

BAB II

KAJIAN PUSTAKA

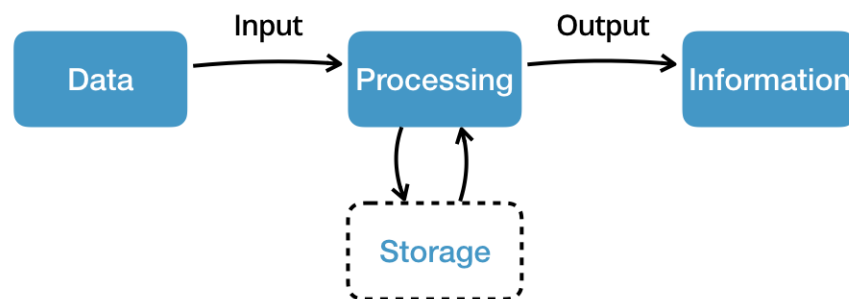
A. Kajian Teoritis

1. Rancang Bangun Sistem Informasi.

Secara terminologi, "rancang bangun" terdiri dari dua kata, yaitu rancang dan bangun. Perancangan merupakan tahapan awal di mana pengembang menerjemahkan spesifikasi kebutuhan pengguna ke dalam sebuah model alur kerja dan basis data, sedangkan pembangunan atau implementasi adalah proses menerjemahkan desain tersebut ke dalam bahasa pemrograman agar menjadi perangkat lunak yang fungsional. Dengan demikian, menurut Al-rizqi et al. (2025), rancang bangun sistem informasi pada hakikatnya adalah proses perancangan dan implementasi aplikasi berbasis teknologi yang bertujuan untuk mengelola dan menyampaikan informasi secara otomatis, akurat, dan terstruktur.

Sebuah sistem informasi tidak hanya sebatas pengembangan perangkat lunak semata, melainkan sebuah kombinasi yang terintegrasi secara utuh antara teknologi informasi, manusia selaku entitas pengguna, dan prosedur kerja operasional. Secara arsitektural, rancang bangun sistem informasi mentransformasikan data masukan (*input*) melalui suatu pemrosesan terkomputerisasi menjadi sebuah keluaran (*output*) informasi yang bernilai, seperti yang diilustrasikan pada Gambar 2.1. Dalam konteks kegiatan operasional rutin seperti

administrasi dan penjadwalan, sistem informasi dibangun untuk menggantikan metode pengelolaan konvensional dan manual yang sangat rentan terhadap kesalahan manusia (*human error*) serta memakan waktu penyusunan yang lama, menjadi sebuah sistem terkomputerisasi yang jauh lebih sistematis dan optimal (Pratiwi et al, 2023). Seperti yang diilustrasikan pada Gambar 2.1.



Gambar 2.1 Ilustrasi Komponen Utama Sistem Informasi

Sejalan dengan hal tersebut, Herlambang et al. (2025) menegaskan bahwa rancang bangun sistem informasi yang diimplementasikan menggunakan platform berbasis situs web (*website*) memiliki keunggulan kompetitif yang sangat tinggi pada aspek aksesibilitas. Pengembangan sistem informasi berbasis web ini bertujuan utama untuk meningkatkan kualitas dan efisiensi pelayanan secara keseluruhan, memastikan bahwa pengelolaan data dan penyampaian informasi dapat diakses secara transparan, mudah, dan *real-time* oleh seluruh elemen masyarakat maupun pengguna yang memiliki hak akses.

Berdasarkan kajian dari berbagai literatur di atas, dapat ditarik kesimpulan bahwa rancang bangun sistem informasi merupakan proses komprehensif, mulai dari tahap desain hingga eksekusi kode program. Untuk menciptakan sebuah perangkat lunak yang fungsional. Peralihan dari sistem konvensional menuju sistem informasi berbasis *web* terbukti mampu mengatasi kelemahan prosedur manual, menekan tingkat kesalahan operasional, serta mempercepat distribusi data secara *real-time*. Dalam konteks penelitian ini, pendekatan tersebut menjadi landasan teoretis yang sangat kuat untuk merancang sistem penjadwalan ibadah yang modern, transparan, dan mampu meningkatkan efektivitas pelayanan secara berkelanjutan.

2. Sistem Informasi Manajemen Pelayanan.

Menurut Masikhoh et al. (2025), sistem informasi manajemen pelayanan merupakan sebuah kerangka kerja terkomputerisasi yang dirancang untuk mengelola, mencatat, dan melacak berbagai proses layanan agar berjalan secara lebih efektif dan efisien. Pemanfaatan sistem ini, khususnya yang diintegrasikan dengan teknologi berbasis *web*, mampu meningkatkan kualitas layanan secara keseluruhan dengan menyederhanakan proses administratif operasional serta memperkuat hubungan interaktif antara penyedia layanan dengan penggunanya.

Raehaneh (dalam Trivalni & Eryando, 2023) mengemukakan dalam konteks pelayanan yang melibatkan agenda atau penjadwalan rutin, sistem informasi manajemen memegang peranan yang sangat

krusial untuk mencegah terjadinya tumpang tindih tugas atau risiko kehilangan jadwal akibat kelalaian pencatatan manual. Siklus operasional pada manajemen penjadwalan ini secara umum mengikuti alur pendataan, alokasi, dan distribusi, seperti yang direpresentasikan pada Gambar 2.2. Sistem ini didesain agar manajemen waktu dapat dilakukan secara tersentralisasi, sehingga penyusunan tugas dapat disesuaikan dengan ketersediaan masing-masing pelaksana tugas maupun anggota yang melayani secara presisi.

