

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Terkait

Dalam penelitian berjudul “Sistem Informasi Pariwisata Kabupaten Karanganyar Berbasis Website” (2021) merancang sistem informasi berbasis *website* untuk mengelola dan menyebarkan informasi pariwisata di Kabupaten Karanganyar. Sistem ini menggunakan *framework* Laravel karena memiliki keunggulan dalam pengelolaan data, kemudahan pengembangan, serta struktur kode yang lebih rapi dibandingkan *framework* lain seperti CodeIgniter. Hasil dari penelitian tersebut menunjukkan bahwa Laravel dapat meningkatkan pengelolaan informasi pariwisata, serta meningkatkan pengalaman pengguna yang lebih baik dan tampilan pengguna yang menarik[6].

Dalam penelitian berjudul “Pengembangan Sistem Informasi Promosi Pariwisata Berbasis Web di Kawasan Ciayumajakuning” (2024) membahas pentingnya sistem informasi dalam mendukung promosi pariwisata dan sektor ekonomi kreatif di wilayah Cirebon, Indramayu, Majalengka, dan Kuningan (Ciyumajakuning). untuk memperkenalkan destinasi wisata, kuliner, dan oleh-oleh khas daerah kepada masyarakat luas. Hasil dari penelitian menunjukkan bahwa penerapan sistem informasi berbasis *website* dapat meningkatkan keterjangkauan informasi wisata serta memudahkan wisatawan dalam merencanakan perjalanan mereka[7].

Dalam penelitian berjudul “Sistem Informasi Pariwisata Berbasis Web pada Kabupaten Nunukan” (2022) menggunakan *framework* Laravel untuk membangun sistem informasi yang mengelola data pariwisata, seperti informasi objek wisata, hotel, rumah makan, serta fitur pemesanan paket wisata. Laravel dipilih karena kemudahannya dalam pengembangan aplikasi *website* yang efisien, dengan fitur *built-in* yang menyederhanakan pengelolaan database dan memberikan antarmuka responsif. Hasil dari penelitian menunjukkan penggunaan Laravel meningkatkan efisiensi pengelolaan data, memungkinkan Dinas Pariwisata untuk menyebarluaskan informasi secara lebih efektif dan profesional[8].

Dalam penelitian berjudul “Perancangan Sistem Informasi Wisata Kopeng Berbasis Web dengan Metode Waterfall” (2022), dikembangkan sistem informasi pariwisata berbasis *website* untuk daerah Kopeng, Kabupaten Semarang. Penelitian ini dibuat karena kurangnya informasi terpadu tentang tempat wisata di Kopeng meskipun banyak situs *website* yang menyediakan informasi namun informasinya sangat minim dan terpisah-pisah. Hasil penelitian menunjukkan bahwa penggunaan *website* sistem informasi memberikan kemudahan kepada masyarakat untuk mengakses informasi lengkap tentang tempat wisata, penginapan, dan toko oleh-oleh di Kopeng dan sekitarnya[9].

Dalam penelitian berjudul “Sistem Informasi Objek Wisata Berbasis Web di Kabupaten Bolaang Mongondow Utara” (2022) membahas pentingnya pengembangan sistem informasi pariwisata berbasis *website*

sebagai upaya meningkatkan pengenalan objek wisata di wilayah tersebut. Penelitian dibuat karena kurangnya informasi terkait lokasi wisata yang ada di kabupaten tersebut meskipun memiliki berbagai jenis wisata alam yang menarik. Penelitian ini mengembangkan sistem informasi berbasis *website* untuk memberikan informasi lokasi objek wisata. Hasil penelitian menunjukkan bahwa pembangunan sistem informasi pariwisata berbasis *website* dapat berperan penting dalam mempromosikan objek wisata dan memberikan kontribusi bagi pendapatan asli daerah[10].

Dari beberapa penelitian-penelitian terdahulu, sistem informasi pariwisata berbasis *website* umumnya dikembangkan untuk Dinas Pariwisata atau promosi destinasi suatu wilayah. Namun, penelitian ini ditujukan untuk mendukung Badan Pusat Statistik (BPS) Kabupaten Tegal dalam pengelolaan data statistik pariwisata. Selain itu, penelitian sebelumnya tidak mengatasi masalah teknis seperti keterbatasan platform AppSheet yang hanya mendukung 10 pengguna. Oleh karena itu, penelitian ini mengusulkan solusi berbasis Laravel yang lebih skalabel, efisien, dan terintegrasi dengan kebutuhan.

