

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang dilakukan oleh Muhamad Dimas Ramdani Fitriani Syaputra et al., (2025) dengan judul " Aplikasi pemesanan kue berbasis web sebagai solusi digital untuk UMKM Donutschaca" Penelitian ini membahas sistem informasi pengelolaan order marketplace berbasis *website*. Sistem yang dikembangkan membantu proses pencatatan data transaksi dan monitoring pesanan secara lebih terstruktur, Fitur utama sistem meliputi registrasi pengguna, katalog produk digital, pemesanan *pre-order*. Hasil pengujian menggunakan *black box* testing menunjukkan bahwa seluruh fitur sistem berjalan dengan baik sesuai kebutuhan pengguna, sehingga aplikasi mampu meningkatkan efisiensi operasional, mempercepat proses transaksi, mengurangi kesalahan pencatatan data, serta membantu UMKM Donutschaca dalam melakukan digitalisasi layanan pemesanan secara lebih modern dan terintegrasi[3].

Penelitian yang dilakukan oleh Ilhan Abdhu Syukur dan Nisa Dienwati Nuris (2023) berjudul "Rancang Bangun Aplikasi Penjualan Kue Berbasis *Web* pada *Home Industry* Maushi Cake Cirebon" bertujuan untuk membangun aplikasi penjualan kue berbasis website guna membantu proses pengelolaan pesanan pelanggan, pencatatan stok bahan baku, serta penjualan produk yang sebelumnya masih dilakukan secara manual dan terbatas secara

offline. Sistem ini dikembangkan menggunakan metode *Waterfall*. Hasil pengujian menggunakan metode *black box testing* menunjukkan bahwa seluruh fitur aplikasi berjalan sesuai kebutuhan pengguna. Dengan adanya aplikasi ini, Home Industry Maushi Cake Cirebon dapat meningkatkan efisiensi pengelolaan penjualan, mempermudah pelanggan dalam melakukan pemesanan secara *online*, memperluas jangkauan pemasaran, serta meningkatkan efektivitas pengelolaan data dan laporan penjualan secara digital[4].

Penelitian yang dilakukan oleh Dwicky Oktavian et al., (2021) berjudul “Pengembangan Sistem Aplikasi Tokokue Berbasis *Web* dengan *PHP*” bertujuan untuk merancang dan mengembangkan sistem informasi penjualan kue berbasis website guna membantu proses penjualan yang sebelumnya masih dilakukan secara konvensional, di mana pelanggan harus datang langsung ke toko untuk melihat dan membeli produk, Sistem ini dikembangkan menggunakan metode *Waterfall*. Aplikasi dibangun menggunakan bahasa pemrograman *PHP* dan *database MySQL* serta didukung pemodelan sistem menggunakan *Use Case Diagram* dan *Activity Diagram*. Fitur utama sistem meliputi registrasi pengguna, login pelanggan, katalog produk, keranjang belanja, pemesanan *online*, pembayaran melalui transfer, pengelolaan data produk oleh admin, laporan penjualan, serta pengelolaan data pelanggan. Hasil penelitian menunjukkan bahwa penerapan aplikasi penjualan berbasis *web* mampu meningkatkan penjualan Tokokue,

serta meningkatkan efisiensi proses transaksi dan pengelolaan data penjualan secara lebih terstruktur[5].

2.2 Landasan Teori

Adapun landasan teori pada “Sistem Informasi Pemesanan Toko Kue berbasis *website* Pada Aqlaya Cake dengan fitur katalog produk, dan *pre-Order*” berpacuan dengan berbagai penelitian terdahulu di antaranya :

2.2.1 Waterfall

Metode penelitian *Waterfall* adalah konsep pengembangan sistem yang menekankan langkah-langkah kerja secara sistematis dan berurutan, dimulai dari identifikasi kebutuhan hingga proses pemeliharaan sistem. Setiap tahap harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya[6].

2.2.2 Sistem Informasi

Sistem informasi merupakan kumpulan komponen dalam sebuah organisasi atau lembaga yang berhubungan dengan proses penciptaan dan pengaliran informasi[7].

2.2.3 Laragon

Laragon adalah perangkat lunak *open source* yang digunakan sebagai *server virtual* atau localhost untuk menjalankan dan mengembangkan aplikasi berbasis *web*. Laragon menyediakan lingkungan pengembangan seperti *web server* dan database sehingga memudahkan pengguna dalam membuat, mengelola, serta menguji aplikasi web sebelum dipublikasikan[8].



Gambar 2. 1 Logo Laragon

2.2.4 Website

Website adalah kumpulan halaman yang berisi informasi, data, atau layanan yang saling terhubung dan dapat diakses melalui internet menggunakan browser. *Website* dapat digunakan untuk menyampaikan informasi, menyediakan layanan, melakukan transaksi, maupun mendukung berbagai kebutuhan pengguna secara *online*[9].

