

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terdahulu**

Penelitian Terdahulu merupakan salah satu acuan untuk mendapatkan hasil penelitian yang optimal dan memperoleh perbandingan dari penelitian sebelumnya. Adapun hasil-hasil penelitian yang dijadikan ialah tidak terlepas dari topik penelitian mengenai aplikasi katalog wisata bersejarah pada Palembang *walking tour* berbasis *website*. Berikut adalah penelitian terdahulu terkait dengan penelitian yang dilakukan :

##### **2.1.1 Rancang Bangun Aplikasi Katalog Elektronik Pariwisata Berbasis Web untuk Banyak Pengelola Objek Wisata.**

Teknologi Informasi digunakan dalam pariwisata agar mempermudah mendapatkan Informasi mengenai wisata, meningkatkan kualitas pariwisata dan alat bantu untuk memenangkan persaingan bisnis pariwisata. Salah satu contoh teknologi Informasi adalah aplikasi katalog elektronik pariwisata. Tujuan dari penelitian ini adalah membangun aplikasi katalog elektronik pariwisata berbasis *web* yang dapat menyajikan Informasi pariwisata sehingga mudah diakses oleh para wisatawan dan pemilik usaha dapat berkontribusi dalam mengelola Informasi secara langsung. Metode yang digunakan yaitu Rational Unified Process, sebagai salah satu tahap pengembangan aplikasi dengan tahapan yang digunakan yaitu inception, elaboration, dan construction dengan menggunakan teknik pemodelan Unified Modeling Language, serta menggunakan metode *black box* dalam tahap pengujian. Hasil penelitian ini yaitu aplikasi katalog elektronik pariwisata berbasis *web* dapat mempermudah wisatawan mengetahui Informasi pariwisata secara lengkap mengenai tempat, sarana serta produk / layanan pariwisata yang ada di suatu daerah, dan dapat digunakan oleh pemilik usaha pariwisata sebagai media untuk membantu mempromosikan tempat usaha serta produk / layanan pariwisata yang dimilikinya (Nurfaizal dkk, 2021).

### **2.1.2 Perancangan dan Pembuatan Sistem Informasi Wisata Berbasis *Website* di Dinas Pariwisata dan Kebudayaan Kabupaten Pidie.**

Penelitian ini dilakukan dengan tujuan untuk merancang dan membangun sistem Informasi wisata yang dapat menyajikan dan memberikan Informasi wisata dan budaya yang ada di Kabupaten Pidie. Adapun latar belakang dari penelitian ini diangkat berdasarkan kebutuhan akan adanya sebuah sistem Informasi yang dapat memberikan Informasi menyeluruh secara cepat dan akurat, karena penyampaian Informasi mengenai wisata di Kabupaten Pidie untuk saat ini masih dengan brosur dengan kata lain penyampaian Informasi semacam ini belum cukup efektif dan belum cukup mampu menarik wisatawan untuk berkunjung ke Kabupaten Pidie. Dan dengan adanya sistem Informasi ini juga dapat mempermudah pemerintah Dinas pariwisata dan kebudayaan setempat dalam mengelola Informasi wisata dan kebudayaan di daerah tersebut. Penelitian ini menggunakan metode pengembangan sistem *waterfall* dan metode pengujian sistem menggunakan metode system usability scale (SUS). Hasil evaluasi penilaian akhir yang dilakukan terhadap sistem Informasi wisata dari responden mendapatkan skor 80,5 dan termasuk grade A berdasarkan grafik percentil rank SUS. Sehingga dapat digunakan dengan mudah oleh pengguna untuk mendapatkan layanan Informasi yang dibutuhkan (Haykal, 2020).

### **2.1.3 Sistem Informasi Wisata Pantai Berbasis *Web* Di Kabupaten Garut.**

Pariwisata merupakan salah satu tujuan yang paling diminati wisatawan saat ini, salah satunya adalah objek wisata pantai. Objek wisata pantai yang ada di kabupaten Garut adalah Pantai Rancabuaya dengan menyimpan begitu pesona pantai yang menawan namun permasalahan yang dihadapi adalah masih terbatasnya Informasi kepada wisatawan seputar objek wisata, informasi yang disampaikan saat ini dari mulut ke mulut orang yang pernah berkunjung kesana dan dari sebuah blog. Teknologi yang dapat digunakan untuk mempromosikan objek wisata pantai Rancabuaya dengan suatu aplikasi berupa Informasi dengan menggunakan teknologi *web* dengan tujuan penelitian ini adalah membuat sistem Informasi untuk menyampaikan Informasi kepada wisatawan yang ingin berkunjung ke objek wisata pantai yang berlokasi di kabupaten Garut. Rational

Unified *Process*(RUP) merupakan metodologi yang digunakan dalam merancang sistem Informasi tersebut. Dengan system Informasi ini nantinya dapat menyampaikan kepada masyarakat luas khususnya wisatawan mengenai fasilitas dan berapa harga yang harus dikeluarkan ketika akan berkunjung ke pantai Rancabuaya beserta fasilitas pendukung lainnya (Kuswara, 2019).

