

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Sistem Informasi

Hartono (2021) menjelaskan sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Definisi sistem adalah suatu kesatuan yang terdiri atas komponen atau elemen yang saling berinteraksi, saling terkait, atau saling bergantung membentuk keseluruhan yang kompleks (Soufitri, 2023). Dengan demikian, sistem informasi adalah integrasi komponen terstruktur untuk mengelola data dan mendukung pengambilan keputusan dalam organisasi.

Setiap komponen dalam system memiliki fungsi dan cara kerja berbeda, namun tetap bekerja terpadu, saling bergantung, dan saling mendukung untuk mencapai tujuan system tanpa saling bertentangan Widarti et al., (2024). Sinergi antar komponen membuat proses sistem berjalan lancar. Karena itu, keterpaduan fungsi menjadi kunci kinerja yang optimal, akurat, dan minim kesalahan.

Sistem informasi adalah kesatuan komponen terintegrasi yang bekerja terstruktur untuk mengolah data dan mendukung pengambilan

keputusan. Sinergi antar komponen memastikan sistem berjalan optimal, akurat, dan berkelanjutan.

2. Sistem Berbasis *Web*

Menurut Arafat et al., (2022) *website* merupakan suatu media informasi yang menawarkan berbagai kemudahan dalam menyajikan informasi dari suatu objek kepada masyarakat luas secara cepat dan dapat diakses oleh pengunjung internet secara global. Kemudahan akses tanpa batas ruang dan waktu menjadikan *website* efektif untuk distribusi data secara *real time*. Karena itu, teknologi ini penting dalam membangun sistem digital yang mampu menjangkau publik secara luas, terstruktur, dan interaktif.

Sistem informasi berbasis *web* system komponen yang menggunakan teknologi *web* untuk mengolah serta menyajikan data sesuai kebutuhan pengguna. Sistem ini dibuat dengan tujuan memberikan kemudahan penggunaan, meskipun bagi pengguna yang tidak memiliki pemahaman teknis mendalam (Zikra et al., 2025). Sedangkan menurut Fauziyah & Sugiarti (2022) sistem informasi berbasis *web* merupakan fasilitas dalam sistem komputerisasi yang dirancang dengan fitur khusus untuk kebutuhan *penginputan* data, dengan tujuan untuk mempermudah, mempercepat, serta menghasilkan pengolahan data yang akurat.

Sistem berbasis *web* system terintegrasi yang memanfaatkan internet untuk mengelola dan menyajikan data secara cepat, akurat, dan global. Dirancang untuk distribusi *real time* dan mudah digunakan, system ini efektif

membangun layanan system yang luas, terstruktur, dan interaktif tanpa batas ruang dan waktu.

3. Layanan Publik

Menurut Mokobombang et al., (2023) layanan publik adalah segala bentuk kegiatan pemenuhan kebutuhan masyarakat oleh penyelenggara negara. Penerapan teknologi terbukti mampu mengubah tata cara dan meningkatkan kualitas responsivitas layanan tersebut. Pelayanan publik merupakan indikator utama dalam menilai keberhasilan kinerja sebuah instansi pemerintahan. Kualitas layanan ini sangat ditentukan oleh kecepatan birokrasi dan kemudahan akses bagi masyarakat luas (Mustaqim et al., 2023).

Adeti & Christiani, (2022) mendefinisikan layanan publik sebagai wujud nyata dari pengabdian aparatur birokrasi kepada warganya. Oleh karena itu, reformasi administrasi menjadi kunci mutlak untuk mewujudkan sistem pelayanan yang transparan. Dengan adanya transparansi, kepercayaan publik terhadap pemerintah akan meningkat karena hak-hak masyarakat dapat terpenuhi secara adil, cepat, dan terukur.

Layanan publik merupakan bentuk pengabdian pemerintah dan menjadi ukuran utama kinerja instansi. Untuk meningkatkan kualitasnya, diperlukan reformasi administrasi yang terintegrasi dengan teknologi informasi. Transformasi ini bertujuan menciptakan sistem yang cepat, transparan, dan mudah diakses, sehingga hak masyarakat terpenuhi secara adil dan meningkatkan kepercayaan publik.

4. Teknologi Pengembangan Sistem

a. HTML

Mardiansyah et al., (2024) menyatakan bahwa *HyperText Markup Language* (HTML) merupakan bahasa yang digunakan untuk membangun struktur dasar halaman *web*, termasuk pengaturan elemen paragraf, gambar, tabel, dan *link*. HTML merupakan pengembangan standar pemformatan dokumen teks *Standard Generalized Markup Language* (SGML). Secara umum, dokumen HTML memiliki tiga tag utama yang berfungsi untuk menampilkan informasi, yaitu *html*, *head*, *tag*, *body*, dan *footer* (Rahmawan et al., 2023).

HTML berfungsi sebagai tulang punggung struktural halaman *web* dengan menerapkan konsep dasar berupa *tag*, elemen, dan atribut dalam penyusunan struktur dokumen (Sinlae et al., 2024). Ketiga konsep ini memungkinkan pengembang mengelompokkan konten seperti teks, gambar, dan *link* secara jelas. Karena itu, struktur HTML yang tepat menjadi fondasi utama sebelum menambahkan desain visual dan fungsi interaktif.