Lebih jauh lagi, Wijaya et al. (2024) menyatakan bahwa implementasi sistem informasi manajemen dalam ranah pelayanan maupun keorganisasian terbukti memberikan dampak positif yang signifikan dalam hal transparansi informasi dan akuntabilitas. Dengan sistem yang terdigitalisasi, anggota atau pengguna layanan dapat mengakses informasi mengenai prosedur, jadwal, serta status kegiatan operasional secara *real-time*. Hal ini tidak hanya mempercepat proses interaksi administratif, tetapi juga memperkuat koordinasi antar departemen dan menekan tingkat keterlambatan dalam pelaksanaan rangkaian kegiatan pelayanan.

Berdasarkan berbagai pandangan empiris tersebut, dapat disintesis bahwa sistem informasi manajemen pelayanan adalah instrumen digital yang esensial untuk mengatur operasional penjadwalan, mencegah tumpang tindih penugasan, dan mendistribusikan data secara transparan. Kehadiran sistem ini menjamin

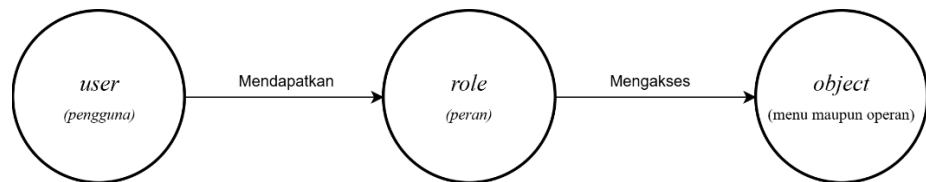
sinkronisasi yang terstruktur antara pihak pengelola dan pelaksana tugas. Dalam kerangka penelitian ini, konsep tersebut diadaptasi sebagai landasan utama untuk membangun sistem manajemen penjadwalan ibadah di Gereja Masehi Advent Hari Ketujuh Kalasan, yang bertujuan untuk mentransformasi pencatatan berbasis *spreadsheet* yang rentan kesalahan menjadi alur kerja yang terpusat, sistematis, dan mudah diakses oleh seluruh jemaat.

3. Aplikasi Berbasis Web dan Role-Based Access Control (RBAC)

Menurut Arbiansyah et al. (2023), kendala utama dalam manajemen administrasi konvensional adalah ketidakefektifan sistem yang masih bersifat manual. Penggunaan aplikasi berbasis *web* yang terintegrasi dengan basis data dapat membantu mengefektifkan proses pengelolaan data secara aman. Lebih lanjut, implementasi sistem berbasis *web* ini tidak hanya mengefektifkan proses perekaman dan temu kembali informasi, melainkan juga mengefektifkan pengolahan data serta akses penyajian informasi kapan saja dan di mana saja. Namun, kemudahan akses yang ditawarkan oleh jaringan internet ini juga berisiko menimbulkan celah kerentanan baru jika tidak dibarengi dengan kontrol keamanan yang tepat. Oleh karena itu, diperlukan sebuah mekanisme kontrol akses (*access control*) yang ideal dan tangguh guna membatasi kewenangan pengguna di dalam sistem.

Secara fundamental, RBAC adalah sebuah sistem keamanan perangkat lunak yang mendukung pengaturan hak akses secara spesifik

berdasarkan peran operasional masing-masing entitas pengguna di dalam suatu organisasi. Model ini memungkinkan setiap modul, fitur, maupun informasi di dalam aplikasi *web* diatur batasannya secara eksplisit. Melalui arsitektur RBAC, pengguna hanya diizinkan untuk mengaktifkan fungsi dan melihat antarmuka yang relevan dengan tanggung jawabnya Rahman (dalam Firmansyah et al. 2025). Adapun ilustrasi konseptual dari pembatasan otorisasi hak akses pada sistem RBAC dapat dilihat pada Gambar 2.2.



Gambar 2.2 Kerangka Konseptual *Role-Based Access Control* (RBAC)
Sumber : (Yuricha & Phan, 2023).

Lebih lanjut, Sahyudi & Susanto. (2025) menjelaskan bahwa dalam arsitekturnya, model keamanan RBAC mengadopsi prinsip hak istimewa terkecil (*least privilege*), di mana sistem secara cerdas hanya memberikan otorisasi seminimal mungkin yang mutlak diperlukan oleh pengguna untuk menjalankan tugasnya. Pendekatan pembatasan peran berbasis *least privilege* ini secara signifikan mampu meningkatkan efisiensi tata kelola akses dan meminimalisasi risiko penyusupan, kebocoran, atau manipulasi data secara ilegal di dalam basis data.

Berdasarkan sintesis dari berbagai literatur di atas, dapat disimpulkan bahwa aplikasi berbasis *web* yang diintegrasikan dengan arsitektur keamanan RBAC merupakan solusi komprehensif untuk

menciptakan perangkat lunak yang fleksibel sekaligus terlindungi. Pendekatan RBAC menjamin tidak adanya tumpang tindih kewenangan melalui pemisahan batas peran yang tegas. Dalam konteks rancang bangun sistem penjadwalan ibadah di Gereja Masehi Advent Hari Ketujuh Kalasan, konsep ini diimplementasikan secara langsung untuk mengisolasi ruang kerja operasional antara Admin, Sekretaris, dan Departemen Komunikasi, sehingga kerahasiaan data jemaat serta integritas manipulasi jadwal pelayanan dapat terjamin keamanannya secara maksimal.

