

BAB II TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Kajian Pustaka adalah proses sistematis untuk mengumpulkan, mengevaluasi, dan menganalisis sumber informasi yang relevan. Tujuannya adalah memahami topik secara mendalam, mengidentifikasi celah pengetahuan, dan melihat kontribusi penelitian sebelumnya. Berikut merupakan daftar penelitian terdahulu yang disajikan dalam Tabel 2.1.

Pengembangan *website* desa menjadi semakin penting dengan berkembangnya teknologi informasi dan komunikasi. Untuk mendukung pengembangan *website* di Desa Paberasan, peneliti membangun *website* dengan pendekatan kerangka OpenSID dalam pengembangannya. OpenSID memberikan desa-desa di Indonesia kemampuan untuk membangun *website* mereka sendiri dengan berbagai fitur yang telah disesuaikan dengan kebutuhan desa. Metode *Rapid Application Development* (RAD) digunakan untuk mempercepat proses pengembangan *website* desa menggunakan pendekatan kerangka OpenSID. Selain itu, pengujian pengembangan *website* juga menjadi hal yang penting dalam pengembangan *website* Desa Paberasan. Salah satu metode pengujian pengembangan *website* yang dapat digunakan adalah *black box testing* dan *heuristic evaluation*, dimana pengujian *black bok testing* digunakan untuk menguji fungsionalitas dari sistem *website*, sedangkan pengujian *heuristic evaluation* digunakan untuk menguji kelayakan penggunaan (*usability*) pada *website*. Berikut merupakan penelitian terdahulu yang menurut penulis mempunyai keterkaitan dengan penelitian yang akan dilakukan.

Penelitian pertama yang dilakukan oleh Puspa Arundini, Rudhy Ho Purabaya, dan Ati Zaidiah pada tahun 2021 dengan judul “Rancang Bangun Sistem Informasi Desa Berbasis Web Menggunakan Metode *Rapid Application Development* (RAD) Pada Desa Sukatani, Kecamatan 18

Cikande, Kabupaten Serang – Banten” bertujuan untuk mendukung implementasi Undang-Undang Nomor 6 Tahun 2014 tentang Desa. Undang-undang tersebut mendorong desa untuk berperan dalam pembangunan ketahanan sosial, layanan dasar, penanggulangan kemiskinan, peningkatan kualitas sumber daya manusia, dan kesejahteraan rakyat. Penelitian ini bertujuan untuk merancang dan membangun sistem informasi desa berbasis web dengan menggunakan metode *Rapid Application Development* (RAD). Hasil penelitian ini adalah sistem informasi desa berbasis web yang menampilkan berbagai fitur informasi seperti Profil Desa, Pemerintahan Desa, Lembaga Masyarakat, Data Kependudukan, Sarana & Prasarana, dan Kegiatan Desa. Fitur-fitur tersebut diharapkan dapat membantu mewujudkan peran desa, khususnya Desa Sukatani, Kecamatan Cikande, Kabupaten Serang – Banten, dalam memberikan informasi yang diperlukan oleh masyarakat dan pemangku kepentingan lainnya [4].

Penelitian selanjutnya dengan judul “Rancangan Sistem Informasi Layanan Administrasi Desa Berbasis Web” merupakan sebuah judul jurnal yang menjelaskan tentang pembuatan sebuah sistem informasi layanan administrasi berbasis *website* di Desa Sukaasih Kecamatan Sukatani, Kabupaten Bekasi. Penulisan jurnal ini dilakukan oleh Susi Susanti, Diego Wahyu, Ridwan Laksmana, dan Yuni Eka Achyani pada tahun 2022. Dalam jurnal ini permasalahan utama yang dihadapi adalah pengelolaan data yang masih manual, menyebabkan proses pengajuan surat menyurat menjadi lama dan memerlukan warga untuk datang ke kantor desa berulang kali. Selain itu, penyimpanan data yang juga masih manual menyebabkan risiko dokumen rusak, hilang, atau tidak terupdate. Tujuan penelitian ini adalah mengimplementasikan sistem informasi teknologi untuk digitalisasi penyerahan dokumen di Desa Sukasih, sehingga dapat meminimalisir kesalahan pencatatan dan mempercepat proses pelayanan administrasi. Pengembangan sistem menggunakan metode *Rapid Application Development* (RAD) dan bahasa pemrograman PHP. Hasil uji validitas menunjukkan bahwa sistem ini valid dengan nilai r hitung yang melebihi nilai r tabel 0,433

untuk setiap pertanyaan yang diuji. Dengan adanya sistem informasi ini, diharapkan pelayanan administrasi di Desa Sukasih menjadi lebih mudah, cepat, dan efisien [12].

Penelitian ketiga yang berjudul “Perancangan Sistem Informasi Desa (Sidesaka) Berbasis Web Pada Desa Karangsalam Kecamatan Kemranjen Kabupaten Banyumas” oleh Johanes Eko Pujiantoro, Aprilian Nurangga Saputra, Ario Mukti Leksono, dan Santoso Setiawan pada tahun 2023. Penelitian ini menjelaskan tentang pembuatan sistem informasi desa berbasis web untuk meningkatkan efisiensi layanan di Desa Karangsalam. Layanan surat menyurat yang masih manual menyebabkan proses pelayanan menjadi lama dan tidak optimal karena keterbatasan jam kerja perangkat desa. Penelitian ini bertujuan untuk mengembangkan Sistem Informasi Desa (SID) yang dapat diakses secara *online* oleh masyarakat. Metode yang digunakan dalam penelitian ini adalah *metode Waterfall*. Sistem yang dihasilkan memiliki berbagai fitur seperti berita desa terkini, kegiatan desa, pengumuman, pengaduan, serta pengajuan surat menyurat seperti pembuatan KTP, surat keterangan lahir, surat keterangan kematian, dan pengajuan SKCK. Diharapkan dengan adanya *website* ini, pelayanan desa menjadi lebih efisien dan optimal, memudahkan masyarakat dalam mendapatkan informasi dan layanan yang dibutuhkan [13].