HTML adalah bahasa yang menjadi kerangka dasar halaman *web*. Dengan elemen, atribut, dan *tag*, HTML menyusun konten secara terstruktur. Karena itu, HTML menjadi fondasi utama sebelum ditambahkan desain visual dan fitur interaktif.

b. CSS

Menurut Orisa, et al., (2023) *Cascading Style Sheets* (CSS) merupakan bahasa yang digunakan untuk mengatur tampilan halaman *web* agar lebih menarik dengan mengendalikan format tampilan pada dokumen HTML. CSS berfungsi untuk mengatur gaya visual halaman *web*, seperti pengaturan tata letak, pemberian warna, serta pengaturan jenis *font*, serta berperan dalam membuat tampilan halaman *web* menjadi lebih menarik dan responsif pada berbagai perangkat (Sinlae et al., 2024). Karena itu, bahasa ini berperan penting dalam mengubah HTML menjadi antarmuka yang estetik, modern, dan responsif bagi pengguna.

Teknologi CSS digunakan untuk menata dan memperindah elemen visual, sehingga pengembang *web* dapat mengubah desain, warna latar belakang, serta tata letak halaman *web* dengan lebih mudah dan responsif (Rahmawan et al., 2023). Hal ini dimungkinkan karena perubahan gaya dapat dilakukan terpusat tanpa mengubah struktur HTML. Karena itu, perancangan antarmuka lebih efisien dan konsistensi visual tetap terjaga.

CSS adalah bahasa pemformatan penting untuk merancang dan memperindah tampilan *web*. Dengan mengatur layout, warna, tipografi, dan responsivitas secara terpusat, CSS memungkinkan perubahan desain secara efisien tanpa mengubah struktur HTML. Karena itu, CSS berperan penting dalam menciptakan tampilan yang konsisten, menarik, dan nyaman di berbagai perangkat.

c. *JavaScript*

Fatikasari & Fargazzi (2025) menyatakan bahwa *JavaScript* merupakan bahasa pemrograman fundamental dalam pengembangan *web* modern yang digunakan secara luas pada sisi klien untuk membangun aplikasi *web* yang bersifat dinamis, kompleks, dan interaktif. Sedangkan menurut Wijaya et al., (2024) *JavaScript* adalah bahasa pemrograman yang dapat digunakan untuk mengembangkan algoritma pengurutan serta menganalisis waktu komputasi dan penggunaan memori. Dengan antarmuka interaktif dan kemampuan menangani logika kompleks, *JavaScript* menjadi penting untuk membangun sistem yang responsif dan berkinerja tinggi.

Menurut Syahputra & Hanifa, (2025) *JavaScript* merupakan bahasa pemrograman dengan sintaks yang fleksibel yang awalnya dikembangkan untuk mendukung interaktivitas *web*, namun kini mendukung model *event driven* yang sangat efisien untuk aplikasi *real time* dan pengembangan yang cepat. Arsitektur *event driven* memungkinkan sistem merespons aksi pengguna, seperti pengisian form atau klik tombol, tanpa perlu memuat ulang halaman. Karena itu, mekanisme ini penting untuk memastikan alur data yang lancar serta pengalaman yang lebih responsif dan dinamis.

JavaScript adalah bahasa pemrograman penting yang menggabungkan pemrosesan logika dengan pembuatan antarmuka *web* interaktif. Sintaks fleksibel dan arsitektur *event driven* memungkinkan

pemrosesan data serta respons *real time* tanpa *reload* halaman. Karena itu, *JavaScript* menjadi fondasi dalam membangun layanan digital yang efisien dan responsif.

d. PHP

Menurut Lase & Alasi (2024) *Hypertext Preprocessor* (PHP) merupakan bahasa pemrograman *web* yang berjalan pada sisi server serta dapat diintegrasikan langsung ke dalam dokumen HTML dalam pengembangan sistem aplikasi. PHP adalah bahasa skrip tingkat tinggi yang dapat disisipkan ke dalam dokumen HTML. Sintaks PHP memiliki kemiripan dengan bahasa C, *Java*, dan *Perl*, dengan tujuan utama untuk merancang aplikasi *web* yang dinamis (Sandria et al., 2022).

PHP merupakan bahasa pemrograman *open source* yang berjalan pada sisi *server*, di mana skrip diproses oleh *server*. Sifat sumber terbuka tersebut memberikan kebebasan kepada pengguna untuk memodifikasi serta mengembangkan aplikasi sesuai kebutuhan (Endra et al., 2022). Selain itu, dukungan ekosistem yang luas memungkinkan pengembang untuk dengan mudah mengintegrasikan berbagai kerangka kerja *framework* guna mempercepat proses perancangan sistem.

PHP adalah bahasa pemrograman *server side open source* untuk membangun aplikasi *web* dinamis. Sintaksnya mudah diintegrasikan dengan HTML, sehingga fleksibel dalam pengembangan sistem. Didukung ekosistem *framework* yang luas, PHP efektif mempercepat pembuatan sistem yang tangguh dan adaptif.

e. Laravel

Menurut Melyani & Aji, (2023) *laravel* adalah *framework* aplikasi *web open source* yang digunakan untuk merancang aplikasi secara cepat. *Framework* ini menyediakan fitur *modern* dengan kode yang singkat, ekspresif, dan mudah dipahami. Karena itu, penggunaannya membantu mengurangi kompleksitas teknis sehingga pengembang dapat fokus pada fitur utama dan logika sistem.

Laravel merupakan *framework* PHP *open source* berlisensi *Massachusetts Institute of Technology* (MIT) yang dibangun menggunakan arsitektur *Model View Controller* (MVC). *Framework* ini dirancang untuk menyederhanakan pengembangan *website* dan meningkatkan kualitas perangkat lunak dengan menekan biaya pemeliharaan (Jatika et al., 2023). Pemisahan logika data dan antarmuka melalui MVC membuat kode lebih rapi dan terstruktur, sehingga memudahkan pengelolaan serta pengembangan sistem di masa depan.