4. Notifikasi WhatsApp Click-to-Chat.

Menurut Nago & Fallo. (2022), aplikasi *WhatsApp* telah berevolusi menjadi platform media sosial dan komunikasi yang paling banyak digunakan oleh berbagai kalangan masyarakat. Keunggulan utama dari platform ini adalah pemanfaatan jaringan internet yang membuatnya jauh lebih hemat dan praktis dibandingkan layanan pesan singkat (SMS) konvensional. Dalam konteks sistem informasi, integrasi dengan *WhatsApp* menjadikan proses penyampaian pemberitahuan maupun distribusi informasi menjadi jauh lebih efektif, cepat, dan mudah diakses oleh penerima kapan saja dan di mana saja.

Lebih lanjut, Sinaga et al. (2025) menjelaskan bahwa sistem pengingat (*reminder*) pada dasarnya merupakan sebuah sistem pesan yang dirancang untuk membantu seseorang dalam mengingat suatu hal. Pemanfaatan sistem pengingat digital ini berfungsi sebagai instrumen

bantu bagi manusia dalam mengingat informasi penting atau melakukan suatu tindakan tertentu agar tidak terlupakan. Kebermanfaatan dari sistem pengingat ini akan jauh lebih optimal apabila terdapat integrasi informasi kontekstual yang memungkinkan penyajian pemberitahuan atau pesan secara presisi pada waktu dan tempat yang tepat.

Sejalan dengan hal tersebut, Fatah et al. (2025) menjelaskan bahwa sebagai alternatif yang efisien dan tanpa biaya tambahan, pengembang sistem dapat memanfaatkan fitur bawaan *WhatsApp Click-to-Chat*. Fitur ini memungkinkan integrasi antara sistem informasi berbasis web dengan platform *WhatsApp* tanpa perlu bergantung pada layanan *Application Programming Interface (API)* pihak ketiga yang berbayar. Mekanismenya bekerja secara *client-side* dengan cara menghasilkan tautan URL secara dinamis yang telah disisipkan parameter nomor telepon dan teks isi pesan, sehingga pengirim hanya perlu menekan tombol kirim tanpa harus menyimpan nomor kontak penerima terlebih dahulu.

Berdasarkan sintesis dari berbagai literatur di atas, dapat disimpulkan bahwa fitur *WhatsApp Click-to-Chat* merupakan solusi notifikasi yang paling efektif, hemat biaya, dan familiar bagi masyarakat umum. Pemanfaatan fitur ini memungkinkan pengiriman pesan pengingat secara instan dengan teks yang sudah terisi otomatis (*pre-filled message*). Dalam kerangka penelitian rancang bangun sistem di Gereja Masehi Advent Hari Ketujuh Kalasan, konsep ini

diimplementasikan untuk menjembatani komunikasi antara pengelola jadwal dengan anggota jemaat, sehingga setiap penugasan pelayanan ibadah dapat diinformasikan secara langsung ke perangkat seluler anggota guna mencegah terjadinya kelalaian tugas.

5. **Framework Laravel dan Basis Data.**

Pengembangan sistem informasi yang dinamis dan terstruktur membutuhkan kerangka kerja (*framework*) yang tangguh dan memiliki skalabilitas tinggi, salah satunya adalah Laravel. Laravel merupakan *framework* berbasis bahasa pemrograman PHP yang dirancang secara khusus untuk mempermudah dan mempercepat proses siklus hidup pengembangan aplikasi web (Nugraha et al., 2024). Penggunaan kerangka kerja ini menawarkan ekosistem pengembangan yang sangat komprehensif, mulai dari sistem perutean (*routing*) yang rapi, perlindungan keamanan terintegrasi terhadap serangan siber, hingga sintaks penulisan kode yang ekspresif, sehingga pengembang dapat berfokus pada penyelesaian logika bisnis operasional tanpa harus membangun sistem dasar dari nol.

Aipina & Witriyono. (2022) menjelaskan bahwa untuk memaksimalkan kinerja aplikasi *web* dan mengoptimalkan siklus pengembangan sistem, para pengembang perangkat lunak saat ini secara fundamental mengadopsi arsitektur *Model-View-Controller* (MVC) pada *framework* modern seperti Laravel. Konsep MVC ini merupakan sebuah pendekatan rekayasa perangkat lunak yang memisahkan secara

tegaskan antara logika pemrosesan data (*Model*), antarmuka pengguna (*View*), dan pusat kendali sistem (*Controller*).

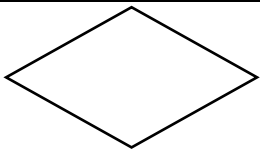
Lebih lanjut, Rifqi et al. (2025) menegaskan bahwa pemanfaatan kerangka kerja berbasis MVC ini memungkinkan pengembangan aplikasi yang jauh lebih terstruktur dan terlindungi dari celah keamanan. Sistem ini dibekali dengan fitur mutakhir seperti sistem perutean dinamis dan *Eloquent Object-Relational Mapping* (ORM) yang secara signifikan menyederhanakan proses penulisan kueri basis data relasional. Pendekatan ini tidak hanya mempercepat proses penulisan kode program, tetapi juga menjamin bahwa manipulasi serta pengelolaan data berskala besar dapat berjalan dengan efisien dan optimal.

