertinggi dari sebuah sistem informasi yang menggambarkan hubungan antara sistem dengan entitas eksternal yang terlibat, serta menunjukkan aliran data yang menghubungkan

keduanya. Diagram ini digunakan untuk memberikan gambaran umum mengenai batasan (*boundary*) sistem, pihak-pihak yang berinteraksi, dan informasi yang dipertukarkan tanpa menampilkan proses internal secara rinci. Menurut Kendall dan Kendall (2014), diagram konteks memuat satu proses utama yang mewakili keseluruhan sistem, dilengkapi dengan panah aliran data yang menghubungkan entitas eksternal dengan sistem.

Dalam penelitian ini, diagram konteks digunakan untuk memvisualisasikan interaksi antara Admin, Pemilik UMKM, dan Pelanggan dengan Sistem *Website* Penjualan Produk UMKM. Admin bertugas mengelola data pengguna dan memberikan konfirmasi pendaftaran. Pemilik UMKM mengelola data produk serta pesanan pelanggan. Pelanggan melakukan pendaftaran, login, memilih produk, melakukan pemesanan, serta menerima *invoice*.

### **2.2.2 Use Case Diagram**

*Use Case Diagram* merupakan salah satu jenis diagram dalam *Unified Modeling Language (UML)* yang menggambarkan interaksi antara pengguna (aktor) dengan sistem perangkat lunak. Untuk *website* UMKM, diagram ini memetakan aktor utama seperti *Admin UMKM*, *Customer* (pelanggan), dan Pengelola Konten, serta layanan sistem seperti registrasi akun, manajemen produk, pemesanan, pembayaran, dan pelaporan penjualan.

No	Gambar	Nama	Keterangan
1		Actor	Peran eksternal (User/Admin/Pelanggan) yang berinteraksi dengan sistem.
2		Use Case	Fungsi/layanan yang diberikan sistem kepada aktor.
3		Association	Hubungan interaksi antara Actor dan Use Case.
4		<<include>>	Relasi bahwa satu use case selalu menyertakan use case lain.
5		<<extend>>	Relasi bahwa use case dapat diperluas secara opsional oleh use case lain.
6		System Boundary	Batasan sistem yang membungkus semua use case yang ditawarkan sistem.

**Gambar 2. 3 Simbol *Use Case Diagram***

### 2.2.3 *Activity Diagram*

*Activity Diagram* adalah diagram *UML* yang menggambarkan alur kerja sistem secara visual dengan menunjukkan tindakan, cabang keputusan, iterasi, dan kemungkinan jalur paralel. Dalam perancangan *website* UMKM, *Activity Diagram* memetakan urutan aktivitas pengguna seperti *login*, pilih produk, pesan, verifikasi stok, hingga konfirmasi pesanan. Ini membantu pengembang memahami logika alur operasional dan kasus-kasus kondisi (misalnya apabila stok tidak mencukupi) sebelum implementasi. Diagram ini menjadi panduan pemodelan proses yang transparan untuk tim pengembang dan pemilik UMKM.

No	Gambar	Nama	Keterangan
1		Initial Node	Titik awal alur aktivitas.
2		Action	Langkah/kegiatan yang dilakukan dalam proses.
3		Decision Node	Percabangan kondisi (mis. ya/tidak).
4		Merge Node	Penggabungan beberapa alur menjadi satu alur.
5		Fork Node	Memecah alur menjadi paralel.
6		Join Node	Menggabungkan kembali alur paralel.
7		Final Node	Titik akhir proses.

**Gambar 2. 4 Activity Diagram**

#### 2.2.4 Sequence Diagram

*Sequence Diagram* adalah jenis diagram *UML* yang memvisualisasikan interaksi antar *objek* atau *aktor* dengan sistem, disusun berdasarkan urutan waktu. Dalam konteks perancangan *website* UMKM, diagram ini menggambarkan alur pesan antara aktor seperti *admin*, pemilik UMKM atau *customer* dan komponen sistem misalnya sistem menerima data *login*, memvalidasi otentikasi, menampilkan halaman utama, menambah produk, atau memproses pesan pembelian.

