

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Rancang Bangun

Menurut Maulana (2023) Aktivitas rancang bangun dapat dipahami sebagai sebuah upaya memetakan, merencanakan, serta menyusun cetak biru dari berbagai komponen terisolasi guna menghasilkan sebuah kesatuan sistem yang aplikatif. Sedangkan menurut Parjito et al. (2022) Tindakan rancang bangun pada dasarnya berorientasi pada visualisasi serta penerapan teknologi demi menginisiasi platform baru, ataupun melakukan modifikasi dan pembenahan pada infrastruktur lama secara parsial maupun menyeluruh. Melalui pendekatan ini, kendala operasional yang sedang dihadapi diharapkan dapat terselesaikan secara efektif.

Jadi dapat disimpulkan bahwa rancang bangun adalah proses terintegrasi yang dimulai dari perencanaan dan penggambaran konsep, hingga tahap implementasi nyata untuk menyatukan berbagai elemen terpisah menjadi satu kesatuan sistem yang utuh. Baik dilakukan untuk menciptakan sesuatu yang baru maupun memperbaiki sistem yang sudah ada, fokus utamanya adalah menghasilkan sistem yang berfungsi optimal dan menjadi solusi konkret atas permasalahan yang dihadapi.

2. Sistem Informasi Manajemen

Menurut Laia et al. (2022) Sistem informasi manajemen merupakan kumpulan elemen yang saling berhubungan satu sama lain yang membentuk

satu kesatuan untuk mengintegrasikan data, memproses data dan menyimpan data. Sedangkan menurut Loilatu et al. (2020) Secara konseptual, Sistem Informasi Manajemen (SIM) diartikan sebagai arsitektur informasi terpadu yang memadukan fungsi pencatatan aktivitas bisnis dengan penyediaan laporan analitis. Perpaduan ini esensial bagi pihak pengambil kebijakan dalam memantau kinerja organisasi sekaligus menentukan arah langkah strategis ke depan.

Jadi dapat disimpulkan bahwa sistem informasi manajemen (SIM) adalah sebuah kesatuan sistemik yang mengintegrasikan berbagai elemen organisasi untuk mengolah, memproses, dan menyimpan data secara terstruktur. Secara substansial, SIM tidak hanya berfungsi sebagai alat mekanis dalam menyederhanakan operasional transaksi harian, melainkan juga bertindak sebagai instrumen strategis yang menyediakan informasi krusial guna mendukung fungsi manajemen dalam pengambilan keputusan yang akurat bagi organisasi.

3. Reservasi

Menurut Rabbani et al. (2023) Aktivitas pemesanan atau yang kerap disebut sebagai *reservation* mengacu pada tindakan proaktif untuk mengamankan ketersediaan sarana di awal. Konsep yang diadopsi dari kata “*to reserve*” ini menekankan pentingnya aspek penyiapan tempat sebelum pengguna benar-benar datang memanfaatkan fasilitas tersebut. Sedangkan menurut Putri (2022) Konstruksi istilah reservasi berakar dari frasa bahasa Inggris *to reserve*, yang mendeskripsikan tindakan menyiagakan atau

mengalokasikan suatu ruang sebelum pemanfaatan sarana tersebut dilakukan. Secara definitif, proses ini merujuk pada aktivitas pemesanan yang bertujuan untuk mengamankan hak pakai atas suatu fasilitas fisik.

Jadi dapat disimpulkan bahwa reservasi secara konseptual adalah tindakan proaktif untuk mengamankan, menyiapkan, atau menyediakan suatu tempat maupun fasilitas tertentu sebelum waktu pemakaian yang direncanakan. Berakar dari istilah bahasa Inggris “*to reserve*”, aktivitas ini menekankan pada aspek persiapan di awal demi menjamin kepastian ketersediaan layanan atau ruang fisik sebelum pengguna benar-benar datang untuk menikmatinya.

4. Website

Menurut Nizar (2021) Pada dasarnya, *website* merupakan sebuah ekosistem digital berbentuk kumpulan halaman yang saling bertautan demi menyajikan beragam informasi multimedia kepada pengguna. Konten yang ditampilkan di dalam platform ini dapat berupa kombinasi dari unsur teks, gambar, rekaman suara, animasi, maupun video bergerak, baik yang dikondisikan secara statis maupun dinamis. Rangkaian elemen tersebut disusun secara terstruktur di mana tiap komponennya dihubungkan secara sistematis melalui jaringan tautan elektronik. Sedangkan menurut Maharani et al. (2021) Sebagai media publikasi digital, sebuah *website* merupakan konvergensi dari lembaran halaman elektronik yang saling terikat demi mendistribusikan data multimedia kepada pengguna. Wadah virtual ini mampu merangkum kombinasi dari teks, gambar, animasi, suara, serta

tayangan video yang dikemas secara interaktif (dinamis) maupun konvensional (statis). Jalinan antarkomponen di dalamnya membentuk satu kesatuan bangunan sistem digital yang terkoneksi secara sistematis memanfaatkan tautan *hyperlink*.