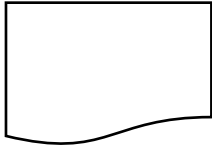
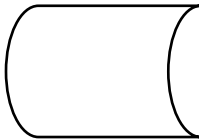
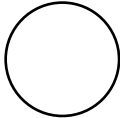
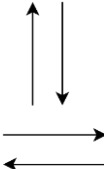
Berdasarkan sintesis dari ketiga literatur di atas, dapat disimpulkan bahwa sinergi antara, kerangka kerja Laravel dan basis data MySQL merupakan standar fondasi teknologi yang sangat tangguh untuk membangun perangkat lunak modern. Pemisahan logika melalui arsitektur MVC menjamin sistem lebih mudah dipelihara dan dikembangkan secara berkelanjutan. Dalam konteks rancang bangun sistem penjadwalan ibadah di Gereja Masehi Advent Hari Ketujuh Kalasan, kombinasi teknologi ini diimplementasikan untuk mengelola kompleksitas data jemaat, memproses komputasi algoritma *Smart Auto-Fill*, serta menangani lalu lintas pertukaran data secara aman, responsif, dan terstruktur.

6. Perancangan Sistem.

Menurut Yuliansyah et al. (2024), dalam merancang alur logika program secara mendetail, pengembang sistem umumnya menggunakan diagram alir atau *flowchart*. *Flowchart* merupakan representasi visual yang terdiri atas simbol-simbol terstandarisasi yang berfungsi untuk menunjukkan bagan aliran proses yang saling terhubung secara logis beserta pendefinisian fungsinya. Adapun simbol-simbol yang digunakan dalam perancangan *flowchart* dijabarkan pada Tabel 2.1.

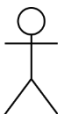
Tabel 2.1 Simbol Flowchart

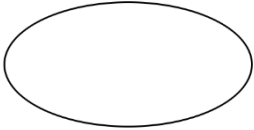
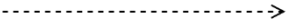
Simbol	Nama	Fungsi
	<i>Process</i>	Menunjukkan suatu proses digunakan untuk melambangkan perhitungan dan perubahan nilai peubah
	<i>Input / Output</i>	Menyatakan suatu proses masukan (<i>input</i>) atau keluaran (<i>output</i>) data tanpa ketergantungan terhadap peralatannya
	<i>Teriminator</i>	Menunjukkan awal atau akhir suatu tahapan dalam <i>flowchart</i>
	<i>Decision</i>	Menyatakan pengujianterhadap suatu kondisi tertentu yang akan menghasilkan

	<i>Document</i>	Menunjukkan dokumen yang dihasilkan atau dibutuhkan dalam proses.
	<i>Stored Data</i>	Menggambarkan alur penyimpanan data
	<i>On-Page Connector</i>	Menyatakan keterkaitan antara satu proses dengan proses lainnya dalam satu halaman yang serupa
	<i>Flow</i>	Menggambarkan bagaimana jalanya suatu arus proses

Selain alur logika dasar, Mahfud et al. (2024) mengemukakan bahwaperancangan sistem berorientasi objek dimodelkan menggunakan *Unified Modeling Language* (UML), yang salah satunya adalah *use case diagram*. Diagram ini berfungsi secara spesifik untuk menggambarkan batasan fungsionalitas sistem serta bentuk interaksi operasional yang terjadi antara aktor (pengguna) dengan perangkat lunak yang dibangun. Standarisasi simbol pada *use case diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol *Use Case Diagram*

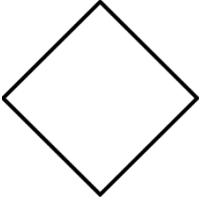
Simbol	Nama	Fungsi
	<i>Actor</i>	Mewakili entitas eksternal (pengguna) yang berinteraksi dengan sistem.

	<i>Use Case</i>	Merepresentasikan fungsi atau layanan yang disediakan oleh sistem.
	<i>Association</i>	Menggambarkan jalur komunikasi antara aktor dengan <i>use case</i> .
	<i>Include / Extend</i>	Menunjukkan relasi ketergantungan atau perluasan fungsi antar <i>use case</i> .

Untuk memetakan prosedur kerja secara lebih rinci, Aryandra & Wardhana. (2024) menjelaskan bahwa pemodelan dilanjutkan menggunakan *activity diagram*. Diagram ini merepresentasikan urutan aktivitas, pergerakan alur kontrol, hingga titik pengambilan keputusan logika (*decision*) dari sebuah proses bisnis yang berjalan di dalam sistem. Simbol-simbol yang digunakan dalam pembentukan *activity diagram* dijelaskan pada Tabel 2.3.

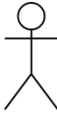
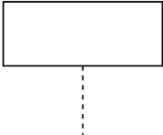
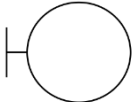
Tabel 2.3 Simbol *Activity Diagram*

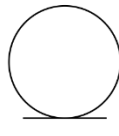
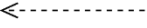
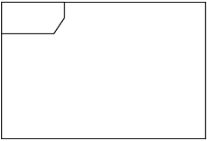
Simbol	Nama	Fungsi
	<i>Initial Node</i>	Menandakan titik awal dimulainya suatu aliran proses kerja.
	<i>Activity</i>	Merepresentasikan suatu aktivitas atau pekerjaan yang dilakukan sistem.

	<i>Decision</i>	Titik percabangan logika yang menghasilkan lebih dari satu kemungkinan alur.
	<i>Final Node</i>	Menandakan titik akhir dari keseluruhan aliran proses pada sistem.

Lebih lanjut, Dalis et al (2024) menjabarkan bahwa pemodelan perilaku sistem antar objek digambarkan melalui *sequence diagram*. Diagram urutan ini secara detail mengilustrasikan siklus hidup interaksi serta pertukaran pesan (*message*) antar entitas di dalam sistem yang direpresentasikan dalam urutan waktu eksekusi tertentu Adapun definisi simbol pada *sequence diagram* ditunjukkan pada Tabel 2.4.