**Tabel 2. 1** Tabel Penelitian Terdahulu

No	Nama, Tahun, Judul Penelitian	Metode Penelitian	Hasil Penelitian
1.	Muhammad Fiqri Nurfaizal, Rinda Cahyana dan Eri Satria. 2021 Rancang Bangun Aplikasi Katalog Elektronik Pariwisata Berbasis <i>Web</i> untuk Banyak Pengelola Objek Wisata.	Penelitian ini dilakukan dengan menggunakan metode Rational Unified Process.	Membangun aplikasi katalog elektronik pariwisata berbasis <i>web</i> yang dapat menyajikan Informasi pariwisata sehingga mudah diakses oleh para wisatawan dan pemilik usaha dapat berkontribusi dalam mengelola Informasi secara langsung.
2.	Muhammad Haykal. 2020 Perancangan dan Pembuatan Sistem Informasi Wisata Berbasis <i>Website</i> di Dinas Pariwisata dan Kebudayaan Kabupaten Pidie	Penelitian ini menggunakan metode pengembangan sistem <i>waterfall</i>	Merancang dan membangun sistem Informasi wisata yang dapat menyajikan dan memberikan Informasi wisata dan budaya yang ada di Kabupaten Pidie.

3.	Ahmad Kuswara, Asep Deddy Supriatna dan Erwin Gunadhi. 2019 Sistem Informasi Wisata Pantai Berbasis <i>Web</i> Di Kabupaten Garut	Metodologi penelitian yang digunakan adalah Rational Unified <i>Process</i> (RUP)	Membuat sistem Informasi untuk menyampaikan Informasi kepada wisatawan yang ingin berkunjung ke objek wisata pantai yang berlokasi di kabupaten Garut.
----	--	--	--

## 2.2 Landasan Teori

Landasan teori adalah seperangkat definisi, konsep, proposisi yang telah disusun rapi dan sistematis tentang variable-variabel dalam sebuah penelitian. Landasan teori merupakan bagian dari penelitian itu sendiri. Landasan teori merupakan bagian dari penelitian yang memuat teori-teori dan hasil-hasil penelitian yang berasal dari studi kepustakaan yang memiliki fungsi sebagai kerangka teori untuk menyelesaikan pekerjaan penelitian (Ramdhan, M, 2021). Seorang peneliti harus menguasai teori-teori ilmiah sebagai dasar bagi argumentasi dalam menyusun pemikiran yang membuahkan hipotesis. Kerangka Pemikiran ini merupakan penjelasan sementara terhadap gejala-gejala yang menjadi objek permasalahan. Dari teori akan terbentuk kerangka pikir. Dari kerangka pikir tersebut kita dapat menyusun modal terlebih dahulu sebelum hipotesis. Kalau teori merupakan bentuk proposisi tentang pandangan mengenai fenomena pada realitas terkait objek penelitian. Adapun model merupakan gambaran abstraksi kenyataan yang merepresentasikan saran pemecahan penelitian (Rahim, 2020).

### 2.2.1 Aplikasi

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpaku pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang

diharapkan (Wiliani, 2018). Aplikasi memberikan kelancaran serta kenyamanan bagi pengguna dalam membantu aaktifitasnya. beberapa fungsi aplikasi diantaranya dibidang bisnis adalah membantu menghitung suatu keuntungan yang diperoleh. Karena dalam beberapa hal apabila kegiatan yang dilakukan secara manual tentunya akan mengandalkan waktu yang lama sehingga membutuhkan suatu sistem (Darnis & Khoirunisa, 2023).

### **2.2.2 Website**

Nofyat & Ambarita (2018) “*Website* adalah kumpulan halaman *web* yang dipublikasikan di Internet dengan nama domain/alamat URL (UniForm Resource Locator) yang dapat diakses oleh setiap pengguna Internet dengan mengetikkan alamatnya. Halaman *web* biasanya adalah dokumen yang ditulis dalam *Format* HTML (Hyper *Text* Markup Language) dan diakses melalui HTTP. HTTPS adalah protokol yang mentransmisikan berbagai Informasi dari *server* suatu *website* yang ditampilkan kepada pengguna atau pengguna melalui *browser*”.

Menurut Nevy (2022) “*Website* adalah kumpulan dari halaman–halaman situs, yang terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam World Wide *Web* ( WWW ) di dalam Internet. Sebuah halaman *web* biasanya berupa dokumen yang ditulis dalam *Format* Hyper *Text* Markup Language (HTML), yang selalu bisa diakses melalui HTTP, yaitu sebuah protokol yang menyampaikan Informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser*”.

Menurut Larno & Anggraini (2019) “*Website* atau sering disingkat *web*, adalah kumpulan halaman *web* dengan topik terkait, terkadang disertai dengan *file* gambar, *video*, atau jenis *file* lainnya. Situs *web* biasanya dihosting di *server web* yang dapat diakses melalui jaringan seperti Internet atau jaringan area lokal (LAN) di alamat Internet yang dikenal sebagai URL. Kombinasi dari semua halaman yang dapat diakses publik di Internet juga disebut sebagai WWW”.

