

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Pada bagian ini akan dijelaskan hasil-hasil penelitian terdahulu yang bisa dijadikan acuan dalam topik penelitian ini. Penelitian terdahulu telah dipilih sesuai dengan metode yang digunakan dalam penelitian ini, sehingga diharapkan mampu menjelaskan maupun memberikan referensi bagi penulis dalam menyelesaikan penelitian ini. Berikut dijelaskan beberapa penelitian terdahulu yang telah dipilih.

Pertama, penelitian oleh Dutio Alpamara dan Yeka Hendriyani yang berjudul “Rancang Bangun Sistem Informasi Ekowisata *AOFarm (Atsiri Organic Farm)* Berbasis *Web* Menggunakan Metode *Waterfall*”. *AOFarm* masih belum didukung dengan sebuah sistem informasi yang dapat memperluas dan membantu dalam hal pengelolaan ekowisata ini. Oleh karena itu, dibuatkanlah sebuah sistem informasi yang dapat memperluas dan membantu pengelolaan untuk kemajuan dari ekowisata *AOFarm* itu sendiri. Metode yang digunakan untuk merancang sistem ini adalah metode *Waterfall*. Tahapan pengembangan siklus hidup pengembangan perangkat lunak dengan pendekatan waterfall adalah analisis kebutuhan, perancangan sistem, implementasi, pengujian dan analisis, dan terakhir pengembangan produk, penarikan kesimpulan, dan saran. Berdasarkan hasil pembahasan, Sistem informasi yang dibangun dapat menyajikan informasi ekowisata yang dibutuhkan oleh pengguna, seperti data klien, data order, data kunjungan, data direktur, data admin, data petugas, data user, dan transaksi[1].

Kedua, penelitian oleh Caroline Christine Putri Tolampi dan Evi Maria yang berjudul “Perancangan Sistem Informasi *Tracking* dan *Monitoring* Posisi Barang Menggunakan Metode *Waterfall*”. Hasil wawancara dengan Kepala Bagian *Shipping & Traffic* PT. Vale Indonesia diketahui bahwa bagian ini seringkali mendapat keluhan dari *user* karena tidak dapat memberikan informasi lokasi barang yang telah dipesan, padahal barang yang di pesan *user* adalah barang yang bersifat mendesak untuk kegiatan operasional perusahaan. Dalam penelitian ini, metode pengembangan sistem yang digunakan adalah metode *Waterfall*. Hasil dari penelitian ini yaitu sistem informasi *tracking* dan *monitoring* posisi barang yang siap untuk diimplementasikan di PT. Vale Indonesia menggunakan metode *Waterfall*. Sistem informasi ini diharapkan dapat menjadi solusi atas permasalahan efisiensi waktu, karena proses pelacakan posisi/lokasi terkini barang dapat dilakukan secara real-time oleh *user* yang meminta barang untuk mendukung kegiatan operasional[2].

Ketiga, penelitian oleh Suprayanta, Dedy Supriadi dan Beni Susanto yang berjudul “Perancangan Sistem Informasi Penggajian Karyawan Dengan metode *Waterfall*” pada CV. Popeye Chiken. Memiliki banyak outlet dan terus bertambahnya cabang maka karyawan juga semakin banyak. Setiap bulan perusahaan selalu melakukan penggajian kepada semua karyawannya. Permasalahannya adalah ada banyak karyawan di berbagai outlet sehingga perhitungan gaji lama, informasi yang disajikan terkadang tidak akurat dan proses auditnya sulit. Metode untuk pengembangan sistem yang di pilih yaitu metode *waterfall*. Sistem usulan masih berupa rancangan, jadi belum dibuatkan programnya, belum bisa langsung di jalankan

dan masih perlu di telaah lebih lanjut sehingga bisa lebih baik lagi terutama dalam mensikapi perkembangan perusahaan yang makin banyak dan tersebar di berbagai daerah[3].

