

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Rancang Bangun

Rancang bangun sistem merupakan salah satu tahap krusial dalam proses pengembangan perangkat lunak, karena pada tahap ini ide dan kebutuhan pengguna mulai diterjemahkan menjadi bentuk desain sistem yang konkrit. Tahapan ini melibatkan perancangan struktur, alur proses, hingga antarmuka pengguna yang nantinya menjadi dasar pembangunan sistem secara teknis.

Musthofa, N., & Adiguna, M. A. (2022) menyatakan bahwa perancangan sistem yang tepat menjadi faktor penting dalam menentukan kualitas aplikasi, spesifikasi kebutuhan yang telah dikumpulkan dapat memenuhi kebutuhan pengguna dan tujuan pengembangannya. Hal ini didukung oleh pendapat Wibowo dan Sari (2021) yang menekankan bahwa desain sistem harus memperhatikan aspek fungsional dan kemudahan penggunaan agar sesuai dengan harapan pengguna.

Rancangan ini mencakup berbagai elemen penting seperti *input*, proses, *output*, basis data, dan antarmuka pengguna. Selain itu, aspek teknis harus diimbangi dengan perhatian terhadap kenyamanan dan kemudahan penggunaan bagi pengguna, sehingga *usability* menjadi elemen yang tidak terpisahkan dalam proses rancang bangun sistem.

Dengan demikian, rancang bangun sistem yang baik akan mendukung implementasi yang efektif dan berkelanjutan.

2. Sistem

Istilah sistem berasal dari bahasa Latin *systema* dan bahasa Yunani *sustēma*, yang berarti suatu kesatuan yang terdiri dari bagian-bagian yang saling berhubungan. Dalam konteks ilmu komputer dan sistem informasi, sistem dapat dipahami sebagai kumpulan elemen yang saling berinteraksi dan bekerja bersama untuk mencapai tujuan tertentu. Dalam konteks sistem informasi, elemen tersebut bisa berupa manusia, perangkat keras, perangkat lunak, prosedur, dan data.

Menurut Ni Wayan Purnawati et al. (2024), sistem informasi berfungsi sebagai fondasi utama dalam organisasi modern, karena mengatur bagaimana data diproses menjadi informasi yang bernilai manfaat dalam berbagai bidang ilmu dan aplikasi nyata. Sementara itu, pemahaman yang komprehensif terhadap komponen sistem termasuk kebutuhan pengguna, alur data, dan fungsi sistem mendukung perancangan sistem informasi yang efektif dan efisien (Permana et al., 2022).

Dari berbagai definisi tersebut, dapat disimpulkan bahwa sistem memiliki unsur-unsur penting seperti adanya *input*, proses, *output*, serta tujuan yang ingin dicapai. Dalam pengembangan sistem informasi, pemahaman yang baik terhadap konsep sistem menjadi dasar utama agar sistem yang dikembangkan dapat berjalan secara efektif dan efisien.

3. Pelaporan

Pelaporan adalah proses penting dalam penyampaian informasi dari suatu pihak ke pihak lain, yang biasanya digunakan sebagai dasar pengambilan keputusan. Menurut Nugraha et al. (2021), pelaporan modern memungkinkan penyajian data yang terstruktur dan akurat, mendukung kebutuhan manajerial secara *real-time*.

Dalam praktiknya, pelaporan sering kali menjadi bagian dari alur komunikasi informasi yang sangat penting, baik dalam kegiatan operasional organisasi maupun dalam sistem informasi digital. Perkembangan teknologi telah mengubah proses pelaporan menjadi lebih cepat dan efisien, memungkinkan informasi disajikan secara *real-time* dan terdokumentasi secara otomatis.

Sistem pelaporan merupakan suatu mekanisme terorganisir yang digunakan untuk mengumpulkan, mengolah, dan menyajikan informasi secara sistematis kepada pihak yang membutuhkan sebagai bahan pertimbangan dalam pengambilan keputusan. Sistem ini dapat bersifat manual atau berbasis teknologi informasi, tergantung pada kebutuhan dan infrastruktur yang tersedia. Erlangga et al (2023) menyebutkan bahwa sistem pelaporan berbasis *web* terbukti mendukung efisiensi alur kerja, mempercepat respon instansi, dan meningkatkan transparansi komunikasi antar pihak.

Dengan demikian, sistem pelaporan tidak hanya berfungsi sebagai alat dokumentasi, tetapi juga sebagai sarana komunikasi formal yang menghubungkan pelaksana kegiatan dengan pengambil kebijakan.

