

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Sebuah penelitian yang diterbitkan dalam jurnal berjudul "Perancangan Sistem Informasi Geografis Objek Wisata Labuhanbatu Berbasis *Web*" oleh Dwi Fitriani, Ibnu Rasyid Munthe, dan Budianto Bangun (2021) bertujuan untuk menciptakan Sistem Informasi Geografis dengan menggunakan teknik *waterfall*. Metode ini dirancang untuk membantu wisatawan mendapatkan informasi yang akurat dan menyeluruh tentang objek wisata Labuhanbatu. Sistem informasi ini diuji melalui pengujian *black box*. Peta tempat wisata akan memudahkan navigasi wisatawan, meningkatkan jumlah wisatawan, dan meningkatkan PAD bagi penduduk lokal dan wilayah tersebut. Pengguna Sistem Informasi Geografis ini dapat dengan mudah mengakses peta ini.[1]

Gabriella Zenia Karundeng, Jullia Titaley, dan Marline S. Paendong melakukan penelitian yang diterbitkan dalam jurnal berjudul "Sistem Informasi Objek Wisata Berbasis *Web* di Kabupaten Bolaang Mongondow Utara" Tujuan dari penelitian ini adalah untuk membuat aplikasi sistem informasi tempat wisata berbasis *web*. Tujuan sistem ini adalah untuk membuat pengunjung lebih mudah menemukan tempat wisata lokal dan internasional. Analisis kebutuhan, desain sistem, pengkodean, pengujian, dan implementasi sistem adalah semua bagian dari pendekatan *waterfall* yang digunakan dalam penelitian ini. Semua informasi tentang rute perjalanan,

akomodasi, tempat makan, dan lokasi menarik dapat ditemukan di sistem ini.[2]

Penelitian yang ditulis oleh Surya Hendra Putra dan Evan Afri (2020) berjudul "Penerapan Sistem Informasi Geografis Berbasis *Web* untuk Pengembangan Pariwisata pada Kabupaten Langkat" bertujuan untuk memanfaatkan aplikasi berbasis *web* untuk menciptakan dan mempromosikan destinasi wisata yang mungkin ada di Kabupaten Langkat. Aplikasi ini menggunakan metodologi GIS (Sistem Informasi Geografis) dan memiliki peta digital yang didukung oleh *Google Maps API*. Aplikasi ini akan mendorong lebih banyak orang untuk mengunjungi Kabupaten Langkat.[3]

Penelitian yang ditulis oleh Ersya Nur Annisa, Nur Hafifah Matondang, dan Sarika Afrizal diterbitkan dalam jurnal berjudul "Sistem Informasi Pariwisata Berbasis *Web* pada Kabupaten Nunukan". Tujuan dari penelitian ini adalah untuk mengatasi tantangan yang dihadapi dalam menyebarkan informasi tentang tempat wisata di Kabupaten Nunukan, Kalimantan Utara. Studi ini membuat sistem informasi pariwisata berbasis *web* dengan menggunakan proses pengembangan *waterfall* dan analisis sistem *PIECES*. *Database MySQL* dan bahasa pemrograman *PHP, HTML, CSS, dan Laravel* digunakan untuk membuat sistem ini. Paket wisata, akomodasi, makanan, informasi budaya, dan lokasi wisata adalah semua fitur yang ditawarkan.[4]

Penelitian yang ditulis oleh Saefudin dan Diah Islamiati diterbitkan dalam jurnal berjudul "Sistem Informasi Geografis Pemetaan Daerah Pariwisata" bertujuan untuk memecahkan masalah ketidakmampuan

penduduk Kabupaten Serang, Provinsi Banten, dalam memperoleh informasi wisata yang jelas dan efektif. Studi ini menciptakan sistem informasi geografis berbasis *web* yang membuat destinasi wisata di Kabupaten Serang mudah diakses dengan menyediakan data dan deskripsi peta yang rinci. Proses pengembangan *waterfall* dan bahasa pemodelan *Unified Modeling Language* (UML) digunakan untuk membuat aplikasi ini, dengan *Leaflet J* sebagai *platform* pemetaan. Aplikasi *Blackbox* diuji untuk memastikan kualitas dan fungsinya. Aplikasi *web* untuk pemetaan kawasan wisata Kabupaten Serang dibuat sebagai hasil dari proyek ini.[5]

2.2 Landasan teori

2.2.1 Sistem Informasi

Sistem informasi adalah gabungan dari komponen-komponen sistem dan informasi yang saling berinteraksi untuk menghasilkan informasi yang bermanfaat bagi penggunanya.[6]

Lebih tepatnya, beberapa ahli mendefinisikan sistem informasi sebagai berikut:

1. Menurut McKeown (1993), sistem informasi adalah kombinasi komputer dan pengguna yang mengatur konversi data menjadi informasi dan penyimpanan data.
2. Menurut John F. Nash (1995:8), sistem informasi terdiri dari orang-orang, infrastruktur teknologi, media, prosedur, dan kontrol. Tujuannya adalah untuk mengelola komunikasi penting,

memproses transaksi rutin, dan membantu membuat keputusan untuk manajemen, pengguna internal, dan eksternal.