Keempat, penelitian oleh Edo Arribe dkk. yang berjudul “Perancangan Sistem Informasi Absensi Menggunakan Metode *Waterfall*: Studi Kasus PT. Nielsen Company”. Penelitian ini berfokus pada pengembangan sistem informasi yang efektif di PT. Nielsen Company, dengan menyoroti masalah pengelolaan absensi karyawan. Metode *Waterfall* digunakan untuk memandu analisis dan perancangan sistem, menawarkan pendekatan sistematis dan berurutan. Masalah utama yang diidentifikasi adalah kebutuhan untuk mengotomatisasi dan memperbaiki proses absensi karyawan, yang sebelumnya dilakukan secara manual dan rentan terhadap ketidakakuratan dan penipuan. Hasil dari penelitian ini menunjukkan bahwa implementasi fitur kehadiran dalam aplikasi OPS PEDIA berhasil mengatasi masalah ini. Fitur baru ini tidak hanya meningkatkan efisiensi dan akurasi dalam pencatatan absensi tetapi juga mengurangi biaya operasional yang terkait dan mengeliminasi potensi penipuan absensi[4].

Kelima, Penelitian oleh Rizal Arianto dkk. yang berjudul “Pengembangan aplikasi sistem informasi inventory pada CV. wijaya las kediri menggunakan model *waterfall*”. Dari hasil penelitian yang dilakukan rizal arianto pada CV. Wijaya Las Kediri pengecekan persediaan material barang bersifat manual dari masalah tersebut maka dibutuhkan sistem informasi inventor. Untuk Mengembangkan Sistem Informasi Inventori kami menggunakan metode *Waterfall*. Hasil pengujian didapatkan bahwa metode *Waterfall* dapat digunakan untuk membuat Sistem Informasi Inventori

sehingga membantu pemilik maupun staff perusahaan dalam meningkatkan pelayanan dan penjualan, mempermudah pekerjaan *staff* dalam mencatat data keluar masuk barang, dan pembuatan laporan[5].

Keenam, penelitian oleh Fajar Mahardika, Ahmad Zulfan dan Suseno yang berjudul “Implementasi Metode *Waterfall* pada Sistem Informasi Kepegawaian Berbasis *Web*”. Penyimpanan data kepegawain yang masih dilakukan menggunakan aplikasi *Microsoft Office Excel*, maka data-data tersebut tidak disimpan dalam *database*. Metode pengembangan sistem yaitu *waterfall*. Dengan hasil yang didapatkan Sistem informasi kepegawaian saat ini memiliki fitur dasar untuk mengelola data pegawai, termasuk informasi pribadi. Sistem ini telah memberikan manfaat dalam mengurangi pekerjaan administratif dan meningkatkan efisiensi dalam pengelolaan data pegawai[6].

Ketujuh, penelitian oleh Nugraha dkk., yang berjudul “Sistem informasi profil perusahaan berbasis *website* menggunakan *laravel 8*”. Tujuan utama dari penelitian ini adalah merancang dan mengimplementasikan aplikasi *web* yang memungkinkan perusahaan menyajikan informasi secara efektif dan interaktif kepada publik. Metode penelitian yang digunakan adalah metode *waterfall*, yang meliputi tahap analisis, desain, implementasi, pengujian, dan pemeliharaan. Hasil penelitian menunjukkan bahwa penggunaan *Laravel 8* dalam pengembangan sistem informasi profil perusahaan berbasis *website* memberikan solusi yang efisien dan efektif untuk memenuhi kebutuhan komunikasi dan informasi perusahaan di era digital[7].

2.2 Landasan Teori

2.2.1 PT. Lentera Yaffa Persada

PT. Lentera Yaffa Persada merupakan perusahaan swasta nasional yang bergerak dibidang konstruksi bangunan, jaringan telekomunikasi, dan maintenance jaringan telekomunikasi. Perusahaan ini berdomisili di Jalan Poros Makale – Rantepao, Kel. Sarira, Kec. Makale Utara - Tana Toraja, Sulawesi Selatan dan Jalan Ampea Lr. 19 No. 23 B, Rt. 03 Rw. 06 Kel. Koja, Kec. Koja, Jakarta Utara.

