

BAB II

KAJIAN PUSTAKA

A. Sistem Informasi

Sistem informasi merupakan serangkaian prosedur-prosedur formal dimana data dikumpulkan dan diproses menjadi informasi untuk mendukung perolehan keputusan dan pengendalian didalam sebuah lembaga (Sitorus & Sakban, 2023). Prosedur ini tentunya juga saling berkaitan satu sama lain dalam proses berjalannya. Hal ini juga diperjelas dengan pendapat Maydianto & Ridho (2021) bahwa sekelompok berbagai komponen yang bekerja sama untuk mengelola data sehingga dapat diubah menjadi informasi yang berguna dan membantu pencapaian tujuan organisasi.

Pendapat lain yang disampaikan oleh Anika *et al.* (2023) menjelaskan bahwa sistem informasi dalam konsep teknologi terdiri dari jaringan, perangkat keras, perangkat lunak, dan sumber daya manusia yang berkolaborasi untuk mengumpulkan, menangani, memproses, dan mendistribusikan data yang dibutuhkan oleh suatu organisasi dengan tujuan utama yaitu untuk meningkatkan kinerja dan membantu dalam pengambilan keputusan. Pendapat ini juga diperkuat oleh Adham (2024) dalam penelitiannya yang menerangkan bahwa sistem informasi adalah sekumpulan elemen yang bekerja sama untuk mengumpulkan, mengelola, menyimpan, memproses, dan menyebarkan informasi yang diperlukan untuk meningkatkan efisiensi dan efektivitas operasional organisasi.

Dengan memperhatikan pendapat-pendapat di atas maka dapat disimpulkan bahwa sistem informasi adalah gabungan atau kumpulan antara prosedur, komponen teknologi dan sumber daya manusia yang terintegrasi dan memiliki fungsi untuk mengolah data yang telah dikumpulkan atau didapat untuk selanjutnya dirangkai menjadi sebuah informasi penting yang membantu dalam pengambilan keputusan dan kinerja.

B. Monitoring

Monitoring pada dasarnya merupakan bentuk pemantauan lebih lanjut *Monitoring* merupakan kegiatan pemantauan yang dilakukan secara sistematis dan berkelanjutan terhadap data, aktivitas, maupun status suatu proses. Kegiatan ini bertujuan untuk memperoleh informasi terkini, mengetahui perkembangan pelaksanaan kegiatan, serta memastikan proses berjalan sesuai dengan rencana dan target yang telah ditetapkan. Dalam pengelolaan kontrak layanan, *monitoring* diperlukan untuk memantau status masa aktif, batas waktu perpanjangan, dan kondisi kontrak agar tindakan yang diperlukan dapat dilakukan sebelum layanan memasuki masa tenggang atau berakhir (Pratama & Nasution, 2023).

Proses *monitoring* yang masih dilakukan secara manual berpotensi menyebabkan keterlambatan pembaruan data, kesulitan pencarian riwayat informasi, serta kesalahan pada pencatatan data. Pembangunan sistem informasi *monitoring* berbasis web dapat membantu pengguna melakukan pemantauan secara lebih terstruktur dan mempermudah penyusunan laporan.

Purnama *et al.* (2022) menjelaskan bahwa sistem informasi *monitoring* dapat digunakan meningkatkan kemudahan dalam pembuatan laporan *monitoring*.

Selain itu, sistem informasi *monitoring* berbasis web dapat mempercepat akses terhadap informasi, mengurangi kesalahan pencatatan manual, dan meningkatkan efisiensi operasional. Informasi yang tersedia secara terpusat memudahkan pihak yang berkepentingan dalam melakukan pengawasan, memperbarui data, serta mengambil keputusan berdasarkan kondisi data yang aktual (Lubis & Husaini, 2025).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa *monitoring* adalah kegiatan pemantauan data dan proses secara terencana, sistematis, dan berkelanjutan untuk mengetahui kondisi aktual suatu kegiatan. Dalam penelitian ini, *monitoring* diterapkan pada kontrak layanan *domain* dan *hosting* untuk memantau data layanan, masa aktif, tanggal berakhir, serta status perpanjangan. Hasil pemantauan tersebut digunakan sebagai dasar untuk menampilkan informasi tenggat waktu dan mengirimkan notifikasi sebelum kontrak layanan berakhir.