4. ODGJ (Orang Dengan Gangguan Jiwa)

ODGJ adalah individu yang mengalami gangguan mental yang mempengaruhi cara berpikir, merasakan, serta berinteraksi dengan lingkungan. Kondisi ini dapat berdampak pada kehidupan sosial dan kemampuan menjalani aktivitas sehari-hari. Menurut Rochmawati (2022), penanganan ODGJ tidak dapat hanya mengandalkan tenaga medis keterlibatan keluarga dan masyarakat dalam memberikan dukungan sosial yang tidak menghakimi menjadi faktor penting dalam proses pemulihan. Yayasan SATUNAMA Yogyakarta (2021), menegaskan bahwa penanganan ODGJ tidak hanya perlu intervensi medis, tetapi juga dukungan lintas sektoral termasuk dari pemerintah dan masyarakat untuk memastikan hak mereka terpenuhi secara adil.

5. *Unified Modeling Language* (UML)

Unified Modeling Language (UML) adalah bahasa pemodelan standar yang digunakan untuk merancang dan mendokumentasikan sistem perangkat lunak. UML menyediakan berbagai jenis diagram yang membantu dalam visualisasi, spesifikasi, dan dokumentasi sistem. Dalam penelitian ini, akan dibahas tiga jenis diagram UML yang penting, yaitu *Use Case Diagram*, *Activity Diagram*, dan *Sequence diagram*.

a. *Use Case Diagram*

Use Case Diagram adalah diagram yang menggambarkan interaksi antara pengguna (aktor) dan sistem. Diagram ini menunjukkan bagaimana aktor berinteraksi dengan sistem untuk mencapai tujuan tertentu. Elemen-elemen utama dalam *Use Case Diagram* meliputi:

- a) Aktor: Entitas eksternal yang berinteraksi dengan sistem, baik itu pengguna atau sistem lain.
- b) *Use Case*: Representasi dari fungsi atau layanan yang disediakan oleh sistem kepada aktor.
- c) Relasi: Hubungan antara aktor dan *Use Case*, yang dapat berupa asosiasi, generalisasi, atau *include*.




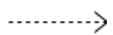
Use Case Diagram membantu dalam memahami kebutuhan pengguna dan fungsionalitas sistem secara keseluruhan. Diagram ini sering digunakan pada tahap awal pengembangan perangkat lunak untuk mengidentifikasi kebutuhan sistem.


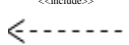
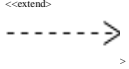
Dalam penelitian oleh (Taufan et al., 2022) menjelaskan bahwa *Use Case diagram* secara visual menggambarkan interaksi pengguna dengan sistem dan seluruh fungsi sistem yang disediakan, meski memerlukan rekayasa kebutuhan dan analisis elisitasi yang lengkap untuk menghasilkan diagram yang akurat. Proses ini diawali dengan rekayasa kebutuhan, yaitu proses memahami dan mendefinisikan yang sebenarnya dibutuhkan pengguna dari sistem. Sedangkan menurut (Ramdany et al., 2024) *Use Case* adalah gambaran tentang fungsi-

fungsi yang diharapkan dari suatu sistem, yang menunjukkan bagaimana interaksi terjadi antara pengguna (aktor) dan sistem. Aktor dalam *Use Case* dapat berupa manusia atau sistem lain yang berperan dalam menjalankan suatu proses di dalam sistem.

Dengan demikian, dapat disimpulkan bahwa *Use Case* adalah gambaran aktivitas yang dilakukan sistem sebagai respons atas perintah dari pengguna dan digunakan untuk memahami bagaimana sistem bekerja dari sudut pandang pengguna. Diagram *Use Case* membantu menunjukkan interaksi antara aktor dan fungsi-fungsi yang tersedia dalam sistem. Proses penyusunannya diawali dengan rekayasa kebutuhan untuk mengetahui apa saja yang dibutuhkan pengguna dari sistem tersebut.

Tabel 2.1 Simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Berperan sebagai sistem lain individu, atau perangkat saat berinteraksi dengan <i>Use Case</i>
2		<i>Use Case</i>	<i>Use Case</i> menunjukkan aktivitas yang dilakukan antara aktor dengan sistem
3		<i>Association</i>	<i>Association</i> menunjukkan hubungan atau komunikasi antara aktor dengan <i>Use Case</i>
4		<i>Dependency</i>	Menunjukkan hubungan di mana perubahan pada suatu elemen independen akan berdampak pada elemen lain yang bergantung padanya dan tidak bersifat <i>independent</i>

No	Simbol	Nama	Keterangan
5		Generalisasi	Interaksi aturan-aturan dengan elemen lain
6		<i>Include</i>	Menunjukkan bahwa <i>Use Case</i> merupakan fungsionalitas dari <i>Use Case</i> lainnya
7		<i>Extend</i>	Menunjukkan bahwa <i>Use Case</i> merupakan tambahan fungsionalitas dari <i>Use Case</i> lainnya





