

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terkait**

Rikardo Sihombing, dkk(2023) dengan judul ” Rancang Bangun Sistem Informasi Pelayanan Restoran Hotel Berbasis Web Dengan Framework Codeigniter” dengan metode R&D (*Research and Development*) yang memudahkan sistem kerja yang ada didalam restoran tersebut, baik dari tamu dan staff Restoran[4].

Endang Setyawati, dkk(2021) dengan judul “Sistem Inforasi Pelayanan Administrasi Surat Pengantar Berbasis Website dengan Framework Codeigniter Guna Meningkatkan Kualitas Pelayanan Pada Desa Tambaksari Kidul Kabupaten Banyumas” dengan menggunakan metode prototype yang dapat meningkatkan efektivitas proses pembuatan surat pengantar di desa Tambaksari Kidul[5].

Janu Ilham Saputro, dkk(2020) dengan judul “Sistem Informasi Surat Pengantar Berbasis Web (Studi Kasus: Kelurahan Sukasari Tangerang)” dengan menggunakan metode analisa SDLC (*System Development Life Cycle*) dan metode perancangan UML (*Unified Modelling Language*) yang dapat memberikan manfaat untuk menunjang kinerja pelayanan yang lebih baik dalam membuat surat pengantar[6].

Ade Rahman, dkk(2023) dengan judul “Perancangan Sistem Informasi Administrasi Penduduk Untuk Validitas Data Kependudukan Menggunakan Framework Codeigniter 4” dengan metode XP (*Extreme Programming*) yang dapat meningkatkan efisiensi kerja dan pelayanan masyarakat[7].

Dony Rakhmad Hidayat, dkk(2022) berjudul “Implementasi Framework Codeigniter Dalam Pembuatan Sistem Informasi Pencatatan dan Pendataan Penduduk Desa Berbasis Web” menggunakan metode *waterfall* dalam merancang sistem yang dapat membantu perangkat desa dalam melakukan pelayanan, pengambilan keputusan, pembagian wilayah administrasi berdasarkan RT/RW dan pelaporan data kependudukan[8].

## 2.2 Landasan Teori

### 2.2.1 Sistem Informasi

Menurut (Sutabri, 2019) sistem informasi dapat diartikan sebagai kumpulan dari berbagai elemen yang saling terkait secara berkesinambungan dalam pengumpulan, penyimpanan, pengolahan, dan penyampaian informasi untuk mendukung berbagai kegiatan dalam suatu organisasi atau entitas. Sistem informasi mencakup perangkat keras, perangkat lunak, data, prosedur, dan orang-orang yang terlibat dalam siklus hidup informasi[9].

### 2.2.2 Website

*Website* adalah sekumpulan halaman web yang saling berhubungan yang umumnya berada pada *server* yang sama berisikan kumpulan informasi yang disediakan secara perorangan, kelompok, atau organisasi. Sebuah situs web biasanya ditempatkan setidaknya pada sebuah *server web* yang dapat diakses melalui

jaringan seperti internet yang dikenali sebagai URL. Unsur-unsur dari *web* adalah sebagai berikut:

1. Nama Domain (*Domain name/URL – Uniform Resource Locator*)

Domain internet adalah bagian dari alamat situs *web* yang mengidentifikasi situs tersebut didunia maya. Ketika *website* dapat diibaratkan sebagai sebuah produk, maka *website* tersebut memiliki merek atau *brand* sendiri. Penulisan domain yang menarik dapat mendorong seseorang untuk mengunjungi situs tersebut[10].

2. *Web Hosting*

*Hosting* berasal dari kata *host*, yang berarti komputer yang terhubung dalam jaringan. *Hosting* adalah penggunaan fasilitas yang tersedia di komputer yang terhubung ke jaringan untuk menyediakan layanan internet seperti penempatan informasi di halaman *web*, penyimpanan email, basis data, dan lain sebagainya. Dengan kata lain, layanan ini adalah solusi untuk menempatkan halaman *web* secara *online*.

Menurut Aliyun(2014) *Hosting* juga memiliki arti layanan berbasis internet sebagai tempat penyimpanan data atau tempat menjalankan aplikasi ditempat terpusat yang disebut dengan *server* dan dapat diakses melalui jaringan internet.

Menurut M. Iqbal Kurniansyah, dkk(2020) Layanan *Hosting*

internet adalah jasa yang menyediakan sumber daya *server* yang dapat disewa oleh organisasi atau individu. Dengan menggunakan layanan ini, mereka dapat menempatkan informasi di internet dalam berbagai bentuk seperti situs web (HTTP), transfer *file* (FTP), penyimpanan email, atau sistem nama domain (DNS) [11].

