

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang dilakukan oleh T. Bayu Kurniawan dan Syarifuddin (2020) bertujuan untuk merancang sistem aplikasi pemesanan makanan dan minuman pada Cafeteria No Caffe, menggunakan pemodelan UML (*Unified Modeling Language*) untuk menggambarkan diagram *use case*, *class*, *sequence*, dan *activity* yang mendukung sistem yang dirancang. Sistem dibangun dengan bahasa pemrograman *PHP* dan *MySQL*. Sistem tersebut dapat membantu mengatasi kendala seperti pencatatan pesanan manual, pengolahan data yang kurang teratur, serta meningkatkan keamanan data. Pengujian terhadap sistem menunjukkan efisiensi dalam proses pemesanan dan pengolahan data. Sistem ini mempermudah konsumen ketika melakukan pemesanan secara *online* tanpa batas ruang dan waktu, serta membantu perusahaan mengelola data dengan lebih akurat dan terorganisir[5].

Penelitian yang dilakukan oleh Wahyu Pudyawardana (2023) bertujuan untuk merancang sistem informasi pemesanan makanan dan minuman berbasis web pada Restoran Lamongan Cahaya. Penelitian ini menggunakan *Data Flow Diagram* (DFD) dan *Entity Relationship Diagram* (ERD) untuk merancang alur data serta relasi basis data. Sistem ini dibangun menggunakan bahasa pemrograman *PHP* dan *MySQL* sebagai basis data.

Aplikasi ini dirancang untuk mengatasi permasalahan seperti tumpang tindih pemesanan, pengantaran tidak berurutan, dan kesulitan mendapatkan menyediakan fitur untuk mencatat pesanan, mengelola data makanan dan minuman, serta mempermudah pelanggan dalam melakukan pemesanan secara *online*[6].

Penelitian yang dilakukan oleh Tuti Handayani, Ipung Gunawan, dan Rohmat Taufiq (2020) bertujuan untuk merancang sistem informasi pemesanan menu makanan berbasis web untuk Restoran Bukit Randu Bandara. Sistem ini dirancang menggunakan *Unified Modeling Language* (UML) dan metode *waterfall*. Sistem ini dibuat untuk mengatasi masalah seperti panjangnya antrian, kesibukan pelayan, dan kurangnya informasi yang tersedia secara *real-time*. Sistem ini memiliki fitur pemesanan makanan secara *online*, pengelolaan data produk dan transaksi oleh admin, serta pembuatan laporan secara *real-time*. Penggunaan sistem ini diharapkan dapat memperluas area pemasaran restoran, memberikan informasi terkini kepada pelanggan, serta meningkatkan kenyamanan pelanggan dalam memesan makanan tanpa harus mengantri. Implementasi sistem berbasis web ini berhasil meningkatkan efisiensi layanan dan membantu pengusaha restoran mengelola data dengan lebih baik[7].

2.2 Landasan Teori

2.2.1 HTML (Hypertext Markup Language)

HTML (HyperText Markup Language) merupakan bahasa markup standar yang digunakan untuk menyusun struktur dasar halaman web, seperti teks, gambar, tautan, serta elemen-elemen *multimedia*. *HTML* berperan sebagai kerangka utama yang memungkinkan *browser* menampilkan isi konten secara terstruktur dan terorganisir. *HTML* menjadi tulang punggung dari desain web modern, karena memungkinkan penyusunan konten yang dinamis dan interaktif dengan cara yang lebih efisien[8].

2.2.2 CSS (Cascading Style Sheets)

CSS digunakan untuk mengatur tampilan visual aplikasi agar menarik dan mudah digunakan. Dengan *CSS*, elemen-elemen seperti warna, tata letak, *margin*, *padding*, serta responsivitas halaman dapat dikendalikan secara efisien, agar antarmuka aplikasi menyesuaikan diri dengan baik pada berbagai perangkat seperti komputer *desktop* dan perangkat *mobile*. Penerapan *CSS* yang tepat membantu menciptakan pengalaman pengguna yang nyaman dan tampilan yang konsisten di seluruh halaman aplikasi[9].

2.2.3 JavaScript

JavaScript merupakan salah satu Bahasa pemrograman yang berperan dalam menciptakan web. Dengan *JavaScript*, elemen-elemen halaman dapat merespons aksi pengguna seperti klik tombol,

pengisian *form*, serta manipulasi konten secara *real-time* tanpa harus memuat ulang halaman. Hal ini sangat penting untuk meningkatkan pengalaman pengguna dalam berinteraksi dengan *website*. *JavaScript* memungkinkan pengembangan aplikasi web yang responsif dan fungsional melalui penggunaan *DOM* (Document Object Model) dan *event handling*[10].

2.2.4 Bootstrap

Bootstrap merupakan *framework front-end* yang sangat populer digunakan untuk membangun *website* dengan desain responsif dan tampilan yang konsisten di berbagai perangkat. *Framework* ini menyediakan kumpulan komponen siap pakai, seperti *grid system*, tombol, navigasi, dan *form*, yang memudahkan pengembang dalam membuat tampilan web secara cepat dan efisien. penggunaan *Bootstrap* memungkinkan penyesuaian *layout* secara otomatis mengikuti ukuran layar perangkat [11].