2.2.5 Laravel

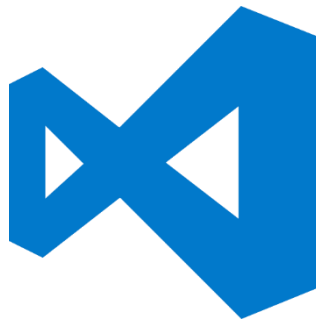
Laravel merupakan salah satu *framework PHP* yang paling diminati dalam pengembangan aplikasi *back end* dengan menggunakan bahasa pemrograman *PHP*. Laravel memiliki banyak fitur yang sangat mendukung dalam mempercepat pembuatan aplikasi atau sistem informasi berbasis *web*, dimana laravel dapat digunakan baik sebagai *back end* dan *front end* atau hanya digunakan sebagai *back end* saja[10].



Gambar 2. 2 Logo Laravel

2.2.6 Visual Studio Code

Visual Studio Code adalah editor teks ringan yang dikembangkan oleh *Microsoft* untuk berbagai sistem operasi. Editor ini mendukung berbagai bahasa pemrograman, termasuk *JavaScript*, *TypeScript*, *Node.js*, *C++*, *C#*, *Python*, *Go*, *Java*, dan *PHP* sistem[11].



Gambar 2. 3 Logo Visual Studio Code

2.2.7 Hypertext Preprocessor (PHP)

HTML adalah bahasa yang digunakan untuk membuat isi dokumen halaman *web*. Tetapi *HTML* bukanlah bahasa pemrograman, melainkan bahasa markup yang adalah sistem untuk mengidentifikasi dan menjelaskan berbagai komponen dokumen seperti *headings*,

paragraf, dan daftar. Markup menunjukkan struktur dasar dokumen. Di dalam *HTML* terdapat *Cascading Style Sheets (CSS)* yang mendeskripsikan bagaimana tampilan atau presentasi dari konten, seperti fonts, warna, gambar latar belakang, spasi baris, dan susunan halaman[12].



Gambar 2. 4 Logo PHP

2.2.8 Tailwind

Tailwind CSS adalah *framework CSS* yang menggunakan konsep *utility-first*, yaitu *styling* langsung pada elemen *HTML* dengan *class* siap pakai sehingga mempermudah dan mempercepat pembuatan tampilan *website* tanpa harus banyak menulis CSS manual[13].



Gambar 2. 5 Logo Tailwind css

2.2.9 Cascading Style Sheet (CSS)

CSS (Cascading Style Sheet) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu *website*, baik tata letaknya, jenis huruf, warna, dan semua yang berhubungan dengan tampilan.

Pada umumnya *CSS* digunakan untuk menformat halaman *web* yang ditulis dengan *HTML* atau *XHTML*[14].



Gambar 2. 6 Logo CSS

2.2.10 Database

Database adalah kumpulan data yang terorganisir dan tersimpan secara sistematis sehingga dapat diakses, dikelola, dimanipulasi, dan diperbaharui dengan mudah. Database digunakan untuk menyimpan informasi secara terstruktur agar dapat diolah oleh pengguna maupun sistem secara efisien. Dalam konteks sistem informasi, database berfungsi sebagai penyimpan utama data transaksi, data pelanggan, produk, maupun aktivitas operasional lainnya[15].

2.2.11 UML (Unified Modeling Language)

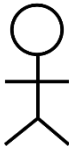
Unified Modeling Language atau yang biasa dikenal dengan singkatan UML, adalah bahasa standar yang digunakan untuk menggambarkan, menjelaskan, dan membangun perangkat lunak. Ini merupakan metode dalam pengembangan sistem berbasis objek yang juga berfungsi sebagai alat bantu dalam proses pengembangan sistem. Beberapa alat bantu yang digunakan dalam perancangan berbasis

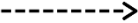
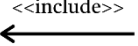
objek dengan menggunakan UML meliputi *diagram use case*, *activity diagram*, *sequence diagram*, dan *class diagram*[16]. Berikut pengertian dan bentuk simbol-simbol dari diagram :

1. Use Case Diagram

Usecase diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. Misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem[17].