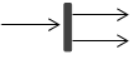
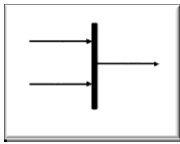

b. Activity Diagram

Activity diagram merupakan salah satu jenis diagram dalam *Unified Modeling Language (UML)* yang menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis (Musthofa & Adiguna, 2022). *Activity diagram* digunakan untuk menggambarkan aktivitas bisnis yang kompleks, yaitu digambarkan hubungan antara satu *Use Case* dengan *Use Case* lainnya. Musthofa & Adiguna (2022) juga menjelaskan bahwa *activity diagram* digunakan untuk menggambarkan alur proses dalam sistem secara menyeluruh dan menunjukkan hubungan antar aktivitas. Fokus utamanya adalah pada proses yang terjadi dalam sistem, bukan pada tindakan aktor. Diagram ini dibuat berdasarkan *Use Case* yang sudah ada dan membantu memahami bagaimana alur kerja sistem berlangsung.

Dengan demikian, dapat disimpulkan bahwa *activity diagram* merupakan jenis diagram UML yang digunakan untuk menggambarkan alur kerja atau proses dalam suatu sistem secara menyeluruh. Diagram

ini memvisualisasikan hubungan antar aktivitas dengan menggunakan simbol-simbol tertentu yang memiliki makna khusus. Fokus utama diagram ini adalah pada proses internal sistem, bukan tindakan pengguna, dan biasanya dibuat berdasarkan *Use Case* yang telah ditentukan sebelumnya. Pada tabel 2.2 berikut menjelaskan tentang simbol, nama, dan fungsi yang terdapat pada *activity diagram*.

Tabel 2.2 Simbol *Activity Diagram*

No	Simbol	Nama	Keterangan
1		<i>Initial Node</i>	Simbol untuk menunjukkan bagaimana objek dibentuk atau dimulai
2		<i>Activities</i>	Menunjukkan kumpulan aktivitas atau aksi
3		<i>Decision</i>	Digunakan untuk menggambarkan keputusan atau tindakan yang harus diambil
4		<i>Activity Final Node</i>	Simbol untuk menunjukkan bagaimana objek diakhiri
5		<i>Fork</i> (Percabangan)	Interaksi aturan-aturan dengan elemen lain
6		<i>Join</i> (Penggabungan)	Menunjukkan bahwa <i>Use Case</i> merupakan fungsionalitas dari <i>Use Case</i> lainnya
7		<i>Extend</i>	Menunjukkan bahwa <i>Use Case</i> merupakan tambahan fungsionalitas dari <i>Use Case</i> lainnya

c. *Sequence diagram*




Sequence diagram merupakan salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang menggambarkan interaksi antar objek di dalam dan di sekitar sistem. Kegunaan *sequence diagram*



adalah untuk menunjukkan rangkaian pesan yang dikirim antar *object* dan interaksi yang terjadi antar *object* (Musthofa & Adiguna, 2022).

Menurut Maulana et al. (2023), *Sequence Diagram* merupakan salah satu diagram UML yang menunjukkan urutan interaksi antar objek dalam sistem berdasarkan alur waktu, menggambarkan bagaimana objek saling berkomunikasi dalam menjalankan suatu proses. Sementara itu, Hendrawan et al. (2024) menjelaskan bahwa *Sequence Diagram* memvisualisasikan interaksi antar objek di dalam maupun di luar sistem melalui pesan-pesan yang disusun berdasarkan urutan waktu.

Dengan demikian, dapat disimpulkan bahwa *sequence diagram* adalah jenis diagram UML yang menggambarkan interaksi antar objek di dalam maupun di luar sistem melalui pengiriman pesan. Pada tabel 2.3 berikut menjelaskan tentang nama, simbol, dan fungsi yang terdapat pada *Sequence diagram*.

Tabel 2.3 Simbol *Sequence diagram*

No	Simbol	Nama	Keterangan
1		<i>Boundary class</i>	Simbol untuk menunjukkan bagaimana objek dibentuk atau dimulai
2		<i>Entity class</i>	Membentuk suatu deskripsi awal sistem yang dapat menjadi dasar penyusunan <i>Database</i>
3		<i>Control Class</i>	Bertanggung jawab terhadap kelas-kelas objek yang berisi logika

No	Simbol	Nama	Keterangan
4		<i>Activation</i>	Mewakili proses durasi aktivasi sebuah operasi
5		<i>LifeLine</i>	Objek entitas atau antarmuka yang berinteraksi dalam suatu diagram

6. *Website*

Website merupakan salah satu bentuk aplikasi berbasis *web* yang memungkinkan pengguna untuk mengakses informasi secara daring melalui jaringan internet. Penggunaan *website* telah berkembang luas dalam berbagai sektor, termasuk layanan publik, pendidikan, hingga promosi usaha kecil dan menengah.