Di bidang Konstruksi Bangunan, PT. Lentera Yaffa Persada menyediakan jasa pembangunan dan renovasi rumah, restoran, lanskap / taman, desain arsitektur, struktural dan lanskap perhitungan RAB dan mekanikal elektrik dan plumbing. Di bidang Jaringan Telekomunikasi, PT. Lentera Yaffa Persada menyediakan jasa konstruksi jaringan FO, jasa perawatan jaringan FO dan jasa pemasangan kabel akses (IKR).

2.2.2 Definisi Website

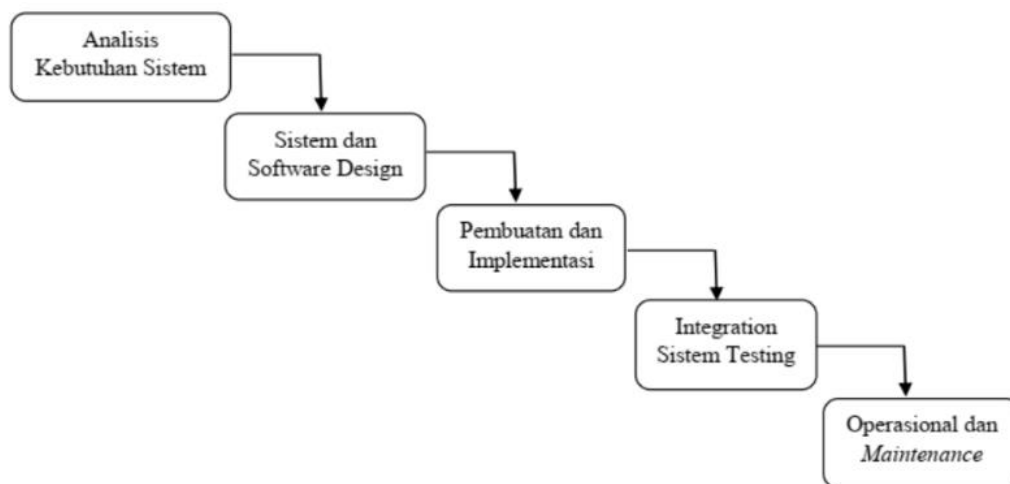
Website dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi *internet* sehingga dapat diakses dan dilihat oleh semua orang di seluruh dunia[8]. Dengan memanfaatkan *website*, PT. Lentera Yaffa Persada dapat mengoptimalkan penyampaian informasi tentang profil perusahaan, layanan, portofolio proyek, dan kontak yang dapat dihubungi sehingga meningkatkan kredibilitas dan daya saing perusahaan di era digital.

2.2.3 Definisi Sistem Informasi

Sistem menurut arti kata adalah kesatuan atau kumpulan dari elemen-elemen atau komponen-komponen atau subsistem-subsistem yang saling berinteraksi untuk mencapai tujuan tertentu. Informasi adalah sekumpulan data fakta yang diorganisasi atau diolah dengan cara tertentu sehingga mempunyai arti bagi penerima[9].

2.2.4 Metode *Waterfall*

Metode *Waterfall* adalah metode pengembangan sistem yang terstruktur di mana setiap tahapan dilakukan secara bertahap dan tidak boleh dilanjutkan sampai tahapan sebelumnya selesai. Metode ini memiliki beberapa keunggulan, termasuk membuat proses perancangan sistem lebih mudah karena tahapan-tahap ini harus dilakukan secara bertahap sampai dengan selesai sehingga proses penelitian tidak terganggu. Gambar berikut menunjukkan rute pengembangan sistem informasi pada penelitian ini[10].



Gambar 2. 1 Tahapan pengembangan dengan metode *waterfall*

1. Analisis Kebutuhan

Tahapan ini berfokus pada proses mengumpulkan informasi tentang apa saja yang diharapkan oleh pengguna dari sistem yang akan dibangun. Informasi tersebut diperoleh melalui komunikasi langsung, observasi, maupun dokumen yang berkaitan. Hasil dari tahap ini menjadi dasar utama bagi langkah-langkah selanjutnya.

2. Desain Sistem

Setelah kebutuhan sistem dipahami, tahap berikutnya adalah merancang solusi teknis. Di sini, peneliti menentukan struktur sistem, arsitektur data, dan rancangan antarmuka. Semua itu disusun secara detail sebagai cetak biru sebelum sistem dikembangkan.