Penelitian selanjutnya dengan judul “Perancangan Aplikasi ‘Studi Al-Qur’an Intensif’ Berbasis *Website* dengan menggunakan Pendekatan Design Sprint (Studi Kasus Studi Al-Qur’an Intensif UINSA)” merupakan sebuah jurnal yang menjelaskan tentang pembuatan aplikasi berbasis website untuk program Studi Al-Qur'an Intensif di Universitas Islam Negeri Sunan Ampel (UINSA). Penulisan jurnal ini dilakukan oleh Anna Retno Indrawati, Yunia Putri Wardayanti, Muhammad Syifaul Qolbi Romadlon, dan Aditya Nur Arif pada tahun 2020. Dalam jurnal tersebut, dijelaskan bahwa pengembangan aplikasi menggunakan metode *Design Sprint*. Metode ini merupakan pendekatan *desain* yang menghasilkan *prototipe* berdasarkan permasalahan

yang ada, yang kemudian diimplementasikan dalam bentuk aplikasi dan diakhiri dengan pengujian. Proses pengujian dilakukan untuk mengetahui kelayakan aplikasi, meliputi *heuristic evaluation*, pengujian *black box testing*, dan *usability testing*. Hasil pengujian *heuristic* menunjukkan beberapa desain *prototipe* yang perlu diperbaiki. Hasil pengujian *black box testing* menunjukkan aplikasi dapat berjalan dengan baik, dan hasil *usability testing* menunjukkan ada beberapa fitur yang perlu dihapus atau ditambahkan. Saran dari penelitian ini adalah perlunya waktu *maintenance* yang lebih untuk penyempurnaan fitur-fitur yang ada [14].

Penelitian terakhir yang dilakukan pada tahun 2021 oleh Reihan Fazri Aldiansyah, Syams Syair Ramadhan, Jaenal Arifin, dan Cahyadi Agustin yang berjudul “*Design of Information System Monitoring Complaints Targeting COVID-19 Vaccination For The General Public Based On The Web (Case Study of Bandung Health Office)*”. Penelitian ini menjelaskan tentang perancangan sistem informasi berbasis web untuk memantau keluhan masyarakat pasca vaksinasi COVID-19. Dinas Kesehatan Kota Bandung memerlukan sistem ini untuk mengelola data keluhan dengan baik, karena belum adanya wadah yang efisien untuk tujuan tersebut. Sistem ini dikembangkan menggunakan *metode waterfall* dan pemodelan berorientasi objek dengan *Unified Modelling Language (UML)*. Hasil pengujian *black box testing* menunjukkan keberhasilan fungsionalitas sistem sebesar 100%, dan evaluasi dengan *System Usability Scale (SUS)* menghasilkan nilai 77.6 dalam kategori *Acceptable* dan *Grade Scale B*. Sistem ini mempermudah petugas kesehatan dalam mengelola data keluhan vaksinasi COVID-19 secara efisien dan sesuai dengan kebutuhan pengguna [15].

Tabel 2. 1 Penelitian Terdahulu

No	Judul, Penulis, dan Tahun Terbit Penelitian	Masalah	Metode	Kesamaan	Perbedaan	Hasil
1	Rancang Bangun Sistem Informasi Desa Berbasis Web Menggunakan Metode <i>Rapid Application Development</i> (RAD) Pada Desa Sukatani, Kecamatan Cikande, Kabupaten Serang–Banten Puspa Arundini, Rudhy Ho Purabaya, Ati Zaidiah 2021 [4]	<i>Implementasi</i> Undang-Undang Nomor 6 Tahun 2014 tentang Desa	<i>Rapid Application Development</i> (RAD)	Merancang sistem informasi desa menggunakan metode pengembangan <i>Rapid Application Development</i> (RAD)	Penelitian ini dalam pengembangannya menggunakan bahasa pemrograman PHP dan tidak mengimplementasikan kerangka OpenSID dalam pengembangannya	Sebuah sistem informasi desa berbasis <i>website</i> yang dapat digunakan untuk pelayanan informasi di Desa Sukatani, Kecamatan Cikande, Kabupaten Serang–Banten
2	Rancangan Sistem Informasi Layanan	Proses pengelolaan data yang masih	<i>Rapid Application</i>	Merancang sistem informasi berbasis	Penelitian ini dalam pengembangannya menggunakan	Sebuah sistem informasi layanan administrasi

No	Judul, Penulis, dan Tahun Terbit Penelitian	Masalah	Metode	Kesamaan	Perbedaan	Hasil
	Administrasi Desa Berbasis Web Susi Susanti, Diego Wahyu, Ridwan Laksmana, dan Yuni Eka Achyani 2022 [12]	manual di Desa Sukaasih, Kecamatan Sukatani, Kabupaten Bekasi	<i>Development</i> (RAD)	<i>website</i> dengan metode pengembangan <i>Rapid Application Development</i> (RAD)	bahasa pemrograman PHP dan tidak mengimplementasikan kerangka OpenSID dalam pengembangannya	berbasis <i>website</i> di Desa Sukaasih, Kecamatan Sukatani, Kabupaten Bekasi
3	Perancangan Sistem Informasi Desa (Sidesaka) Berbasis Web Pada Desa Karangsalam Kecamatan Kemranjen Kabupaten Banyumas Johanes Eko Pujiantoro, Aprilian Nurangga Saputra, Ario Mukti	Proses layanan surat menyurat yang masih manual di Desa Karangsalam, Kecamatan Kemranjen, Kabupaten Banyumas	<i>Waterfall</i>	Merancang sistem Informasi desa berbasis <i>website</i>	Metode yang digunakan adalah metode pengembangan <i>waterfall</i>	Sebuah sistem informasi layanan administrasi berbasis <i>website</i> di Karangsalam Kecamatan Kemranjen Kabupaten Banyumas.