Tabel 2.4 Simbol *Sequence Diagram*

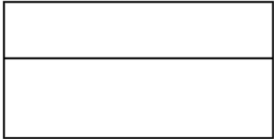
Simbol	Nama	Fungsi
	<i>Actor</i>	Mewakili pengguna yang memicu urutan interaksi pengiriman pesan.
	<i>Lifeline</i>	Menggambarkan siklus hidup dari sebuah objek selama interaksi berlangsung.
	<i>Boundary Class</i>	Mewakili View atau antarmuka sistem tempat aktor berinteraksi (contoh: halaman website atau formulir).

	<i>Control Class</i>	Mewakili <i>Controller</i> yang memuat logika program untuk memproses aksi dan mengoordinasikan aliran pesan.
	<i>Entity Class</i>	Mewakili Model atau entitas basis data yang digunakan untuk menyimpan atau mengambil data sistem.
	<i>Activation</i>	Menunjukkan durasi waktu atau periode saat sebuah objek sedang aktif melakukan eksekusi proses atau operasi.
	<i>Message</i>	Menunjukkan pengiriman pesan atau instruksi dari satu objek ke objek lainnya.
	<i>Return Message</i>	Menunjukkan pesan kembalian atau hasil balasan dari objek yang telah diproses.
	<i>Combined Fragment</i>	Membingkai dan mengelompokkan serangkaian pesan yang memiliki kondisi logika tertentu, seperti <i>loop</i> (perulangan) atau <i>alt</i> (percabangan IF-ELSE).

Terakhir, pada tahap perancangan basis data dan struktur fisik aplikasi, Khairunnisa & Voutama. (2024) menjelaskan bahwa pemodelan divisualisasikan melalui *class diagram*. Diagram ini merupakan salah satu jenis diagram UML yang secara spesifik digunakan untuk menggambarkan struktur sistem secara detail,

termasuk pendefinisian kelas-kelas yang ada di dalam sistem, rincian atribut, operasi metode, serta hubungan relasi antar objek tersebut. Adapun simbol-simbol pemodelan *class diagram* dijabarkan pada Tabel 2.5.

Tabel 2.5 Simbol *Class Diagram*

Simbol	Nama	Fungsi
	<i>Class</i>	Merepresentasikan entitas tabel yang berisi nama kelas, atribut, dan metode.
	<i>Association</i>	Menunjukkan hubungan statis antar kelas dalam basis data.
	<i>Directed Association</i>	Menunjukkan arah aliran hubungan antar kelas secara spesifik.
1..*	<i>Multiplicity</i>	Menunjukkan relasi ketergantungan atau perluasan fungsi antar <i>use case</i> .

Berdasarkan sintesis dari berbagai literatur di atas, dapat disimpulkan bahwa perpaduan antara *flowchart* dan instrumen UML (*use case*, *activity*, *sequence*, *class diagram*) merupakan kerangka pemodelan yang sangat komprehensif. Rangkaian diagram visual ini menjadi fondasi yang mutlak diperlukan untuk meminimalisasi kesalahan logika sebelum masuk ke tahap implementasi pengkodean. Dalam konteks rancang bangun sistem di Gereja Masehi Advent Hari

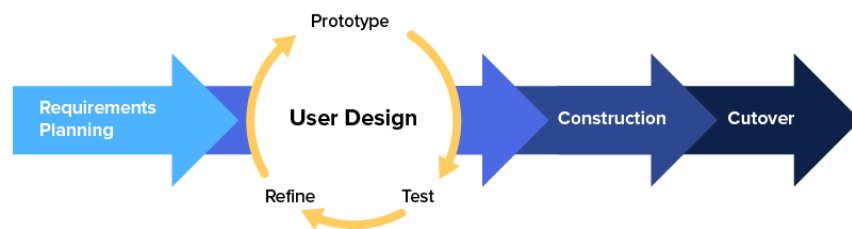
Ketujuh Kalasan, seluruh pemodelan ini digunakan sebagai cetak biru (*blueprint*) utama untuk merancang alur logika hak akses (RBAC), memetakan komputasi algoritma *Smart Auto-Fill*, dan membangun struktur rancangan basis data MySQL sebelum aplikasi dikembangkan secara utuh menggunakan kerangka kerja Laravel.

7. Metode Pengembangan Perangkat Lunak (*Rapid Application Development*).

Menurut Umar (dalam Dalis et al. 2024), dalam rekayasa perangkat lunak, pemilihan metodologi pengembangan sangat menentukan keberhasilan dan efisiensi penyelesaian sebuah sistem. Metodologi *Rapid Application Development* (RAD) merupakan sebuah metode pengembangan sistem informasi yang mengutamakan pendekatan iteratif dengan waktu pengerjaan yang relatif singkat. Pendekatan iteratif ini berarti bahwa model kerja sistem dibangun pada awal tahap pengembangan untuk mengidentifikasi kebutuhan pengguna secara langsung, sehingga model tersebut dapat beradaptasi, ditingkatkan, atau diubah seiring dengan berjalannya proses pengembangan proyek.

Keunggulan utama dari metode ini terletak pada fleksibilitas dan kecepatan eksekusinya. Pengembangan sistem dengan RAD menekankan pada penciptaan purwarupa (*prototyping*), pengumpulan umpan balik (*feedback*) pengguna yang intensif, serta iterasi yang berjalan cepat, seperti yang diilustrasikan pada Gambar 2.3. Dengan

skema iterasi dan purwarupa tersebut, pengembang dapat dengan cepat menghasilkan model antarmuka yang dapat diuji langsung oleh pengguna, sehingga modifikasi sistem benar-benar terarah pada penyelesaian masalah tanpa memakan waktu yang lama seperti pada metode tradisional (Haviz et al., 2025).