3. Menurut Whitten (2004), sistem informasi adalah kumpulan terpadu dari orang-orang, aktivitas, data, jaringan, dan teknologi yang mendukung operasi sehari-hari bisnis dan menyediakan informasi yang diperlukan untuk pemecahan masalah dan pengambilan keputusan.

Dengan demikian, sistem informasi dapat didefinisikan sebagai integrasi antara pengetahuan manusia dan teknologi untuk memenuhi kebutuhan informasi dalam pengambilan keputusan dan pemecahan masalah.

2.2.2 Website

Website adalah serangkaian halaman yang terintegrasi dalam domain tertentu dan dirancang untuk menyediakan informasi yang dapat diakses dan dilihat oleh pengguna internet melalui mesin pencari. Konten yang dapat disertakan dalam sebuah *website* termasuk teks, gambar, ilustrasi, dan video, yang dimaksudkan untuk memenuhi berbagai kebutuhan informasi.[7]

2.2.3 PHP (PHP: Hypertext Preprocessor)

Berbeda dengan *JavaScript* yang biasanya bekerja di sisi klien (di browser), *PHP* (*PHP: Preprocessor Hypertext*) adalah bahasa pemrograman yang ditujukan untuk dijalankan di sisi *server*. *Java*, *Perl*, dan *C* adalah bahasa pemrograman yang digunakan untuk membuat

PHP, dan semua pekerjaan dilakukan di sisi *server*, bukan di klien. Oleh karena itu, *PHP* disebut sebagai bahasa pemrograman sisi *server*. *PHP* adalah program *open source*, jadi pengguna dapat membuat dan mengubah kode dengan bebas. Memproses dan menghasilkan halaman *HTML* secara dinamis untuk setiap klien adalah fungsi utama *PHP*. [8]



Gambar 2.1 PHP (PHP: Hypertext Preprocessor)

2.2.4 XAMPP

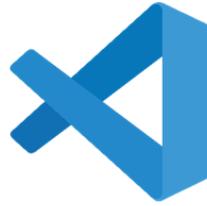
XAMPP adalah aplikasi *server web* lokal yang berfungsi sebagai pembuat *server* lokal di komputer dan memungkinkan membuat dan mengembangkan situs *web* secara *offline*. Aplikasi ini juga disebut sebagai "*localhost*" *XAMPP*. *XAMPP* merupakan singkatan dari *X* (*cross-platform*), *A* (*Apache*), *M* (*MySQL*), *P* (*PHP*), dan *P* (*Perl*). Huruf *X* menunjukkan bahwa aplikasi ini dapat digunakan secara lintas *platform*, artinya dapat berjalan di berbagai sistem operasi seperti *Windows*, *Linux*, *macOS*, dan *Solaris*. Huruf *A* menunjukkan *server web* bawaan *XAMPP*, dan huruf *M* menunjukkan aplikasi *server* basis data *MySQL* dan *MariaDB*. Sementara huruf *P* menunjukkan bahwa *XAMPP* menggunakan *PHP* dan *Perl* sebagai bahasa pemrograman. [7]



Gambar 2.2 XAMPP

2.2.5 Visual Studio Code

Visual Studio Code adalah sebuah aplikasi editor teks yang dikembangkan oleh *Microsoft*. Aplikasi ini diperkenalkan sebagai perangkat lunak gratis. Fungsi utama *Visual Studio Code* tidak hanya sebagai editor kode yang sederhana, tetapi juga didesain untuk mempermudah penulisan kode bagi pengembang yang bekerja di berbagai *platform* seperti *web*, *mobile*, dan *cloud*. *Visual Studio Code* mendukung berbagai sistem operasi, seperti *Windows*, *Linux*, dan *macOS*, yang memungkinkan pengembang bekerja tanpa terbatas pada lingkungan pengembangan yang tertutup. Tujuan utama dari *Visual Studio Code* bukanlah untuk menghasilkan *file* biner seperti *.exe* dan *.dll*, melainkan untuk menyederhanakan proses penulisan dan pengelolaan kode. Hal ini sangat krusial terutama dalam konteks pengembangan aplikasi yang bersifat lintas *platform*. Sebagai contoh, *ASP.NET Core 1.0* adalah teknologi yang bersifat lintas *platform* dan sumber terbuka yang memungkinkan pembuatan aplikasi *web* portabel yang dapat berjalan di *Windows*, *Linux*, dan *macOS*. [9]