No	Judul, Penulis, dan Tahun Terbit Penelitian	Masalah	Metode	Kesamaan	Perbedaan	Hasil
	Leksono, dan Santoso Setiawan 2023 [13]					
4	Perancangan Aplikasi ‘Studi Al-Qur’an Intensif’ Berbasis <i>Website</i> dengan menggunakan Pendekatan Design Sprint (Studi Kasus Studi Al-Qur’an Intensif UINSA) Anna Retno Indrawati, Yunia Putri Wardayanti, Muhammad Syifaul Qolbi Romadlon, dan Aditya Nur Arif 2020 [14]	Belum adanya pengolahan data yang mempermudah jalannya program Studi Al-Qur’an <i>Intensif</i>	<i>Design Sprint</i>	Merancang aplikasi berbasis <i>website</i> dengan pengujian <i>black box testing</i> dan <i>heuristic evaluation</i>	Metode yang di gunakan adalah metode pengembangan <i>Design Sprint</i>	Sebuah aplikasi berbasis <i>website</i> untu mempermudah pengolahan data dalam jalannya program Studi Al-Qur’an Intensif di Universitas Islam Negeri Sunan Ampel (UINSA)

No	Judul, Penulis, dan Tahun Terbit Penelitian	Masalah	Metode	Kesamaan	Perbedaan	Hasil
5	<i>Design of Information System Monitoring Complaints Targeting COVID-19 Vaccination For The General Public Based On The Web (Case Study of Bandung Health Office)</i> Reihan Fazri Aldiansyah, Syams Syair Ramadhan, Jaenal Arifin, dan Cahyadi Agustin 2021 [15]	Belum adanya pengolahan data keluhan masyarakat pandemi COVID-19 pada Dinas Kesehatan Kota Bandung	<i>Waterfall</i>	Merancang sistem informasi berbasis <i>website</i> dengan pengujian <i>black box testing</i>	Metode yang di gunakan adalah metode pengembangan <i>Waterfall</i>	Sebuah sistem informasi berbasis <i>website</i> yang digunakan untuk mengelola data pelaporan keluhan vaksinasi COVID-19 pada Dinas Kesehatan Kota Bandung

2.2 Landasan Teori

Berikut merupakan kajian terkait beberapa teori yang digunakan penelitian ini yaitu sebagai berikut:

2.2.1 Website

Website adalah sekumpulan halaman yang terhubung secara elektronik dan dapat diakses melalui jaringan internet. *Website* biasanya berisi informasi, konten multimedia, atau layanan interaktif yang disajikan dalam bentuk teks, gambar, video, atau elemen lainnya. Tujuan utama dari sebuah *website* adalah menyediakan informasi yang dapat diakses oleh pengguna internet dari berbagai lokasi. *website* juga merupakan kumpulan sistem halaman yang saling berhubungan antara halaman lain dan dari tempat manapun yang tidak dibatasi oleh jenis komputer, *smartphone*, dan sistem operasi [16].

2.2.2 Sistem Informasi

Sistem informasi merupakan kombinasi dari orang, proses, dan teknologi yang saling berinteraksi untuk mengumpulkan, mengelola, menyimpan, dan menyebarkan informasi yang diperlukan untuk mendukung pengambilan keputusan, koordinasi, kontrol, analisis, dan tindakan di dalam suatu organisasi. Sistem informasi mengintegrasikan elemen-elemen ini untuk memfasilitasi pengumpulan, pemrosesan, penyimpanan, dan distribusi data dan informasi yang relevan [17].

2.2.3 Sistem Manajemen Informasi

Sistem manajemen informasi adalah sistem yang mengelola dan mengatur data serta informasi yang bermanfaat untuk mendukung pelaksanaan tugas dalam suatu organisasi. Sistem manajemen informasi atau *Management Information System (MIS)* dalam bahasa Inggris merupakan bagian dari perencanaan dan

pengendalian internal suatu bisnis yang melibatkan pemanfaatan sumber daya manusia, dokumen, teknologi, dan prosedur oleh manajemen akuntansi untuk menyelesaikan masalah bisnis seperti biaya produk, layanan, atau strategi bisnis. Sistem manajemen informasi berbeda dari sistem informasi biasa karena digunakan untuk menganalisis sistem informasi lain yang diterapkan pada kegiatan operasional organisasi [18].

2.2.4 Internet

Internet adalah jaringan komputer *global* yang menghubungkan jutaan perangkat komputer di seluruh dunia. Ini merupakan infrastruktur yang meliputi kabel, satelit, dan konektivitas nirkabel yang memungkinkan pengiriman dan pertukaran data antara perangkat yang terhubung. Internet memberikan akses ke berbagai sumber daya dan layanan, termasuk situs web, email, komunikasi real-time, transfer file, streaming media, dan banyak lagi [19].