C. Kontrak Layanan *Domain* dan *Hosting*

Layanan *domain* dan *hosting* merupakan bagian penting dalam penyelenggaraan *website* maupun aplikasi berbasis internet. *Domain* digunakan sebagai identitas atau alamat layanan pada internet, sedangkan *hosting* menyediakan sumber daya server yang digunakan untuk menyimpan data dan menjalankan *website* atau aplikasi. Dalam praktiknya, *domain* dan *hosting* dapat dikelola melalui platform berbasis web yang menyediakan fitur

pendaftaran *domain*, pembelian *hosting*, pengelolaan akun, serta bantuan layanan pengguna (Fahmi *et al.*, 2023).

Layanan *hosting* memiliki peran terhadap kualitas akses *website* atau aplikasi. Performa layanan *hosting* dapat dipengaruhi oleh beberapa parameter, seperti *load time*, *Domain Name System* (DNS), koneksi, dan *Secure Socket Layer* (SSL). Oleh sebab itu, pemilihan serta pemantauan layanan *hosting* diperlukan untuk menjaga kualitas akses dan ketersediaan *website* bagi pengguna.

Selain performa, pengelolaan layanan *hosting* juga perlu memperhatikan aspek keamanan. Khare & Badholia (2022) menjelaskan bahwa layanan *self-web-hosting* maupun *cloud-web-hosting* dapat dianalisis berdasarkan parameter keamanan, seperti autentikasi, perlindungan jaringan, pencatatan aktivitas sistem, *spam filtering*, CAPTCHA, dan pengelolaan kata sandi. Berdasarkan uraian tersebut, kontrak layanan *domain* dan *hosting* dalam penelitian ini dipahami sebagai data berlangganan layanan yang memuat informasi *domain* atau *hosting*, penyedia layanan, masa aktif, biaya, serta tanggal berakhir. Data tersebut perlu dikelola dan dimonitor secara terstruktur agar proses perpanjangan layanan dapat dilakukan sebelum masa aktif berakhir.

D. *Deadline Tracking*

Deadline Tracking merupakan mekanisme pemantauan waktu yang digunakan untuk mencatat, melacak, dan mengevaluasi tanggal penting dalam suatu kontrak secara berkala. Dalam konteks pengelolaan kontrak, proses

pemantauan diperlukan untuk mendukung pengambilan keputusan terkait kelanjutan dan perpanjangan layanan. Organisasi perlu melakukan pelacakan terhadap faktor-faktor kontrak secara berkelanjutan agar dapat merespons perubahan kebutuhan layanan dan mengambil keputusan perpanjangan secara tepat.

Pada penelitian ini, *Deadline Tracking* diterapkan pada kontrak layanan *domain* dan *hosting*. Layanan *domain* dan *hosting* perlu dikelola secara terstruktur melalui sistem berbasis web agar informasi layanan dapat dicatat dan dikelola secara terpusat. Setiap layanan memiliki tanggal mulai, tanggal berakhir, biaya, serta periode perpanjangan yang perlu diperhatikan oleh Admin.

Melalui fitur *Deadline Tracking*, sistem mencatat tanggal mulai dan tanggal berakhir layanan, menghitung sisa hari secara otomatis, serta menampilkan status layanan berdasarkan kedekatannya dengan tanggal jatuh tempo. Informasi tersebut membantu Admin memprioritaskan layanan *domain* atau *hosting* yang perlu segera diperpanjang. Pemantauan tersebut penting karena kualitas layanan *hosting* berhubungan dengan akses dan performa website, termasuk *load time*, DNS, dan koneksi. Apabila masa aktif layanan berakhir dan tidak segera ditindaklanjuti, *website* atau aplikasi dapat berisiko mengalami gangguan akses.