## **2.2 Landasan Teori**

### **2.2.1 Sistem Informasi**

Sistem Informasi merupakan suatu sistem yang dirancang untuk mendukung operasional dalam suatu organisasi atau instansi, sehingga dapat menyajikan informasi kepada pihak lain[11]. Dalam

konteks pariwisata, sistem informasi berfungsi untuk mengumpulkan, mengolah dan menyimpan informasi terkait destinasi wisata, fasilitas, dan layanan lainnya.

### **2.2.2 Pariwisata**

Pariwisata merupakan berbagai kegiatan terkait pariwisata didukung oleh beragam fasilitas dan layanan yang ditawarkan oleh pemerintah setempat, komunitas bisnis, dan para pengusaha. Pariwisata memiliki peran penting dalam pembangunan ekonomi karena mampu mendorong pertumbuhan, meningkatkan devisa, dan membuka lapangan kerja. Sistem informasi pariwisata dapat membantu menyajikan data secara efisien untuk mendukung pengelolaan dan promosi destinasi wisata[12].

### **2.2.3 Website**

*Website* adalah kumpulan halaman yang menampilkan informasi dalam berbagai format, seperti teks, gambar diam atau bergerak, animasi, suara, dan kombinasi dari semuanya, baik bersifat statis maupun dinamis. *Website* berfungsi sebagai media informasi yang dapat diakses secara daring, serta memudahkan pengguna dalam mencari dan memperoleh informasi dengan cepat. Dalam sistem informasi pariwisata, website digunakan untuk menyajikan data destinasi, fasilitas, dan layanan yang tersedia secara interaktif dan terstruktur terhubung[13].

#### 2.2.4 *Framework*

Dalam pengembangan *software*, *framework* merupakan struktur yang digunakan dalam pengembangan *software* untuk mempercepat pembangunan aplikasi atau sistem[13]. *Framework* memiliki beberapa fungsi dalam pengembangan aplikasi atau sistem diantaranya mempercepat proses, menjaga konsistensi kode, meningkatkan kualitas sistem, dan mempermudah pemeliharaan aplikasi atau sistem[14].

#### 2.2.5 **Laravel**

Laravel dikembangkan dengan menggunakan bahasa pemrograman PHP. Laravel terus berkembang dengan komunitas dan pengguna yang semakin aktif. Laravel menerapkan konsep MVC (*Model, View, Controller*), yaitu pendekatan yang memisahkan manajemen data antara bagian antarmuka pengguna (*front-end*) dan bagian pemrosesan di sisi belakang (*back-end*)[14].

#### 2.2.6 **UML**

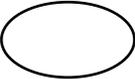
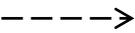
UML (*Unified Modeling Language*) merupakan alat yang dapat meningkatkan kualitas analisis dan desain sistem serta membantu menciptakan sistem informasi yang lebih signifikan[15]. Berikut beberapa jenis diagram yang digunakan dalam pengembangan sistem informasi

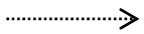
## 1. Use Case Diagram

Dalam pengembangan sistem, *Use Case Diagram* merupakan deskripsi dari langkah-langkah atau urutan yang menjelaskan apa yang dilakukan pengguna dengan sistem[16].

Simbol-simbol *Use Case Diagram* bisa di lihat pada Tabel 2.1

Tabel 2.1 Simbol Use Case Diagram

No	Simbol	Nama	Keterangan
1.		<i>System</i>	Menspesifikan paket yang menampilkan sistem secara terbatas.
2.		<i>Actor</i>	Mendefinisikan peran atau fungsi yang dilakukan aktor dalam hubungannya dengan <i>use case</i> sistem.
3.		<i>Use Case</i>	Menggambarkan interaksi antara aktor dan sistem yang menghasilkan suatu tujuan atau hasil tertentu.
4.		<i>Association</i>	Simbol penghubung antara objek satu dengan objek lainnya
5.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
6.		<i>Extend</i>	Menspesifikan bahwa <i>Use Case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.