Laravel adalah kerangka kerja pemrograman *open source* yang membantu pengembang memaksimalkan penggunaan PHP dalam pengembangan *website*. *Laravel* dilengkapi dengan berbagai fitur unggulan yang memudahkan proses, seperti *template engine*, *routing*, dan modularitas (Aipina & Witriyono, 2022). Karena itu, fitur terintegrasi *Laravel* menjadikannya efektif untuk mempercepat pengembangan aplikasi sekaligus menghasilkan kode yang terstruktur dan mudah dikelola.

Laravel adalah *framework* PHP *open source* berbasis MVC yang dirancang untuk mempercepat pengembangan aplikasi *web*. Dengan *sintaks* ekspresif dan fitur terintegrasi seperti *templating*, *routing*, serta pemisahan logika dan tampilan, *Laravel* menyederhanakan kompleksitas pemrograman. Karena itu, *Laravel* efektif menghasilkan kode yang rapi, memudahkan pengelolaan, serta mendukung efisiensi pemeliharaan dan pengembangan sistem di masa depan.

f. MySQL

Menurut Siregar et al., (2024) MySQL merupakan sistem manajemen basis data untuk pengolahan data yang dipilih karena kemampuannya dalam mengelola dan memproses informasi secara efisien, sehingga sangat relevan dengan kebutuhan industri. Sedangkan Wahyudi et al., (2022) mendefinisikan MySQL sebagai sistem manajemen basis data relasional RDBMS *open source* yang berfungsi untuk membuat, mengelola, serta menyampaikan informasi dari basis data berskala kecil hingga sangat besar. Perpaduan efisiensi pemrosesan dan fleksibilitas penyimpanan menjadikan MySQL andal dalam menjaga integritas, kecepatan, dan keamanan data pada sistem aplikasi.

MySQL merupakan perangkat lunak sistem manajemen basis data yang sangat fleksibel digunakan. Sistem ini mendukung aktivitas pengelolaan seperti membuat, menyimpan, mengubah, menghapus, serta memulihkan data dalam tabel (Mukhlis & Santoso, 2023). Kelengkapan fungsionalitas dalam manipulasi data tersebut menjadikan

MySQL sebagai fondasi penyimpanan yang tangguh, memastikan setiap informasi dapat dikelola secara terstruktur dan diakses dengan cepat sesuai dengan dinamika kebutuhan sistem.

MySQL adalah RDBMS *open source* yang efisien dan fleksibel untuk mengelola data dalam berbagai skala. Dengan fitur lengkap untuk manipulasi dan pemulihan data, MySQL memastikan informasi tersimpan dan terakses secara terstruktur. Kombinasi kecepatan dan kapasitasnya menjadikannya andal dalam menjaga integritas, keamanan, dan kelancaran arus data pada sistem aplikasi.

5. Perancangan Sistem

a. *Flowchart*

Flowchart adalah diagram yang menggambarkan urutan proses secara sistematis dengan simbol grafis. Diagram ini membantu memahami, menganalisis, dan merancang alur kerja agar lebih terstruktur, efisien, serta mudah dievaluasi dan dikembangkan. (Listyoningrum et al., 2023). Sedangkan menurut Haumahu et al., (2023) *flowchart* merupakan urutan logika dari prosedur pemecahan masalah yang dituliskan menggunakan 16 system-simbol standar guna menggambarkan tahapan penyelesaian masalah secara sederhana, terurai, rapi, dan jelas.

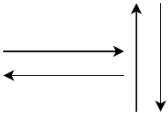
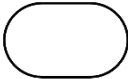
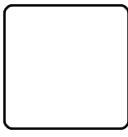
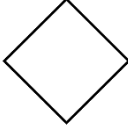
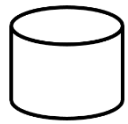
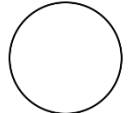
Tuasamu et al., (2023) mendefinisikan bagan alur *flowchart* sebagai teknik analitis bergambar yang digunakan untuk menjelaskan berbagai aspek sistem informasi secara jelas, ringkas, dan logis melalui

pencatatan jalannya proses bisnis perusahaan. Penggambaran dilakukan dengan simbol grafis terstandar yang saling terhubung untuk memetakan urutan instruksi dari *input*, proses, hingga *output*. Karena itu, *flowchart* penting sebagai alat komunikasi visual agar logika sistem mudah dipahami, dievaluasi, dan dirancang secara terstruktur sebelum diimplementasikan ke dalam kode program.

Flowchart adalah teknik visual yang menggunakan simbol standar untuk memetakan alur logika dan pemecahan masalah secara sistematis. Diagram ini menyederhanakan proses kompleks dari *input*, keputusan, hingga *output* menjadi alur yang jelas dan mudah dipahami. Karena itu, *flowchart* penting sebagai alat komunikasi perancangan untuk memastikan kelayakan, struktur, dan efisiensi sistem sebelum diimplementasikan ke dalam kode program.

Sebagai alat visual, *flowchart* memanfaatkan simbol-simbol khusus untuk menggambarkan setiap tahapan proses serta hubungan antar langkah. Simbol-simbol tersebut membantu analis dan pengembang dalam memahami, mendokumentasikan, dan merancang alur kerja sistem secara sistematis. Adapun bentuk, nama, dan fungsi masing-masing simbol *flowchart* dijelaskan pada Tabel 2.1.

Tabel 2.1. Simbol-Simbol Flowchart

Simbol	Nama	Fungsi
	<i>Flow</i>	Menunjukkan arah jalannya alur proses.
	<i>Terminator</i>	Menandai awal dan akhir dari sebuah proses.
	<i>Process</i>	Menunjukkan proses pengolahan data atau tindakan oleh sistem.
	<i>Decision</i>	Menunjukkan titik pengambilan keputusan pilihan Ya atau Tidak.
	<i>Input / Output</i>	Menunjukkan proses memasukkan data (<i>input</i>) atau menampilkan hasil (<i>output</i>).
	<i>Database</i>	Menunjukkan proses penyimpanan atau pemanggilan data dari <i>database</i> .
	<i>On-Page Reference</i>	Menyambungkan alur proses yang terputus di halaman yang sama.