### **2.2.3 Katalog Wisata**

Wisata adalah suatu kegiatan atau aktivitas yang dilakukan dengan tujuan untuk mengunjungi, mengeksplorasi, dan menikmati keindahan dan keunikan suatu

tempat atau daerah. Wisata dapat meliputi perjalanan ke tempat-tempat alam seperti pantai, gunung, dan taman nasional, atau ke tempat-tempat bersejarah, budaya, dan rekreasi seperti museum, kuil, dan taman hiburan (Juliati, 2023). Katalog adalah suatu daftar atau buku yang berisi Informasi tentang produk, layanan, atau Informasi lain yang disajikan secara teratur dan sistematis (Prayoga dkk, 2020).

Dalam konteks pariwisata, katalog wisata adalah daftar atau buku yang berisi Informasi tentang tempat wisata, penginapan, transportasi, dan layanan lain yang tersedia di suatu daerah atau destinasi wisata (Jocom dkk, 2021). Katalog wisata biasanya dilengkapi dengan gambar, deskripsi, dan Informasi lain yang berguna bagi pengunjung untuk memilih dan merencanakan perjalanan mereka. Bentuk penyajian katalog wisata dapat berupa buku cetak, brosur, atau dalam bentuk digital seperti situs *web* atau aplikasi. Dalam era digital, katalog wisata digital atau aplikasi dapat lebih mudah diakses dan diperbarui secara real-time, sehingga dapat memberikan Informasi yang lebih akurat dan up-to-date bagi pengunjung.

#### 2.2.4 XAMPP

XAMPP adalah *software* atau aplikasi komputer yang banyak digunakan dalam dunia *web developer* yang juga bisa dipelajari untuk membuat *website*. XAMPP adalah perangkat lunak berbasis *web server* bersifat *open source* (bebas) serta mendukung di berbagai sistem operasi seperti OS Linux, OS Windows, Mac OS, dan juga Solaris. XAMPP bisa dilakukan untuk menghemat anggaran karena mampu menggantikan peran *web hosting* dengan cara menyimpan *file website* ke dalam *hosting local* agar bisa dipanggil lewat *browser*.

Menurut Nevy (2022) Xampp adalah alat yang menggabungkan tiga aplikasi dalam satu paket, yaitu Apache, MySQL, dan PHPMyAdmin. Xampp membuat pekerjaan lebih mudah dengan memungkinkan mengunduh dan mengkonfigurasi ketiga aplikasi ini secara bersamaan dan otomatis. Sedangkan Supono & Putratama (2018) mendefinisikan bahwa XAMPP terdiri dari apache *web server*, MySQL, PHP, Perl, FTP *server* dan phpMyAdmin.

Menurut Novendri (2019) “XAMPP adalah perangkat lunak gratis yang mendukung banyak sistem operasi dan merupakan kumpulan dari beberapa

program. Seperti Apache, MYSQL, PHP dan Perl. Xampp adalah alat yang menyediakan paket perangkat lunak dalam satu paket. Paket Xampp termasuk Apache (*server web*), Mysql (*database*), PHP (*side scripting*), Perl, server FTP, PhpMyAdmin, dan beberapa pustaka pendukung lainnya.”.



Gambar 2. 1 Xampp

### 2.2.5 PHP MyAdmin

Menurut Suprianto (2021), PhpMyAdmin merupakan bagian untuk mengelola *database* MySQL yang ada dikomputer. Eko siswanto et al (2021), PhpMyAdmin adalah alat untuk manajemen visual yang mudah dari *database* MySQL dan *server* MySQL, tanpa perlu menulis query SQL setiap kali menjalankan perintah operasi *database*.

Menurut Nevy (2022), PhpMyadmin adalah sebuah aplikasi pemrograman yang digunakan untuk manajemen *database* melalui *browser (web)* untuk mengontrol data mereka dan isi *web* yang akan ditampilkan dalam sebuah *website* yang mereka buat tanpa harus menggunakan perintah (command) SQL. Menurut Wijaya (2020), “PHPMYAdmin adalah aplikasi perangkat lunak yang digunakan untuk mengelola *database* melalui *browser (web)* yang mengontrol data dan konten *web* untuk ditampilkan di situs *web* yang dibuat tanpa menggunakan perintah SQL”.



Gambar 2. 2 PHP MyAdmin

### 2.2.6 Database

Menurut Larno & Anggraini (2019), Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah untuk menyimpan data yang telah diproses dan memberikan Informasi pada saat dibutuhkan. Pada dasarnya basis data merupakan alat untuk menyimpan Informasi agar dapat diakses dengan mudah dan cepat. Nofyat & Ambarita (2018) “Basis data didefinisikan sebagai gudang data tempat data bersarang atau dikumpulkan. Prinsip utama *database* adalah manajemen Informasi, yang tujuan utamanya adalah fleksibilitas dan kecepatan dalam mengambil Informasi. Sasaran basis data meliputi efisiensi, yang mencakup kecepatan, ruang penyimpanan, dan akurasi, penanganan data dalam jumlah besar, berbagi, dan replikasi.”.

