

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terkait**

Penelitian yang dilakukan oleh Ririn Ayu Ardila dan Magdalena A. Ineke Pakereng (2022), dalam jurnal yang berjudul Sistem Informasi Penjualan Roti Mutiara *Bakery* Berbasis *Website*. Pada penelitian ini terdapat masalah yaitu sistem transaksi penjualan yang masih dilakukan secara manual. Proses tersebut melibatkan pencatatan data penjualan menggunakan buku nota yang kemudian datanya disimpan begitu saja sehingga memungkinkan terjadinya kehilangan data. Oleh karena itu, pada penelitian ini membuat sistem informasi berbasis web yang dapat memudahkan karyawan untuk melakukan pencatatan transaksi penjualan.[2]

Penelitian yang dilakukan oleh Ahmad Heri Setiawan dan Rony Wijanarko (2021), dalam jurnal yang berjudul Sistem Informasi Penjualan Roti Berbasis Web (Studi Kasus di CV Mams *Bakery*). Pada penelitian ini terdapat masalah yaitu mengharuskan pelanggan untuk datang langsung saat ingin melakukan pemesanan serta proses pengelolaan penjualan yang masih dilakukan secara konvensional. Oleh karena itu, sistem informasi penjualan roti berbasis web ini dapat menjadi solusi untuk memudahkan pelanggan dalam melakukan pemesanan dan juga memudahkan petugas dalam mengelola data. Dengan adanya sistem ini, pelanggan dapat melakukan

pemesanan secara *online* dengan memanfaatkan jaringan internet yang kini semakin berkembang dengan cepat.[3]

Penelitian yang dilakukan oleh Eva Lestari, Agus Nugroho, dan Despita Meisak (2023), dalam jurnal yang berjudul Perancangan Sistem Informasi Penjualan Berbasis Web Pada Toko Kue JP *Bakery And Cake*. Pada penelitian ini terdapat permasalahan pada proses jual beli yaitu pelanggan masih datang ke toko untuk membeli dan memilih kue. Selain itu, pencatatan data juga masih dilakukan secara manual, memungkinkan terjadinya kesalahan dalam pencatatan. Proses perhitungan juga masih menggunakan alat hitung, yang sering kali mengalami kesalahan baik dalam hal pencatatan data maupun pada proses perhitungan. Pada penelitian ini menggunakan bahasa pemrograman PHP dan MySQL. Dari hasil implementasi, sistem ini memberikan kemudahan kepada pelanggan dalam mendapatkan informasi barang secara detail dan mempermudah toko dalam mengelola informasi penjualan secara efisien.[4]

## **2.2 Landasan Teori**

### **2.2.1 Visual Studio Code**

*Visual Studio Code* adalah aplikasi *code editor* gratis buatan Microsoft yang dapat berjalan di semua perangkat desktop. Kelengkapan fitur dan ekstensi menjadikan *code editor* ini menjadi pilihan pertama bagi pengembang. *Visual Studio Code* mendukung hampir semua sistem operasi termasuk Windows, Mac OS, dan Linux.

Berdasarkan survey dari *Stack Overflow*, *Visual Studio Code* adalah *editor* paling populer di kalangan pengembang profesional. Dari 21 aplikasi *text editor* pesaing, *Visual Studio Code* menduduki peringkat pertama dengan persentase user sebesar 71,07%.

*Visual Studio Code* didesain seringan dan senyaman mungkin sehingga pengguna tidak membutuhkan perangkat berspesifikasi tinggi. Aplikasi ini juga bisa digunakan untuk membuat atau mengedit kode sumber dalam berbagai bahasa pemrograman seperti Node.js, JavaScript, TypeScript, dan lain-lain.

Selain itu, *Visual Studio Code* menawarkan beragam ekstensi dan ekosistem yang cukup luas. Ini memberikan kompatibilitas tinggi dengan bahasa atau *runtime environment* lain seperti bahasa pemrograman Python, PHP, .NET, dan Java.