Menurut Ambara et al. (2024), *website* tidak hanya menjadi media penyampaian informasi, tetapi juga sebagai sarana pelayanan masyarakat yang efektif, khususnya dalam konteks pemerintahan desa. Dalam studi mereka, pengembangan sistem informasi berbasis *web* di Desa Sibetan terbukti membantu meningkatkan transparansi dan keterjangkauan informasi.

Sejalan dengan itu, Gumiwang dan Santoso (2023) menekankan bahwa penggunaan *website* dalam konteks ketenagakerjaan, seperti sistem informasi pencarian kerja, mempermudah pelamar dalam mengakses lowongan secara cepat dan akurat. Desain sistem yang interaktif dan

responsif menjadi elemen kunci agar *website* dapat digunakan secara optimal oleh berbagai lapisan masyarakat.

Sementara itu, Di Kabupaten Karimun, *e-government* yang diintegrasikan di DPMPTSP memperkuat efektivitas dan efisiensi dalam layanan perizinan (Sulaiman et al., 2022). Hal ini mencerminkan pentingnya tata kelola informasi digital di sektor publik.

Dari berbagai penelitian tersebut, dapat disimpulkan bahwa *website* kini tidak hanya berfungsi sebagai alat penyebar informasi, melainkan telah berevolusi menjadi platform layanan yang mendukung berbagai aspek kehidupan masyarakat, mulai dari layanan publik hingga pendidikan dan pekerjaan.

7. PHP : Hypertext Preprocessor

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman skrip bersifat *server-side* yang secara luas digunakan untuk membangun aplikasi *web* yang dinamis dan interaktif. Sebagai bahasa pemrograman yang dijalankan di sisi *server*, PHP mengeksekusi perintah dan menghasilkan *output* dalam bentuk HTML yang kemudian dikirimkan ke sisi klien, yaitu *browser* pengguna. Dengan demikian, pengguna hanya melihat hasil pemrosesan tanpa mengetahui logika program yang ada di belakangnya.

PHP dirancang untuk terintegrasi langsung dengan HTML. Pengembang *web* sering menggunakan HTML sebagai kerangka antarmuka (*layout*), sementara PHP berperan dalam mengelola logika aplikasi dan koneksi ke basis data. Pemisahan komponen logika dan

tampilan dalam aplikasi *web*, seperti difasilitasi oleh pola MVC, memungkinkan pemeliharaan menjadi lebih mudah dan efisien (Softjourn, 2024).

Salah satu keunggulan PHP adalah kemudahannya dalam dipelajari dan digunakan. PHP memiliki sintaks yang sederhana, didukung oleh dokumentasi lengkap dan komunitas pengembang yang besar. Bahasa ini juga memiliki fleksibilitas tinggi, memungkinkan integrasi dengan berbagai sistem basis data seperti MySQL, PostgreSQL, dan SQLite. Selain itu, PHP juga memiliki banyak *framework* modern seperti *Laravel*, *CodeIgniter*, dan *Symfony* yang mempercepat proses pengembangan perangkat lunak.

Seiring berkembangnya teknologi, PHP terus mengalami pembaruan. Versi terbaru PHP 8.0 ke atas membawa peningkatan signifikan dalam hal performa dan keamanan, seperti penerapan *Just-in-Time (JIT) compiler*, properti *readonly*, dan dukungan untuk *union types*. Hal ini menunjukkan bahwa PHP tetap relevan dan kompetitif di tengah munculnya berbagai bahasa pemrograman *web* modern lainnya.

8. *Framework Laravel*

Laravel adalah salah satu *framework* PHP yang paling populer dan banyak digunakan untuk pengembangan aplikasi *web*. Dikenal karena sintaksisnya yang elegan dan kemudahan penggunaannya, *Laravel* dirancang untuk membantu pengembang membangun aplikasi *web* yang bersih dan terstruktur dengan baik. Sinlae et al. (2024) mengungkapkan

bahwa *Laravel* menyediakan dokumentasi lengkap dan didukung komunitas aktif yang memudahkan pengembang menemukan solusi saat *coding* dan mempercepat proses pengembangan, sehingga cocok digunakan pada proyek yang menuntut efisiensi dan keandalan.

Laravel pertama kali diperkenalkan oleh Taylor Otwell pada tahun 2011. Sejak saat itu, *Laravel* telah mengalami banyak pembaruan dan peningkatan, dengan versi terbaru yang dirilis secara berkala. *Laravel* memiliki komunitas yang aktif dan banyak dokumentasi yang mendukung pengembang dalam mempelajari dan menggunakan *framework* ini (*Laravel*, 2023).