Dengan demikian, *Deadline Tracking* pada penelitian ini merupakan mekanisme sistem informasi untuk memantau masa aktif kontrak layanan *domain* dan *hosting* secara berkala, menghitung sisa waktu menuju tanggal

berakhir, serta menyediakan informasi bagi Admin agar perpanjangan layanan dapat dilakukan sebelum masa aktif berakhir.

E. *Threshold Notification*

Threshold Notification merupakan mekanisme notifikasi otomatis yang diaktifkan ketika suatu kondisi telah mencapai batas atau ambang tertentu (*threshold*). Dalam sistem monitoring, penetapan *threshold* digunakan untuk menentukan kapan sistem perlu memberikan peringatan kepada pengguna agar tindakan dapat dilakukan sebelum kondisi yang tidak diinginkan terjadi. Sistem peringatan berbasis *threshold* memerlukan nilai ambang yang jelas agar notifikasi dapat dikirim secara tepat waktu dan tidak menimbulkan peringatan yang berlebihan.

Pada penelitian ini, *Threshold Notification* diterapkan untuk memberikan peringatan kepada Admin ketika masa aktif layanan *domain* dan *hosting* mendekati tanggal berakhir. Nilai ambang notifikasi dapat dikonfigurasi, misalnya pada H-30, H-14, dan H-3 sebelum tanggal berakhir layanan. Ketika sisa masa aktif layanan mencapai salah satu nilai tersebut, sistem secara otomatis membuat dan mengirimkan notifikasi kepada Admin melalui WhatsApp.

Penerapan notifikasi otomatis berbasis jadwal dapat membantu mengurangi ketergantungan pada pengingat manual. Fatah *et al.* (2025) mengembangkan sistem pengingat berbasis WhatsApp yang dapat mengirimkan notifikasi otomatis berdasarkan jadwal yang telah ditentukan

oleh pengguna sistem. Sistem pengingat otomatis dapat dikirimkan sebelum waktu kegiatan yang telah dijadwalkan, sehingga pengguna memperoleh informasi lebih awal untuk melakukan tindakan yang diperlukan.

Dengan demikian, *Threshold Notification* pada penelitian ini merupakan mekanisme peringatan dini yang mengirimkan notifikasi WhatsApp secara otomatis ketika sisa masa aktif kontrak layanan *domain* dan *hosting* mencapai nilai ambang yang telah ditetapkan. Fitur ini bertujuan membantu Admin mempersiapkan proses perpanjangan layanan secara lebih tepat waktu serta mengurangi risiko keterlambatan akibat kelalaian dalam memantau tanggal berakhir layanan.

F. Website

Website merupakan media berbasis internet yang dapat digunakan untuk menyajikan informasi, menyediakan layanan, serta mendukung interaksi pengguna melalui peramban (*web browser*). Dalam pengembangan sistem informasi, platform berbasis website dapat digunakan untuk membantu proses pengelolaan data dan penyampaian informasi secara lebih efektif serta efisien (Zeva, 2023).

Sistem informasi berbasis *website* memungkinkan pengguna mengakses fitur dan informasi melalui jaringan internet tanpa harus melakukan instalasi aplikasi pada setiap perangkat pengguna. Pengembangan sistem berbasis *website* umumnya memanfaatkan teknologi seperti Hypertext Markup Language (HTML), Hypertext Preprocessor (PHP), serta basis data

MySQL untuk mengelola data dan menyediakan informasi bagi pengguna (Rosa *et al.*, 2024).

Penerapan sistem berbasis *website* mampu meningkatkan efisiensi proses pengelolaan data, mempercepat pencarian informasi, mengurangi kesalahan pencatatan manual, serta memudahkan pembuatan laporan (Bakhar *et al.*, 2025). Oleh karena itu, *website* digunakan dalam penelitian ini sebagai media untuk membangun Sistem Informasi Monitoring Kontrak Layanan *Domain* dan *Hosting* . Melalui sistem tersebut, Admin dapat mengelola data layanan, memantau masa aktif kontrak, menerima notifikasi, serta melihat riwayat perpanjangan secara terpusat.