Gambar 2.1 *Visual Studio Code*

### 2.2.2 **Laravel**

Laravel adalah framework PHP yang menggunakan pola arsitektur *Model View Controller* (MVC). Dengan menggunakan pola arsitektur MVC, struktur kode dibagi menjadi tiga bagian, lebih terorganisir dan memudahkan pengembang untuk mengembangkan

kodenya. Arsitektur MVC meningkatkan kinerja, menyediakan beberapa fungsi bawaan dan dapat diterapkan secara langsung, serta penulisan dokumentasi yang lebih baik.[5]



Gambar 2.2 Laravel

### 2.2.3 PHP

PHP (*Hypertext Preprocessor*), bahasa pemrograman *open source server side*. *Server side* adalah *script* yang diinput dan diproses oleh server. PHP memiliki keuntungan karena bersifat *open source*. Artinya, pengguna bebas memodifikasi dan mengembangkan aplikasi dan sistem.[6]



Gambar 2.3 PHP (*Hypertext Preprocessor*)

### 2.2.4 MySQL

MySQL adalah *database* atau penyimpanan data yang memudahkan seorang *programmer* menyimpan data yang telah dimasukkan. MySQL sendiri memiliki arti yaitu *software database open source* dan banyak digunakan untuk mengolah *database* dengan

menggunakan bahasa SQL.[7] MySQL adalah *database server* gratis yang dilisensikan di bawah GNU *General Public License* (GPL), sehingga dapat digunakan untuk keperluan pribadi maupun komersial tanpa membayar lisensi yang sudah ada.



Gambar 2.4 MySQL

### 2.2.5 XAMPP

XAMPP adalah perangkat lunak yang dikembangkan pada tahun 2002 oleh sekelompok tim Apache Friend dan tersedia secara gratis dengan label *General Public License* (GNU). Sebagai *software open source* berbasis web server, XAMPP memiliki beragam program dan mendukung berbagai sistem operasi yang umum digunakan seperti Linux, Windows, atau MacOS.[8] Aplikasi ini bertindak sebagai server lokal yang berisi program Apache, MySQL, dan PHP.

Dengan kata lain, XAMPP merupakan media atau *web server localhost* yang dapat digunakan secara *offline*. Melalui XAMPP, pengguna dapat mengelola *database* yang terletak di *localhost* tanpa memerlukan akses Internet.



Gambar 2.5 XAMPP

### 2.2.6 *Database*

*Database* adalah kumpulan basis data atau data yang dikelola sesuai ketentuan tertentu dan saling berhubungan supaya mudah dikelola. Pengelolaan *database* memudahkan siapa saja untuk mencari, menyimpan, dan menghapus informasi.[9] *Database* juga dapat diartikan sebagai suatu sistem yang dirancang untuk mengumpulkan data, arsip, atau tabel yang disimpan dan dihubungkan dengan media elektronik seperti aplikasi dan *website*. Basis data membuat penyimpanan dan pengelolaan data menjadi lebih efisien.

Definisi lain dari database menurut Oracle adalah kumpulan terorganisir dari informasi terstruktur atau data yang disimpan secara elektronik dalam sistem komputer. *Database* biasanya dikendalikan oleh DBMS (sistem manajemen *database*). Data dan DBMS aplikasi yang terkait biasanya disebut sebagai sistem *database*. Data dalam *database* biasanya dimodelkan sebagai kumpulan baris dan kolom tabel. Tujuannya agar pengolahan dan pengambilan data menjadi lebih efisien. Data kemudian dapat diakses, dikelola, dimodifikasi, diperbarui, dikontrol, dan diatur. Sebagian besar *database*

menggunakan *Structured Query Language* (SQL) untuk menulis dan meminta data.



Gambar 2.6 *Database*

### 2.2.7 **PhpMyAdmin**

PhpMyAdmin adalah perangkat lunak gratis yang ditulis dalam bahasa PHP dan merupakan perangkat lunak paling populer untuk mengelola tabel dan data pada database melalui web. PHPMyAdmin mendukung berbagai operasi *database* seperti MySQL dan MariaDB. Tugas yang dimaksudkan antara lain mengelola *database*, tabel, kolom, indeks, *user*, *permission*, dan lain-lain.[10] Semua tugas ini dapat dilakukan melalui antarmuka yang mudah digunakan. Namun, walaupun phpMyAdmin memiliki antarmuka pengguna, pengguna juga dapat menjalankan MySQL *statement* dan *query* secara langsung.

PhpMyadmin sangat populer untuk mengelola *database* karena dapat diakses melalui *web browser*. Selain itu, *software* ini juga memungkinkan pengguna membuat, memperbarui, mengubah, menghapus, mengimpor, dan mengekspor tabel *database* MySQL. Pengguna juga dapat menjalankan *query* MySQL, memperbaiki,

mengoptimalkan, dan memeriksa tabel, serta melakukan perintah manajemen *database* lainnya. PhpMyAdmin juga dapat digunakan untuk melakukan tugas administratif seperti membuat *database* dan menjalankan *query*.