*Database* atau basis data merupakan sekumpulan data atau Informasi yang tersimpan secara sistematis. *Database* memiliki peran penting dalam perangkat untuk mengumpulkan Informasi, data, atau *file* secara terintegrasi. *Database* membuat penyimpanan dan pengelolaan data menjadi lebih efisien. *Database* berwujud tabel yang terdiri dari kolom dan baris yang memuat atribut dan nilai tertentu. Adapun jumlah kolom dan baris dalam suatu *database* tergantung pada jumlah kategori atau jenis Informasi yang perlu disimpan.

### 2.2.7 MySQL

Hermiati dkk (2021), MySQL adalah jenis *server* basis data yang terkenal. MySQL adalah jenis RDBMS (Relational *Database* Management System). MySQL mendukung bahasa pemrograman PHP, bahasa query terstruktur, karena ada beberapa aturan penggunaan SQL yang dibakukan oleh asosiasi yang disebut ANSI. MySQL adalah *server* RDBMS (*Relational Database Management System*). RDBMS adalah program yang memungkinkan pengguna basis data untuk membuat, mengelola, dan mengakses data dalam model relasional. Dengan demikian, tabel-tabel dalam *database* memiliki hubungan antara satu tabel dengan tabel lainnya.

Menurut Tampubolon (2018), MySQL adalah untuk semua pengguna *database* yang menggunakan Structured Query Language. MySQL di *server* client menyertakan daemon *server* MySQL di *server* dan berbagai program dan library yang berjalan di client. MySQL dapat menangani data yang cukup besar, yang



merupakan perangkat lunak sistem manajemen basis data (*Database Management System - DBMS*) yang sangat populer dalam pemrograman. MySQL merupakan *database* yang paling populer digunakan pada aplikasi *web* yang menggunakan *database* sebagai pengolah data.

### 2.2.8 *Visual studio code*

*Visual studio code* adalah kode *editor* sumber yang dikembangkan oleh Microsoft untuk Windows, Linux dan macOS. Ini termasuk dukungan untuk debugging, control git yang tertanam dan GitHub, penyorotan syntax, penyelesaian kode cerdas, snippet, dan refactoring kode. Ini sangat dapat disesuaikan, memungkinkan pengguna untuk mengubah tema, pintasan keyboard, preferensi, dan menginstal ekstensi yang menambah fungsionalitas tambahan (Agustini dan Kurniawan, 2019). 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, dst).

### 2.2.9 HTML

Menurut Sulistiono (2018), HTML (*HyperText Markup Language*) adalah bahasa markup yang digunakan untuk membuat halaman *web*, menampilkan berbagai Informasi seperti gambar, teks, *video* dan suara pada penjelajah *web* internet, yang ditulis dalam *file Format ASCII* untuk menghasilkan tampilan yang terintegrasi. Pada tahun 1980, fisikawan Tim Berners Lee, seorang kontraktor di CERN, mengusulkan dan merancang INQUIRE, sebuah sistem bagi peneliti CERN untuk menggunakan dan berbagi dokumen. Pada tahun 1989, Berners Lee menulis sebuah memo yang mengusulkan sebuah system *hypertext* berbasis internet.

Putra dkk (2019), menjelaskan bahwa *Hypertext Markup Language* adalah bahasa markup yang digunakan untuk membuat aspek online, berbagai Informasi saat menjelajah Internet dan *Format hypertext* sederhana yang ditulis dalam *file ASCII* untuk menghasilkan layar yang terintegrasi. *File* dimuat ke perangkat lunak komputer dan disimpan dalam *Format ASCII* sehingga dapat diubah menjadi halaman *web* menggunakan perintah HTML.

### 2.2.10 PHP

Menurut Putra dkk (2019), PHP adalah bahasa pemrograman yang digunakan untuk membuat pengkodean *website* dinamis yang memungkinkan kita untuk memperbarui *website* setiap saat. PHP adalah skrip sisi *server* yang dapat digunakan di berbagai sistem operasi seperti Linux, Windows, dll. Elgama (2020) menjelaskan bahwa PHP telah menjadi bahasa pemrograman *web* yang banyak digunakan untuk membuat halaman *web* dinamis. PHP awalnya dikembangkan pada tahun akhir 1994 oleh Rasmus Lerdorf, tetapi sekarang diambil oleh The PHP Grup. Awalnya PHP adalah singkatan dari Personal *Home* Page, tetapi di pengembangan diubah menjadi PHP: *hypertext* preprocessor. PHP menjadi bahasa pemrograman yang interpreter, dalam arti membaca setiap instruksi dan sintaks (*coding*) dengan membaca satu per satu atau baris baris kode program”.

Menurut Novendri (2019) “Bahasa pemrograman PHP adalah bahasa pemrograman untuk membangun *website* yang bersifat *server-side script*. PHP bersifat dinamis. PHP dapat digunakan di beberapa sistem operasi seperti Windows, Linux dan Mac Os”. PHP (Personal *Hypertext* Preprocessor) adalah sebuah bahasa pemrograman *server side scripting* yang bersifat *open source*. Sebagai sebuah *scripting language*, PHP menjalankan instruksi pemrograman saat proses *runtime*. Hasil dari instruksi tentu akan berbeda tergantung data yang diproses. PHP merupakan bahasa pemrograman *server-side*, maka *script* dari PHP nantinya akan diproses di *server*. Jenis *server* yang sering digunakan bersama dengan PHP antara lain Apache, Nginx, dan LiteSpeed.