Jadi dapat disimpulkan bahwa *website* secara konseptual merupakan sebuah arsitektur digital yang terdiri dari kumpulan halaman yang saling terintegrasi dan dihubungkan oleh jaringan *hyperlink*. Platform ini berfungsi sebagai media penyaji data dan informasi berbasis multimedia meliputi kombinasi teks, gambar, animasi, suara, hingga video yang dikemas secara statis maupun dinamis dalam satu kesatuan sistem yang utuh.

5. *WhatsApp Gateway*

Menurut Ismawati et al. (2023) Sebagai media penghubung antar-platform, WhatsApp Gateway berfungsi sebagai peranti lunak pengelola lalu lintas pesan berbasis API. Infrastruktur ini beroperasi sebagai jembatan komunikasi cerdas yang menerima input permintaan dari pengguna, menginstruksikan sistem inti untuk melakukan pemrosesan data, serta menyampaikan kembali jawaban atau notifikasi balasan secara instan. Sedangkan menurut Rahmawati et al. (2021) *WhatsApp Gateway* merupakan peranti aplikasi yang memfasilitasi integrasi pengiriman dan penerimaan pesan WhatsApp secara dua arah dari platform eksternal seperti website ke gawai pengguna. Melalui arsitektur ini, sinkronisasi antar-gawai maupun aplikasi dapat terwujud untuk mendukung efisiensi transfer data. Kendali komunikasi tersebut berpusat pada API yang beroperasi sebagai

perantara pesan; bertugas menerima *input* perintah pengguna, mengarahkan jalannya sistem, serta menghasilkan *output* respons yang sejalan dengan instruksi awal.

Jadi dapat disimpulkan bahwa *whatsapp gateway* adalah sebuah sistem aplikasi terintegrasi yang berfungsi sebagai jembatan komunikasi lintas platform untuk mengirim dan menerima pesan WhatsApp antara situs web atau aplikasi pihak ketiga dengan perangkat target. Dengan memanfaatkan *Application Programming Interface* (API) sebagai arsitektur utamanya, sistem ini bekerja secara interaktif sebagai perantara cerdas yang menerima permintaan pengguna, menginstruksikan sistem untuk mengeksekusi perintah tersebut, dan menghasilkan respons balik berupa pertukaran data atau informasi yang relevan.

6. *Payment Gateway*

Menurut Gibran et al. (2024) *Payment gateway* merupakan infrastruktur portal digital yang bertugas menghubungkan rekening bank dengan lembaga pemroses pembayaran melalui transmisi data virtual berbasis API dan web, ataupun terminal transaksi langsung. Peranti ini memegang peran vital dalam mengamankan serta mengesahkan multi-metode transaksi daring. Lingkup pembayarannya mencakup konvergensi instrumen keuangan modern, mulai dari kartu kredit, *internet banking*, *digital wallet*, hingga jaringan UPI. Sedangkan menurut Fatman et al. (2023) Midtrans dikenal secara luas sebagai salah satu gerbang pembayaran (*payment gateway*) yang eksis sejak 2012 untuk melayani kebutuhan

transaksi elektronik para pelaku usaha. Melalui arsitektur integrasi yang dinamis dan mudah diimplementasikan, platform ini berhasil merangkum beraneka jenis instrumen pembayaran *online* ke dalam satu ekosistem digital yang efisien.

Jadi dapat disimpulkan bahwa *payment gateway* adalah portal infrastruktur digital yang berfungsi sebagai penghubung antara rekening bank konsumen dengan pihak pemroses pembayaran (*payment processor*) melalui integrasi API atau layanan web guna mengotorisasi berbagai moda transaksi elektronik secara aman. Dalam ranah aplikatif, teknologi ini direpresentasikan oleh penyedia layanan seperti Midtrans, yang sejak tahun 2012 telah memfasilitasi para pelaku usaha dengan menyediakan ekosistem pembayaran multipanduan yang adaptif dan mudah diintegrasikan.