1. *Framework*

Dalam konsep pembuatan sebuah sistem, *framework* menjadi salah satu tumpuan yang bisa membantu dan memberikan kemudahan dalam implementasinya. *Framework* merupakan kumpulan fungsi, *Class*, dan aturan yang ada, beda dengan library yang sifatnya hanya untuk tujuan tertentu, *framework* lebih mengatur pada bagaimana aplikasi dibuat. Pendapat ini didukung oleh Rahmawati & Sumarsono (2024) yang memaparkan bahwa *framework* adalah sebuah kerangka kerja yang dibuat untuk mempermudah proses perancangan sebuah *website*. Dari pendapat diatas, dapat disimpulkan bahwa *framework* adalah alat yang menyediakan infrastruktur dan aturan pengembangan sebuah aplikasi atau sistem. Tujuan dari *framework* sendiri meliputi untuk mempermudah proses perancangan sistem, meningkatkan

efisiensi, serta memastikan konsistensi pembuatan sistem tersebut. Untuk menyelamatkan programmer dari keharusan menulis skrip yang sama berulang-ulang, *framework* juga merupakan komponen pemrograman yang selalu siap digunakan.

2. Laravel

Laravel juga merupakan salah satu *framework* open source yang populer untuk membangun sebuah sistem *website*. *Framework* ini berbasis bahasa pemrograman PHP dengan menerapkan pola MVC dalam implementasinya. *Framework* laravel ini sendiri memiliki fitur routing yang berfungsi menentukan respons aplikasi web terhadap permintaan dari pengguna, dengan menghubungkan URL yang diakses ke kode program yang akan dieksekusi di dalam aplikasi. (Siswanto, 2023). Pendapat diatas juga diperjelas oleh Rahayu *et al.* (2023) yang menerangkan bahwa laravel merupakan sebuah *framework* PHP (*hypertext preprocessor*) dengan agar bisa bersifat open source digunakanlah sebuah konsep berupa *MVC (Model View Controller)* yang digunakan untuk membangun *website* atau sistem. Model *MVC* sendiri merupakan teknik dalam pemrograman yang memisahkan antara alur pikir (*bisnis logic*), penyimpanan data (*data logic*), dan tampilan antarmuka (*presentation logic*).

Dari pendapat di atas, dapat disimpulkan bahwa laravel adalah *framework* yang sangat populer yang memiliki open source yang sering digunakan saat ini dengan berbasis bahasa pemrograman PHP dengan

memanfaatkan pola *MVC (Model View Controller)* untuk membangun *website* atau sistem serta memberikan kerangka kerja yang terstruktur dan mudah dikelola.

3. Laragon

Laragon merupakan suatu server virtual yang bisa digunakan untuk menjalankan sebuah program. Andarsyah *et al.* (2022) menerangkan bahwa laragon adalah sebuah perangkat lunak yang berjenis *open source* yang dapat mendukung banyak sistem operasi yang dapat dijalankan didalamnya yang mana juga bertugas sebagai server virtual atau localhost. Dalam aplikasi ini, terdapat banyak layanan, alat dan fitur pendukung berupa apache, MySQL, composer, PhpMyAdmin dan masih banyak fitur lain yang dapat digunakan.

Sehingga dapat disimpulkan bahwa laragon cukup menjadi pilihan bagi pengembang aplikasi karena memudahkan pengguna dalam menjalankan dan mengembangkan program secara lokal. Laragon juga menjadi solusi efisien untuk kebutuhan pengembangan perangkat lunak karena dilengkapi dengan fitur-fitur yang terdapat didalamnya yang tentunya dapat memperkecil kebutuhan instalasi sistem.