Gambar 2.3 Visual Studio Code

2.2.6 Composer

Composer adalah alat untuk mengelola dependensi bahasa pemrograman PHP. Melalui *Composer*, pengembang dapat mendefinisikan dan mengelola pustaka-pustaka PHP yang dibutuhkan dalam proyek secara terstruktur dan efisien. *Composer* memungkinkan pengguna untuk menentukan dependensi yang spesifik, versi pustaka, dan sumber dari mana pustaka tersebut akan diunduh. Dengan menggunakan *file* konfigurasi *composer.json*, *Composer* dapat mengotomatiskan proses instalasi, pembaruan, dan manajemen dependensi, serta menyediakan *autoloading* yang mempermudah penggunaan pustaka-pustaka tersebut dalam kode *PHP* tanpa perlu memuat manual setiap *file* kelas secara terpisah.[10]



Gambar 2.4 Composer

2.2.7 *Laravel*

Laravel adalah sebuah *framework PHP open source* yang dibuat untuk mempermudah pengembangan aplikasi *web* dengan menyediakan struktur yang teratur dan berbagai fitur bawaan untuk mempercepat proses pengembangan. Taylor Otwell meluncurkan *framework* ini sebagai alternatif dari *CodeIgniter* pada 9 juni 2011. *Laravel* beroperasi di sisi *back-end* atau *server-side* dan mengadopsi pola desain *model-view-controller (MVC)*, yang membantu dalam mempercepat pengembangan aplikasi *web*. Dengan *MVC*, pengembangan dapat dipercepat karena memungkinkan pengembang untuk fokus secara terpisah pada setiap komponen, seperti *model* (yang mengelola basis data), *view* (yang menangani tampilan untuk pengguna), dan *controller* (yang menghubungkan antara model dan *view* berdasarkan permintaan pengguna).[11]



Gambar 2.5 Laravel

2.2.8 *Bootstrap*

Bootstrap adalah sebuah *framework HTML, CSS, dan JavaScript* yang digunakan untuk merancang dan mengembangkan antarmuka pengguna (UI) yang responsif dan menarik. *Framework* ini menyediakan berbagai komponen UI seperti tombol, formulir, navigasi, dan lainnya, serta sistem *grid* yang memungkinkan pengaturan tata

letak halaman *web* secara fleksibel dan konsisten di berbagai perangkat dan ukuran layar.[7]



Gambar 2.6 Bootstrap

2.2.9 UML (*Unified Modelling Language*)

Unified Modeling Language (UML), sebuah metode visual yang dibuat oleh *Object Management Group*, dimaksudkan untuk membantu dalam pengembangan sistem berbasis objek. *UML*, sering disebut sebagai *blueprint* perangkat lunak, pertama kali dirilis dalam versi 1.0 pada Januari 1997 dan digunakan sebagai bahasa standar untuk merencanakan, memvisualisasikan, dan mendokumentasikan sistem. [12]

Beberapa diagram fungsi *UML* yang digunakan untuk membangun sistem adalah sebagai berikut:

1. *Use Case Diagram*

Dalam *Unified Modeling Language (UML)*, *Use Case* diagram adalah jenis diagram yang menunjukkan bagaimana aktor berinteraksi dengan sistem dalam berbagai skenario penggunaan.

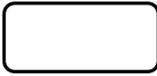
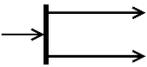
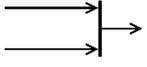
Tabel 2.1 Simbol *Use Case* Diagram

No.	Simbol	Keterangan
1.	<p><i>Actor</i></p> 	<i>Actor</i> adalah Perangkat, manusia yang berperan dalam sistem yang dibangun
2.	<p><i>Use Case</i></p> 	Hubungan <i>actor</i> dan sistem yang terdapat didalamnya
3.	<p><i>Association</i></p> 	Penghubung antara sistem dan <i>actor</i> dengan <i>use case</i>
4.	<p><i>Generalisation</i></p> 	Pengkhususan <i>actor</i> yang bisa bergabung dengan <i>use case</i>
5.	<p><<<i>Include</i>>></p> 	Untuk menunjukkan seluruh sistem yang dibangun dari fungsional sistem lainnya
6.	<p><<<i>Extend</i>>></p> 	Untuk menunjukkan bagian dari sistem diagram tambahan dengan yang lainnya jika kondisi terpenuhi

2. Activity Diagram

Dalam *Unified Modeling Language (UML)*, *Activity* diagram adalah jenis diagram yang menunjukkan proses-proses yang terjadi dalam sistem. Ini menunjukkan urutan proses secara visual, seringkali secara vertikal, untuk menunjukkan bagaimana aktivitas berhubungan satu sama lain dan berlangsung dalam sistem tersebut.