Gambar 2.7 PhpMyAdmin

### 2.2.8 *Hypertext Markup Language (HTML)*

*Hypertext Markup Language (HTML)* adalah bahasa *markup* standar yang digunakan untuk membuat halaman situs web dan aplikasi web. HTML diciptakan oleh Tim Berners-Lee, seorang fisikawan di laboratorium CERN di Swiss.[11] Versi HTML pertama diterbitkan oleh Tim Berners Lee pada tahun 1991 dan memiliki 18 tag. Sejak itu, versi baru selalu menyertakan tag dan atribut baru.

Spesifikasi HTML dikelola dan dikembangkan oleh *World Wide Web Consortium (W3C)*. Berdasarkan *HTML Element Reference* milik *Mozilla Developer Network*, saat ini terdapat 140 tag HTML, meskipun beberapa tidak lagi didukung di versi terbaru beberapa *browser*. Kemudian, pada tahun 2014, HTML5 diperkenalkan. Pengenalan HTML5 menambahkan semantik baru seperti `<article>`,



<header>, dan <footer> yang membuat beberapa bagian konten menjadi lebih jelas.



Gambar 2.8 HTML (*Hypertext Markup Language*)

### 2.2.9 Javascript

Javascript adalah bahasa pemrograman tingkat tinggi yang awalnya dirancang untuk menambahkan interaktivitas pada halaman web. Bersama dengan HTML dan CSS, Javascript menjadi pilihan utama dalam mengembangkan aplikasi web yang dinamis. Sebagai bahasa *scripting*, javascript memiliki kemampuan untuk memberikan logika dan fungsionalitas tambahan pada halaman web tanpa memerlukan proses kompilasi. Artinya, kode javascript dapat dieksekusi langsung oleh *interpreter* tanpa perlu melalui tahap kompilasi terlebih dahulu. Dengan menggunakan *interpreter*, javascript mengubah kode yang ditulis menjadi instruksi yang dapat dimengerti oleh mesin. Karena itu, bahasa *scripting* seperti javascript memungkinkan pengembang untuk menciptakan program yang kompleks dengan menggunakan sedikit kode.[12]

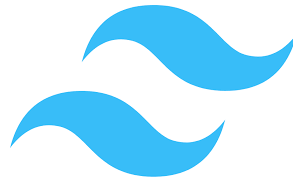


Gambar 2.9 Javascript

### 2.2.10 Tailwind CSS

Tailwind CSS adalah *framework Cascading Style Sheet* yang digunakan untuk menyesuaikan atau mendesain antarmuka pengguna pada sebuah web. *Framework* ini berbasis *utility* dan hanya terdiri dari *utility class* dan tanpa *utility* komponen lainnya. *Framework* CSS ini cukup populer di kalangan *developer* karena dilengkapi dengan fitur-fitur yang membuat pembuatan tampilan web lebih mudah dan cepat.

Tailwind CSS dikembangkan oleh Adam Wathan pada tahun 2017. *Framework* CSS ini memiliki konsep *utility-first classes*. Utilitas ini mempermudah pengguna dengan *class-class* yang telah dipetakan. *Framework* ini memiliki ratusan *class* CC yang bisa dipanggil dengan mudah melalui dokumen HTML. Pengembang cukup memberi nama setiap *utility-class* sesuai kebutuhan. Konsep ini memungkinkan pengembang web untuk lebih mudah menyesuaikan *layout*, *background*, warna, tipografi, dan aset lainnya tanpa menuliskan kode HTML atau CSS khusus lainnya.