Beberapa jenis layanan *hosting* yaitu *shared hosting*, VPS atau *Virtual Private Server*, *dedicated server*, *colocation server* .

### 3. Bahasa Program

Menurut Mohammad Farid Naufal(2018) dalam karya Terttiaavini(2023) pemrograman merupakan seni menggabungkan satu atau lebih algoritma dengan menggunakan bahasa pemrograman tertentu untuk menciptakan sebuah program komputer. Bahasa pemrograman sangat penting dalam merancang aplikasi berbasis web sesuai dengan kebutuhan yang ada [12].

Penelitian ini menggunakan HTML (*Hyper Text Markup Language*), PHP (*Hypertext Preprocessor*) sebagai bahasa pemrograman utama untuk merancang dan mengembangkan aplikasi web yang diperlukan.

### 4. Desain *website*

Desain website merujuk pada proses merencanakan, membuat, dan mengatur tampilan visual serta fungsionalitas suatu situs

web. Desain website tidak hanya tentang aspek visual, tetapi juga tentang kegunaan (*usability*) dan pengalaman pengguna (*user experience*). Desain web harus memastikan bahwa situs dapat diakses dengan mudah, intuitif untuk dinavigasi, dan memberikan nilai tambah kepada pengguna.

#### 5. Program Transfer data ke pusat data

Program transfer data ke pusat data adalah aplikasi atau sistem yang dirancang untuk menstrafer data dari lokasi tertentu (misalnya, mesin pengguna atau server local) ke pusat data atau data center. Tujuan utama dari program tersebut adalah untuk menyimpan data secara terpusat di lokasi yang aman dan dapat diakses dengan mudah.

Program tersebut mungkin melibatkan penggunaan protocol komunikasi tertentu seperti FTP (File Transfer Protocol), SCP (Secure Copy Protocol), SFTP (SSH File Transfer Protocol), atau protocol khusus lainnya yang sesuai dengan kebutuhan dan keamanan.

#### 6. Publikasi *website*

Publikasi website adalah proses membuat situs web secara tersedia untuk diakses oleh pengguna melalui internet. Proses publikasi website melibatkan beberapa tahap seperti memuat konten ke server web, mengonfigurasi domain dan hosting, serta mengatur izin akses agar situs dapat ditemukan dan diakses

secara luas oleh pengguna.

### 2.2.3 *Database*

Basis data adalah tempat penyimpanan data yang menggantikan sistem tradisional seperti penyimpanan dokumen dalam file. Database merupakan kumpulan data yang terkait dan dirancang untuk memenuhi kebutuhan informasi suatu organisasi. Untuk mengolah database, strategi pencarian ditentukan oleh query agar dapat dieksekusi secara optimal. Beberapa sistem manajemen basis data (DBMS) menggunakan heuristik tertentu dalam memilih strategi pemrosesan query. Meskipun bahasa query SQL memiliki kelemahan, optimalisasi diperlukan untuk memilih strategi yang efisien dalam mengevaluasi ekspresi relasional.

### 2.2.4 *Xampp*

*XAMPP* merupakan sebuah paket perangkat lunak server web yang mencakup Apache, server basis data MySQL, dan mendukung Bahasa pemrograman PHP. Singkatan *XAMPP* sendiri berasal dari empat komponen utamanya: X untuk empat sistem operasi yang didukung (Windows, Linux, MacOS), Apache, MySQL, PHP, dan Perl [13].



Gambar 2.1 *Xampp*

### 2.2.5 *Framework*

*Framework* adalah sebuah struktur yang terdiri dari kelas-kelas dan fungsi-fungsi yang telah dikumpulkan dengan tujuan untuk mempermudah para pengembang dalam memanggilnya tanpa perlu mengulang-ulang sintaks program yang sama, serta membantu dalam menghemat waktu [14].

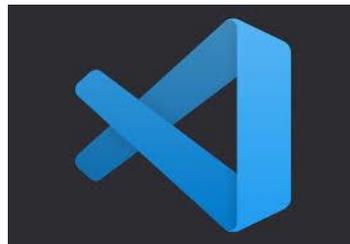
### 2.2.6 *Codeigniter*

*Codeigniter* adalah sebuah *framework* yang didasarkan pada pola desain Model View Controller (MVC). Pola desain merupakan kumpulan penjelasan mengenai metode-metode untuk menyelesaikan masalah umum yang muncul dalam perancangan perangkat lunak [14].