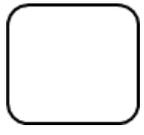
No	Simbol	Nama	Keterangan
7.		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya.
8.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi <i>element</i> yang bergantung padanya elemen yang tidak mandiri.

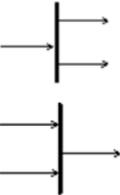
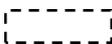
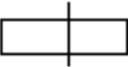
## 2. Activity Diagram

Diagram yang menggambarkan aliran kerja atau aktivitas dalam sebuah sistem pada perangkat lunak. *Activity Diagram* merepresentasikan proses yang dijalankan oleh sistem, bukan tindakan yang dilakukan oleh aktor, sehingga aktivitas yang ditampilkan adalah yang dilakukan oleh sistem itu sendiri[17].

Simbol dalam *activity diagram* bisa di lihat pada Tabel 2.2

Tabel 2.2 Simbol Activity Diagram

No.	Simbol	Nama	Keterangan
1.		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
2.		<i>Final Node</i>	Bagaimana objek dibentuk dan diakhiri.
3.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.

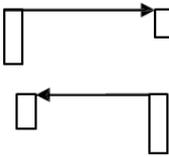
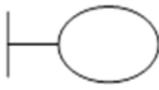
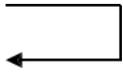
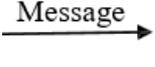
No.	Simbol	Nama	Keterangan
4.		<i>Action</i>	State dari sistem yang mencerminkan eksekusi suatu aksi.
5.		<i>Decision</i>	Pilihan untuk mengambil keputusan
6.		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.
7.		<i>Fork/Join</i>	Digunakan untuk mendefinisikan eksekusi beberapa aktivitas secara bersamaan atau untuk mengintegrasikan aktivitas paralel ke dalam satu alur.
8.		<i>Rake</i>	Menunjukkan adanya dekomposisi
9.		<i>Send</i>	Simbol pengiriman
10.		<i>Time</i>	waktu

### 3. Sequence Diagram

Diagram urutan (*sequence diagram*) merupakan diagram yang menjelaskan interaksi antar objek dan menunjukkan komunikasi antara objek-objek tersebut. *Sequence Diagram* digunakan untuk menggambarkan perilaku dalam suatu sistem serta menjelaskan bagaimana entitas dan sistem saling berinteraksi, termasuk pesan yang digunakan dalam interaksi tersebut. Pesan ditampilkan sesuai dengan urutan eksekusi[18].

Simbol *Sequence Diagram* bisa dilihat pada Tabel 2.3

Tabel 2.3 Simbol *Sequence diagram*

No	Simbol	Nama	Keterangan
1.		<i>Activation</i>	Sebagai sebuah objek yang akan melakukan sebuah aksi
2.		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi – informasi tentang aktifitas yang terjadi.
3.		<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem
4.		<i>Boundary Class</i>	Menggambarkan penggambaran dari form
5.		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan
6.		<i>Control Class</i>	Menggambarkan penghubung antara <i>Boundary</i> dengan tabel
7.		<i>Self Message</i>	Mengindikasikan komunikasi kembali kedalam sebuah objek itu sendiri
8.		<i>LifeLine</i>	Objek <i>entity</i> , antar muka yang saling berinteraksi.
9.		<i>Message</i>	Mengindikasikan komunikasi antara objek dengan objek

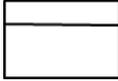
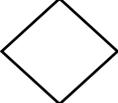
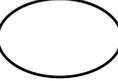
#### 4. *Class Diagram*

*Class Diagram* adalah jenis pemodelan yang digunakan untuk menggambarkan struktur basis data dan kelas objek dalam sebuah sistem. Tujuannya adalah untuk mendefinisikan kelas-kelas pada basis data. Diagram ini memperlihatkan bagaimana kelas-kelas saling berhubungan dan berinteraksi untuk

menciptakan kerangka dasar dari sebuah sistem software[19].

Simbol *Class* Diagram bisa dilihat pada Tabel 2.4

Tabel 2.4 Simbol Class Diagram

No.	Simbol	Nama	Keterangan
1.		<i>Class</i>	Himpunan dari objek - objek yang berbagi atribut serta operasi yang sama.
2.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
3.		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk.
4.		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
5.		<i>Colaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
6.		<i>Dependency</i>	Operasi yang benar - benar dilakukan oleh suatu objek.
7.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.