2.2.5 Hypertext Preprocessor (PHP)

PHP (Hypertext Preprocessor) merupakan salah satu bahasa pemrograman yang paling sering digunakan dalam pembangunan *website*[12]. *PHP* disebut sebagai pemrograman *Server Side Programming*, hal ini dikarenakan seluruh proses nya dijalankan di server. *PHP* pertama kali dikembangkan oleh Rasmus Lerdorf pada tahun 1994 sebagai serangkaian skrip *CGI* (Common Gateway Interface) sederhana untuk melacak kunjungan ke situs pribadinya.

Awalnya disebut *Personal Home Page Tools*, *PHP* kemudian berkembang menjadi bahasa pemrograman yang lebih kompleks dan dirilis sebagai *open-source* pada tahun 1995. *PHP* dirancang untuk pengembangan web dan dapat disisipkan langsung ke dalam kode *HTML*, sehingga memudahkan pengembang untuk membuat situs web dinamis dengan cepat[13].

2.2.6 Laravel

Framework atau kerangka kerja menjadi wadah bagi pengembangan sebuah *website*. Dengan menggunakan *framework*, proses pembuatan *website* menjadi lebih efisien karena waktu pengerjaan lebih singkat, dan perbaikan dapat dilakukan dengan lebih mudah[14]. *Laravel* juga merupakan salah satu *framework* pemrograman berbasis *open source* yang populer di kalangan *developer* di seluruh dunia. *Framework* ini dirancang untuk membantu pengembang dalam penggunaan *PHP* untuk pengembangan *website*. *Laravel* pertama kali dirilis oleh Taylor Otwell pada tahun 2011 sebagai alternatif untuk *framework CodeIgniter*, dengan fokus pada fitur yang lebih modern seperti autentikasi dan *routing*[15].

2.2.7 Midtrans

Midtrans merupakan solusi *payment gateway* yang berfungsi sebagai sistem verifikasi pembayaran pada berbagai platform digital. Midtrans memudahkan proses transaksi pembayaran secara

online dengan menyediakan layanan yang aman, efisien, dan terintegrasi dengan berbagai metode pembayaran. Sistem Midtrans tidak hanya memproses pembayaran, tetapi juga memberikan notifikasi status pembayaran secara *real-time* sehingga mempercepat proses verifikasi dan meminimalisir kesalahan manual. Dengan begitu, integrasi Midtrans pada *website* meningkatkan kepercayaan pengguna dan kenyamanan dalam melakukan transaksi[16].

2.2.8 MySQL

MySQL adalah sistem manajemen basis data (DBMS) *open source* yang mendukung multi-user dan multi-threaded. *MySQL* menggunakan bahasa *SQL* untuk mengelola data, termasuk membuat, membaca, memperbarui, dan menghapus data. Sebagai bagian dari *Relational Database Management System* (RDBMS), *MySQL* menyimpan data dalam tabel-tabel yang saling berhubungan, pengelolaan data yang terstruktur dan efisien. Fleksibilitasnya mendukung berbagai fitur seperti replikasi dan clustering, serta mudah diintegrasikan dengan bahasa pemrograman seperti *PHP*, *Python*, dan *Java*[17].

2.2.9 UML (Unified Modelling Language)

UML merupakan bahasa yang umum digunakan di dunia industri untuk mencari kebutuhan, analisis dan desain, serta menjelaskan arsitektur dalam pemrograman berorientasi objek[17].

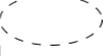
Dalam proses pembuatan sistem terdapat beberapa diagram *UML* yang sering digunakan yaitu :

a. Use Case Diagram

Use Case Diagram digunakan untuk menggambarkan kegiatan antara pengguna (aktor) dan sistem dalam skenario tertentu. *Use Case* memiliki simbol-simbol untuk penyampaian pesan, berikut beberapa simbol *use case* diagram:

Tabel 2.1. Simbol *Use Case* Diagram

No.	Simbol	Nama	Penjelasan
1		<i>Actor</i>	Orang atau sistem yang melakukan berbagai hal dengan sistem lain
2		<i>Dependency</i>	Hubungan dimana perubahan pada suatu elemen yang berdisi sendiri
3		<i>Generalization</i>	Hubungan generalisasi antar dua <i>use case</i> umum ke kusus
4		<i>Include</i>	Hubungan tambahan, dimana <i>use case</i> yang ditambahkan mamp berdiri sendiri.
5		<i>Extend</i>	Untuk menghubungkan <i>use case</i> tambahan dan <i>use case</i> tambahan butuh <i>use case</i> ini untuk bisa berjalan