7. HTML

Menurut Nasution et al. (2022) HTML (*Hypertext Markup Language*) direpresentasikan sebagai rangkaian kode struktural yang menjadi basis utama wujud visual dari sebuah halaman web. Arsitektur ini menyimpan kumpulan data dan informasi di dalam barisan penanda khusus atau tag. Melalui mekanisme tersebut, tiap-tiap tag akan mengeksekusi perintah pemformatan tertentu agar informasi yang disajikan tampil secara teratur dan rapi saat diakses. Sedangkan menurut Ramdani & Agustin (2023) Sebagai media penyusun arsitektur digital, *Hypertext Markup Language* (HTML) digunakan untuk memproduksi halaman website dan menampilkan berbagai materi informasi secara daring. Penggunaan format berkas ASCII pada bahasa markah ini memungkinkan sistem untuk mengolah data hiperteks, sehingga menghasilkan wujud visualisasi antarmuka yang utuh, fungsional, serta terstruktur dengan baik saat diakses lewat browser.

Jadi dapat disimpulkan bahwa HTML (*Hypertext Markup Language*) adalah bahasa markah standar berupa rangkaian kode struktural yang berfungsi sebagai fondasi dasar dalam membangun halaman web. Dengan memanfaatkan sistem pengkodean berbasis tag yang ditulis dalam format berkas teks (ASCII), HTML bertugas untuk mengorganisasi, memformat, dan mengintegrasikan berbagai elemen informasi agar dapat diterjemahkan oleh penjelajah internet (*web browser*) menjadi sebuah representasi visual yang utuh dan terstruktur.

8. CSS

Menurut Mardiansyah et al. (2025) Pengertian CSS (*Casading Style Sheet*) adalah framework memiliki konsep terstruktur serta terorganisir pada satu library. Dalam proses perancangan aplikasi, penerapan CSS memfasilitasi pengayaan tata visual halaman dengan mengizinkan pengembang menyunting komponen grafis secara dinamis. Hal ini meliputi kendali penuh atas struktur tata letak dokumen, penentuan warna latar belakang, serta pengelolaan karakteristik teks seperti ukuran dan ragam jenis huruf yang digunakan. Sedangkan menurut I. P. Sari, Azzahrah, et al. (2022) Secara definitif, *Cascading Style Sheet* (CSS) merupakan bahasa pemformatan tampilan yang diterapkan untuk mengarsiteki aspek estetika dan desain sebuah halaman web. Peranti ini mengontrol berbagai komponen visual meliputi penentuan palet warna, tipografi (*font*), aksentuasi garis tepi (*outline*), serta arsitektur latar belakang (*background*).

Jadi dapat disimpulkan bahwa CSS (*Cascading Style Sheet*) adalah bahasa pemformatan tampilan yang berfungsi untuk mengatur, menstrukturkan, dan mengorganisasi elemen visual pada halaman web secara sistematis. Berkolaborasi secara integral dengan HTML, CSS memungkinkan pengembang untuk mentransformasikan kerangka teks murni menjadi antarmuka yang menarik melalui kontrol tata letak (*layout*), warna, tipografi, serta kemampuan adaptasi antarmuka yang responsif terhadap berbagai ukuran layar perangkat.

9. Javascript

Menurut Sahi (2020) Sebagai bahasa pemrograman yang beroperasi secara melekat di dalam dokumen HTML, JavaScript merupakan peranti berbasis kumpulan skrip yang dikembangkan pertama kali oleh Netscape pada tahun 1995. Sebelum menyandang nama besarnya yang sekarang, teknologi ini awalnya dilahirkan dengan sebutan “*LiveScript*”. Pada masa awal rilisnya tersebut, fungsinya masih terbatas sebagai bahasa pendukung tingkat sederhana untuk mengoptimalkan peramban Netscape Navigator 2. Sedangkan menurut I. P. Sari, Jannah, et al. (2022) Sebagai bahasa pemrograman web berkategori *Client-Side*, JavaScript menjalankan seluruh instruksi kodenya secara mandiri di sisi pengguna tanpa harus bergantung pada respons server. Pendekatan arsitektural ini memungkinkan skrip program diterjemahkan secara langsung oleh aplikasi penjelajah internet (*web browser*) yang terpasang pada gawai klien, termasuk di antaranya Google Chrome, Mozilla Firefox, hingga Opera Mini.

Secara garis besar, JavaScript adalah bahasa skrip terintegrasi HTML yang dirilis Netscape pada 1995 dengan nama awal *LiveScript*. Sebagai *Client-Side Programming Language*, komputasi kodenya dialihkan dari server ke perangkat lokal, sehingga dapat dieksekusi instan langsung oleh peramban web pengguna seperti Chrome, Firefox, atau Opera Mini.