### 2.2.11 CSS

Salah satu bahasa desain *web* yang dapat mengatur *Format* tampilan sebuah halaman *web* dengan perancangan desain *text* berupa *font* , color, margins, size dan lain-lain. Menurut Hasan (2020) “*Cascading Style Sheet* (CSS) adalah fitur yang diperkenalkan sejak HTML versi 4.0 yang menangani masalah tampilan HTML seperti jenis, *font* dan warna, posisi teks, border atau batas teks, warna *background*, dll. Penting untuk diperhatikan bagaimana CSS dan bahasa berbasis *web* lainnya dibuat untuk mempermudah pengelolaan, *pengeditan*, dan penyimpanan file”.

Menurut Tampubolon (2018) “CSS adalah bahasa desain *web* (*Style Sheet Language*) yang mengontrol *Format* tampilan halaman *web* yang ditulis dalam bahasa markup. Selain itu, CSS juga dapat menyesuaikan posisi, warna, dekorasi atau apapun yang berhubungan dengan tampilan dokumen HTML.

Menurut Mahdi (2021) “CSS kepanjangan dari *Cascading Style Sheet* adalah bahasa-bahasa yang merepresentasikan halaman *web* seperti warna, *Layout*, dan *font* . CSS dapat mengontrol ukuran gambar, warna, bagian tubuh teks, warna tabel, ukuran border, warna border, warna *hyperLink*, warna mouse over, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya”. CSS adalah bahasa gaya yang digunakan untuk mengatur tampilan dokumen. Dengan CSS memungkinkan kita untuk menampilkan halaman dengan *Format* yang berbeda (Sulistiono, 2018).

### **2.2.12 Bootstrap**

Bootstrap terdiri dari CSS dan HTML untuk menghasilkan *Grid*, *Layout*, *Typography*, *Tabel*, *Form*, *Navigation*, dan lain-lain. Di dalam Bootstrap juga sudah terdapat *JQuery plugins* untuk menghasilkan komponen UI yang cantik seperti *Transitions*, *Modal*, *Dropdown*, *Scrollspy*, *Tooltip*, *Tab*, *Popover*, *Alert*, *Button*, *Carousel* dan lain-lain. Menurut (Nugroho & Setiyawati, 2019), Bootstrap adalah *framework* CSS untuk membuat tampilan *web*. Bootstrap menyediakan *class* dan komponen yang sudah siap dipakai. Menurut Maulana (2021) “Bootstrap adalah *library* (pustaka / kumpulan fungsi-fungsi) dari *Framework* CSS yang dibuat khusus untuk bagian pengembangan *frontend* dari suatu *website*. Didalam *library* tersebut terdapat berbagai jenis *file* yang diantaranya HTML, CSS, dan Javascript”.

Menurut Tampubolon (2018) “Bootstrap adalah *library framework* CSS yang dibuat khusus untuk bagian pengembangan *front-end website*. Bootstrap juga merupakan salah satu *framework* HTML, CSS dan javascript yang paling populer di kalangan *web developer* yang digunakan untuk mengembangkan sebuah *website* yang *responsive*” .

Bootstrap adalah *framework web* development berbasis HTML, CSS, dan JavaScript yang dirancang untuk mempercepat proses pengembangan *web responsive* dan *mobile-first*.



**Gambar 2. 3** Bootstrap

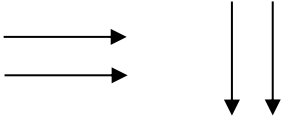
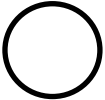
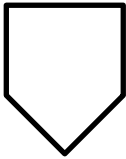

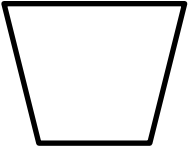
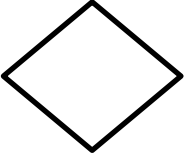
### **2.2.13 Javascript**

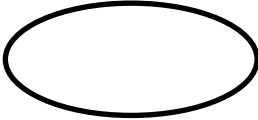
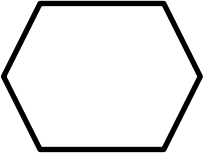

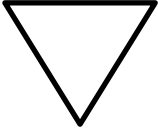


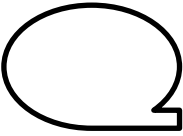
Menurut Sahi (2020), “Javascript adalah bahasa yang berupa kumpulan skrip, yang tugasnya dijalankan dalam dokumen HTML. Bahasa ini adalah bahasa skrip pertama di *web*. Bahasa ini merupakan bahasa pemrograman yang memberikan fungsionalitas tambahan pada HTML dengan memungkinkan eksekusi perintah di sisi pengguna, yaitu di sisi *browser*, bukan *server web*”. JavaScript adalah bahasa untuk membuat program yang digunakan untuk membuat dokumen HTML yang digunakan, yang ditampilkan di *browser* lebih interaktif. JavaScript menyediakan beberapa fungsionalitas ke halaman *web*, sehingga bisa menjadi program yang disajikan melalui antarmuka *web*. JavaScript adalah bahasa yang tidak memerlukan kompiler, cukup dengan interpreter (Hermiati dkk, 2021)