### 2.2.7 *Visual Studio Code*

*Visual Studio Code* adalah sebuah editor kode yang dikembangkan oleh Microsoft. Salah satu editor kode yang paling populer dan banyak digunakan oleh para pengembang perangkat lunak. *Visual Studio Code* didesain untuk memberikan pengalaman pengembangan yang produktif dan efisien dengan fitur-fitur seperti penyorotan sintaksis, penyelesaian otomatis, debugging yang terintegrasi, control versi yang terintegrasi, dan banyak lagi. Selain itu, *Visual Studio Code* mendukung berbagai Bahasa pemrograman dan platform, membuatnya menjadi pilihan yang serbaguna bagi para pengembang dari berbagai latar belakang.

*Visual Studio Code* adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks Editor ini secara langsung mendukung Bahasa pemrograman Javascript, Typescript, dan Node. Js, serta Bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via marketplace *Visual Studio Code* seperti : C++, C#, Python, Go, Java, PHP, dst [15].



Gambar 2.2 *Visual Studio Code*

#### 2.2.8 *Bootstrap*

*Bootstrap* adalah sebuah kerangka kerja yang digunakan untuk mengembangkan desain web yang responsif. Desain web yang dibangun menggunakan *bootstrap* akan secara otomatis menyesuaikan tampilannya dengan ukuran layar yang digunakan oleh pengguna, baik itu desktop, tablet, maupun perangkat seluler. *Bootstrap* menyediakan beragam plugin CSS, HTML, dan juga JQuery untuk berbagai elemen seperti tipografi, formulir, tombol, tabel, grid, navigasi, serta banyak komponen antarmuka pengguna (UI) lainnya.

*Bootstrap* sendiri merupakan *library* yang biasa digunakan untuk membuat aplikasi web ataupun situs web responsif secara cepat, mudah dan gratis [16].



Gambar 2.3 *Bootstrap*

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

UML adalah salah satu alat atau model yang digunakan untuk merancang pengembangan perangkat lunak berorientasi objek. UML memberikan standar dalam penulisan blueprint sistem, yang mencakup konsep proses bisnis, struktur kelas dalam Bahasa pemrograman tertentu, desain skema basis data, serta komponen-komponen yang diperlukan dalam pengembangan sistem perangkat lunak [17].

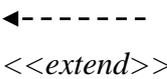
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis *UML* adalah sebagai berikut:

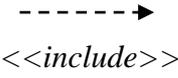
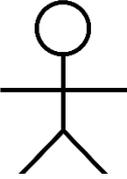
1. *Use Case Diagram*

*Diagram use case* adalah model pemodelan perilaku sistem informasi yang akan dibangun. Digunakan untuk mengidentifikasi fungsi-fungsi yang ada di dalam sistem informasi dan untuk menentukan siapa yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang

digunakan dalam *Use Case Diagram* yaitu [18]:

Tabel 2.1 *Use Case Diagram*

No	Simbol	Pengertian	Keterangan
1.		<i>Use Case</i>	Menggambarkan fungsi-fungsi yang disediakan oleh sistem yang memungkinkan pertukaran pesan antara unit atau actor biasanya dijelaskan menggunakan frasa nama <i>Use Case</i> yang mengandung kata kerja.
2.		Asosiasi/ <i>Association</i>	Komunikasi antara actor dan <i>Use Case</i> yang terlibat pada penggunaan atau interaksi <i>Use Case</i> melibatkan partisipasi actor dalam proses tersebut.
3.		Generalisasi/ <i>generalization</i>	Asosiasi antara actor dan <i>Use Case</i> yang menggunakan panah terbuka mengindikasikan bahwa actor berinteraksi secara pasif dengan sistem.
4.		Ekstensi/ <i>Extend</i>	Merupakan perluasan dari <i>use case</i> lain terjadi ketika kondisi atau syarat tertentu terpenuhi.

No	Simbol	Pengertian	Keterangan
5.		<i>Include/uses</i>	Merupakan bagian dari <i>use case</i> lain ( <i>required</i> ) atau dipanggil oleh <i>use case</i> lain terjadi ketika sebuah <i>use case</i> membutuhkan atau memanggil <i>use case</i> lain untuk mencapai tujuan tertentu, seperti pemanggilan sebuah fungsi program dalam sebuah <i>use case</i> .
6.		<i>Actor/Aktor</i>	Aktor dalam konteks sistem informasi merujuk pada orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibangun, tetapi berada di luar sistem tersebut. Meskipun symbol dari actor seringkali berupa gambar orang, namun sebenarnya actor tidak selalu mewakili manusia. Biasanya, kata yang berbeda digunakan di awal frasa nama actor untuk menjelaskan perannya dalam interaksi dengan sistem. Penting

No	Simbol	Pengertian	Keterangan
			untuk dicatat bahwa actor berinteraksi dengan <i>Use Case</i> , namun actor tidak memiliki kendali atau control langsung terhadap <i>Use Case</i> tersebut.