10. XAMPP

Menurut Kurniawansyah et al. (2025) Dalam rekayasa aplikasi berbasis web, XAMPP diadopsi sebagai perangkat lunak bersumber terbuka

(*open-source*) yang mengintegrasikan berbagai instrumen esensial. Paket ini sudah merangkum komponen vital seperti web server Apache, basis data MySQL, bahasa pemrograman PHP, serta antarmuka grafis phpMyAdmin untuk mempermudah manajemen data.

XAMPP beroperasi sebagai server mandiri, memungkinkan pengeditan, desain, dan pengembangan perangkat lunak secara lokal. *XAMPP* berfungsi sebagai perangkat lunak sumber terbuka yang beroperasi sebagai server lokal, memungkinkan pengguna untuk mengembangkan dan menguji aplikasi web di komputer pribadi tanpa memerlukan server online (Abdila et al., 2025).

Jadi dapat disimpulkan bahwa *XAMPP* adalah paket perangkat lunak sumber terbuka (*open-source*) yang mengintegrasikan berbagai komponen esensial pengembangan web meliputi Apache (web server), MySQL (database), PHP (bahasa pemrograman), dan phpMyAdmin (antarmuka grafis) ke dalam satu ekosistem *local server*. Beroperasi sebagai server mandiri (*localhost*), perangkat lunak ini berfungsi sebagai lingkungan simulasi luring yang memungkinkan pengembang untuk merancang, mengedit, dan menguji aplikasi secara aman di komputer pribadi tanpa ketergantungan pada jaringan internet atau server online.

11. MySql

Menurut Winanjar & Susanti (2021) *MySQL* menjadi salah satu *database server* yang sangat populer karena mengadopsi bahasa SQL sebagai standar dasar untuk memanipulasi data di dalamnya. Nilai lebih dari

peranti ini adalah sifatnya yang terbuka karena dilengkapi dengan kode sumber (*source code*) pembuatan aplikasinya secara langsung. Sedangkan menurut Sudaria et al. (2021) Sebagai peranti DBMS *open-source* terpopuler saat ini, MySQL hadir untuk menyediakan infrastruktur data yang tangkas, andal, dan ramah pengguna. Kehadiran fitur *multithreaded* dan kemampuan *multi-user* pada sistem ini memungkinkannya mengelola lalu lintas data secara paralel dengan efisien. Seluruh aktivitas pengaksesan dan manipulasi informasi di dalam lingkungan database ini dijalankan secara terstruktur melalui perintah *Structured Query Language* (SQL).

Kesimpulannya, MySQL merupakan sistem manajemen basis data (*open-source*) terpopuler yang mengadopsi sintaksis SQL sebagai standar dasar operasionalnya. Sistem ini dikonstruksikan khusus guna menghasilkan performa data yang cepat, andal, dan mudah digunakan, serta dibekali fitur *multithreaded* dan *multi-user* untuk memproses banyak instruksi data secara bersamaan.

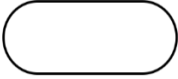
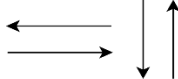
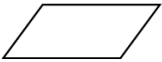
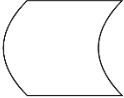
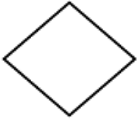
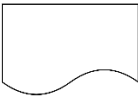
12. Flowchart

Menurut Listyoningrum et al. (2023) Dalam pemodelan sistem, flowchart (diagram alir) diklasifikasikan sebagai representasi grafis terstruktur yang menggambarkan urutan langkah atau prosedur secara sistematis dalam eksekusi sebuah program. Keberadaan diagram ini berfungsi sebagai visualisasi pemandu yang mempermudah tahapan analisis, perancangan arsitektur, hingga pemetaan logika pengodean untuk menyelesaikan berbagai problematik operasional secara mendetail.

Sedangkan menurut Tuasamu et al. (2023) Bagan alir (*flowchart*) merupakan teknik analisis bergambar yang memudahkan pemahaman terhadap komponen sistem informasi secara lugas, ringkas, dan sistematis. Diagram alir ini bekerja dengan cara merekam setiap tahapan proses bisnis yang berjalan sekaligus memperlihatkan alur peredaran dokumen dari satu bagian ke bagian lainnya di dalam organisasi.

Jadi dapat disimpulkan bahwa flowchart adalah instrumen analitis berbasis visual yang menggambarkan urutan prosedur dan logika sistem secara sistematis, ringkas, dan logis. Sebagai alat pemecahan masalah, flowchart mengonversikan kompleksitas operasional menjadi representasi grafis yang mudah dipahami, baik untuk kebutuhan teknis pengembangan perangkat lunak (analisis, perancangan, dan pengkodean) maupun untuk pemetaan tata kelola bisnis (alur kerja dan sirkulasi dokumen di dalam organisasi).