4. PHP

PHP (*Hypertext Preprocessor*) adalah sebuah bahasa pemrograman yang populer diaplikasikan dalam sebuah *framework* yang bersifat *open source*. Hermiati *et al.* (2021) menerangkan bahwa

PHP adalah bahasa skrip yang digunakan untuk mengembangkan halaman web dinamis yang terintegrasi dengan tag HTML dan berjalan di server. Sudaria *et al.* (2021) dalam penelitiannya menjelaskan PHP adalah bahasa pemrograman yang mengubah kode program menjadi kode mesin yang dapat disisipkan ke dalam HTML, diinterpretasikan, dan dijalankan oleh komputer sisi server.

Berdasarkan sudut pandang yang disebutkan di atas, PHP adalah bahasa pemrograman sumber terbuka yang banyak digunakan untuk membuat aplikasi web. PHP mengubah kode program menjadi bahasa mesin sehingga dapat dijalankan oleh komputer.

G. Basis Data

Basis data adalah komponen penting dalam penyusunan program *website*. Menurut Syamsudin & Ali (2024) basis data adalah kumpulan informasi yang disimpan secara sistematis di komputer sehingga informasi dapat diambil darinya menggunakan program komputer. Sedangkan menurut Allisya & Sari (2023) dalam penelitiannya menerangkan basis data adalah kumpulan dari beberapa data yang terhubung satu sama lain untuk memungkinkan pencarian data yang lebih cepat. Kumpulan data yang saling berhubungan dan perangkat lunak yang digunakan untuk mengelola serta mengakses data tersebut membentuk sistem manajemen basis data (DBMS) yang bertujuan menyediakan cara untuk menyimpan, mengelola, dan mengakses data yang telah dikumpulkan secara terstruktur (Billah *et al.*, 2023)

Sehingga dapat disimpulkan bahwa basis data merupakan komponen kunci dalam pengembangan suatu sistem yang berisikan kumpulan informasi yang tersimpan secara rapi dan terstruktur dalam komputer sehingga nantinya mempermudah dalam pencarian data. Basis data juga bisa disebut sebagai otak dalam penyimpanan dan pengelolaan data pada program *website*. Sedangkan *database management system* atau DBMS sendiri juga dapat diartikan sebagai sekumpulan program yang dirancang untuk mengakses, menyimpan, dan mengelola data secara efektif. Sehingga basis data yang didukung oleh DMBS menjadi bagian yang penting untuk pengembangan suatu sistem informasi dalam hal pengolahan data.

H. MySQL

MySQL merupakan konsep utama dalam *database* yang merupakan turunan dari SQL (*Structured Query Language*), sedangkan SQL sendiri merupakan konsep *tool database management system* (DBMS) *open source* yang mendukung banyak pengguna, banyak thread, dan gratis. Berdasarkan teori di atas, SQL adalah bahasa permintaan *database* yang memiliki subbahasa yang dapat membuat dan mengubah data dalam *database* (Rina Noviana, 2022).

Berdasarkan sudut pandang yang disebutkan di atas, SQL adalah sebuah konsep utama dalam operasi *database* yang mempermudah pemilihan, pemasukan dan pengolahan data secara otomatis. Sedangkan MySQL sebagai salah satu sistem basis data yang mendukung konsep SQL berfungsi sebagai

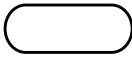
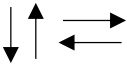
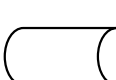
alat praktis untuk mengelola basis data relasional dan memudahkan pengguna dalam pengolahan data secara efisien.

I. Flowchart

Flowchart dapat diartikan sebagai bagan alur suatu program. Dalam penjelasannya Rasiban *et al.* (2022) mengungkapkan bahwa bagan alur adalah diagram yang menunjukkan pilihan dan tindakan yang diperlukan untuk menyelesaikan prosedur program yang digambar dan dihubungkan dengan garis atau panah. Pendapat ini didukung oleh Listyoningrum *et al.* (2023) yang menerangkan bahwa *Flowchart* merupakan gambaran grafik yang memiliki urutan suatu proses atau langkah-langkah yang berjalan secara sistematis untuk menjalankan sebuah program, dapat juga diartikan sebagai suatu diagram yang menggunakan simbol-simbol grafis yang menggambarkan alur suatu proses.