### **2.2.14 Flowchart**

*Flowchart* atau yang biasa disebut dengan diagram alir merupakan suatu jenis diagram yang mempresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem (Rosaly & Prasetyo, 2019). *Flowchart* merupakan alternatif lain untuk menggantikan algoritma maupun pseudokode. Seorang analis sistem menggunakan *flowchart* sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang akan dibangun ke *programmer*. Pada dasarnya *flowchart* digambarkan dengan menggunakan simbol-simbol yang mewakili suatu proses tertentu. Untuk menghubungkan satu proses ke proses selanjutnya digambarkan dengan menggunakan garis penghubung. Dengan adanya *flowchart*, setiap urutan proses dapat digambarkan menjadi lebih jelas.

Tabel 2. 2 Simbol *Flowchart*

NO	SIMBOL	KETERANGAN
1.		Simbol arus / <i>flow</i> , berfungsi untuk menyatakan jalannya arus suatu proses
2.		Simbol <i>connector</i> , berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
3.		Simbol <i>offline connector</i> , berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
4.		Simbol <i>process</i> , berfungsi untuk menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
5.		Simbol manual, berfungsi untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
6.		Simbol <i>decision</i> , berfungsi untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya/tidak

7.		Simbol terminal, berfungsi untuk menyatakan permulaan atau akhir suatu program
8.		Simbol <i>predefined process</i> , berfungsi untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
9.		Simbol <i>keying operation</i> , berfungsi untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i> .
10.		Simbol <i>offline-storage</i> , berfungsi untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu
11.		Simbol manual <i>input</i> , berfungsi untuk memasukkan data secara manual dengan menggunakan <i>online keyboard</i>
12.		Simbol <i>input/output</i> , berfungsi untuk menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya
13.		Simbol <i>magnetic tape</i> , berfungsi untuk menyatakan <i>input</i> berasal dari pita magnetis atau <i>output</i> disimpan ke pita magnetis

14.		Simbol <i>disk storage</i> , berfungsi untuk menyatakan <i>input</i> berasal dari disk atau <i>output</i> disimpan ke <i>disk</i>
15.		Simbol <i>document</i> , berfungsi untuk mencetak keluaran dalam bentuk dokumen (melalui <i>printer</i> )
16.		Simbol <i>punched card</i> , berfungsi untuk menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu


### 2.2.15 Use case Diagram




Komponen pembentuk diagram *use case* di dalam jurnal (Yunita, 2018) adalah :

- a. Aktkor (*actor*) menggambarkan pihak-pihak yang berperan dalam sistem.
- b. *Use case*, aktivitas/sarana yang disiapkan oleh bisnis/sistem.

Diagram *Use case* menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. Pada *Use case diagram*, ada beberapa simbol sebagai berikut :

**Tabel 2. 3** Daftar Simbol *Use case diagram*

No	Simbol	Keterangan
1.		<i>Actor</i> , mewakili peran orang, sistem lain, atau alat ketika berkomunikasi atau berinteraksi dengan <i>use case</i> .


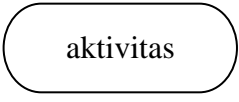
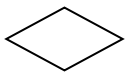


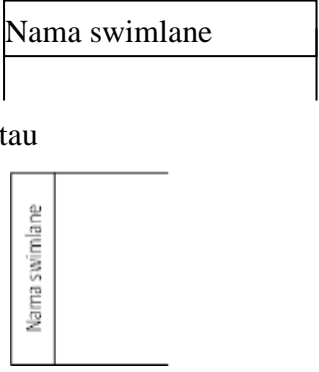

2.		<i>Use case</i> , adalah sebuah aksi atau fungsi yang dilakukan oleh sistem.
3.		<i>Association</i> , hubungan antara <i>actor</i> atau objek lain dengan <i>use case</i> tapi tidak mengendalikan <i>use case</i> tersebut.
4.		<i>Generalization</i> , menunjukkan spesialisasi <i>actor</i> untuk dapat berpartisipasi dengan <i>use case</i> .
5.	<code>&lt;&lt;extend&gt; &gt;</code>	<i>Extend</i> , menunjukkan bahwa sebuah <i>use case</i> opsional tergantung pada <i>use case</i> lainnya jika suatu kondisi terpenuhi.
6.	<code>&lt;&lt;include&gt; &gt;</code>	<i>Include</i> , menunjukkan bahwa sebuah <i>use case</i> membutuhkan aksi dari <i>use case</i> lain untuk dapat digunakan.

### 2.2.16 Activity Diagram

Menurut Sucipto di dalam jurnal (Listiono & Sintaro, 2021) diagram aktivitas menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *use case* atau iteraksi. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada *business modeling* untuk memperlihatkan urutan aktivitas proses bisnis. Struktur diagram ini mirip *flowchart* atau *data flow diagram* pada perancangan terstruktur simbol dalam *activity diagram*.