Tabel 2.1 Simbol-Simbol *Flowchart*

No	Simbol	Fungsi
1	 Terminator	Permulaan/akhir program
2	 Garis alir(flow line)	Arah aliran program
3	 Proses	Proses perhitungan/proses pengolahan data
4	 Input output data	Proses input/output data, parameter, informasi
5	 Disk Storage	Menggambarkan suatu inputan dari disk storage
6	 Decision	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
7	 Document	Sebuah dokumen atau laporan. Dokumen dapat dibuat dengan tangan atau cetak oleh komputer

Sumber : (Zalukhu et al., 2023)

13. Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan metodologi standar yang digunakan dalam pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak berbasis objek. Sebagai bahasa pemodelan grafik, UML menyediakan sekumpulan diagram yang standar guna membantu para pengembang, arsitek sistem, serta pemangku kepentingan dalam memvisualisasikan struktur statis dan perilaku dinamis dari sebuah sistem yang kompleks. Pemodelan ini tidak hanya mempermudah pemahaman alur kerja dan interaksi komponen di dalam sistem, tetapi juga berfungsi sebagai cetak biru (*blueprint*) yang menjembatani komunikasi teknis selama proses analisis, perancangan, hingga implementasi kode program (Nugroho, 2010a, pp. 6–9).

Berdasarkan penjelasan tersebut, dapat disimpulkan bahwa UML bertindak sebagai instrumen krusial dalam rekayasa perangkat lunak yang mentransformasikan konsep sistem abstrak menjadi model visual yang terstruktur. Melalui pendekatan yang berbasis objek, penerapan UML mampu meminimalisasi risiko kesalahan desain sejak dini, memastikan setiap fungsi sistem terdefinisi dengan jelas, serta mempermudah pemeliharaan sistem di masa mendatang demi tercapainya efisiensi dan kualitas pengembangan aplikasi yang optimal.

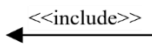
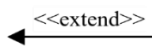
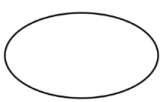
a. *Use Case Diagram*

Use Case Diagram merupakan bagian dari *Unified Modeling Language* (UML) yang digunakan untuk memodelkan aspek perilaku (*behavioral*) dari sebuah sistem melalui interaksi antara pengguna dan sistem itu sendiri. Diagram ini memvisualisasikan fungsionalitas sistem dari sudut pandang pengguna luar yang disebut sebagai aktor, serta bagaimana aktor tersebut memanfaatkan sistem melalui serangkaian fungsi atau kasus penggunaan (*use case*). Di dalam pengembangannya, *use case diagram* tidak menggambarkan detail teknis atau urutan proses internal, melainkan fokus pada apa yang dapat dilakukan oleh sistem (*what the system does*), sehingga dapat mempermudah identifikasi kebutuhan fungsional awal serta batasan sistem yang sedang dibangun (R. F. Sari & Utami S, 2019a, pp. 114–115).

Adapun uraian simbol *Use Case Diagram* dapat dilihat pada tabel bawah ini.

Tabel 2.2 Simbol-Simbol *Use Case Diagram*

No	Simbol	Nama	Fungsi
1		Actor	Menspesifikasikan himpunan peran yang dimainkan oleh pengguna atau entitas luar ketika berinteraksi dengan sistem (<i>use case</i>).
2		Generalization	Hubungan di mana perubahan pada suatu elemen mandiri (<i>independent</i>) akan

			memengaruhi elemen yang bergantung padanya.
3		Include	Menspesifikasikan bahwa <i>use case</i> sumber secara mutlak (eksplisit) memerlukan <i>use case</i> lain untuk menjalankan fungsinya.
4		Extend	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber jika suatu kondisi tertentu terpenuhi.
5		Association	Menghubungkan antara objek/aktor dengan <i>use case</i> yang saling berinteraksi.
6		Use case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.