3. Implementasi

Pada tahap ini, desain yang telah disiapkan mulai diwujudkan menjadi program nyata. Pengembang menulis kode, mengembangkan fitur, dan mengintegrasikan komponen sistem agar bisa berjalan sebagaimana direncanakan.

4. Pengujian

Sebelum sistem digunakan secara luas, tahap pengujian dilakukan untuk memastikan semua fungsinya berjalan baik dan bebas dari kesalahan. Proses ini penting agar sistem benar-benar stabil, aman, dan sesuai harapan pengguna.

5. Penerapan program dan pemeliharaan

Setelah sistem selesai diuji dan dinyatakan siap, sistem mulai dijalankan di lingkungan nyata dan digunakan oleh pengguna. Proses ini dikenal sebagai penerapan atau implementasi langsung. Namun, pekerjaan tidak berhenti di situ. Seiring waktu,

bisa saja ditemukan kekurangan, kesalahan, atau kebutuhan tambahan dari pengguna. Oleh karena itu, pemeliharaan dilakukan untuk memastikan sistem tetap berfungsi optimal, menyesuaikan dengan perubahan, serta memperbaiki masalah yang mungkin muncul setelah sistem digunakan secara aktif.

2.2.5 Bahasa pemrograman *PHP*

PHP (*Hypertext Preprocessor*) adalah bahasa berbentuk *script* yang ditempatkan dalam *server* dan diproses *server*. Selain itu juga *PHP* merupakan salah satu dari sekian banyak bahasa pemrograman *HTML* (*Hypertext Markup Language*). Dibuat oleh Rasmus Lerdorf diawali dengan membuatnya sebagai *Personal Project* dan disempurnakan oleh *Group Six of Developer* dan lahir kembali dengan nama *PHP* [11].

2.2.6 *Framework*

Framework atau dalam bahasa Indonesia kerangka kerja dapat diartikan sebagai sekumpulan fungsi/prosedur dan kelas-kelas yang mempunyai tujuan tertentu dan siap digunakan untuk mempermudah dan mempercepat pekerjaan *programmer* tanpa harus membuat fungsi atau kelas tertentu[12].

2.2.7 *Framework Laravel*

Laravel adalah kerangka kerja yang dibangun dengan basis bahasa pemrograman *PHP*. *Laravel* memiliki komunitas dan pengguna yang terus berkembang hingga sekarang, *Laravel* sendiri tidak terlepas dari konsep *MVC* (*Model, View, Controller*) dimana merupakan sebuah konsep modern yang memisahkan bagian tampilan (*front-end*) dan juga bagian pengelolaan data atau biasa

disebut controller (*back-end*). Ada pula beberapa kelebihan yang dimiliki oleh *Laravel* diantaranya yaitu, *Laravel* memiliki fitur untuk mengatur rute atau disebut dengan *routing*, fitur ini akan mengatur rute *request* yang ditentukan berdasarkan kebutuhannya. Selanjutnya *Laravel* juga memiliki *query builder* dan *ORM* yang dapat memberikan kemudahan dalam melakukan operasi *database* dan mendukung berbagai *database*. lalu, *Laravel* juga sudah terhubung dengan *composer* yang tentu akan mempermudah pengembang ketika membutuhkan *modul* atau *library* tambahan dan *composer* dapat melakukan pembaruan pada modul tersebut. *Laravel* juga dikenal memiliki komunitas yang sangat aktif sehingga *Laravel* sendiri memiliki banyak *library* tambahan yang dapat di *install* dan disesuaikan dengan kebutuhan aplikasi[13].

2.2.8 Xampp

XAMPP adalah perangkat lunak *open source*, yang mendukung untuk banyak sistem operasi, yang merupakan kompilasi dari beberapa program. Fungsi *XAMPP* sendiri adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri beberapa program antara lain : *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl* [14].