**Tabel 2. 4** Activity diagram



No	Simbol	Deskripsi
1.	Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane  atau 	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

### 2.2.17 Class Diagram

*Class* di dalam jurnal (Prayuda, 2019) adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan

deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok:

1. Nama (dan *stereotype*)
2. Atribut
3. Metode

Atribut dan metoda dapat memiliki salah satu sifat berikut:

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
3. *Public*, dapat dipanggil oleh siapa saja.

*Class* dapat merupakan implementasi dari sebuah *Interface* , yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus di implementasikan dahulu menjadi sebuah *class*. Dengan demikian *Interface* mendukung resolusi metoda pada saat *runtime*. Sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package*.

### 2.2.18 *Entity Relationship Diagram*

Pengertian ERD Menurut Sukamto dan Shalahuddin (2018:50), menyatakan bahwa ERD merupakan “pemodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basis data relasional. Simbol-simbol dalam ERD (*Entity Relationship Diagram*) adalah sebagai berikut:

- a. Entitas: suatu yang nyata atau abstrak yang mempunyai karakteristik dimana kita akan menyimpan data.
- b. Atribut: ciri umum semua atau sebagian besar instansi pada entitas tertentu.
- c. Relasi: hubungan alamiah yang terjadi antara satu atau lebih entitas.
- d. *Link*: garis penghubung atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi.

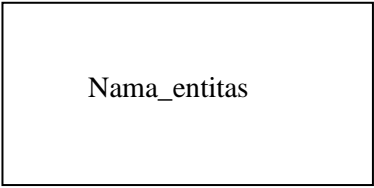
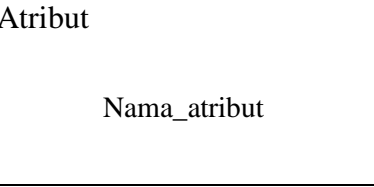
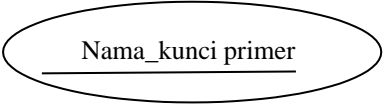
Kardinalitas Relasi :

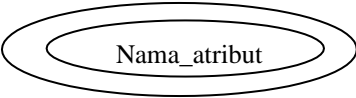
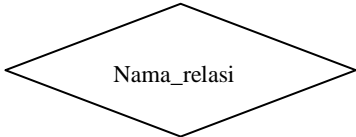
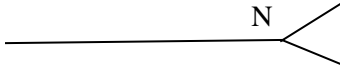
- a. Satu ke satu (*One to One*) Setiap elemen dari Entitas A berhubungan paling banyak dengan elemen pada Entitas B. Demikian juga sebaliknya setiap elemen B berhubungan paling banyak satu elemen pada Entitas A.

- b. Satu ke banyak (*One to Many*) Setiap elemen dari Entitas A berhubungan dengan maksimal banyak elemen pada Entitas B. Dan sebaliknya setiap elemen dari Entitas B berhubungan dengan paling banyak satu elemen di Entitas A.
- c. Banyak ke satu (*Many to One*) Setiap elemen dari Entitas A berhubungan paling banyak dengan satu elemen pada Entitas B. Dan sebaliknya setiap elemen dari Entitas B berhubungan dengan maksimal banyak elemen di entitas A.
- d. Banyak ke banyak (*Many to Many*) Setiap elemen dari Entitas A berhubungan maksimal banyak elemen pada Entitas B demikian sebaliknya.

Berikut merupakan simbol-simbol yang digunakan pada ERD dengan notasi Chen:

**Tabel 1.1 Simbol-simbol pada ERD (notasi Chen)**

No	Simbol	Deskripsi
1.	Entitas / <i>Entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal table pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi computer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel
2.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
3.	Atribut Kunci Primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).

4.	Atribut Multinilai / Multivalue 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
6.	Asosiasi / Association 	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah. Pemakaian kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B.

Sumber: Sukamto dan Shalahuddin (2018:50-51)