Sumber : (Rasiban et al., 2024)

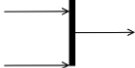
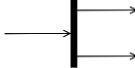
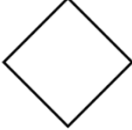
b. Activity Diagram

Activity Diagram atau diagram aktivitas merupakan salah satu pemodelan grafis di dalam *Unified Modeling Language* (UML) yang berfungsi untuk menggambarkan alur kerja (*workflow*), aliran aktivitas sistem, atau proses bisnis pada suatu perangkat lunak. Fokus utama diagram ini adalah memvisualisasikan aktivitas-aktivitas apa saja yang dapat dilakukan dan dialami oleh sistem, bukan detail fungsionalitas yang dijalankan oleh aktor individual. Dalam implementasinya, *activity*

diagram digunakan untuk mendefinisikan urutan proses bisnis terstruktur, memetakan rancangan antarmuka visual berdasarkan pengelompokan tampilan sistem, serta menjadi landasan bagi pemetaan skenario pengujian fungsionalitas perangkat lunak. Struktur alur ini dibangun menggunakan serangkaian simbol standar seperti status awal, aksi (*activity*), percabangan (*decision*), penggabungan (*join*), status akhir, serta *swimlane* untuk memisahkan tanggung jawab dari elemen organisasi atau sistem yang terlibat (Rosa A.S & Shalahuddin, 2013a, pp. 161–163).

Adapun uraian simbol *Activity Diagram* dapat dilihat pada tabel bawah ini.

Tabel 2.3 Simbol-Simbol *Activity Diagram*

No	Simbol	Nama	Fungsi
1		Activity	Mempresentasikan sebuah tindakan, pekerjaan, atau aktivitas yang dilakukan di dalam aliran kerja (<i>workflow</i>) sistem.
2		Control flow	Menunjukkan urutan eksekusi atau arah perpindahan dari satu aktivitas ke aktivitas lainnya.
3		Start point	Menandakan titik awal dimulainya suatu aliran kerja atau aktivitas pada sistem.
4		End point	Menandakan titik akhir atau selesainya keseluruhan aliran kerja dalam suatu <i>Activity Diagram</i> .
5		Join node	Menggabungkan kembali dua atau lebih aliran kerja paralel (berbarengan) menjadi satu aliran eksekusi tunggal.
6		Fork node	Memecah satu aliran kerja tunggal menjadi dua atau lebih aliran kerja paralel yang dieksekusi secara bersamaan.
7		Decision node	Titik percabangan yang menggambarkan pemilihan tindakan berdasarkan suatu kondisi atau syarat tertentu.

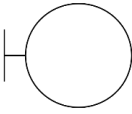
Sumber : (Syahrizal, 2025)

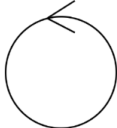
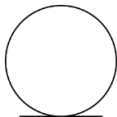
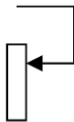
c. *Sequence Diagram*

Sequence Diagram atau diagram urutan merupakan pemodelan grafis dua dimensi dalam *Unified Modeling Language* (UML) yang digunakan untuk mendokumentasikan komunikasi, interaksi, dan pertukaran pesan (*message*) antar-objek atau antar-kelas yang terlibat di dalam suatu *use case*. Matra visual diagram ini terbagi menjadi dua sumbu utama, yaitu sumbu vertikal yang merepresentasikan dimensi waktu yang bergerak maju dari atas ke bawah, serta sumbu horizontal yang merepresentasikan objek-objek mandiri dalam kolaborasi. Di dalam strukturnya, elemen aktor dan kelas ditempatkan secara berurutan dari kiri ke kanan pada bagian atas diagram dengan garis panduan vertikal (*lifeline*) sebagai penanda kehadiran objek tersebut. Aliran pesan antar-objek kemudian digambarkan melalui tanda panah horizontal secara kronologis, sehingga urutan waktu kejadian dari setiap instruksi dapat terlihat dengan jelas (R. F. Sari & Utami S, 2019b, p. 119) dan (Nugroho, 2010b, pp. 42–43).

Adapun uraian simbol *Sequence Diagram* dapat dilihat pada tabel bawah ini.

Tabel 2.4 Simbol-Simbol *Sequence Diagram*

No	Simbol	Nama	Fungsi
1		Boundary class	Objek yang mewakili antarmuka pengguna (<i>user interface</i>). Tempat di mana <i>Actor</i> berinteraksi langsung

			dengan sistem (contoh: halaman <i>form</i> reservasi, <i>dashboard</i>).
2		Control class	Objek yang berisi logika bisnis dan aturan aplikasi. Bertugas menerima perintah dari <i>Boundary</i> dan mengatur <i>Entity</i> (contoh: proses validasi pembayaran, perhitungan total).
3		Entity class	Objek yang merepresentasikan struktur data atau tabel di dalam <i>database</i> (contoh: data pengguna, data jadwal).
4		Lifeline	Garis putus-putus yang mewakili eksistensi atau "garis hidup" suatu objek selama proses interaksi berlangsung.
5		Activation	Mewakili periode waktu di mana sebuah objek sedang aktif menjalankan sebuah operasi atau fungsi.
6		Message	Simbol pengiriman pesan atau pemanggilan metode dari satu objek ke objek lainnya secara sinkron.
7		Recursive	Menggambarkan pemanggilan metode atau operasi yang dikirimkan oleh sebuah objek kepada dirinya sendiri.