Gambar 2.10 Tailwind CSS

### 2.2.11 Node JS

Node JS adalah perangkat lunak yang dirancang untuk mengembangkan aplikasi berbasis web dengan menggunakan sintaks bahasa pemrograman javascript. Meskipun selama ini javascript dikenal sebagai bahasa pemrograman yang berjalan di sisi klien (*browser*), Node.js memperluas fungsinya sehingga dapat digunakan sebagai bahasa pemrograman disisi server, seperti halnya PHP, Ruby, atau Perl. Node.js dapat dijalankan pada berbagai sistem operasi, termasuk Windows, Mac OS X, dan Linux, tanpa memerlukan perubahan pada kode program. Dengan pustaka server HTTP bawaan, Node.js memungkinkan pengembangan server web tanpa perlu menggunakan program server web seperti Apache atau Nginx.[13]



Gambar 2.11 Node JS

### 2.2.12 ReactJS

ReactJS atau React adalah *library* JavaScript populer dari *Facebook* yang digunakan dalam proses pengembangan aplikasi

*mobile* dan web. React berisi kumpulan *snippet* kode JavaScript (disebut "komponen") yang dapat digunakan berulang kali untuk mendesain antarmuka pengguna.[14] ReactJS bukanlah *framework* JavaScript karena hanya bertugas untuk merender komponen area tampilan aplikasi. React adalah alternatif untuk *framework* seperti Angular dan Vue yang sama-sama dapat digunakan untuk membuat fungsi yang kompleks.



Gambar 2.12 ReactJS

### 2.2.13 ViteJS

Sebenarnya, Vite dikembangkan untuk *Vue Single File Components* (SFC) sebagai server pengembangan. Namun seiring berjalannya waktu, Vite telah berevolusi menjadi server pengembangan JavaScript tanpa *bundle*. Vite saat ini mendukung sebagian besar kerangka web. Selain itu, menawarkan alur kerja yang lebih cepat, lebih ramping, dan lancar untuk mengembangkan aplikasi web *modern*. Singkatnya, Vite mirip dengan Vue CLI, namun lebih ramping dan lebih cepat.

Vite menggunakan kode sumber melalui ESM asli. Selain itu, dengan menggunakan ESBuild dan ESM, vite meningkatkan kecepatan *startup* 10-100x. Namun, *browser* menggabungkan kode

sumber. Oleh karena itu, Vite hanya menyediakan dan mengubah kode sumber jika *browser* memintanya. Ini mempercepat proses menyimpan kode yang dimodifikasi dan melihat perubahan di *browser*.



Gambar 2.13 ViteJS

#### **2.2.14 Payment Gateway**

*Payment gateway* adalah istilah bahasa Inggris yang berasal dari kata *payment* artinya pembayaran dan *gateway* berarti gerbang. Bila digabungkan artinya gerbang pembayaran adalah teknologi yang memungkinkan bisnis apa pun menerima transaksi pembayaran dari berbagai tempat dan waktu tak terbatas menggunakan Internet. Jadi pengertian *payment gateway* adalah alat pembayaran untuk transaksi pada layanan aplikasi *e-commerce* dengan fungsi mengotorisasi berbagai proses pembayaran baik perbankan, kartu kredit, transfer bank maupun secara langsung dari konsumen.[15]

#### **2.2.15 Figma**

Figma merupakan salah satu alat berbasis web yang digunakan untuk proses desain yang dapat diakses secara daring, memungkinkan pengguna melakukan desain kapan pun dan di mana pun melalui

internet. Secara umum, Figma sering digunakan untuk merancang antarmuka dari aplikasi yang sedang dikembangkan. Dalam pengembangan aplikasi baru, Figma dapat dimanfaatkan untuk kolaborasi tim dalam membangun aplikasi tersebut. Figma dapat dijalankan pada sistem operasi Windows dan macOS untuk desktop.

Fitur utama Figma terfokus pada pengembangan desain *User Interface* (UI) dan *User Experience* (UX). Proses pengembangan antarmuka aplikasi dan pengalaman pengguna yang dihasilkan dalam penggunaan aplikasi dapat dikembangkan melalui platform Figma. Meskipun terdapat alat lain yang memiliki fungsi serupa seperti Sketch dan Adobe XD, perbedaan utama antara Figma dengan alat lainnya terletak pada fitur yang disediakan oleh masing-masing alat tersebut.