Dapat disimpulkan bahwa *Flowchart* secara keseluruhan merupakan model grafis yang memaparkan alur tahapan dari proses yang berjalan dalam sebuah sistem. Tidak hanya itu, *Flowchart* juga memberikan kemudahan dalam pemahaman alur program dan menjadi alat komunikasi yang efektif pada perancangan sistem untuk memastikan setiap tahapan proses dapat di implementasikan.

Tabel 2.1 Simbol *Flowchart*

No	Simbol	Nama	Keterangan
1		<i>Terminator</i> (Terminal)	Menyatakan awal (Mulai) dan akhir (Selesai) dari suatu program atau proses.
2		<i>Flow Line</i> (Garis Alir)	Menunjukkan arah aliran proses dari satu langkah ke langkah berikutnya.
3		<i>Process</i> (Proses)	Menyatakan proses pengolahan atau perhitungan data.
4		<i>Decision</i> (Keputusan)	Menyatakan percabangan atau pengambilan keputusan berdasarkan kondisi (ya/tidak).
5		<i>Input/Output</i> (Data)	Menyatakan proses masukan (input) atau keluaran (output) data.
6		<i>Preparation</i> (Persiapan)	Menyatakan pemberian nilai awal atau persiapan suatu variabel/penghitung.
7		<i>Connector</i> (Penghubung)	Penghubung bagian-bagian <i>flowchart</i> pada halaman yang sama.
8		<i>Off-Page Connector</i>	Penghubung <i>flowchart</i> yang berlanjut ke halaman yang berbeda.
9		<i>Document</i> (Dokumen)	Menyatakan masukan/keluaran berupa dokumen atau laporan tercetak.
10		<i>Disk / on-line Storage</i>	Menyatakan input yang berasal dari <i>disk</i> atau disimpan ke <i>disk</i>

Sumber: (Rasiban *et al.*, 2022)**J. Extreme Programming (EP)**

Metode ini merupakan model pengembang perangkat lunak yang bertujuan untuk membuat banyak fase proses pengembangan menjadi lebih sederhana untuk meningkatkan efektivitas dan fleksibilitasnya (Rudianto, 2023). Model pengembangan ini tidak hanya berfokus pada pengkodean namun juga mencakup semua lini pengembangan perangkat lunak. Model ini adalah suatu model yang dapat digunakan untuk membangun atau mengembangkan suatu sistem informasi yang dilakukan secara efisien dan pada setiap tahapannya dilakukan perubahan yang cepat serta fleksibel dibandingkan dengan model lainnya (Taufiq *et al.*, 2022).

Dalam metode *extreme programming*, terdapat pengulangan proses yang dilakukan berulang kali sesuai dengan kebutuhan, dengan salah satu kelebihanannya yaitu memberikan tawaran fase dengan cepat dan diulang untuk berbagai bagian berdasarkan tingkat perhatian yang dicapai (Fazrin *et al.*, 2021). Metode ini memiliki beberapa tahapan pelaksanaan adalah perencanaan (*planning*), pemodelan (*designing*), pengkodean (*coding*), dan pengujian (*testing*).

Selanjutnya adalah penjelasan tentang tahapan pelaksanaannya:

1. Perencanaan (*planning*)

Perencanaan adalah tahap pertama dari pengembangan sistem, di mana sejumlah tugas perencanaan diselesaikan, termasuk analisis kebutuhan, identifikasi masalah, dan membuat jadwal pengembangan sistem (Insani *et al.*, 2022).

2. Pemodelan (*designing*)

Langkah selanjutnya adalah desain, ketika tugas pemodelan selesai, mulai dari pemodelan basis data hingga pemodelan sistem dan pemodelan arsitektur (Insani *et al.*, 2022).