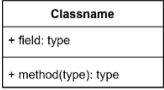
Sumber : (Maria & Efendi, 2021)

d. *Class Diagram*

Class Diagram atau diagram kelas merupakan diagram struktural dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan elemen-elemen statis serta hubungan keterhubungan antar-kelas di dalam sistem. Komponen utama dari pemodelan ini adalah kelas (*class*), yang merepresentasikan sebuah unit dalam sistem yang terbagi ke dalam tiga bagian utama, yaitu nama kelas, atribut (*field*), dan metode atau operasi. Hubungan yang bersifat tetap dan statis antar-kelas tersebut kemudian dipetakan melalui beberapa jenis relasi, seperti *Association* (asosiasi) yang menggambarkan bahwa suatu kelas memiliki keterhubungan atau menyadari keberadaan kelas lainnya. Selain itu, terdapat relasi *Aggregation* (agregasi) sebagai bentuk asosiasi spesifik yang menunjukkan hubungan "seluruh-bagian" (*whole-part*), dengan karakteristik di mana kelas bagian tetap dapat berdiri sendiri secara independen dari kelas utamanya. Hubungan struktural ini juga disempurnakan oleh relasi *Generalization* (generalisasi) yang bermakna hubungan generalisasi-spesialisasi, di mana kelas anak (*child class*) secara langsung mewarisi seluruh atribut dan metode dari kelas induknya (*parent class*) (Rosa A.S & Shalahuddin, 2013b, pp. 141–147).

Berikut simbol-simbol *class diagram* dan fungsinya masing-masing.

Tabel 2.5 Simbol-Simbol *Class Diagram*

No	Simbol	Nama	Fungsi
1		Class	Merupakan sebuah kelas dalam sistem yang terdiri dari tiga bagian utama, yaitu nama kelas, atribut, dan metode (operasi).
2		Association	Adalah hubungan antar kelas yang tetap statis, menggambarkan kelas yang memiliki keterhubungan atau menyadari keberadaan kelas lain.
3		Aggregation (agregasi)	Bentuk asosiasi yang menunjukkan hubungan "seluruh-bagian" (<i>whole-part</i>), di mana kelas bagian dapat berdiri sendiri secara independen dari kelas utamanya.
4		Generalization (generalisasi)	Yaitu hubungan antar kelas dengan makna generalisasi-spesialisasi, di mana kelas anak mewarisi atribut dan metode dari kelas induk.

Sumber : (Subrata & Halifah, 2025)