2.2.5 Kerangka OpenSID

OpenSID adalah sebuah sistem informasi desa yang sengaja dirancang terbuka dan dapat dikembangkan secara bersama-sama oleh komunitas peduli Sistem Informasi Desa (SID) [20]. OpenSID juga merupakan sebuah *tools* dan kerangka kerja yang digunakan untuk pengelolaan informasi desa secara *online opensource* dan dapat berfungsi sebagai *website* profil desa. OpenSID dikembangkan oleh komunitas peduli desa dengan tujuan mempermudah pelayanan sistem informasi desa kepada masyarakat. OpenSID didesain untuk digunakan oleh pemerintah desa dalam mengelola data administrasi desa, menyampaikan informasi kepada masyarakat, dan memfasilitasi partisipasi aktif masyarakat dalam pengembangan desa. Sistem informasi desa yang bermakna dan

bermanfaat bagi masyarakat indonesia baik secara internal maupun eksternal desa [7].

Komponen fitur *website* OpenSID menyediakan berbagai fitur antara lain [21].

1. Administrasi kependudukan: Fitur yang memuat pengolahan data kependudukan desa seperti data keluarga, penduduk, data pemilih desa dan lain-lain.
2. Laporan penduduk: Fitur yang memuat pembuatan laporan jumlah penduduk, kepemilikan KTP dan KK dan lain-lain.
3. Data keluarga: Fitur yang memuat pengelompokkan data keluarga yang digunakan pemerintah dalam memantau untuk menargetkan program dari pemerintah.
4. Mutasi penduduk: Fitur yang memuat pendataan perubahan jumlah penduduk desa jika terjadi penambahan atau pengurangan penduduk yang disebabkan oleh kelahiran, kematian dan perpindahan penduduk.
5. Statistik penduduk: Fitur yang memuat pelayanan pengolahan data penduduk dan pengelompokkan data penduduk berdasarkan usia, pendidikan, pekerjaan, gender dan lain-lain.
6. Fitur layanan terpadu: Fitur yang memuat pelayanan pembuatan surat secara otomatis tanpa harus mengetik secara manual di computer [22].

Komponen Teknologi yang digunakan dalam kerangka bangun OpenSID [23].

- Bahasa Pemrograman: OpenSID dibangun menggunakan bahasa pemrograman PHP, dan bahasa pemrograman pendukung lainnya seperti JavaScript dan Blade.
- Database: Database OpenSID menggunakan MySQL sebagai sistem manajemen basis datanya.

- *Framework*: OpenSID memanfaatkan *framework CodeIgniter* versi 3, yang merupakan salah satu *framework* PHP yang ringan dan cepat. *CodeIgniter* mempermudah pengembangan aplikasi web dengan menyediakan berbagai *library* dan *helper* yang siap digunakan.

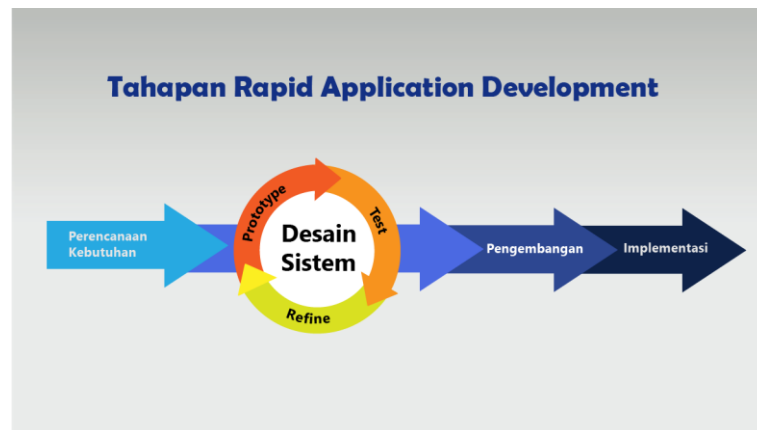
Tujuan utama dari OpenSID adalah untuk memberdayakan desa-desa dengan alat yang diperlukan untuk menghadapi era digital [24]. Beberapa tujuan spesifik OpenSID adalah:

- a) Memudahkan akses informasi: OpenSID menyediakan *platform* untuk mempublikasikan berbagai informasi terkait dengan pemerintahan desa, kegiatan masyarakat, kebijakan, dan program-program pembangunan.
- b) Transparansi dan akuntabilitas: pengelolaan data dan informasi dan yang baik akan mendorong transparansi dan akuntabilitas, yang merupakan pilar dasar tata kelola yang baik.
- c) Meningkatkan keterlibatan masyarakat: OpenSID memberikan ruang bagi masyarakat untuk berpartisipasi dalam proses pembangunan desa. Melalui forum diskusi, pengaduan *online*, dan fitur-fitur interaktif lainnya, masyarakat dapat berkontribusi aktif dalam meningkatkan kualitas hidup di desa mereka.
- d) Pelayanan, administrasi, dan pelaporan yang akurat: sistem pengelolaan data dan informasi yang mendukung pelayanan, administrasi, dan pelaporan; yakni mempunyai kegunaan praktis untuk mempermudah pekerjaan pemerintah desa.

Dengan menggunakan pendekatan kerangka OpenSID, diharapkan sistem manajemen informasi desa yang dikembangkan dapat memenuhi kebutuhan di desa Paberasan secara efektif dan efisien.