No	Gambar	Nama	Keterangan
1		Actor	Entitas eksternal yang memulai interaksi.
2		Object / Lifeline	Objek dalam sistem dan garis hidup interaksinya.
3		Message	Pesan (sinkron/asinkron) yang dikirim antar objek.
4		Activation Bar	Periode saat objek sedang mengeksekusi operasi.
5		Return Message	Pesan balasan dari objek setelah memproses.
6		Destroy	Mengakhiri keberadaan objek (tanda X).

**Gambar 2. 5 Sequence Diagram**

### 2.2.5 Class Diagram

*Class Diagram* menunjukkan struktur kelas dalam sistem dan hubungan antar kelas. Diagram ini penting untuk merancang basis data dan logika aplikasi.

No	Gambar	Nama	Keterangan
1		Class	Representasi struktur dengan atribut dan operasi.
2		Attribute	Data/variabel milik kelas.
3		Method / Operation	Fungsi/perilaku milik kelas.
4		Association	Hubungan struktural antar kelas.
5		Inheritance / Generalization	Relasi pewarisan antar kelas.
6		Aggregation	Relasi bagian-keseluruhan (lemah) ditandai diamond kosong.
7		Composition	Relasi bagian-keseluruhan (kuat) ditandai diamond terisi.

**Gambar 2. 6 Class Diagram**

### 2.3 Penelitian Terdahulu

No	Peneliti & Tahun	Judul Penelitian	Metode Penelitian	Hasil / Temuan	Relevansi dengan Penelitian Ini
1	Wahyu Pratama, M., et al. (2024)	Efektivitas <i>Digital Marketing</i> dalam Meningkatkan <i>Turnover UMKM</i>	Kuantitatif	Digital marketing melalui media sosial, optimasi konten, dan <i>e-commerce</i> meningkatkan penjualan UMKM hingga 50% <i>turnover</i>	Mendukung penggunaan strategi pemasaran digital untuk meningkatkan omzet UMKM melalui <i>website</i> penjualan
2	Strategi, A., et al. (2024)	Optimalisasi <i>E-commerce</i> dalam Promosi Digital UMKM <i>Tas BlewUp</i>	Deskriptif	Promosi digital yang terstruktur melalui <i>e-commerce</i> mampu meningkatkan kinerja penjualan meskipun ada keterbatasan SDM dan modal	Relevan dalam konteks pengembangan fitur promosi digital pada <i>website</i> UMKM
3	Salsabila, A., et al. (2025)	Pemanfaatan <i>Platform Digital</i> dalam Peningkatan Penjualan Kedai Wigung	Studi Kasus	<i>Instagram, ShopeeFood, GoFood, dan WhatsApp</i> meningkatkan penjualan 20–39% selama pandemi dan pasca-pandemi	Memberi referensi integrasi media digital dengan <i>e-commerce</i> UMKM
4	Hutasoit, R., et al. (2024)	Penerapan <i>TQM</i> pada UMKM <i>Eito Japanese Curry</i>	Studi Kasus	<i>TQM</i> meningkatkan efisiensi operasional dan kualitas layanan sehingga penjualan meningkat konsisten	Relevan untuk peningkatan kualitas layanan pada <i>website</i> penjualan <i>online</i>
5	Koç, T.,	Relevansi <i>UML</i>	Kajian	<i>UML</i> tetap	Mendukung

	et al. (2021)	dalam Perancangan Sistem Perangkat Lunak Modern	Literatur	dominan dalam desain sistem karena kemampuannya memodelkan arsitektur perangkat lunak secara jelas	penggunaan UML dalam perancangan sistem <i>e-commerce</i> UMKM
6	Chen, Y., et al. (2025)	Integrasi LLM dalam Pembuatan Diagram UML Otomatis	Eksperimen	<i>AI</i> dapat membantu menghasilkan model <i>UML</i> dari deskripsi teks, mengurangi kesalahan desain dan mempercepat pembuatan diagram	Relevan dengan pengembangan diagram sistem berbasis <i>UML</i> di penelitian ini
7	Gomez, R., et al. (2023)	Kolaborasi <i>UML</i> dalam Lingkungan <i>Virtual Reality</i>	Eksperimen Komparatif	<i>VR</i> meningkatkan keterlibatan tim pengembang walau efisiensi awal lebih rendah dibanding metode konvensional	Memberi inspirasi alternatif kolaborasi dalam desain sistem
8	Santos, L., & Oliveira, M. (2023)	Pengaruh Dokumentasi UML terhadap Pemeliharaan Sistem	Kuantitatif	Dokumentasi UML yang diperbarui dapat mempercepat pemeliharaan sistem hingga 30%	Relevan dalam pemeliharaan sistem <i>e-commerce</i> jangka panjang
9	Rahman, A., et al. (2024)	Deteksi Masalah Desain dengan <i>UML</i> pada Tahap Awal	Kuantitatif	<i>UML</i> mampu mengurangi biaya perbaikan pasca-rilis hingga 40%	Mendukung perancangan awal sistem untuk meminimalkan revisi besar
10	Kumar, P., et al. (2025)	Integrasi <i>MDD</i> dan <i>AI</i> untuk Pembaruan Otomatis	Eksperimen	<i>AI</i> dapat memperbarui diagram <i>UML</i> secara otomatis pada pengembangan	Relevan untuk mempercepat pembaruan model sistem