B. Kajian Empiris

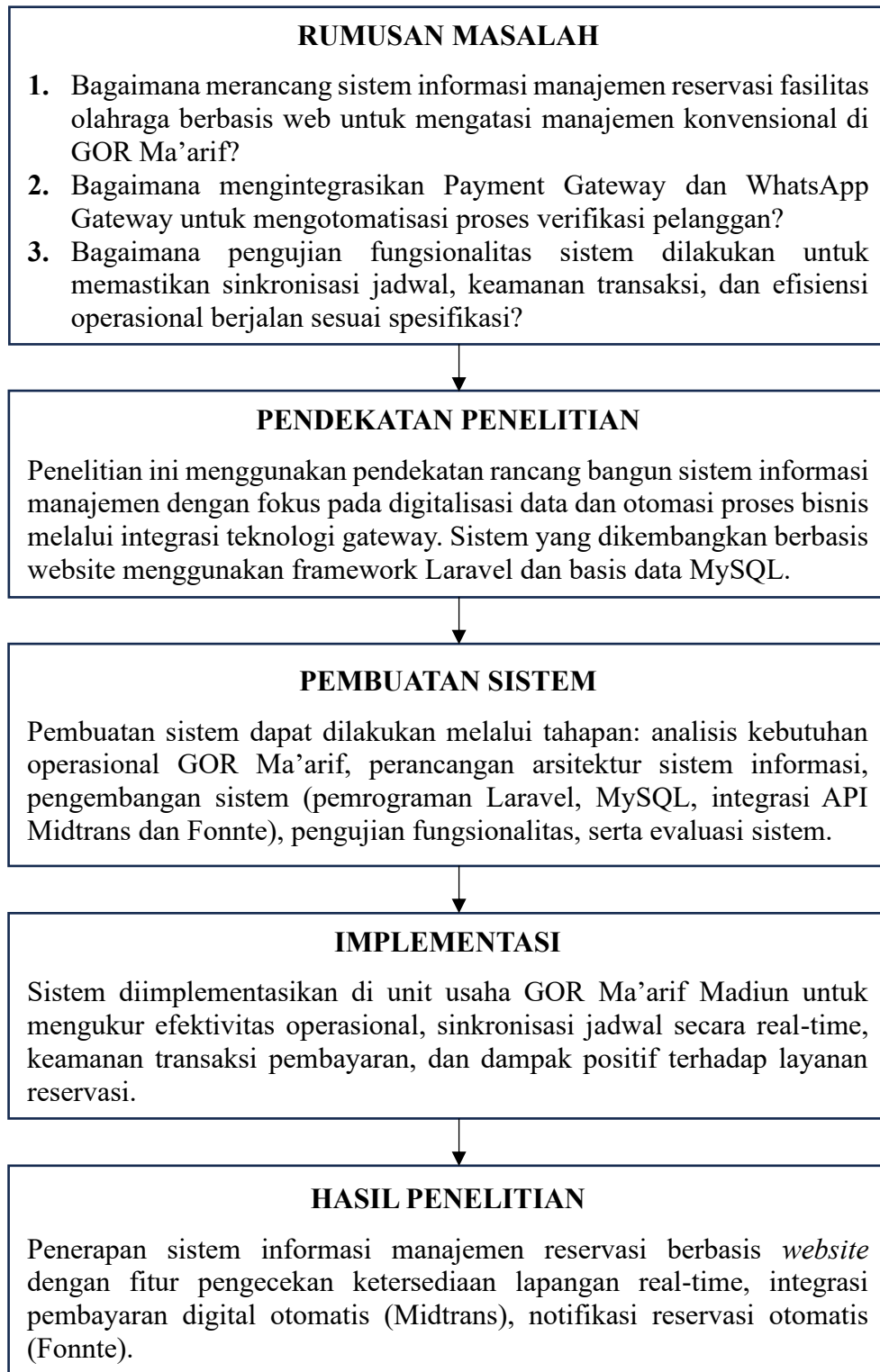
Beberapa penelitian terdahulu yang memiliki relevansi dengan topik rancang bangun sistem manajemen reservasi dan integrasi teknologi *gateway* adalah sebagai berikut:

1. Dwiatno (2022) melakukan penelitian mengenai digitalisasi sistem reservasi pada fasilitas olahraga melalui implementasi sistem *booking* berbasis web. Fokus utama penelitiannya adalah mengganti mekanisme pencatatan manual menjadi digital guna memusatkan data penyewaan dalam basis data. Namun, penelitian tersebut masih menyisakan celah operasional karena proses verifikasi pembayarannya belum otomatis dan belum ada fitur notifikasi langsung kepada pelanggan. Di sinilah penelitian ini hadir untuk menutup kekurangan tersebut melalui otomatisasi validasi dan komunikasi *real-time*.
2. Ismawati et al. (2023) dalam penelitiannya membahas mengenai penggunaan *WhatsApp Gateway* sebagai aplikasi untuk mengirim dan menerima pesan melalui *API (Application Programming Interface)*. Penelitian ini menekankan bahwa sistem *gateway* bertindak sebagai penyedia pesan yang mampu merespons permintaan pengguna secara otomatis. Hal ini relevan dengan rencana implementasi fitur notifikasi pada sistem reservasi GOR Ma'arif agar setiap transaksi mendapatkan respons instan melalui perangkat WhatsApp penyewa.
3. Gibran et al. (2024) melakukan kajian terkait *Payment Gateway* sebagai portal yang menghubungkan rekening bank ke pihak *payment processor*

melalui layanan web dan API. Penelitian tersebut menjelaskan bahwa layanan ini mampu mengesahkan berbagai mode transaksi *online* seperti transfer bank dan dompet digital (*e-wallet*) secara virtual. Referensi ini memperkuat landasan peneliti untuk menggunakan sistem pembayaran yang lebih aman dan terverifikasi otomatis.

4. Fatman et al. (2023) secara spesifik meneliti tentang penggunaan *Midtrans* sebagai penyedia layanan *payment gateway* yang telah ada sejak 2012. Hasil kajiannya menunjukkan bahwa Midtrans memiliki integrasi yang mudah digunakan untuk berbagai jenis pembayaran *online* bagi pelaku usaha. Hal ini menjadi rujukan teknis yang sangat relevan bagi peneliti dalam mengintegrasikan sistem pembayaran pada reservasi di GOR Ma'arif agar lebih efisien.

C. Kerangka Berpikir



Gambar 2.1 Kerangka Berpikir