3. Pengkodean (*coding*)

Langkah ini melibatkan pemanfaatan bahasa pemrograman untuk menggabungkan model yang dikembangkan ke dalam antarmuka pengguna (Insani *et al.*, 2022).

4. Pengujian (*testing*)

Setelah fase pengkodean selesai, sistem diuji untuk melihat apakah sistem yang dikembangkan memenuhi persyaratan pengguna dan untuk mengidentifikasi kesalahan yang mungkin timbul saat program dijalankan.

Setelah fase pengkodean selesai, fase pengujian sistem adalah untuk memastikan apakah terdapat kesalahan selama eksekusi program dan apakah sistem yang dikembangkan memenuhi persyaratan pengguna (Insani, *et al.*, 2022).

Dari pendapat yang ada, maka dapat disimpulkan bahwa *extreme programming* adalah model pengembangan perangkat lunak dengan tujuan menyederhanakan suatu tahapan pengembangan sehingga lebih efektif dan sangat fleksibel yang tidak hanya berfokus pada pengkodean namun juga seluruh aspek dalam pengembangan. Secara keseluruhan, model pengembangan

extreme programming memberikan pendekatan yang dinamis dan responsif terhadap pengembangan dalam sistem informasi.

K. *Unified Modeling Language (UML)*

Pada sebuah sistem, *Unified Modeling Language (UML)* berfungsi sebagai alat bantu untuk mendeskripsikan dan merancang sistem yang dibangun menggunakan konsep pemrograman berorientasi objek. Suharni *et al.* (2023) menjelaskan bahwa UML merupakan metode pemodelan berorientasi objek yang menggunakan simbol-simbol tertentu untuk merepresentasikan objek, proses, serta hubungan dalam sistem. Dengan penggunaan UML, rancangan sistem dapat divisualisasikan secara lebih terstruktur sebelum dilakukan proses implementasi.

UML merupakan bahasa pemodelan standar yang digunakan untuk menggambarkan, merancang, serta mendokumentasikan sistem perangkat lunak. UML membantu pengembang menerjemahkan kebutuhan pengguna ke dalam model visual, termasuk rancangan proses bisnis, struktur data, fungsi sistem, dan interaksi antarobjek.

Dalam penelitian ini, diagram UML yang digunakan meliputi *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram*. *Use Case Diagram* digunakan untuk menggambarkan interaksi antara aktor dengan fungsi sistem. *Activity Diagram* digunakan untuk menggambarkan alur aktivitas sistem. *Class Diagram* digunakan untuk menggambarkan struktur kelas, atribut, serta hubungan antarobjek.

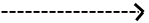
Sementara itu, *Sequence Diagram* digunakan untuk menggambarkan urutan interaksi atau pertukaran pesan antarobjek dalam menjalankan proses sistem (Ramdany *et al.*, 2024).

a. *Use Case diagram*

Use Case diagram merupakan suatu interaksi antara sistem dengan aktor yang memiliki alur dan kemudian diterapkan pada sistem yang nantinya dibuat, yang mana nantinya interaksi ini menjadi dasar perintah saat aktor melakukan tindakan maka sistem akan merespon tindakan tersebut dan menjalankan sesuai sebagaimana fungsinya. *Use Case diagram* juga berfungsi sebagai alat untuk melihat atau menggambarkan kebutuhan yang diharapkan sistem, memodelkan layanan yang disediakan oleh sistem, memetakan kebutuhan sistem, merepresentasikan interaksi antar aktor, serta mengetahui kebutuhan eksternal dari sistem (Puturuhi, 2024).