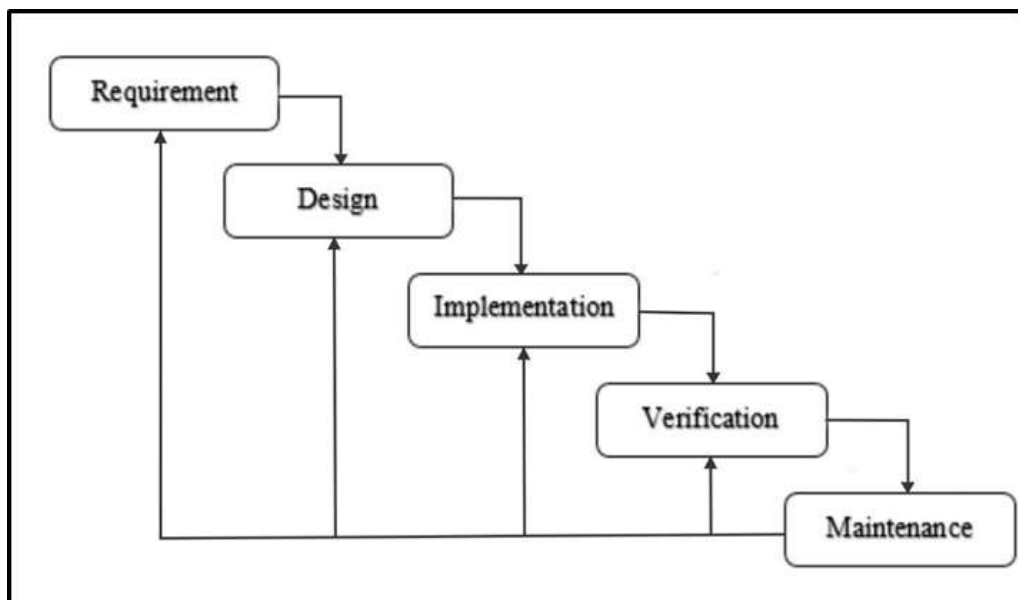
### 2.3 Metode Pengembangan Sistem

Metode *Waterfall* adalah salah satu model SDLC (*System Development Life Cycle*) yang sering digunakan dalam pengembangan sistem Informasi atau perangkat lunak. Model *Waterfall* menggunakan pendekatan sistematis dan berurutan. Tahapan model *Waterfall* antara lain *requirement*, *design*, *implementation*, *verification*, dan *maintenance* (Wahid, 2020). Kelebihan menggunakan metode *Waterfall* adalah kualitas dari sistem yang dihasilkan akan baik karena pelaksanaannya dilakukan secara bertahap, sementara untuk kekurangannya adalah proses pengembangan sistem membutuhkan waktu yang lama sehingga biaya yang diperlukan juga mahal. Adapun kelebihan menggunakan metode *waterfall* diantaranya:

1. Urutan proses pengerjaan menggunakan metode ini menjadi lebih teratur

dari satu tahap ke tahap yang selanjutnya.

2. Dari sisi *user* juga lebih menguntungkan karena dapat merencanakan dan menyiapkan seluruh kebutuhan data dan proses yang akan diperlukan.
3. Jadwal menjadi lebih menentu karena jadwal setiap proses dapat ditentukan secara pasti. Dengan adanya urutan yang pasti, dapat dilihat pula progress untuk setiap tahap secara pasti.



**Gambar 2. 4** Metode *Waterfall*

### 2.3.1 *Requirement*

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

### 2.3.2 *Design*

Pada tahap ini, pengembang membuat desain sistem yang dapat membantu menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

### **2.3.3 Implementation**

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit testing.

### **2.3.4 Verification**

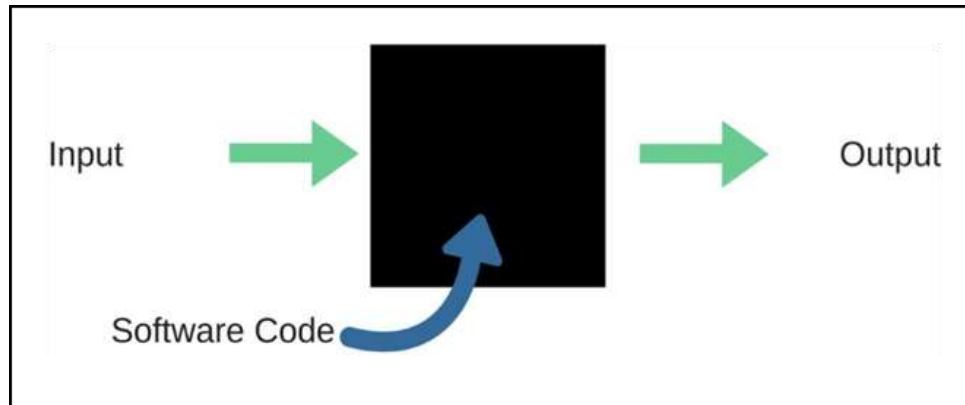
Pada tahap ini, sistem dilakukan verifikasi dan pengujian apakah sistem sepenuhnya atau sebagian memenuhi persyaratan sistem, pengujian dapat dikategorikan ke dalam unit testing (dilakukan pada modul tertentu kode), sistem pengujian (untuk melihat bagaimana sistem bereaksi ketika semua modul yang terintegrasi) dan penerimaan pengujian (dilakukan dengan atau nama pelanggan untuk melihat apakah semua kebutuhan pelanggan puas).

### **2.3.5 Maintenance**

Ini adalah tahap akhir dari metode *waterfall*. Perangkat lunak yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

## **2.4 Blackbox testing**

*Black-Box Testing* merupakan Teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. *Blackbox testing* bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya difokuskan pada Informasi domain. *Black box testing* memungkinkan pengembang *software* untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program (Jaya, 2018).



**Gambar 2. 5** Sistem Pengujian *Black box testing*

Pengujian *black box* hanya melibatkan *database* antara *input* dan *output*. Pengujian ini menangani kebutuhan pelanggan dari *input* yang *valid* maupun tidak *valid*. Pengujian *black box* memiliki peran penting dalam pengujian perangkat lunak yaitu untuk memvalidasi fungsi keseluruhan sistem apakah telah bekerja dengan baik (Parlika dkk, 2020).