Sumber: (Yuliawati et al., 2024)

Pemahaman dan penerapan yang tepat terhadap simbol-simbol *flowchart* di atas sangat krusial dalam merancang alur logika program. Melalui visualisasi pada Tabel 2.1, setiap tahapan proses bisnis dapat dipetakan secara terstruktur, sehingga meminimalisasi terjadinya kesalahan logika pada saat tahap pengkodean sistem diimplementasikan.

b. Unified Modelling Language

Menurut Wahyuni et al., (2022) *Unified Modelling Language* (UML) adalah suatu metode pemodelan secara visual yang digunakan sebagai bahasa standar maupun sarana dalam perancangan dan dokumentasi sebuah sistem yang berorientasi objek. UML menyediakan standar untuk membuat *blueprint* sistem yang mencakup konsep proses bisnis, perancangan *class* yang dapat diimplementasikan dalam sistem pemrograman tertentu, desain basis data, serta berbagai komponen yang diperlukan dalam pengembangan sistem (Narulita et al., 2024). Dengan *blueprint* yang komprehensif, UML berperan penting dalam menjembatani proses bisnis dan implementasi teknis, sehingga pengembangan sistem berjalan terarah, konsisten, dan minim ambiguitas.

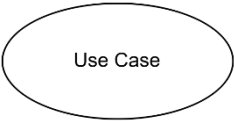
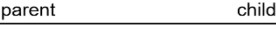
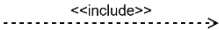
UML merupakan standarisasi bahasa pemodelan visual yang berfungsi untuk menspesifikasikan, menggambarkan, merancang, membangun, serta mendokumentasikan sistem perangkat lunak menggunakan teknik pemrograman berorientasi objek (Salimu et al., 2023). Melalui pendekatan ini, UML menyediakan berbagai diagram untuk merepresentasikan sistem dari sisi struktur dan perilaku. Standarisasi ini membantu menyamakan pemahaman tim sehingga kebutuhan dan spesifikasi sistem jelas sebelum implementasi.

UML adalah bahasa pemodelan visual standar untuk merancang, mendokumentasikan, dan merepresentasikan sistem

berorientasi objek. Melalui berbagai diagram, UML memetakan aspek sistem mulai dari struktur, basis data, hingga proses bisnis. Karena itu, UML penting untuk menyamakan pemahaman tim dan menjembatani kebutuhan bisnis dengan implementasi teknis secara terarah dan konsisten.

Use case diagram merupakan model hasil analisis perancangan sistem yang bertujuan untuk mendeskripsikan kebutuhan fungsional sistem secara visual (Ramdany et al., 2023). Diagram ini menggambarkan kebutuhan sistem dan interaksi antara aktor dan sistem secara jelas. Untuk memvisualisasikannya, digunakan notasi standar yang mewakili elemen sistem. Simbol-simbol utama *use case* diagram disajikan pada Tabel 2.2.

Tabel 2.2. Simbol-Simbol Use Case Diagram

Simbol	Nama	Keterangan
 Actor	<i>Actor</i>	Menunjukkan pihak yang berinteraksi dengan sistem, seperti pengguna atau pihak luar.
 Use Case	<i>Use Case</i>	Fitur, fungsionalitas, atau aktivitas tertentu yang bisa dilakukan oleh pengguna di dalam sistem.
 parent child	<i>Association</i>	Menunjukkan hubungan antara <i>actor</i> dengan <i>use case</i> .
 System name [Subsystem: System]	<i>System Boundary</i>	Kotak pembatas yang memisahkan antara bagian dalam ruang lingkup aplikasi berupa sistem dengan bagian luarnya berupa <i>Actor</i> .
 <<include>>	<i>Include</i>	Menunjukkan bahwa suatu <i>use case</i> selalu

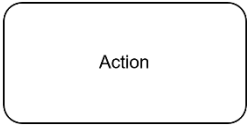
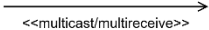
membutuhkan proses *use case* lain.

Sumber: (Yuliawati et al., 2024)

Melalui pengaplikasian simbol-simbol *use case* diagram pada Tabel 2.2 batasan hak akses dan interaksi dari masing-masing aktor terhadap sistem dapat didefinisikan dengan tegas. Hal ini akan sangat memudahkan pengembang dalam memvalidasi apakah seluruh spesifikasi kebutuhan fungsional perangkat lunak telah terakomodasi dengan baik.

Activity diagram merupakan salah satu jenis diagram dalam UML yang berfungsi untuk memvisualisasikan alur kerja atau rangkaian aktivitas dalam suatu sistem secara runtut (Yuliawati, 2024). Diagram ini digunakan untuk menggambarkan alur proses bisnis dari awal hingga akhir. Simbol-simbol *activity* diagram dijelaskan pada Tabel 2.3.

Tabel 2.3. Simbol-Simbol *Activity* Diagram

Simbol	Nama	Keterangan
	<i>Initial Node</i>	Menunjukkan awal dari suatu aktivitas atau proses.
	<i>Activity / Action</i>	Menunjukkan aktivitas atau proses yang dilakukan dalam sistem.
	<i>Control Flow</i>	Menunjukkan arah aliran proses dari satu aktivitas ke aktivitas lainnya.
	<i>Decision Node</i>	Menunjukkan percabangan berdasarkan kondisi tertentu.