2.2.6 *Rapid Application Development (RAD)*

Metode *Rapid Application Development* atau RAD merupakan sebuah metode yang digunakan dalam pengembangan sistem yang memiliki tujuan untuk mempercepat waktu dalam perencanaan, perancangan, dan pengimplementasian agar dapat diselesaikan dengan cepat dalam waktu yang singkat [25]. Gambar 2.1 menunjukkan tahapan yang ada di dalam metode RAD.



Gambar 2. 1 Tahapan dalam Metode *Rapid Application Development* [26].

Terdapat empat tahapan pada metode RAD, yaitu tahap perencanaan kebutuhan, desain sistem, pengembangan dan tahap implementasi. Salah satu keunggulan metode RAD dibandingkan metode lainnya, yakni pengembangan perangkat lunak dapat dilakukan dengan lebih cepat dan fleksibel karena pengembang dapat dengan cepat menguji dan memodifikasi *prototipe*, serta menggunakan kembali komponen yang ada untuk mempercepat pengembangan [26].

2.2.6.1 Perencanaan Kebutuhan

Pada tahap ini, masalah diidentifikasi dan data dikumpulkan dari pihak terkait atau pengguna untuk menetapkan tujuan akhir sistem dan kebutuhan informasi. Keterlibatan aktif antara pengembangan dan *stakeholder*

menjadi kunci untuk menentukan kebutuhan pengembangan sistem [26].

Metode perencanaan kebutuhan melibatkan wawancara langsung dengan instansi terkait atau narasumber yang terkait dengan isu penelitian yang sedang diteliti oleh peneliti, observasi langsung di Desa Paberasan, serta referensi studi Pustaka dari buku dan jurnal penelitian terdahulu sebagai dasar referensi.

2.2.6.2 Desain Sistem

Pada tahap desain sistem partisipasi aktif dari pengguna yang terlibat sangat penting untuk mencapai tujuan, karena pada tahap ini terjadi proses desain dan perbaikan desain yang berulang sampai sesuai dengan kebutuhan pengguna. Hasil dari tahap ini adalah perancangan atau pembuatan *prototipe* dengan kemungkinan adanya perbaikan desain yang dilakukan berulang kali apabila terdapat kesalahan atau ketidaksesuaian antara desain dengan kebutuhan pengguna yang telah diidentifikasi pada tahap sebelumnya. Tujuan dari proses ini adalah membantu mengurangi kesalahan dan melakukan perbaikan yang diperlukan. Pengguna juga dapat melakukan uji coba dan memberikan umpan balik. Hasil dari perancangan sistem atau desain perangkat lunak yang mencakup organisasi sistem secara umum, struktur data, dan informasi yang akan dihasilkan [26].

2.2.6.3 Proses Pengembangan

Dalam tahap pengembangan sistem, dilakukan pengkodean berdasarkan *prototipe* yang telah dibuat dan kemudian dilakukan *slicing* terhadap *prototipe* tersebut

untuk menghasilkan *website*. Apabila proses berjalan dengan lancar, pengembang dapat melanjutkan ke langkah berikutnya. Namun, jika aplikasi yang dikembangkan belum memenuhi persyaratan sepenuhnya, pengembangan akan kembali ke tahap desain sistem [26]

Metode proses pengembangan dan pengumpulan *feedback* yaitu identifikasi kebutuhan adalah pengembang harus berinteraksi dengan pengguna atau pemangku kepentingan untuk mengidentifikasi kebutuhan, Perancangan *prototipe*, pengembangan *iterative*, pengumpulan *feedback*, evaluasi dan penyesuaian evaluasi terhadap aplikasi berdasarkan *feedback* yang diterima, penyelesaian dan peluncuran berdasarkan *feedback* dan evaluasi, pengembang melanjutkan ke tahap finalisasi, yang meliputi penyelesaian fitur, pengujian akhir, dan persiapan untuk peluncuran aplikasi, iterasi lanjutan.

2.2.6.4 Implementasi atau penyelesaian produk

Pada tahap implementasi atau penyelesaian produk, sangat penting bagi pengembang untuk menangani potensi masalah yang mungkin muncul selama proses pengembangan aplikasi. Tujuan dari tahap ini adalah untuk mengoptimalkan stabilitas aplikasi, meningkatkan antarmuka pengguna, melakukan pemeliharaan, dan menyusun dokumentasi. Meskipun demikian, langkah terakhir ini dilakukan sebelum aplikasi diserahkan kepada klien [26].

Metode Implementasi atau penyelesaian produk yaitu analisis *bug* untuk mengetahui kesalahan yang terdeteksi, prioritas dan pengelompokan *bug* pengembang akan

memprioritaskan *bug* berdasarkan dampaknya terhadap aplikasi, Perbaikan kode pengembang akan memperbaiki kode yang menyebabkan *bug* atau kesalahan, pengujian ulang, peningkatan interface pengguna, pemeliharaan dan optimalisasi, pembuatan dokumentasi dokumentasi ini akan membantu pengguna dalam penggunaan dan pemeliharaan aplikasi.

2.2.7 PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman sisi server yang dirancang terutama untuk pengembangan web. Dibuat oleh Rasmus Lerdorf pada tahun 1994, PHP telah berkembang menjadi bahasa yang kuat yang digunakan untuk membangun halaman *website* dan aplikasi *website* yang dinamis. PHP berfungsi dengan memproses kode di server dan menghasilkan *output* HTML untuk ditampilkan di *browser website*. PHP juga merupakan bahasa *scripting* umum yang populer, terutama cocok untuk pengembangan *website*. PHP digunakan dalam segala hal mulai dari *blog* hingga situs *website* paling populer di dunia [27].