Tabel 2. 1 Use Case Diagram

No.	Gambar	Simbol	Keterangan
1		Aktor	Mewakili peran seorang <i>system</i> yang lain, atau alat ketika berkomunikasi dengan <i>Use Case</i>
2		<i>Use Case</i>	Mwakiili Abstraksi dan interaksi antara sebuah <i>System</i> dan <i>Actor</i>
3		<i>Association</i>	Abstraksi penghubung antar <i>Actor</i> dengan <i>Use Case</i>

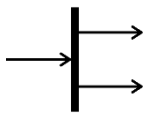
No.	Gambar	Simbol	Keterangan
4.		<i>Generalisasi</i>	Menunjukkan spesialisasi actor untuk dapat berpartisipasi dengan <i>Use Case</i>
5.		<i>Include</i>	Menunjukkan bahwa suatu <i>Use Case</i> seluruhnya merupakan fungsionalitas dari <i>Use Case</i> lainnya
6.		<i>Extend</i>	Menunjukkan bahwa suatu <i>Use Case</i> merupakan tambahan fungsional dari <i>Use Case</i> lainnya jika suatu kondisi terpenuhi.

2. Activity Diagram

Activity Diagram adalah salah satu jenis diagram dalam *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan alur aktivitas atau proses bisnis dalam sebuah sistem secara berurutan. Diagram ini menunjukkan bagaimana aktivitas dimulai, pilihan keputusan yang mungkin terjadi, hingga aktivitas berakhir. *Activity Diagram* berfokus pada *flow of control* atau alur kerja (*workflow*) dari suatu proses, bukan pada struktur data atau entitas[18].

Tabel 2. 2 Activity Diagram

No.	Gambar	Simbol	Keterangan
1.		<i>Initial node</i>	Merupakan awal aktivitas yang akan dibuat
2.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi
3.		<i>Decision</i>	Menggambarkan suatu keputusan atau tindakan yang harus diambil pada kondisi tertentu.

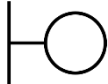
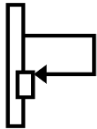
No.	Gambar	Simbol	Keterangan
4.		<i>Fork node</i>	Satu aliran pada tahap yang berubah menjadi beberapa aliran.
5.		<i>Fork Join node</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
6.		<i>Activity Final node</i>	Merupakan akhir aktivitas yang telah dibuat.

3. Sequence Diagram

Sequence Diagram adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Selain itu *sequence diagram* juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya. Objek-objek yang berhubungan dengan berjalannya proses operasi biasanya diurutkan dari kiri ke kanan. diagram ini terdiri dari dua dimensi, yaitu dimensi vertikal yang menunjukkan waktu dan dimensi horizontal yang menunjukkan objek-objek. Tiap-tiap objek, termasuk actor, memiliki waktu aktif yang digambarkan dengan kolom vertikal yang disebut dengan *lifeline*. Sementara itu, pesan

atau perintah digambarkan sebagai garis panah dari satu *lifeline* ke *lifeline* yang lain[19]. Berikut simbol-simbol dari *sequence diagram*:

Tabel 2. 3 Sequence Diagram

No.	Gambar	Simbol	Keterangan
1.		<i>Entity Class</i>	Menggambarkan awal dari sebuah sistem, menjadi dasar penyusunan basisdata.
2.		<i>Boundary Class</i>	Menggambarkan komunikasi antara sistem seperti <i>Form</i>
3.		<i>Control Class</i>	Penghubung antara <i>boundary</i> dengan <i>table</i> .
4.		<i>Recursive</i>	Mengirimkan pesan yang akan dikirim ke dirinya sendiri
5.		<i>Activation</i>	Menghubungkan proses durasi aktivasi sebuah proses
6.		<i>Life Line</i>	Menghubungkan tiap-tiap objek yang mana terdapat <i>activation</i>
7.		<i>Message</i>	Sebagai pengirim pesan ke tiap-tiap <i>Class</i>

4. Class Diagram

Class Diagram adalah salah satu jenis diagram struktur pada *UML* yang menggambarkan dengan jelas struktur serta deskripsi *class*, atribut, metode, dan hubungan dari setiap objek. Desain model dari diagram kelas ini sendiri dibagi menjadi dua bagian. Bagian pertama merupakan penjabaran dari *database*. Bagian kedua merupakan bagian dari modul MVC, yang memiliki *class interface*, *class control*, dan *class entity*[20]. Berikut simbol-simbol dari *class diagram* :

Tabel 2. 4 Class Diagram

No.	Gambar	Simbol	Keterangan
1.		<i>Class</i>	Menggambarkan kumpulan objek-objek dengan atribut dan operasi yang sama
2.		<i>Nary Association</i>	Sebagai upaya menghindari asosiasi dengan lebih dari 2 objek
3.		<i>Collaboration</i>	Urutan aksi-aksi yang ditampilkan oleh sistem yang menghasilkan suatu hasil yang tertukar bagi suatu <i>actor</i>
4.		<i>Generalization</i>	Menggambarkan dimana hubungan objek <i>descendent</i> berperilaku dan struktur data objek yang di atasnya objek induk <i>ancestor</i>

No.	Gambar	Simbol	Keterangan
5.	←-----	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6.	-----→	<i>Dependenc y</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang tidak mandiri.