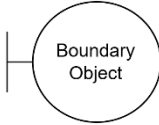
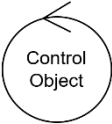
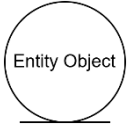
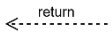
	<i>Swimlane</i>	Membagi aktivitas berdasarkan pelaku atau bagian yang menjalankan proses.
	<i>Activity Final Node</i>	Menunjukkan akhir dari suatu aktivitas atau proses.

Sumber: (Yuliawati et al., 2024)

Penggunaan elemen-elemen *activity* diagram pada Tabel 2.3 ini sangat efektif untuk memecah proses bisnis yang kompleks menjadi serangkaian tindakan yang lebih detail. Dengan adanya pemisahan peran melalui elemen *swimlane*, alur kerja antara pengguna dan sistem dapat dieksekusi secara berurutan tanpa adanya tumpang tindih proses.

Sequence diagram menggambarkan bagaimana interaksi berlangsung antara objek-objek di dalam aplikasi dengan penekanan pada urutan waktu pengiriman pesan, sehingga alur komunikasi, proses pertukaran data, serta hubungan antar objek dalam sistem dapat dipahami secara lebih sistematis dan terstruktur (Nawassyarif et al., 2021). Diagram ini memvisualisasikan komunikasi antar komponen secara kronologis, sehingga alur pesan, hubungan objek, dan urutan interaksi mudah dipahami. Dengan demikian, proses komunikasi sistem menjadi lebih jelas dan terstruktur. Simbol *sequence* diagram dijelaskan pada Tabel 2.4.

Tabel 2.4. Simbol-Simbol *Sequence Diagram*

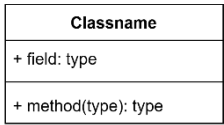
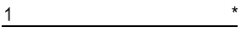
Simbol	Nama	Keterangan
	<i>Actor</i>	Menunjukkan pihak yang berinteraksi dengan sistem
	<i>Lifeline</i>	Menunjukkan objek atau bagian sistem yang terlibat dalam proses.
	<i>Boundary</i>	Mewakili antarmuka pengguna atau halaman <i>website</i> yang dilihat pengguna.
	<i>Control</i>	Mengatur jalannya proses bisnis atau logika sistem.
	<i>Entity</i>	Menunjukkan data atau objek yang disimpan dalam <i>database</i> .
	<i>Activation Bar</i>	Menunjukkan bahwa suatu objek sedang aktif menjalankan proses tertentu.
	<i>Message</i>	Menunjukkan perintah, permintaan, atau komunikasi antar objek dalam sistem.
	<i>Return Message</i>	Menunjukkan balasan atau hasil dari proses yang telah dilakukan.

Sumber: (Yuliawati et al., 2024)

Dengan memanfaatkan komponen-komponen *sequence diagram* pada Tabel 2.4 arsitektur komunikasi antar objek dari waktu ke waktu dapat divisualisasikan secara dinamis. Detail pertukaran pesan dari antarmuka pengguna hingga ke basis data ini akan menjadi panduan utama dalam membangun logika *Controller* pada saat tahapan implementasi sistem perangkat lunak.

Class diagram merupakan diagram yang menggambarkan struktur statis dari sebuah sistem dengan mendeskripsikan himpunan kelas, atribut, metode, serta hubungan relasi antar objek satu sama lain (Harlina et al., 2025). Pemodelan ini berfungsi untuk memperlihatkan *blueprint* sistem serta memberikan gambaran mengenai bagaimana elemen-elemen penyusunnya saling terhubung dan berinteraksi dalam suatu kesatuan sistem. Adapun simbol-simbol yang digunakan dalam *class* diagram dijabarkan pada Tabel 2.5.

Tabel 2.5. Simbol-Simbol *Class* Diagram

Simbol	Nama	Keterangan
	<i>Class</i>	Menunjukkan objek utama atau entitas dalam sistem yang memiliki atribut data dan operasi / fungsi.
	<i>Association</i>	Menunjukkan hubungan antara satu <i>class</i> dengan <i>class</i> lainnya yang saling berinteraksi.
	<i>Multiplicity</i>	Menunjukkan jumlah hubungan antar <i>class</i> atau kardinalitas relasi.
	<i>Composition</i>	Menunjukkan hubungan bagian dari yang sangat kuat, di mana satu <i>class</i> bergantung penuh pada <i>class</i> lainnya.

Sumber: (Yulawati et al., 2024)

Pemodelan yang komprehensif menggunakan simbol-simbol *class* diagram pada Tabel 2.5 akan menjadi fondasi utama dalam merancang arsitektur basis data. Pendefinisian atribut, operasi, beserta kardinalitas relasi yang jelas antar entitas akan memastikan integritas

data dan mempermudah proses integrasi *database* ke dalam bahasa pemrograman yang digunakan.

6. Metode Prototype

Menurut Fauziyah & Sugiarti (2022) metode *prototype* merupakan metode pengembangan perangkat lunak berupa model fisik yang berfungsi sebagai representasi awal sistem untuk memberikan ide fungsionalitas aplikasi bagi pengembang dan pengguna secara cepat. Pemodelan *prototype* adalah metode yang digunakan pengembang perangkat lunak untuk membuat dan mensimulasikan rancangan program kepada pengguna guna memastikan kesesuaian sistem dengan spesifikasi kebutuhan pengguna (Syarif et al., 2024). Melalui simulasi dan evaluasi interaktif, metode ini efektif mengurangi kesalahan sejak awal perancangan, sehingga hasil akhir perangkat lunak sesuai dengan kebutuhan pengguna.