		Diagram <i>UML</i>		<i>Agile</i>	
11	Indrayani, L., et al. (2020)	Perancangan Sistem Informasi Pemasaran UMKM Berbasis <i>Web</i> Menggunakan <i>UML</i>	Research and Development	Menghasilkan sistem pemasaran UMKM berbasis <i>web</i> dengan pemodelan <i>UML</i> lengkap termasuk <i>Use Case Diagram</i>	Memberikan acuan pemodelan fitur <i>e-commerce</i> UMKM
12	Prasetyo, R., & Putra, A. (2021)	Sistem Pembelian Bahan Baku UMKM Menggunakan <i>UML</i>	<i>Waterfall</i>	Pemodelan dengan <i>Use Case</i> , <i>Sequence</i> , dan <i>Class Diagram</i> memperjelas kebutuhan sistem	Relevan dalam penggunaan <i>UML</i> untuk analisis kebutuhan
13	Seki, K., et al. (2020)	Otomatisasi Deteksi <i>Bad Smells</i> pada <i>Use Case</i>	Eksperimen	Deteksi otomatis dapat meningkatkan kualitas deskripsi <i>Use Case</i> dan mengurangi kesalahan implementasi	Relevan untuk meningkatkan kualitas spesifikasi kebutuhan sistem
14	Susianto, D., et al. (2022)	Perancangan Sistem Informasi Pemasaran Produk UMKM dengan <i>Activity Diagram</i>	<i>Research and Development</i>	<i>Activity Diagram</i> memetakan proses pemesanan dan manajemen produk secara detail	Mendukung perancangan alur proses pada sistem <i>e-commerce</i> UMKM
15	Tim Desa Kalijati (2024)	Perancangan Sistem Penjualan Produk Desa Berbasis <i>Web</i>	Studi Kasus	<i>Activity Diagram</i> digunakan untuk memodelkan aktivitas pembelian dan riwayat pesanan	Relevan dalam pemodelan alur transaksi penjualan produk lokal

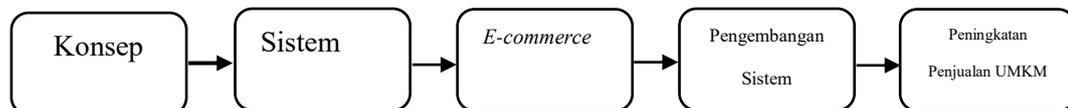
Tabel 2. 1 Penelitian terdahulu

## 2.4 Kerangka Berpikir

Kerangka berpikir dalam penelitian ini berfokus pada pengembangan *website e-commerce* untuk UMKM menggunakan *framework CodeIgniter*. Dengan memahami konsep sistem, sistem informasi, dan *e-commerce*, penelitian ini bertujuan untuk merancang *website* yang efektif dan efisien. Penelitian terdahulu menunjukkan bahwa penggunaan *website e-commerce* dapat

meningkatkan penjualan dan daya saing UMKM. Oleh karena itu, penelitian ini akan mengintegrasikan teori-teori yang relevan dan hasil penelitian sebelumnya untuk membangun dasar pemikiran yang kuat dalam merancang *website* penjualan *online*.

Bagan kerangka berpikir dapat disusun sebagai berikut:



**Gambar 2. 7 Kerangka Berpikir**