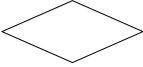
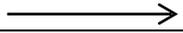
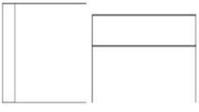
No.	Simbol	Nama	Penjelasan
6		<i>Association</i>	Komunikasi aktor dan <i>use case</i> yang ada di <i>use case</i> diagram
7		<i>System</i>	Merujuk pada paket yang menunjukkan sistem yang terbatas.
8		<i>Use Case</i>	Deskripsi dari rangkaian Aksi yang dilakukan oleh sistem
9		<i>Collaboration</i>	Hubungan antar aturan dan elemen yang saling berkollaborasi menghasilkan perilaku sistem yang kompleks
10		<i>Note</i>	komponen fisik yang hadir selama aplikasi beroperasi

b. Activity Diagram

Activity Diagram merupakan diagram yang menjelaskan secara lanjut langkah-langkah dari sebuah *use case*. Berikut merupakan beberapa simbol yang digunakan di Activity Diagram.

Tabel 2.2. Simbol *Activity* Diagram

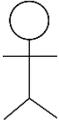
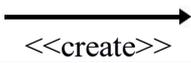
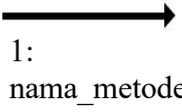
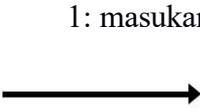
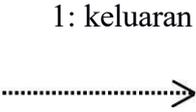
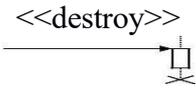
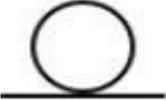
No.	Simbol	Nama	Penjelasan
1		<i>Activity</i>	Menjelaskan interaksi antar kelas

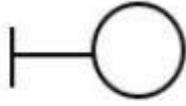
No.	Simbol	Nama	Penjelasan
2		<i>Decision</i>	Asosiasi percabangan jika pilihan lebih dari satu
3		<i>Initial Node</i>	Awal dari sebuah langkah-langkah dalam <i>activity diagram</i>
4		<i>Activity Final Node</i>	Akhir dari suatu langkah-langkah dalam <i>activity diagram</i>
5		<i>Fork Node</i>	Satu aliran yang bisa berubah menjadi beberapa aliran
6		<i>Control Flow</i>	Menampilkan urutan aliran aktivitas.
7		<i>Partition</i>	Simbol yang membatasi aktivitas antar orang, organisasi, sistem atau kelompok

c. Sequence Diagram

Sequence Diagram merupakan salah satu diagram *UML* yang digunakan untuk memodelkan hubungan dari berbagai objek dalam sistem yang berurutan. Diagram tersebut menunjukkan bagaimana pesan dikirim dan diterima antar objek, serta menggambarkan alur eksekusi suatu proses secara berurutan. Berikut beberapa simbol yang digunakan dalam *sequence diagram*.

Tabel 2.3. Simbol *Sequence Diagram*

No.	Simbol	Nama	Penjelasan
1		<i>Actor</i>	Orang yang berinteraksi dengan sistem
2		<i>Lifeline</i>	menunjukkan kehidupan suatu objek
3		<i>Object</i>	entitas yang berinteraksi melalui pesan
4		<i>Timelife</i>	keberadaan objek selama pertukaran pesan
5		Pesan tipe <i>Create</i>	Menunjukkan objek dalam kondisi aktif dan berinteraksi
6		Pesan tipe <i>Call</i>	objek melakukan pemanggilan proses yang ada pada objek lain atau dirinya
7		Pesan tipe <i>Send</i>	Menunjukkan objek mengirimkan pesan ke objek lain
8		Pesan tipe <i>Return</i>	Menunjukkan suatu objek telah berhasil prosesnya dan menghasilkan kembalian
9		Pesan tipe <i>Destroy</i>	Menyatakan suatu objek telah menjalankan suatu operasi
10		Entity Class	Kelas yang menyimpan data atau informasi

No.	Simbol	Nama	Penjelasan
11		Boundary Class	<i>user interface</i> atau alat yang berinteraksi dengan aktor dan sistem
12		Control Class	Mengendalikan jalannya informasi suatu skenario

d. Class Diagram

Class Diagram digunakan untuk memodelkan struktur kelas dalam suatu sistem. Diagram ini menunjukkan kelas yang terdapat, atribut, metode, serta relasi antar kelas seperti asosiasi, pewarisan (inheritance), agregasi, dan komposisi.

Tabel 2.4. Simbol *Class Diagram*

No.	Simbol	Nama	Penjelasan
1		Kelas	Kelas yang terdapat pada sistem
2	 Nama_interface	Antarmuka/ <i>Interface</i>	Tampilan antarmuka dari sistem
3		Asosiasi/ <i>Association</i>	Keterkaitan antar kelas dengan makna umum
4		Asosiasi berarah	keterkaitan antar kelas dengan makna kelas
5		Generalisasi	hubungan berbagai kelas dengan makna khusus

No.	Simbol	Nama	Penjelasan
6		Kebergantungan / <i>Dependency</i>	Hubungan berbagai kelas yang kebergantungan
7		Agregasi / <i>Aggregation</i>	Hubungan berbagai kelas dengan makna whole- part