Gambar 2.3 Siklus Metode *Rapid Application Development* (RAD)
Sumber : (Iqbal, 2023)

Secara prosedural, Iqbal (2023) menjelaskan bahwa implementasi metode RAD dalam pengembangan perangkat lunak terdiri atas empat tahapan utama yang saling berkesinambungan. Tahapan tersebut diawali dengan perencanaan kebutuhan (*requirements planning*) untuk mengidentifikasi masalah, dilanjutkan dengan perancangan pengguna (*user design*) untuk membangun purwarupa antarmuka dan basis data. Setelah desain disetujui, tahapan berlanjut pada konstruksi (*construction*) berupa penulisan kode program, dan diakhiri dengan fase peralihan (*cutover*) yang mencakup pengujian keseluruhan sistem sebelum diserahkan dan dioperasikan oleh pengguna akhir.

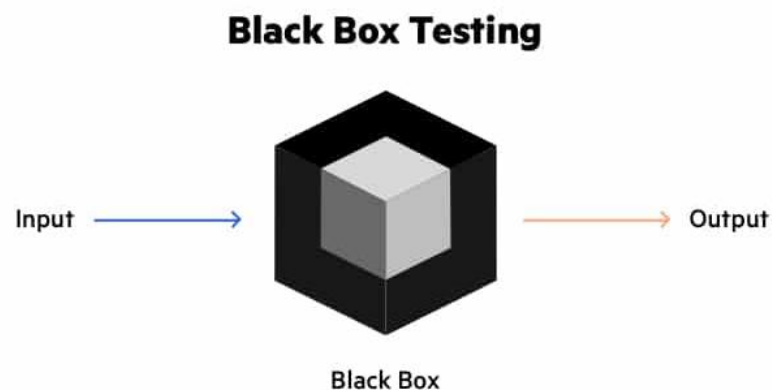
Berdasarkan sintesis dari berbagai literatur di atas, dapat disimpulkan bahwa metodologi *Rapid Application Development* (RAD) merupakan pendekatan yang paling adaptif dan efisien untuk merancang sistem yang membutuhkan interaksi pengguna secara intensif. Fleksibilitasnya dalam mengakomodasi perubahan desain di tengah proses pengembangan menjadikannya sangat ideal dibandingkan metode sekuensial. Dalam konteks penelitian di Gereja Masehi Advent Hari Ketujuh Kalasan, metode RAD dipilih agar pengembang dapat secara cepat menyesuaikan fungsionalitas sistem seperti alur otomatisasi jadwal dan notifikasi *WhatsApp* berdasarkan umpan balik dari Sekretaris dan Departemen Komunikasi, sehingga aplikasi yang dihasilkan benar-benar relevan dengan kebutuhan operasional gereja.

8. Pengujian Perangkat Lunak (*Black Box Testing*).

Menurut Cahyono et al. (2025), Pengujian perangkat lunak merupakan tahapan esensial di akhir fase pengembangan untuk memastikan bahwa sistem informasi yang dibangun dapat berfungsi sesuai dengan spesifikasi kebutuhan dan bebas dari kesalahan operasional. Dalam pengembangan aplikasi berbasis *web*, pendekatan pengujian yang paling umum dan efektif digunakan adalah *Black Box Testing*. Metode ini berfokus pada pengujian aspek fungsionalitas dari perangkat lunak berdasarkan hasil masukan (*input*) dan keluaran (*output*) tertentu, tanpa perlu memperhatikan, menganalisis, atau

memeriksa struktur kode internal maupun algoritma pemrograman di baliknya.

Menurut Mahfud et al. (2024), secara operasional, *Black Box Testing* dilakukan berdasarkan skenario yang telah disusun sebelumnya oleh pengembang atau penguji. Penguji akan memposisikan dirinya sebagai pengguna akhir (*end-user*) untuk memasukkan berbagai variasi data masukan guna memeriksa apakah sistem menghasilkan luaran yang sesuai dengan ekspektasi. Konsep pengujian fungsionalitas yang mengabaikan struktur internal ini diilustrasikan pada Gambar 2.4. Melalui pengujian skenario ini, pengembang dapat memastikan bahwa fitur-fitur kritis, seperti pengelolaan data hingga pengiriman laporan, mampu menangani interaksi pengguna secara akurat.



Gambar 2.4 Konsep Pengujian Fungsionalitas *Black Box*
Sumber : (Rahma et al., 2025)

Lebih lanjut, Aryandra & Wardhana. (2024) menegaskan bahwa meskipun terlihat sederhana karena tidak menyentuh lapisan kode, teknik ini sangat ampuh untuk menemukan berbagai anomali pada

tingkat antarmuka dan basis data. *Black Box Testing* dirancang secara spesifik untuk mengidentifikasi fungsi-fungsi yang tidak berjalan dengan benar, melacak kesalahan pada desain antarmuka pengguna, mendeteksi kegagalan saat sistem mencoba mengakses struktur basis data, hingga mengevaluasi kesalahan performa pada saat inisialisasi maupun terminasi sebuah proses.

Berdasarkan sintesis dari berbagai literatur di atas, dapat disimpulkan bahwa *Black Box Testing* merupakan metode evaluasi perangkat lunak yang paling efisien untuk memvalidasi fungsionalitas sistem dari sudut pandang pengguna. Metode ini memastikan bahwa aplikasi benar-benar siap beroperasi secara nyata. Dalam konteks rancang bangun sistem penjadwalan ibadah di Gereja Masehi Advent Hari Ketujuh Kalasan, metode ini akan diaplikasikan untuk memvalidasi fungsi-fungsi krusial, seperti pembatasan hak akses antar pengguna (RBAC), pencegahan duplikasi jadwal saat eksekusi *Smart Auto-Fill*, pembuatan dokumen PDF, serta validasi ketepatan pemicu pengiriman pesan pada fitur *WhatsApp Click-to-Chat*, sehingga meminimalisasi risiko kegagalan sistem saat digunakan oleh pengurus gereja.