Metode *prototype* adalah suatu teknik analisis dan perancangan yang memungkinkan pengguna untuk ikut serta secara langsung dalam menentukan spesifikasi kebutuhan dan pembentukan sistem yang akan dikerjakan (Sari et al., 2023). Keterlibatan aktif dilakukan melalui evaluasi dan umpan balik berulang terhadap rancangan awal. Kolaborasi ini mengurangi risiko perubahan di akhir serta memastikan sistem sesuai dengan kebutuhan pengguna.

Metode *prototype* adalah pendekatan pengembangan perangkat lunak interaktif yang melibatkan pengguna melalui pembuatan model awal sistem. Metode ini memungkinkan evaluasi berulang untuk mengidentifikasi

kebutuhan secara cepat dan akurat. Karena itu, pendekatan ini efektif mengurangi kesalahan dan risiko perubahan di akhir, serta memastikan sistem sesuai dengan kebutuhan pengguna.

7. Pengujian Sistem

a. *Black Box Testing*

Menurut Praniffa et al., (2023) *Black Box Testing* disebut sebagai pengujian fungsional atau perilaku. Penguji mengevaluasi sistem berdasarkan spesifikasi kebutuhan tanpa mengetahui struktur interior serta logika kode program perangkat lunak tersebut. Karena itu, pendekatan ini penting untuk menilai kualitas dan kelayakan perangkat lunak dari sudut pandang pengguna, serta memastikan setiap *input* menghasilkan *output* yang tepat dan fungsi sistem berjalan sesuai rancangan.

Black Box Testing adalah metode uji fungsionalitas aplikasi yang difokuskan untuk menemukan kesalahan fungsi, detail tampilan, serta kesesuaian alur kerja tanpa mengharuskan penguji menguasai bahasa pemrograman (Ichsanudin et al., 2022). Sedangkan menurut Wahyudi et al., (2023) *Black Box Testing* merupakan teknik pengujian yang digunakan untuk menguji fungsionalitas sistem terhadap spesifikasinya guna memastikan apakah sistem menghasilkan *output* yang sesuai harapan pengguna. Dengan evaluasi antarmuka dan validasi *output*, pengujian ini menjadi tahap akhir untuk memastikan sistem berjalan optimal, bebas *bug*, dan siap digunakan.

Menurut Hardika et al., (2024) perhitungan hasil pengujian *Black Box Testing* menggunakan rumus sebagai berikut.

$$\text{Persentase Validasi} = \left(\frac{\text{Jumlah Test Case Valid}}{\text{Total Test Case}} \right) \times 100\%$$

Keterangan:

- 1) Persentase validasi: Hasil perhitungan ditulis dalam persentase.
- 2) Jumlah test *case* valid: total test *case* yang sesuai harapan (valid).
- 3) Total test *case*: Seluruh test *case* yang dilakukan.

Black Box Testing adalah metode pengujian yang berfokus pada fungsi, perilaku, dan antarmuka sistem tanpa melihat kode program. Pengujian ini memvalidasi *input* dan *output* untuk memastikan sistem sesuai kebutuhan. Karena itu, *Black Box Testing* menjadi tahap akhir yang penting untuk menjamin sistem berjalan optimal, bebas *bug*, dan siap digunakan.

b. User Acceptance Test

Menurut Yakub et al., (2024) *User Acceptance Testing* (UAT) adalah pengujian oleh pengguna akhir untuk mengumpulkan masukan, mengidentifikasi masalah, dan memastikan perangkat lunak telah memenuhi kebutuhan pelanggan sebelum diimplementasikan. Tahapan ini dilakukan pada akhir pengembangan dengan melibatkan pengguna untuk mencoba aplikasi dalam skenario nyata. Melalui uji penerimaan ini, pengembang memperoleh validasi akhir bahwa sistem berjalan optimal dan siap digunakan.

UAT merupakan tahap pengujian oleh pengguna akhir secara langsung untuk memverifikasi dan memastikan perangkat lunak telah memenuhi fungsionalitas dan harapan pengguna sebelum diluncurkan (Hartono et al., 2025). Sedangkan menurut Wahyudi et al., (2023) UAT adalah fase pengujian akhir di mana pengguna berinteraksi langsung dengan sistem guna memverifikasi fungsionalitas dan memastikan sistem telah memenuhi kebutuhan sebenarnya. Melalui verifikasi ini, pengembang memperoleh persetujuan akhir bahwa perangkat lunak telah berjalan dengan baik dan siap digunakan di lingkungan nyata.

Berdasarkan penelitian yang dilakukan oleh Salsabila et al., (2024), metode pengolahan data yang digunakan adalah kuesioner dengan skala *Likert*. Perhitungan hasil kuesioner dilakukan menggunakan rumus sebagai berikut.

$$P = \frac{\sum(f \times \text{bobot})}{\text{Skor tertinggi} \times \text{Jumlah responden}} \times 100\%$$

Keterangan:

- 1) P: Persentase tingkat persetujuan responden terhadap pertanyaan
- 2) f: Jumlah responden yang memilih skor tertentu
- 3) Bobot: Nilai skor pada Skala *Likert* (1–5)
- 4) Skor tertinggi: Skor maksimal pada Skala *Likert* , yaitu 5
- 5) Jumlah responden: Total responden yang mengisi kuesioner

UAT adalah tahap pengujian akhir yang melibatkan pengguna untuk memastikan sistem sesuai kebutuhan dan ekspektasi. Pengguna mencoba aplikasi dengan skenario nyata untuk memberi masukan dan

menemukan masalah. Karena itu, UAT menjadi validasi akhir bahwa sistem telah berjalan optimal dan siap digunakan.

B. Kajian Empiris

Penelitian terdahulu penting sebagai dasar dan referensi untuk mengetahui perkembangan studi serupa serta menentukan *novelty*. Berdasarkan penelusuran, terdapat beberapa penelitian yang relevan dengan Rancang Bangun Sistem Informasi Profil Desa Warurejo Berbasis *Web* sebagai Media Layanan Publik.