Dalam penggunaannya, *Framework Laravel* memiliki beberapa keuntungan ketika digunakan untuk pengembangan *website*, diantaranya adalah:

- a. *Website* menjadi lebih *scalable* (mudah dalam pengembangannya).
- b. Terdapat *namespace* dan tampilan yang dapat membantu untuk mengorganisir dan mengatur sumber daya *website*.
- c. Proses pengembangan menjadi lebih cepat sehingga dapat menghemat waktu karena *Laravel* dapat dikombinasikan dengan beberapa komponen dari *framework* lain.

Laravel merupakan salah satu *framework* PHP yang populer karena kemudahan dalam pengembangan aplikasi *web*. Namun, selain keunggulannya, *Laravel* juga memiliki beberapa keterbatasan teknis. Salah satunya adalah struktur folder dan file yang kompleks, terutama bagi

pengembang pemula. Struktur *laravel* memberikan fleksibilitas bagi pengembangan aplikasi skala besar, namun untuk proyek kecil, kerumitan dan banyaknya konfigurasi dapat menjadi beban (NextGen Inovasi Indonesia, 2023).

Selain itu, proses instalasi *Laravel* sangat bergantung pada *Composer*, sebuah *dependency manager* yang membutuhkan koneksi internet untuk mengunduh seluruh *library* dan paket pendukung. Jika koneksi internet tidak stabil atau terputus, proses instalasi dapat terganggu, bahkan gagal sepenuhnya (NextGen Inovasi Indonesia, 2023). Kondisi ini menyulitkan pengembang pemula yang belum terbiasa dengan sistem manajemen paket berbasis *command-line*.

Laravel juga memerlukan *server* lokal seperti XAMPP, Laragon, atau Nginx untuk menjalankan aplikasi serta mengelola koneksi *Database* melalui ekstensi seperti PDO atau MySQL. Hal ini memperpanjang proses konfigurasi awal yang dapat menjadi tantangan tersendiri, terutama pada perangkat yang belum terinstal *software* pendukung. Meskipun *Laravel* menyediakan *built-in server* (*php artisan serve*) untuk keperluan pengujian, pengguna tetap harus melakukan konfigurasi lebih lanjut untuk *deployment* atau integrasi dengan sistem *Database*.

Secara keseluruhan, keterbatasan *Laravel* ini tidak menjadikannya kurang layak digunakan, tetapi menuntut pengembang untuk memiliki pemahaman teknis dasar yang cukup agar dapat memanfaatkan *framework* ini secara optimal.

9. *Flowchart*

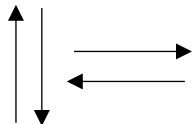




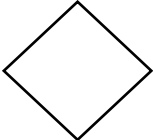
Diagram alir atau *flowchart* adalah representasi visual dari urutan langkah-langkah dan keputusan yang diperlukan untuk menjalankan suatu proses dalam sebuah program. Setiap langkah direpresentasikan dalam bentuk diagram dan dihubungkan oleh garis atau panah yang menunjukkan arah alur. *Flowchart* memiliki peran krusial dalam menentukan langkah-langkah atau fungsi dari suatu proyek pembuatan program yang melibatkan berbagai pihak secara bersamaan. Selain itu, menggunakan diagram alur untuk merinci proses program dapat memberikan kejelasan, kesingkatan, dan mengurangi risiko penafsiran yang salah. Penerapan *flowchart* dalam konteks pemrograman merupakan metode yang efektif untuk menghubungkan aspek teknis, seperti logika dan alur program, dengan aspek nonteknis, seperti pemahaman konsep dan komunikasi antar anggota tim.


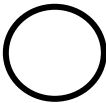

Flowchart sistem menunjukkan proses pengolahan data lengkap dengan media *input*, *output*, dan penyimpanan. Dalam membuat *flowchart*, sangat penting untuk mengikuti aturan seperti arah alur dari atas ke bawah, kejelasan aktivitas, serta penggunaan simbol standar. Menurut Firmansyah et al. (2023), *flowchart* digunakan untuk menjelaskan aliran aktivitas dalam sebuah organisasi, berupa bagan yang mencakup aktivitas manual dan aliran dokumen yang digunakan dalam sistem. Umumnya *flowchart* terdiri dari tiga bagian utama, yaitu *input*, *output*, dan proses. *Flowchart* memiliki beberapa karakteristik, yaitu menggunakan simbol sebagai

representasi tugas, hanya memuat perintah penting, dan efektif untuk merancang program secara terstruktur.