B. Kajian Empiris

Penelitian mengenai perancangan sistem informasi manajemen, algoritma penjadwalan, serta pemanfaatan notifikasi *WhatsApp* telah banyak dilakukan oleh para peneliti sebelumnya. Tinjauan terhadap literatur

terdahulu ini menjadi landasan empiris yang sangat penting untuk mengetahui perkembangan teknologi terkini, mengidentifikasi kelemahan dari sistem yang sudah ada, serta menentukan posisi dan nilai tambah (kebaruan) dari penelitian yang akan dilakukan. Beberapa penelitian terdahulu yang memiliki relevansi dengan rancang bangun sistem informasi penjadwalan pelayanan ibadah ini diuraikan sebagai berikut.

Penelitian pertama dilakukan oleh Solikin et al. (2025), dengan judul "Perancangan Sistem Informasi B2B Untuk Meningkatkan Efisiensi Operasional di CV. X". Penelitian ini berfokus pada digitalisasi sistem informasi berbasis *Business-to-Business* (B2B) untuk meningkatkan efisiensi operasional. Menggunakan metode pengembangan perangkat lunak *Waterfall*, hasil dari penelitian ini membuktikan bahwa implementasi sistem berbasis *web* berhasil mempercepat efisiensi operasional, meminimalkan kesalahan pencatatan data transaksi, dan memberikan fleksibilitas dalam pengelolaan data.

Penelitian kedua dilakukan oleh Pratiwi et al. (2023), yang berjudul "Rancang Bangun Sistem Informasi Penjadwalan Mata Pelajaran dengan Metode *Waterfall*". Fokus utama dari penelitian ini adalah pada perancangan sistem yang mampu mengatasi kesalahan penyusunan jadwal untuk mencegah terjadinya tumpang tindih (*clashing*). Melalui pendekatan *Waterfall*, penelitian tersebut menghasilkan sebuah sistem terpusat yang memudahkan pengelola dalam menyusun jadwal secara akurat, sehingga

menjamin tidak ada jam maupun penugasan pengajar yang saling berbenturan.

Penelitian ketiga dilakukan oleh Fatah et al. (2025), dengan kajian berjudul "Sistem Informasi Pengingat Obat melalui Notifikasi Whatsapp". Kajian empiris ini berfokus pada efektivitas mekanisme pengingat otomatis (*reminder*) yang dikirimkan melalui platform *WhatsApp*. Dikembangkan dengan pendekatan rekayasa perangkat lunak model *Waterfall*, hasil penelitian tersebut mengonfirmasi bahwa pengiriman pesan pengingat secara tepat waktu terbukti sangat efektif dalam mempermudah komunikasi dan meningkatkan kepatuhan penggunaanya tanpa perlu instalasi aplikasi tambahan.

Berdasarkan tinjauan terhadap ketiga penelitian terdahulu di atas, dapat ditarik kesimpulan bahwa sistem informasi berbasis *web* untuk manajemen penjadwalan dan pemanfaatan notifikasi *WhatsApp* telah terbukti efektif. Namun, terdapat celah penelitian (*research gap*) yang signifikan. Ketiga penelitian sebelumnya sepenuhnya bertumpu pada metode pengembangan *Waterfall* yang cenderung kaku terhadap perubahan. Selain itu, penelitian terkait penjadwalan sebelumnya belum dilengkapi fitur pengingat berbasis pesan, dan integrasi *WhatsApp* umumnya masih membutuhkan infrastruktur *server* atau API yang lebih kompleks.

Untuk mengisi celah penelitian tersebut, penelitian ini menawarkan kebaruan dengan menerapkan metodologi *Rapid Application Development* (RAD) yang iteratif dan adaptif terhadap umpan balik pengguna. Inovasi

lainnya terletak pada pemanfaatan *Framework* Laravel serta implementasi arsitektur *WhatsApp Click-to-Chat* yang mampu menjalankan fungsi pengingat jadwal pelayanan secara efisien tanpa membebani organisasi dengan biaya layanan pihak ketiga (*zero-cost API*). Lebih lanjut, penelitian ini dikontekstualisasikan secara spesifik pada Gereja Masehi Advent Hari Ketujuh Kalasan dengan menerapkan *Role-Based Access Control* (RBAC) yang ketat dan terstruktur antara Admin, Sekretaris, dan Departemen Komunikasi.