2.2.9 Visual Studio Code

Visual Studio Code adalah kode editor sumber yang dikembangkan oleh *Microsoft* untuk *Windows*, *Linux* dan *mac OS*. *Visual studio Code* memudahkan dalam penulisan kode yang mendukung beberapa jenis bahasa pemrograman yang digunakan dan memberi variasi warna sesuai dengan fungsi dalam rangkaian kode tersebut. Selain itu, fitur lainnya adalah kemampuan untuk menambah ekstensi dimana para

pengembang dapat menambah ekstensi untuk menambah fitur yang tidak ada di *Visual Studio Code*[15].

2.2.10 Mysql

MySQL adalah sistem manajemen database SQL yang sifatnya *open source* (terbuka) dan paling banyak digunakan saat ini. Sistem database *MySQL* mampu mendukung beberapa fitur seperti *multi threaded*, *multi-user*, dan *SQL database management system (DBMS)*[16].

2.2.11 Framework Bootstrap

Bootstrap adalah salah satu jenis *framework* gabungan dari *CSS* dan *JavaScript* yang ditawarkan sebagai alternatif diantara *framework* lainnya dimana awal *framework* ini dikembangkan oleh Mark Otto dan Jacob Thornton di kantor *Twitter* dengan maksud untuk menghadirkan konsistensi terhadap tahap *interface development* dalam membangun sebuah *website*. Saat ini *bootstrap* sudah menjadi aplikasi yang *open-source* dan mendukung platform seperti *HTML5* dan *CSS3*. Dalam *Bootstrap* itu sendiri seperti fungsinya yaitu memudahkan *developer* dalam membangun *interface* dalam *website*-nya, terdapat template untuk *font* atau *typography*, tombol, navigasi, dan *interface* lainnya yang *responsive* ketika diakses melalui aplikasi yang berbasis *desktop*[17].

2.2.12 Composer

Composer adalah sebuah *tools dependency manager* khusus *PHP* yang mempunyai fungsionalitas seperti *Gem (Ruby)* atau *Maven (Java)*. Suatu *library* dapat di-*install* di *Composer* dan *Composer* akan secara otomatis bisa meng-

install ataupun *meng-update library* yang sangat dibutuhkan. *Dependency manager* dalam *composer* adalah bahwa *composer* akan mengelola “ketergantungan” antara *library-library PHP* (yang disebut *package*), yang dibuat oleh seseorang untuk digunakan orang lain[18].

2.2.13 ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram (ERD) adalah diagram berbentuk notasi grafis yang berada dalam pembuatan *database* yang menghubungkan antara data satu dengan yang lain. Fungsi *ERD* adalah sebagai alat bantu dalam pembuatan database dan memberikan gambaran bagaimana kerja *database* yang akan dibuat[19]. *ERD* memiliki tiga konsep utama yaitu :

1. Entitas

Sebuah entitas dapat berupa orang, tempat, objek, atau kejadian yang dapat dianggap penting bagi sebuah organisasi atau perusahaan. Setiap entitas memiliki beberapa atribut yang mendeskripsikan karakteristik dari objek. Entitas pada komponen *ERD* dapat dibedakan menjadi dua macam yaitu *strong entity* dan *weak entity*. *Strong entity* merupakan entitas yang tidak bergantung pada entitas lain atau entitas yang dapat berdiri sendiri. Sedangkan untuk *weak entity* merupakan entitas yang keberadaannya tergantung pada entitas lain.

2. Atribut

Setiap entitas memiliki karakteristik tertentu yang disebut dengan atribut. Atribut berfungsi untuk mendeskripsikan karakteristik yang ada pada entitas yang disimpan dalam basis data

3. Relasi

Relasi adalah sebuah hubungan antara dua atau lebih entitas yang saling berkaitan. Relasi pada *ERD* dapat digambarkan dengan menggunakan simbol belah ketupat (*diamond*). Relasi memiliki beberapa jenis relasi yaitu *unary*, *binary*, *ternary*[20].