Berdasarkan uraian di atas, dapat disimpulkan bahwa *flowchart* adalah representasi grafis yang menunjukkan langkah-langkah dalam suatu program atau sistem secara terstruktur, menggunakan simbol khusus untuk menggambarkan alur proses. Berikutnya pada tabel 2.4 akan dijelaskan tentang nama, simbol, dan fungsi yang terdapat pada *flowchart*.

Tabel 2.4 Simbol-Simbol *Flowchart* dan Penjelasannya

No	Simbol	Nama	Keterangan
1		<i>Flow line</i>	Menghubungkan simbol-simbol dalam diagram.
2		<i>Terminal</i>	Menandai awal dan akhir dari rangkaian proses komputasi
3		<i>Preparation</i>	Menunjukkan proses inisialisasi.
4		<i>Input/Output</i>	Menampilkan proses <i>input</i> atau <i>output</i> data.
5		<i>Process</i>	Menunjukkan proses yang dilakukan sistem komputer.
6		<i>Decision</i>	Memilih proses berdasarkan kondisi yang ada.

No	Simbol	Nama	Keterangan
7		<i>On-line Storage</i>	Menyatakan <i>input</i> yang berasal dari <i>disk</i> atau disimpan ke <i>disk</i> .
8		<i>On-page connector</i>	Menghubungkan bagian-bagian dari <i>flowchart</i> yang dilanjutkan ke halaman yang sama.
9		<i>Off-page connector</i>	Menghubungkan bagian-bagian dari <i>flowchart</i> dilanjutkan ke halaman terpisah.

Sumber : (Zalukhu et al., 2024)

10. Database

a. Pengertian Database

Database adalah tempat menyimpan data dalam jumlah besar yang mencakup data operasional serta uraian mengenai data tersebut. Menurut Aulia et al. (2023), efektivitas sistem *database* dalam sistem informasi manajemen sangat mempengaruhi kualitas keseluruhan, sehingga pengelolaan *database* secara baik berkontribusi terhadap efisiensi operasional.

Efisiensi sangat penting dalam desain *Database* karena terlalu banyak redundansi data akan mengurangi efektivitas. *Database* juga harus digunakan oleh seluruh bagian organisasi sebagai pusat data bersama, bukan hanya dimiliki oleh satu departemen. Untuk pemanfaatan optimal, penggunaan *Database Management System* (DBMS) diperlukan agar pengelolaan data berjalan dengan baik.

b. MySQL

Salah satu sistem manajemen *Database* yang populer adalah MySQL, yang dinilai ringan dan kompatibel dengan berbagai platform pengembangan. MySQL mendukung operasi data secara cepat dan aman, menjadikannya pilihan utama dalam berbagai pengembangan aplikasi web. Bahasa ini digunakan pula oleh sistem seperti *SQL Server*, *Oracle*, dan *PostgreSQL*. Dalam pembangunan aplikasi berbasis bahasa pemrograman PHP, MySQL merupakan *Database server* yang paling umum dimanfaatkan. MySQL berfungsi menyimpan data dalam *Database* dan memanipulasi data dalam *Database*. Bentuk manipulasi data yang bisa dilakukan di dalam MySQL diantaranya adalah menambah, mengubah, dan menghapus data.

11. Sistem Informasi Geografis

Pengertian *Geographic Information System* atau Sistem Informasi Geografis (SIG) sangatlah beragam. Hal ini terlihat dari banyaknya definisi SIG yang beredar di berbagai sumber pustaka. SIG merupakan suatu bidang kajian ilmu dan teknologi yang digunakan oleh berbagai bidang atau disiplin ilmu, dan berkembang dengan cepat. SIG adalah sistem berbasis komputer yang digunakan untuk mengelola dan menganalisis data yang berhubungan dengan lokasi geografis. Dalam sistem pelaporan, SIG sangat membantu untuk menampilkan lokasi kejadian secara akurat. Menurut Mentik et al (2024), perangkat lunak SIG seperti QGIS sangat bermanfaat untuk pemetaan dan analisis spasial yang

dapat mendukung pengambilan keputusan strategis. Sistem Informasi Geografis (SIG) digunakan secara luas di bidang pemerintahan untuk merancang kebijakan publik yang lebih tepat sasaran, misalnya dalam pemeliharaan infrastruktur jalan di Kabupaten Langkat Hilir (Irfandi et al., 2023).