2.2.8 CodeIgniter

Codeigniter adalah sebuah *framework* yang mengadopsi aturan MVC (*Model, View, Controller*) berbasis PHP. Codeigniter mempercepat pengerjaan aplikasi tanpa perlu menulis semua kode dari awal. Codeigniter sangat ringan digunakan, dan juga memiliki dokumentasi yang lengkap dengan besertakan contoh penggunaannya [28].

2.2.9 HTML (*Hypertext Markup Language*)

HTML (*Hypertext Markup Language*) adalah sebuah bahasa yang biasa digunakan untuk menampilkan sebuah halaman yang berisi konten pada halaman *website*. Pada HTML, terdapat fungsi-

fungsi utama yang dapat dilakukan yaitu, mengatur dan mendesain halaman *website*, membuat tabel, dapat melakukan input dari user, menampilkan gambar, dan dapat dipublikasikan secara online [27].

2.2.10 CSS (*Cascading Style Sheets*)

CSS (*Cascading Style Sheets*) merupakan sebuah teknologi dalam pengembangan sebuah *website* yang berfokus pada pembuatan tampilan *website*. CSS dapat difungsikan sebagai pengatur dalam tampilan *website* seperti dalam teks, gambar dan memposisikan setiap elemen pada *website* [29].

2.2.11 MySQL

MySQL merupakan sistem manajemen basis data atau DBMS yang terstruktur dan bersifat relasional dengan mempunyai bentuk baris dan kolom pada suatu tabel dengan menggunakan bahasa atau perintah SQL (*Structured Query Language*) untuk mengeksekusinya. Secara garis besar mempunyai fungsi untuk mengelola dan membuat *database* dari sisi server dengan memuat berbagai sumber informasi yang ditata secara terstruktur, sehingga pengguna dapat mengakses dataset menggunakan perintah query dalam jumlah besar untuk menyeleksi, agresri, filterisasi dan penggabungan data [27].

2.2.12 Bootstrap

Bootstrap adalah *framework* pengembangan *front end* yang dirancang untuk membantu menciptakan antarmuka pengguna *website* yang efisien dan menarik. Dikembangkan oleh Twitter dan dirilis pada tahun 2011, Bootstrap menyediakan *template desain* berbasis HTML dan CSS untuk berbagai elemen antarmuka pengguna, serta ekstensi *JavaScript* opsional. Mendukung desain *responsif*, bootstrap memastikan bahwa *layout website* dapat menyesuaikan diri dengan berbagai ukuran layar. Ini mudah

digunakan dan memiliki komunitas pengguna yang besar, menjadikannya populer di kalangan pengembang *website* dari semua tingkat keahlian [29].





2.2.13 Unified Modeling Language (UML)

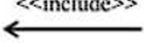
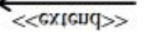
UML (*Unified Modeling Language*) adalah bahasa pemodelan visual yang digunakan untuk merancang, menggambarkan, dan mendokumentasikan sistem perangkat lunak. UML menyediakan seperangkat notasi grafis yang dapat digunakan untuk menggambarkan berbagai aspek sistem yang akan dikembangkan. Berikut adalah penjelasan tentang beberapa diagram UML yang umum digunakan dalam tugas akhir dan referensi situs yang relevan untuk rujukan lebih lanjut [30]:

2.2.13.1 Use Case Diagram

Use Case Diagram digunakan untuk menggambarkan interaksi antara aktor (pengguna sistem) dengan sistem itu sendiri. Diagram ini menunjukkan fungsionalitas sistem dan kasus penggunaan yang terkait dengan sistem tersebut [30].

Tabel 2. 2 Simbol *Use Case Diagram*

Simbol	Keterangan
	Aktor: Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan use case.
	<i>Use Case</i> : Abstraksi dan interaksi antara sistem dan aktor.
	<i>Association</i> : Abstraksi penghubung antara aktor dengan <i>use case</i> .
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .

	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainya.
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainya jika suatu kondisi terpenuhi.

2.2.13.2 Class Diagram

Class diagram menjelaskan kelas - kelas yang digunakan dalam desain sistem dan menggambarkan struktur sistem tersebut. Diagram ini mencakup dua komponen utama, yaitu metode (fungsi atau tindakan yang dapat dilakukan oleh kelas) dan atribut (variabel atau data yang dimiliki oleh kelas). Dengan menggunakan *Diagram Class*, kita dapat memvisualisasikan hubungan antara kelas-kelas dalam sistem dan bagaimana mereka saling berinteraksi. Diagram ini membantu dalam memahami organisasi dan hierarki kelas - kelas serta bagaimana mereka saling terkait [30].



Gambar 2. 2 Komponen *Class Diagram* [30].

Class diagram memiliki beberapa komponen yang dapat digunakan untuk merancang diagram seperti yang ditunjukkan pada gambar 2.2 diantaranya


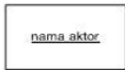
1. Komponen atas merupakan komponen yang berfungsi untuk memberikan nama class.





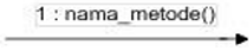
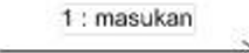
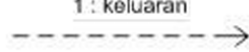
2. Komponen tengah merupakan komponen yang berisi kumpulan atribut untuk menjelaskan suatu class.
3. Komponen bawah merupakan komponen yang berisi sebuah metode metode yang berfungsi untuk mendeskripsikan operasi pada suatu class sebagai interaksi dengan data.