Penelitian oleh, Valovi et al., (2025) membangun sebuah *website* Desa Balayan di Kecamatan Kisam Tinggi yang difungsikan sebagai media komunikasi digital untuk penyampaian informasi. Sistem ini menggunakan *CodeIgniter* dan MySQL, dengan fitur profil, sejarah, berita, galeri, dan kegiatan. Hasilnya, sistem meningkatkan efisiensi dan transparansi informasi, dengan tingkat kepuasan pengguna 85,93% dengan kategori sangat baik pada UAT. Perbedaannya terletak pada fitur dan *framework* yang digunakan. Penelitian tersebut belum menyediakan fitur pengaduan publik dan masih menggunakan *CodeIgniter*, sedangkan penelitian ini menggunakan *Laravel* dengan fitur lebih lengkap serta mendukung sistem yang lebih interaktif.

Dalam penelitian yang lain, Sari et al., (2023) merancang sebuah sistem informasi pelayanan masyarakat berbasis *web* pada Kelurahan Sipagimbar menggunakan metode *prototype*. Sistem ini menangani layanan seperti pengurusan surat untuk mengatasi duplikasi data dan kurangnya informasi persyaratan. Hasilnya, sistem mempermudah pendokumentasian dan

mempercepat pelayanan dibandingkan proses manual. Penelitian tersebut berfokus pada digitalisasi layanan yang masih bersifat satu arah, sedangkan penelitian yang diusulkan dikembangkan secara interaktif dua arah.

Pendekatan berbeda ditunjukkan oleh, Susanto et al., (2024) yang mengimplementasikan metode Rapid Application Development (RAD) dalam membangun Sistem Pengaduan Masyarakat (SIPMAS) berbasis *web* di Desa Logung. Sistem ini mendigitalisasi alur pengaduan yang sebelumnya manual, sehingga lebih terstruktur, terpantau, dan efisien. Hasil pengujian menunjukkan sistem berfungsi dengan baik melalui *Black Box Testing* dan memperoleh tingkat penerimaan UAT sebesar 86%. Persamaannya terletak pada fitur pengaduan masyarakat. Namun, penelitian tersebut hanya berfokus pada sistem pengaduan, tanpa integrasi profil desa maupun publikasi dokumen transparansi dana desa.

Selanjutnya, Syahputra et al., (2025) merancang sebuah platform layanan pengaduan masyarakat berbasis *web* pada Kantor Desa Helvetia menggunakan *framework Laravel*. Sistem ini meningkatkan efisiensi dan responsivitas pemerintah desa, karena masyarakat dapat menyampaikan pengaduan secara *online* tanpa datang langsung ke kantor desa. Hasilnya, alur pelaporan menjadi lebih sederhana, pengarsipan lebih mudah, dan pengaduan dapat dipantau serta ditindaklanjuti secara terstruktur. Perbedaannya, penelitian ini mengintegrasikan fitur pengaduan dengan publikasi dokumen APBDes, RKPDDes, RPJMDes, serta promosi UMKM dalam satu portal *web* desa terpadu.

Sementara itu, Zikra et al., (2025) merancang sebuah sistem informasi berbasis *web* untuk pengelolaan permohonan data pada instansi pemerintah guna

mendukung transformasi digital dalam layanan publik. Sistem ini bertujuan menciptakan alur pengajuan data yang lebih efisien, transparan, dan berorientasi pada kepuasan pengguna, sehingga mengurangi hambatan birokrasi dalam layanan informasi. Hasil penelitian ini menghasilkan kerangka kerja berbasis UML untuk menjaga integritas data dan mempermudah akses informasi publik. Penelitian ini relevan dalam digitalisasi layanan publik, namun berbeda pada ruang lingkup dan objek penelitian. Penelitian ini difokuskan pada tingkat pemerintahan desa dengan tambahan pengujian terstruktur menggunakan *Black Box Testing* dan *User Acceptance Testing*.

Berdasarkan tinjauan penelitian terdahulu, sistem informasi desa umumnya masih bersifat parsial, seperti hanya menyajikan informasi satu arah, belum menggunakan *Laravel*, atau terbatas pada modul pengaduan tanpa integrasi profil desa. Karena itu, *novelty* penelitian ini terletak pada pengembangan sistem informasi desa terpadu berbasis *Laravel* yang menggabungkan profil desa, informasi, transparansi dokumen, dan layanan pengaduan interaktif dalam satu platform. Selain itu, penggunaan metode *Prototype* memungkinkan evaluasi langsung oleh pengguna sebelum pengembangan final, sehingga hasil lebih tepat guna. Kelayakan sistem selanjutnya diuji dengan *Black Box Testing* dan *User Acceptance Testing*.