## 2. *Activity Diagram*

*Activity Diagram* adalah representatif grafis dari aliran kerja atau serangkaian kegiatan dalam sebuah sistem atau proses. Menunjukkan urutan langkah-langkah atau aktivitas yang harus dilakukan, keputusan yang dibuat selama proses, dan hubungan antara aktivitas-aktivitas tersebut. Diagram aktivitas biasanya digunakan untuk memodelkan alur kerja dalam sebuah sistem atau proses bisnis, dan membantu untuk memahami bagaimana pekerjaan atau informasi bergerak dari satu aktivitas ke aktivitas lainnya dalam sistem tersebut. Simbol-simbol yang digunakan dalam *Activity Diagram* yaitu:

Tabel 2.2 *Activity Diagram*

No	Simbol	Pengertian	Keterangan
1.		<i>Activity</i>	Menggambarkan cara setiap kelas antarmuka berinteraksi satu sama lainnya.

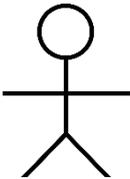
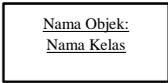
No	Simbol	Pengertian	Keterangan
2.		<i>Activity Final Mode</i>	Suatu kegiatan yang menandai akhir dari alur aktivitas dalam diagram aktivitas.
3.		Initial Node	Titik awal atau permulaan dari alur aktivitas dalam sebuah diagram aktivitas.
4.		Fork Node	Titik dalam sebuah diagram aktivitas yang digunakan untuk membagi alur kerja menjadi dua atau lebih alur kerja paralel.

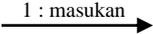
### 3. *Sequence Diagram*

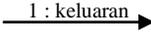
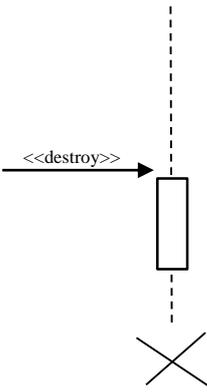
*Sequence Diagram* menggambarkan interaksi antara objek-objek di dalam dan di sekitar sistem (termasuk pengguna, tampilan, dan sebagainya) dalam bentuk pesan yang diperlihatkan seiring berjalannya waktu. Diagram ini mencakup dimensi vertikal (waktu) dan horizontal (objek-objek yang saling berhubungan)[19].

Berikut ini adalah simbol-simbol *sequence diagram*.

Tabel 2.3 Sequence Diagram menurut (Irfan, 2023)

No	Simbol	Pengertian	Keterangan
1.		Aktor	<p>Proses, orang atau sistem yang berbeda yang terhubung dengan sistem data yang sedang dikembangkan berada di luar sistem informasi yang sedang dibuat, sehingga aktor tidak selalu berupa manusia meskipun simbol aktor adalah gambar seseorang. Biasanya, kata benda digunakan di depan frasa nama aktor untuk mengungkapkannya.</p>
2.		<i>Lifeline</i>	Mewakili siklus hidup suatu objek.
3.		Objek	Mengidentifikasi objek yang berinteraksi melalui pesan.

No	Simbol	Pengertian	Keterangan
4.		Waktu Aktif	Mengatur bahwa objek tersebut aktif dan interaktif, serta semua hal yang terkait dengan jangka waktu aktivitasnya merupakan langkah yang diambil di dalamnya.
5.		Pesan tipe <i>Create</i>	Menunjukkan bahwa sebuah objek menciptakan objek lain, dengan panah yang mengarah ke objek yang dibuat. Panah ini menunjuk ke objek yang memiliki operasi/metode tersebut. Operasi/metode yang dipanggil harus ada dalam diagram kelas sesuai dengan kelas dari objek yang berinteraksi.
6.		Pesan tipe <i>Send</i>	Merupakan objek yang mengirimkan

No	Simbol	Pengertian	Keterangan
			<i>input</i> data, atau informasi ke objek lain, dengan panah yang mengarah ke objek penerima.
7.		Pesan tipe <i>return</i>	Mengindikasikan bahwa suatu objek kembali ke objek tertentu setelah menjalankan operasi atau metode tertentu, dengan panah yang mengarah ke objek yang menerima pengembalian.
		Pesan tipe <i>destroy</i>	Menetapkan bahwa sebuah objek mengakhiri keberadaan objek lain, dengan panah yang mengarah ke objek yang diakhiri. Idealnya, jika ada <i>create</i> , harus ada juga <i>destroy</i> .

#### 4. Class Diagram

*Class Diagram* adalah cara untuk memvisualisasikan

struktur sistem dengan mendefinisikan kelas-kelas yang menyusunnya[20].