Gambar 2.14 Figma





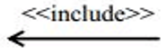
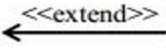
### **2.2.16 Unified Modeling Language (UML)**

*Unified Modeling Language* (UML) adalah bahasa pemodelan perangkat lunak yang distandarisasi sebagai media penulisan cetak biru perangkat lunak (*Pressman*). UML bisa digunakan untuk visualisasi, spesifikasi, konstruksi dan dokumentasi beberapa bagian dari sistem yang ada dalam perangkat lunak. Dengan kata lain, sama

seperti seorang arsitek membuat dokumen cetak biru yang digunakan perusahaan konstruksi untuk membangun sebuah bangunan, arsitek perangkat lunak membantu pemrogram/pengembang membangun perangkat lunak.[16] Terdapat beberapa diagram UML yang sering digunakan dalam pengembangan sebuah sistem yaitu:

1. *Use Case Diagram* adalah jenis diagram UML (*Unified Modeling Language*) yang menggambarkan hubungan interaksi antara sistem dan aktor. Sebuah *use case* dapat menggambarkan tipe interaksi antara pengguna sistem dengan sistem. *Use Case* merupakan sesuatu yang mudah dipelajari. Langkah awal untuk melakukan pemodelan perlu adanya suatu diagram yang mampu menjabarkan aksi aktor dengan aksi dalam sistem itu sendiri, seperti yang terdapat pada *Use Case*.


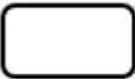



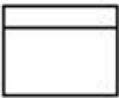
Tabel 2.1 Simbol *Use Case Diagram*

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor.
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i> .
	Generalisasi : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.

2. *Activity diagram* atau diagram aktivitas dalam bahasa Indonesia, adalah diagram yang dapat memodelkan berbagai proses yang terjadi dalam suatu sistem. Seperti layaknya runtutan proses berjalannya suatu sistem dan digambarkan secara vertikal. *Activity diagram* adalah salah satu contoh diagram dari UML dalam pengembangan dari *Use Case*.



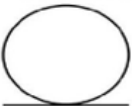
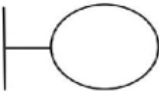
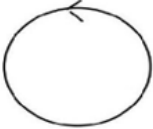
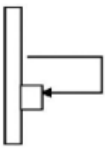


Tabel 2.2 Simbol *Activity Diagram*

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / <i>Decision</i>	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / <i>Join</i>	Penggabungan dimana yang lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	<i>Swimlane</i>	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

3. *Sequence diagram* adalah diagram yang menjelaskan interaksi objek berdasarkan urutan waktu. *Sequence* dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk

dapat menghasilkan sesuatu, seperti yang tertera pada *Use Case diagram*.






Tabel 2.3 Simbol *Sequence Diagram*

Gambar	Nama	Keterangan
	<i>Entity class</i>	Gambaran sistem sebagai landasan dalam menyusun basis data.
	<i>Boundary class</i>	Menangani komunikasi antar lingkungan sistem.
	<i>Control class</i>	Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika.
	<i>Recursive</i>	Pesan untuk dirinya.
	<i>Activation</i>	Mewakili proses durasi aktivasi sebuah operasi.
	<i>Life line</i>	Komponen yang digambarkan garis putus terhubung dengan objek.

4. *Class diagram* atau diagram kelas adalah suatu diagram yang digunakan untuk menampilkan kelas-kelas berupa pake-paket

untuk memenuhi salah satu kebutuhan paket yang akan digunakan nantinya.

Tabel 2.4 Simbol *Class Diagram*

No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
2		<i>Nary association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.

6	----->	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri.
7	—————	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

### 2.2.17 Pengujian Sistem

#### 1. *Black Box Testing*

*Black box testing* adalah pengujian yang bertujuan untuk menunjukkan kesalahan dalam sistem aplikasi, seperti kesalahan pada fungsi sistem aplikasi dan hilangnya menu aplikasi. *Black box testing* merupakan metode pengujian fungsionalitas sistem aplikasi. Dalam pengujian ini, digunakan data acak sebagai masukan untuk mendapatkan hasil yang pasti. Hasil yang pasti berarti jika terdapat kesalahan, maka akan ditolak oleh sistem informasi, atau data input tersebut tidak akan disimpan dalam basis data. Sebaliknya, jika data input benar, maka data tersebut akan diterima dan masuk kedalam basis data sistem informasi.[17]

## 2. *White Box Testing*

*White box testing* adalah metode pengujian yang berfokus pada sistem internal, yaitu kode sumber program. Tujuan utama dari pengujian *white box* adalah untuk mengukur kompleksitas kode program. Bagi *programmer*, *white box testing* sangat penting untuk menilai tingkat kompleksitas suatu kode. Selain itu, pengujian ini juga digunakan untuk memvalidasi apakah kode sumber sesuai dengan desain yang telah ditetapkan, memenuhi kebutuhan fungsional, dan tidak memiliki kerentanan.[18]