Berikut adalah gambar atau simbol dari ke tiga konsep *ERD*:



Gambar 2. 2 Konsep ERD

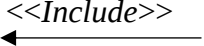
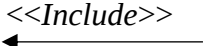
2.2.14 UML (*Unified Modelling Language*)

Unified Modeling Language (UML) adalah bahasa pemodelan grafis untuk menggambarkan, mendeskripsikan, mengkonstruksikan, dan mendokumentasikan artifak-artifak dari sebuah sistem piranti lunak. *UML* dapat dijadikan sebagai sebuah standar untuk membuat blueprint sebuah sistem, yang mencakup hal-hal yang bersifat konseptual seperti proses bisnis dan fungsi-fungsi sistem, dan juga mencakup hal-hal yang bersifat konkrit seperti statemen bahasa pemrograman, skema basis data, dan juga komponen sistem[21]. Komponen diagram *UML* yang digunakan ialah *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*.

1. Use case diagram

Use Case Diagram adalah salah satu jenis diagram dalam *UML* yang digunakan untuk menggambarkan interaksi antara pengguna (aktor) dengan sistem yang sedang dikembangkan. Diagram ini membantu memvisualisasikan fungsi-fungsi utama (*use case*) yang dapat dilakukan oleh aktor terhadap sistem.

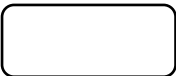
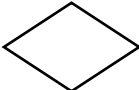
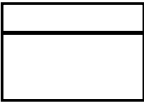
Tabel 2. 1 Simbol Use Case Diagram

No	Simbol	Nama	Keterangan
1		Aktor	Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .
2		<i>Use case</i>	Abstraksi dan interaksi antara sistem dan aktor.
3		<i>Association</i>	Abstraksi dari penghubung antara aktor dengan <i>use case</i> .
4		Generaliasi	Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .
5		<i>Include</i>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
6		<i>Extend</i>	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsionalitas dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.

2. Activity Diagram

Activity Diagram adalah salah satu diagram perilaku dalam *UML* yang digunakan untuk memodelkan alur kerja (*workflow*) atau urutan aktivitas dalam suatu proses bisnis atau sistem perangkat lunak.

Tabel 2. 2 Simbol Activity Diagram

No	Simbol	Nama	Keterangan
1		<i>Initial</i>	Titik awal untuk memulai suatu aktifitas.
2		<i>Final</i>	Titik akhir untuk mengakhiri suatu aktifitas.
3		<i>Activity</i>	Menandakan suatu aktifitas
4		<i>Decision</i>	Pilihan untuk mengambil keputusan.
5		<i>Fork</i> atau <i>Join</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
6		<i>Flow Final</i>	Untuk mengakhiri suatu aliran.
7		<i>Swimlane</i>	Untuk mengelompokkan <i>activity</i> berdasarkan <i>Actor</i> .

3. *Sequence diagram*

Sequence Diagram adalah salah satu diagram perilaku dalam *UML* yang digunakan untuk menggambarkan interaksi antar objek dalam sistem berdasarkan urutan waktu. Diagram ini menunjukkan bagaimana pesan dikirim antar objek untuk menyelesaikan suatu proses atau use case.