B. Kajian Empiris

Kajian empiris merupakan bagian penting dalam penelitian ilmiah yang menyajikan hasil-hasil penelitian terdahulu yang relevan. Kajian ini bertujuan untuk memberikan landasan nyata mengenai bagaimana penelitian sejenis telah dilakukan, serta memberikan perbandingan dan referensi terhadap penelitian yang sedang dilakukan. Berikut ini adalah beberapa kajian empiris yang relevan:

1. **Kusuma et al. (2023)** Penelitian ini menunjukkan bahwa sistem informasi pengaduan masyarakat berbasis *web* dapat meningkatkan efektivitas pelayanan publik. Dalam studi ini, sistem “Lapor Wak Dul” yang dirancang menggunakan metode *Waterfall* berhasil memberikan kemudahan dalam penyampaian laporan serta meningkatkan transparansi proses penanganan aduan. Salah satu hasil penting dari penelitian ini adalah kemampuan sistem untuk memberikan notifikasi *real-time* kepada petugas, sehingga laporan dapat ditindaklanjuti dengan lebih cepat dan terstruktur. Hal ini menunjukkan bahwa sistem berbasis *web* tidak hanya mempermudah akses masyarakat, tetapi juga meningkatkan akuntabilitas dan efisiensi proses pelayanan.

2. **Erlangga et al. (2023)** Penelitian ini merancang sebuah sistem informasi layanan pengaduan masyarakat berbasis *web* yang bertujuan untuk meningkatkan pelayanan publik di Dinas Sosial Kabupaten Pasaman Barat. Dengan adanya sistem ini, masyarakat dapat dengan mudah menyampaikan aduan, dan instansi terkait dapat mengelola serta menindaklanjuti laporan secara digital. Hasil dari perancangan ini menunjukkan bahwa penggunaan *platform* berbasis *web* berhasil menciptakan alur komunikasi yang lebih efisien dan transparan antara masyarakat dan pemerintah daerah.
3. **Ikhwan dan Lubis (2023)** Penelitian ini mengembangkan sistem informasi pengaduan masyarakat untuk Dinas ESDM Provinsi Sumatera Utara. Dengan menggunakan pendekatan berbasis *web* dan metode *Waterfall*, sistem ini mampu meningkatkan efisiensi dalam manajemen laporan masyarakat. Hasil kajian menunjukkan bahwa data laporan masyarakat dapat dikategorikan dan ditindaklanjuti secara lebih terstruktur. Selain itu, pengelolaan laporan menjadi lebih mudah diakses oleh petugas dan pimpinan, sehingga proses pengambilan keputusan menjadi lebih cepat dan tepat. Sistem ini juga berkontribusi dalam menciptakan pelayanan publik yang lebih responsif dan akuntabel.
4. **Zunidar dan Wiyono (2023)** Kajian ini membahas pengembangan sistem informasi pengaduan masyarakat di Kecamatan Rajeg. Sistem yang dikembangkan menggantikan proses manual dengan formulir kertas menjadi platform digital yang terintegrasi. Hasil penelitian