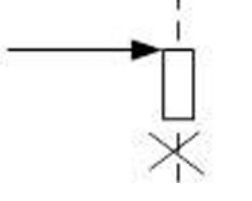
2.2.13.3 Sequence Diagram

Sequence Diagram digunakan untuk mengilustrasikan objek-objek berinteraksi di dalam dan sekitar suatu sistem. Diagram ini memperlihatkan pesan-pesan yang saling dikirim antara objek-objek tersebut seiring berjalannya waktu. *Sequence Diagram* memiliki dimensi vertikal yang menggambarkan waktu dan dimensi *horizontal* yang mencakup objek-objek yang terlibat dalam interaksi tersebut [30].

Tabel 2. 3 Simbol *Sequence Diagram*

Simbol	Deskripsi
<p>Aktor</p>  <p>Atau</p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.</p>
Garis Hidup/ <i>Lifeline</i>	Menyatakan kehidupan suatu objek.


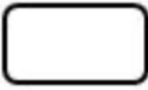
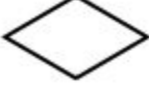

	
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan.</p>
<p>Waktu Aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.</p>
<p>Pesan Tipe <i>Create</i></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan Tipe <i>Call</i></p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.</p>
<p>Pesan Tipe <i>Send</i></p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>Pesan Tipe <i>Return</i></p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima.</p>
<p>Pesan Tipe <i>Destroy</i></p>	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek</p>


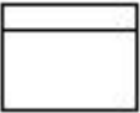
	<p>yang diakhiri, sebaiknya jika ada create maka ada destroy.</p>
---	---

2.2.13.4 Activity Diagram

Activity Diagram merupakan bagian penting dari UML, menunjukkan elemen dinamis sistem. Aktivitas diagram dapat dengan mudah digunakan untuk menjelaskan logika prosedural, proses bisnis, dan aliran kerja suatu perusahaan. *Activity Diagram* memiliki tujuan yang sama seperti *flowchart*, tetapi mereka bisa mendukung perilaku paralel sementara *flowchart* tidak bisa. Dengan menunjukkan aliran pesan yang mengalir dari satu aktifitas ke aktifitas lainnya, *Activity Diagram* dirancang dengan tujuan untuk mengidentifikasi tingkah laku dinamis dari sistem [30].

Tabel 2. 4 Simbol *Activity Diagram*

Simbol	Nama	Keterangan
	Status Awal	Sebuah diagram aktivitas memiliki status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan/ <i>Decision</i>	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan <i>/ Join</i>	Penggabungan dimana yang mana lebih dari

		satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	Swimlane	Swimlane organisasi bertanggung memisahkan bisnis yang jawab.

2.2.14 Database

Database adalah sekumpulan data yang tersimpan pada perangkat komputer untuk dapat dikelola seperti diakses, diperbaharui dan diorganisir dengan menggunakan *Database Manajemen System* (DBMS). Salah satu model yang paling terkenal pada *database* yaitu *relation data model* yang memungkinkan sebuah data akan disimpan pada tabel yang memiliki value dan berelasi satu sama lain. Salah satu keunggulan *database* ialah kemudahan dalam proses identifikasi data dimana *database* akan menyajikan data yang diminta bersamaan dengan data – data yang memiliki keterkaitan melalui *database managemen sistem*. Dari segi konsep *database* merupakan sekumpulan data yang membentuk sebuah file yang memiliki keterkaitan satu sama lain dengan aturan tertentu dan membentuk sebuah data atau informasi baru. Relasi pada tabel di *database* biasanya ditandai dengan kunci (key) dari setiap file [31].

2.2.15 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah suatu visualisasi yang digunakan dalam perancangan basis data untuk mengilustrasikan secara rinci hubungan atau koneksi antara objek atau entitas beserta atribut-atributnya. Dengan menggunakan ERD, sistem basis data yang dikembangkan dapat digambarkan secara lebih terstruktur dan terorganisir [31]. Dalam ERD, ada tiga elemen dasar di antaranya adalah entitas, atribut, dan relasi:

2.2.15.1 Entitas

Entitas merupakan objek yang menjadi fokus dalam suatu *database*. Mereka dapat mewakili manusia, tempat, objek, atau kondisi yang terkait dengan informasi yang dikelola. Representasi simbolis untuk entitas adalah bentuk persegi Panjang [31].

2.2.15.2 Atribut

Atribut merupakan segmen informasi yang terdapat dalam suatu entitas. Setiap entitas umumnya memiliki *primary key* sebagai identitas khasnya, disertai dengan atribut deskriptif. Atribut ini biasanya disimpan dalam tabel entitas atau dapat berdiri sendiri dari tabel. Simbol yang digunakan untuk merepresentasikan atribut adalah bentuk elips [31].

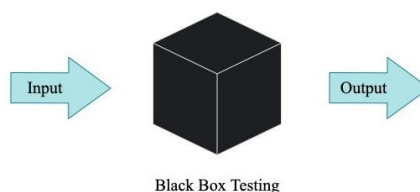
2.2.15.3 Relasi

Relasi atau hubungan dalam ERD merepresentasikan koneksi antara dua entitas atau lebih. Simbol hubungan adalah bentuk belah ketupat [31]

2.2.16 *Black Box Testing*

Black Box Testing merupakan aktivitas pengujian perangkat lunak yang memverifikasi *output* dari *software* berdasarkan *input* yang diberikan untuk memastikan fungsionalnya sudah sesuai dengan persyaratan. *Black Box Testing* (*Behavioral Testing*) adalah

jenis pengujian perangkat lunak yang hanya mengevaluasi hasil input dan output tanpa memerhatikan struktur kode di dalamnya [32].