2.2.15 Blackbox Testing

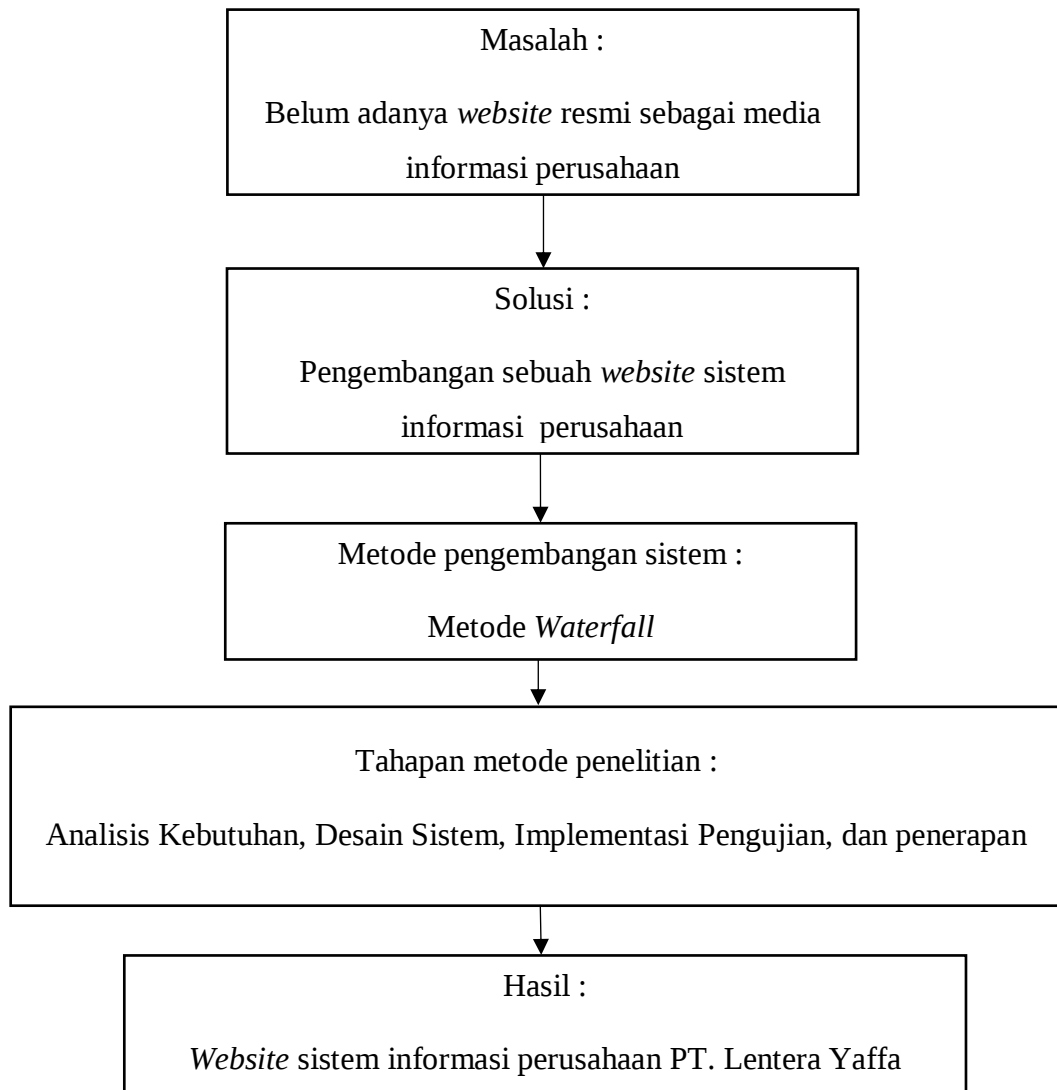
Blackbox testing adalah pengujian untuk mengetahui fungsional pada perangkat lunak dengan memberi masukan dan melihat keluaran seperti yang diharapkan atau tidak. Pengujian *Black box testing* cenderung dapat menemukan beberapa hal seperti fungsional yang tidak benar atau tidak ada, kesalahan basis data, kesalahan struktur data, kesalahan akses data, kesalahan antar muka, kesalahan pengguna, kesalahan performa, serta kesalahan inisialisasi dan terminasi[22]. *Blackbox Testing* tidak melibatkan pengujian terhadap kode sumber program (*source code*).

2.2.16 Metode System Usability Scale (SUS)

System Usability Scale (SUS) merupakan metode dalam mengukur pengujian terhadap pengguna ketika menggunakan produk. Metode ini ada pada tahun 1989, metode untuk menguji pengguna yang digunakan sebagai “*quick and dirty*”, dimana *SUS* dipergunakan dalam mengukur ketergunaan sistem komputer berdasarkan sudut pandang subjektif pengguna. Metode kuesioner *SUS (System Usability Scale)* mempunyai 5 jawaban, dimana responden dapat memilih jawaban “Sangat Tidak Setuju” bernilai 1 poin, “Tidak Setuju” bernilai 2 poin, “Netral/Ragu-ragu” bernilai 3

poin, “Setuju” bernilai 4, serta “Sangat Setuju” yang bernilai 5[23]. Dalam penelitian ini metode pengujian *system usability scale* digunakan sebagai pengujian pendukung dalam tingkat keberhasilan penelitian ini.

2.3 Kerangka Berpikir



Gambar 2. 3 Kerangka Berpikir