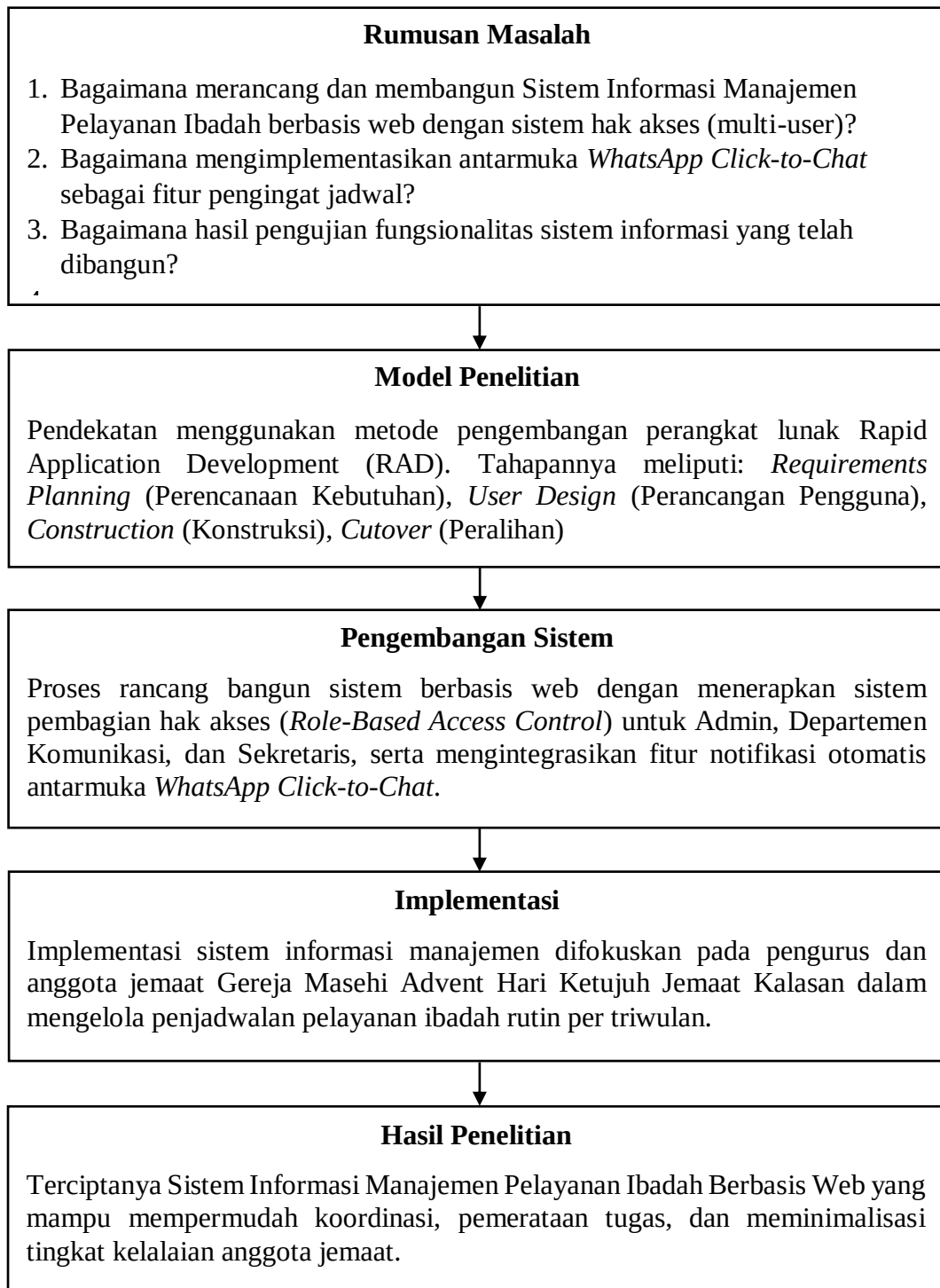
C. Kerangka Berpikir

Penelitian ini dilakukan untuk menjawab permasalahan dalam manajemen pelayanan ibadah di Gereja Masehi Advent Hari Ketujuh Jemaat Kalasan yang saat ini masih dikelola secara manual menggunakan perangkat lunak *spreadsheet*. Berdasarkan hasil observasi dan kajian empiris, metode konvensional tersebut menyulitkan pemerataan tugas pelayanan, rentan memicu duplikasi jadwal, serta memiliki keterbatasan dalam penyampaian informasi secara *real-time* kepada anggota jemaat. Ketidadaan sistem pengingat yang terstruktur juga sering kali mengakibatkan anggota jemaat melalaikan tugas pelayanannya. Hal ini menjadi landasan utama perlunya pengembangan sistem informasi manajemen terpusat berbasis web yang didukung oleh pembagian hak akses (*Role-Based Access Control*) untuk memfasilitasi kolaborasi pembagian jadwal antar pengurus.

Dalam penelitian ini, model pengembangan perangkat lunak yang digunakan adalah pendekatan *Rapid Application Development* (RAD).

Tahapan dalam metode ini meliputi perencanaan kebutuhan (*requirements planning*), perancangan pengguna (*user design*), konstruksi (*construction*), dan peralihan (*cutover*). Pendekatan ini dipilih karena memberikan alur kerja yang iteratif, kolaboratif, dan memungkinkan penyesuaian fungsionalitas sistem secara langsung berdasarkan umpan balik dari pengguna. Fitur unggulan yang ditanamkan ke dalam sistem ini adalah manajemen *multi-user* (Admin, Departemen Komunikasi, dan Sekretaris) serta integrasi notifikasi pengingat otomatis menggunakan antarmuka *WhatsApp Click-to-Chat* guna memastikan informasi tersampaikan secara langsung kepada anggota jemaat.

Subjek dari implementasi sistem ini adalah pengurus dan anggota jemaat Gereja Masehi Advent Hari Ketujuh Jemaat Kalasan. Adapun luaran dari hasil penelitian ini adalah terciptanya rancang bangun Sistem Informasi Manajemen Pelayanan Ibadah berbasis web yang fungsional. Melalui implementasi luaran tersebut, diharapkan pengelolaan dan pendistribusian jadwal ibadah menjadi lebih efisien, transparan, dan mampu meminimalisasi tingkat kelalaian anggota jemaat terhadap tugas pelayanan. Berdasarkan keseluruhan tahapan dan arah pengembangan sistem tersebut, maka dapat disusun kerangka berpikir penelitian yang direpresentasikan dalam gambar 2.6



Gambar 2.5 Kerangka Berpikir