Tabel 2.2 Simbol Activity Diagram

No.	Simbol	Keterangan
1.	<p><i>Activity</i></p> 	Untuk memvisualisasikan UI dari setiap kelas yang saling terhubung
2.	<p><i>Action</i></p> 	Keadaan suatu sistem yang menggambarkan implemementasi dari suatu aksi
3.	<p><i>Start</i></p> 	Penggambaran dari suatu sistem untuk membentuk objek
4.	<p><i>End</i></p> 	Penggambaran dari suatu sistem untuk membentuk objek dan dihancurkan
5.	<p><i>State Transitio</i></p> 	Aksi selanjutnya
6.	<p><i>Fork</i></p> 	Cabang dari aliran pada diagram <i>activity</i> .
7.	<p><i>Join</i></p> 	Arah aliran diagram <i>activity</i> yang digabung

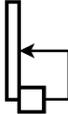
8.	<p><i>Decision</i></p> 	Opsi dalam sistem untuk membuat keputusan
9.	<p><i>Flow Final</i></p> 	Akhir dari aliran

3. *Sequence* Diagram

Dalam *Unified Modeling Language (UML)*, *Sequence* diagram adalah jenis diagram yang menunjukkan pesan yang dikirimkan antara objek dan urutan pelaksanaannya. Ini menunjukkan interaksi antar objek dalam sebuah skenario atau proses tertentu. Bagaimana objek berinteraksi satu sama lain sepanjang waktu dapat digambarkan dengan *Sequence* diagram.

Tabel 2.3 Simbol *Sequence* Diagram

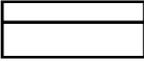
No.	Simbol	Keterangan
1.	<p><i>Entity Class</i></p> 	Gambaran sistem sebagai dasar untuk merancang basis data
2.	<p><i>Boundary Class</i></p> 	Menyusun interaksi antar lingkungan sistem

3.	<p><i>Control Class</i></p> 	Memiliki tanggung jawab terhadap kelas-kelas yang mengandung logika objek
4.	<p><i>Recursive</i></p> 	Pesan untuk dirinya
5.	<p><i>Activation</i></p> 	Menggambarkan durasi aktivasi sebuah operasi
6.	<p><i>Life Line</i></p> 	Komponen yang diindikasikan dengan garis

4. *Class Diagram*

Dalam *Unified Modeling Language (UML)*, *Class diagram* adalah jenis diagram yang digunakan untuk merancang struktur data dan perilaku objek dalam sistem perangkat lunak. Mengidentifikasi hubungan antara kelas-kelas dalam diagram ini memungkinkan pengembang untuk merencanakan implementasi sistem secara lebih terstruktur dan efisien.

Tabel 2.4 Simbol *Class Diagram*

No.	Simbol	Keterangan
1.	<p data-bbox="667 416 860 450"><i>Generalization</i></p> 	<p data-bbox="938 416 1351 595">Hubungan antara objek turunan (<i>descendant</i>) dan struktur data objek induk (<i>ancestor</i>)</p>
2.	<p data-bbox="651 640 876 674"><i>Nary Association</i></p> 	<p data-bbox="938 640 1351 819">Suatu Aksi untuk terhindar dari asosiasi yang memiliki lebih dari dua objek</p>
3.	<p data-bbox="727 864 799 898"><i>Class</i></p> 	<p data-bbox="938 887 1303 987">Kumpulan objek atribut dan operasi yang sama</p>
4.	<p data-bbox="687 1055 839 1088"><i>Realization</i></p> 	<p data-bbox="938 1088 1278 1122">Tindakan dari suatu objek</p>
5.	<p data-bbox="679 1211 847 1245"><i>Dependency</i></p> 	<p data-bbox="938 1211 1318 1391">Perubahan pada elemen yang memengaruhi elemen yang bergantung padanya</p>
6.	<p data-bbox="679 1435 839 1469"><i>Association</i></p> 	<p data-bbox="938 1435 1302 1536">Hubungan antara satu objek dengan objek lainnya</p>