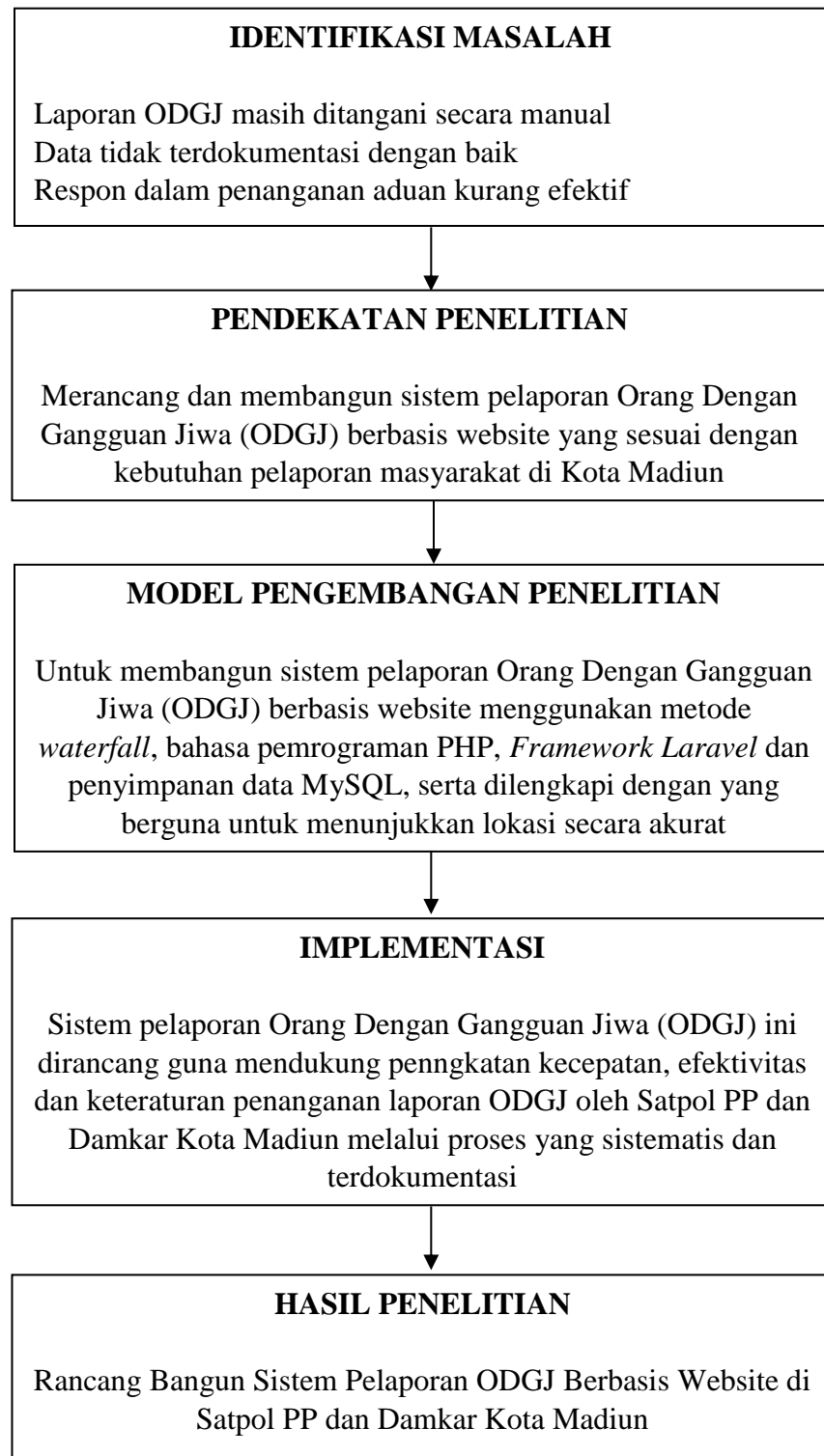
menunjukkan adanya peningkatan dalam efektivitas pengelolaan aduan, baik dari segi kecepatan maupun akurasi data. Selain itu, sistem ini memungkinkan koordinasi yang lebih baik antar bagian dalam kecamatan, sehingga penanganan aduan menjadi lebih optimal. Implementasi sistem juga meningkatkan dokumentasi laporan dan meminimalisasi risiko kehilangan data.

5. **Kurniasih dan Mulyono (2022)** Penelitian ini menekankan pentingnya digitalisasi layanan aduan di tingkat desa, khususnya di Desa Ladang Peris. Sistem yang dikembangkan berbasis *web* ini memberikan kemudahan bagi masyarakat dalam menyampaikan keluhan, sekaligus memfasilitasi perangkat desa dalam melakukan pencatatan, pengolahan, dan pelaporan masalah secara digital. Salah satu temuan penting dari penelitian ini adalah peningkatan akurasi dan kecepatan dalam proses pelaporan dan penyampaian tindak lanjut kepada masyarakat. Sistem ini juga berfungsi sebagai media transparansi dan partisipasi warga dalam pengawasan pelayanan publik di lingkup desa.

Berdasarkan kajian-kajian empiris terbaru, pemanfaatan sistem informasi berbasis *web* dalam layanan pengaduan masyarakat terbukti efektif dalam meningkatkan kualitas pelayanan publik. Sistem digital tidak hanya mempercepat respon petugas, tetapi juga memberikan kemudahan dalam dokumentasi laporan secara terstruktur dan transparan. Beberapa penelitian menunjukkan bahwa integrasi fitur pelaporan *real-time*, notifikasi otomatis, serta pelacakan status aduan mampu mendorong efisiensi kerja dan

memperkuat akuntabilitas instansi terkait. Selain itu, pemanfaatan teknologi geospasial seperti GPS dan peta interaktif mendukung validasi lokasi kejadian secara akurat, yang sangat penting dalam menangani kasus-kasus sosial mendesak, termasuk laporan Orang Dengan Gangguan Jiwa (ODGJ). Dalam konteks Kota Madiun, kebutuhan akan sistem pelaporan ODGJ yang cepat, tepat, dan terkoordinasi lintas lembaga menjadi semakin mendesak. Oleh karena itu, pengembangan sistem pelaporan ODGJ berbasis *website* pada lingkungan Satpol PP dan Damkar sangat relevan dan didukung oleh temuan-temuan penelitian sebelumnya. Sistem ini diharapkan mampu menjadi solusi strategis untuk mempercepat penanganan kasus di lapangan melalui pelaporan digital, pemetaan lokasi, dan dokumentasi yang tertata dengan baik.

C. Kerangka Berpikir Penelitian



Gambar 2.1 Kerangka Berpikir Penelitian