Gambar 2. 3 Alur *Black Box Testing*.

Berikut merupakan teknik *Black Box Testing* yang digunakan diantaranya [32]:

1. *Equivalence Class Testing* merupakan teknik untuk meminimalkan jumlah pengujian sistem dengan tujuan pengujian yang optimal dengan mempertahankan cakupan uji yang sedang.
2. *Boundary Value Testing* merupakan teknik yang berfokus pada nilai nilai pada batas dengan menentukan rentang nilai yang diterima oleh sistem.
3. *Decision Table Testing* merupakan tabel yang mempunyai isi sebuah keputusan dengan menempatkan penyebab serta pengaruh yang ditimbulkan.

2.2.17 *Heuristic Evaluation*

Heuristic evaluation merupakan salah satu metode yang digunakan untuk mengukur usability sebuah *website*. Metode pengujian ini dikembangkan oleh Jakob Nielsen pada awal 1990-an dan telah menjadi alat penting dalam evaluasi usability [33]. *Heuristic evaluation* merupakan teknik pengujian *usability* yang dilakukan oleh ahli (*expert*) dan biaya cenderung lebih mahal yang

disebabkan keterlibatan ahli (*expert*) namun lebih cepat mendapatkan umpan balik. *Heuristic Evaluation* dilakukan dengan melihat suatu interface dan mencoba untuk mendapatkan pendapat mengenai interface tersebut, apakah baik atau tidak. *Heuristic Evaluation* biasanya dilakukan oleh *usability specialist* atau orang yang setidaknya memiliki keterlibatan pada proyek tersebut. Idealnya seorang ahli akan melakukan evaluasi atau ulasan berdasarkan prinsip kegunaan dari badan penelitian dan pengalaman profesional sebelumnya [34]. Tujuan dari *Heuristic Evaluation* adalah untuk menemukan masalah *usability* dalam *user interface design* sehingga dapat dijadikan sebagai bagian dari proses desain berulang. Sudut pandangannya adalah target *spesifik* yang akan menggunakan produk tersebut. *Heuristic evaluation* memiliki beberapa jenis *instrumen* pengujian dan penentuan hasil pengujian yang diskala dari 0 (nol) sampai dengan 4 (empat), 0 (nol) berarti tidak memiliki masalah *usability*, 1 (satu) berarti memiliki masalah *cosmetic problem*, 2 (dua) berarti *minor usability problem*; perlu perbaikan, 3 (tiga) berarti *major usability problem*; perlu perbaikan yang mempengaruhi proses, dan 4 (empat) berarti *usability catastrophe*; perlu tindakan desain ulang [35]. Tabel 2.5 menggambarkan 10 item variabel pernyataan *instrumen* pengujian *heuristic evaluation* dari Jakob Nielsen [36].

Tabel 2. 5 Pernyataan instrument *Heuristic Evaluation*

No	Pernyataan	Keterangan
1.	<i>Visibility of system status</i>	instrumen pengujian <i>heuristic evaluation</i> (HE) untuk mengetahui apakah perangkat lunak dapat selalu menyediakan informasi kepada pengguna berkaitan proses yang sedang berlangsung.

2.	<i>Match between system and the real world</i>	instrumen pengujian ini untuk mengetahui apakah perangkat lunak menggunakan bahasa yang umum dan sesuai dengan umum bagi pengguna termasuk penggunaan kata, prasa dan konsep.
3.	<i>User control and freedom</i>	instrumen pengujian <i>heuristic evaluation</i> (HE) untuk mengetahui apakah pengguna dalam menjalankan perangkat lunak dapat dilakukan dengan nyaman dan leluasa seperti adanya <i>undo</i> dan <i>redo</i> .
4.	<i>Consistency and standards</i>	instrumen pengujian <i>heuristic evaluation</i> (HE) untuk mengetahui apakah perangkat lunak tidak ada makna yang ambigu baik penggunaan kata dan <i>icon</i> .
5.	<i>Error prevention</i>	instrumen pengujian <i>heuristic evaluation</i> (HE) untuk mengetahui bagaimana perangkat lunak mengatasi atau menangani ketika terjadi <i>error</i> atau kesalahan yang dilakukan oleh pengguna.
6.	<i>Recognition rather than recall</i>	instrumen pengujian <i>heuristic evaluation</i> (HE) untuk mengetahui apakah perangkat lunak dapat meminimalisir penggunaan memori (mengingat) oleh pengguna dalam hal makna gambar, keterangan atau ketika melakukan pilihan.

7.	<i>Flexibility and efficiency of use</i>	instrumen pengujian <i>heuristic evaluation</i> (HE) untuk mengetahui apakah perangkat lunak dapat membuat pekerjaan lebih cepat dan apakah proses pengerjaannya memiliki jalan pintas.
8.	<i>Aesthetic and minimalist design</i>	instrumen pengujian <i>heuristic evaluation</i> (HE) untuk mengetahui apakah perangkat lunak memiliki menu, informasi dan bagian yang kurang relevan dengan keinginan pengguna.
9.	<i>Help users recognize, diagnose, and recover from errors</i>	instrumen pengujian ini untuk mengetahui apakah perangkat lunak dapat menampilkan pesan jika terjadi kesalahan dan memiliki informasi bagaimana mengatasinya.
10.	<i>Help and documentation</i>	instrumen pengujian <i>heuristic evaluation</i> (HE) untuk mengetahui apakah perangkat lunak dapat dijalankan tanpa harus membaca petunjuk penggunaan.