C. Kerangka Berpikir

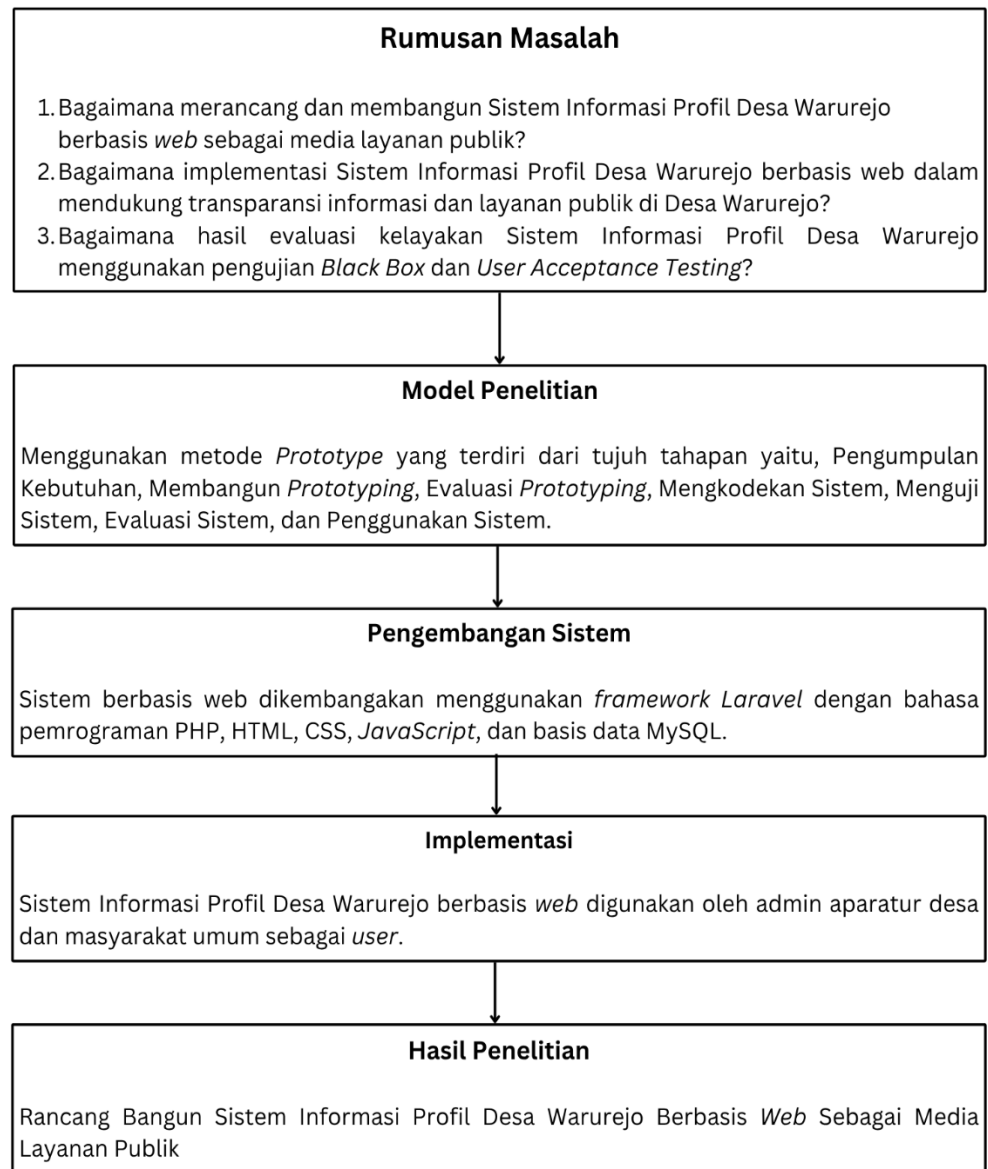
Penelitian ini dilatarbelakangi oleh tiga permasalahan utama di Desa Warurejo, yaitu belum tersedianya sistem informasi digital terintegrasi sebagai media publikasi resmi desa sehingga pengelolaan informasi masih bergantung

pada grup *Whatsapp* yang rawan tertumpuk dan kedaluwarsa, terbatasnya transparansi akses dokumen publik seperti APBDes, RPJMDes, dan RKPDes, serta belum adanya wadah pengaduan publik yang terpusat dan interaktif. Berdasarkan kajian empiris, penelitian-penelitian terdahulu menunjukkan bahwa sistem informasi desa yang telah dikembangkan masih bersifat parsial, yakni hanya menyajikan informasi satu arah atau hanya berfokus pada pengaduan tanpa integrasi profil desa dan transparansi publik secara menyeluruh.

Mengacu pada celah tersebut, penelitian ini mengembangkan Sistem Informasi Profil Desa Warurejo berbasis *web* menggunakan metode *Prototype* dengan *framework Laravel* dan basis data MySQL, yang kelayakannya diuji melalui *Black Box Testing* dan *User Acceptance Testing*. Objek penelitian adalah sistem informasi yang mencakup modul, profil desa, informasi desa, publikasi dokumen transparansi, dan forum pengaduan, dengan subjek pengguna terdiri atas *Admin* selaku aparatur desa dan *User* selaku masyarakat umum. Luaran yang diharapkan adalah sebuah *website* resmi desa yang terstruktur, transparan, dan *user friendly*, sehingga mampu meningkatkan efisiensi pengelolaan informasi, keterbukaan tata kelola pemerintahan, serta mendorong partisipasi aktif masyarakat melalui layanan pengaduan yang terpusat dan interaktif.

Hasil dari penelitian ini adalah terwujudnya Rancang Bangun Sistem Informasi Profil Desa Warurejo Berbasis *Web* sebagai Media Layanan Publik menggunakan metode *Prototype* dengan *framework Laravel* dan basis data MySQL. Sistem ini memungkinkan aparatur desa untuk mengelola dan mempublikasikan informasi secara terpusat, sekaligus memungkinkan

masyarakat untuk mengakses profil desa, melihat transparansi dokumen publik seperti APBDes, RPJMDes, dan RKPDes secara terbuka, serta menyampaikan aspirasi dan pengaduan secara *online* dan interaktif melalui forum publik yang terintegrasi dalam sistem. Keseluruhan alur dalam pengembangan sistem informasi yang dibangun ini dirangkum ke dalam sebuah bagan kerangka berpikir yang disusun oleh peneliti, sebagaimana dijelaskan pada Gambar 2.1 berikut:



Gambar 2.1 Kerangka Berpikir