Tabel 2.2 Simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
1		Aktor	Mewakili orang
Abstraksi antara sistem dan aktor			
2		<i>Use Case</i>	
Penghubung use case			

No	Simbol	Nama	Keterangan
3		<i>Association</i>	dengan aktor
4		Generalisas i	berhubungan dengan <i>Use Case</i>
5		<i>Include</i>	Simbol yang mewakilkan bahwa suatu <i>Use Case</i> seluruhnya merupakan fungsionalitas dari <i>Use Case</i> lainnya.
6		<i>Extend</i>	Simbol yang mewakilkan bahwa suatu <i>Use Case</i> merupakan tambahan fungsional dari <i>Use Case</i> lainnya jika suatu kondisi terpenuhi.

Sumber : (Harlina *et al.*, 2025)

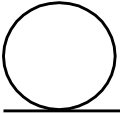
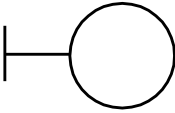
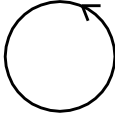
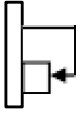
b. Activity diagram

Activity diagram merupakan gambaran visualisasi dari kegiatan yang terjadi dalam sistem yang menjelaskan tentang bagaimana sistem merespon dan menampilkan hasil dari perintah dengan urutan yang sesuai (Sutrisno & Karnadi, 2021).

c. Sequence diagram

Sequence diagram merupakan salah satu jenis diagram pada *UML* yang dapat menjelaskan urutan waktu pemrosesan sistem (Sumiati *et al.*, 2021) . *Sequence diagram* menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunannya untuk menunjukkan rangkaian pesan yang dikirim antara objek dan interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi.

Tabel 2.4 Simbol *Sequence Diagram*

No	Simbol	Nama	Keterangan
1		<i>Entity Class</i>	Simbol yang menggambarkan sistem sebagai dasar untuk kompilasi basis data.
2		<i>Boundary Class</i>	Simbol yang menangani komunikasi antar lingkungan sistem.
3		<i>Control Class</i>	Simbol yang bertanggung jawab atas tipe objek yang berisi logika.
4		<i>Recursive</i>	Simbol yang melambangkan pesan terhadap dirinya.
5		<i>Activation</i>	Simbol yang menunjukkan berapa lama prosedur yang

No	Simbol	Nama	Keterangan
			diperlukan untuk mengaktifkan sistem.
6		<i>Life Line</i>	Simbol yang menggambarkan aktivitas dari objek.

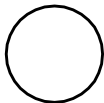
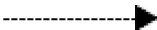
Sumber : (Sumiati *et al.*, 2021).

d. *Class diagram*

Ilustrasi struktur sistem yang menentukan kelas-kelas yang akan digunakan untuk membangun sistem disebut diagram kelas. Salah satu interpretasi yang mungkin dari diagram ini adalah bahwa diagram tersebut menggambarkan struktur statis sistem aplikasi berorientasi objek. *Class diagram* mendefinisikan kerangka dasar yang menunjukkan interaksi antar kelas (Martono *et al.*, 2023).

Dengan visualisasi struktur kelas yang jelas, tentunya diagram ini memungkinkan pengembang merancang arsitektur sistem sebelum melakukan implementasi kode dan memastikan konsistensi dalam pengembangan sistem yang berorientasi objek.

Tabel 2.5 Simbol *Class Diagram*

No	Simbol	Nama	Keterangan
1		<i>Class</i>	Kelas pada struktur sistem
2		<i>Interface</i>	Sama dengan konsep interface dalam pemrograman berorientasi
3		<i>Association</i>	Multiplisitas biasanya dikaitkan dengan interaksi antarkelas secara umum
4		<i>Direct Association</i>	Multiplisitas biasanya dikaitkan dengan hubungan antara kelas dan makna suatu kelas atau bagaimana kelas lain menggunakannya
5		<i>Generalisasi</i>	Hubungan antara kelas dan apa arti generalisasi- spesialisasi (umum-spesifik)
6		<i>Dependency</i>	Hubungan antar kelas dan apa arti ketergantungan antar kelas

No	Simbol	Nama	Keterangan
7		Aggregation	Relasi antar kelas dengan makna semua-bagian (whole-part)

Sumber : (Ramdany *et al.*, 2024).

L. Kajian Empiris

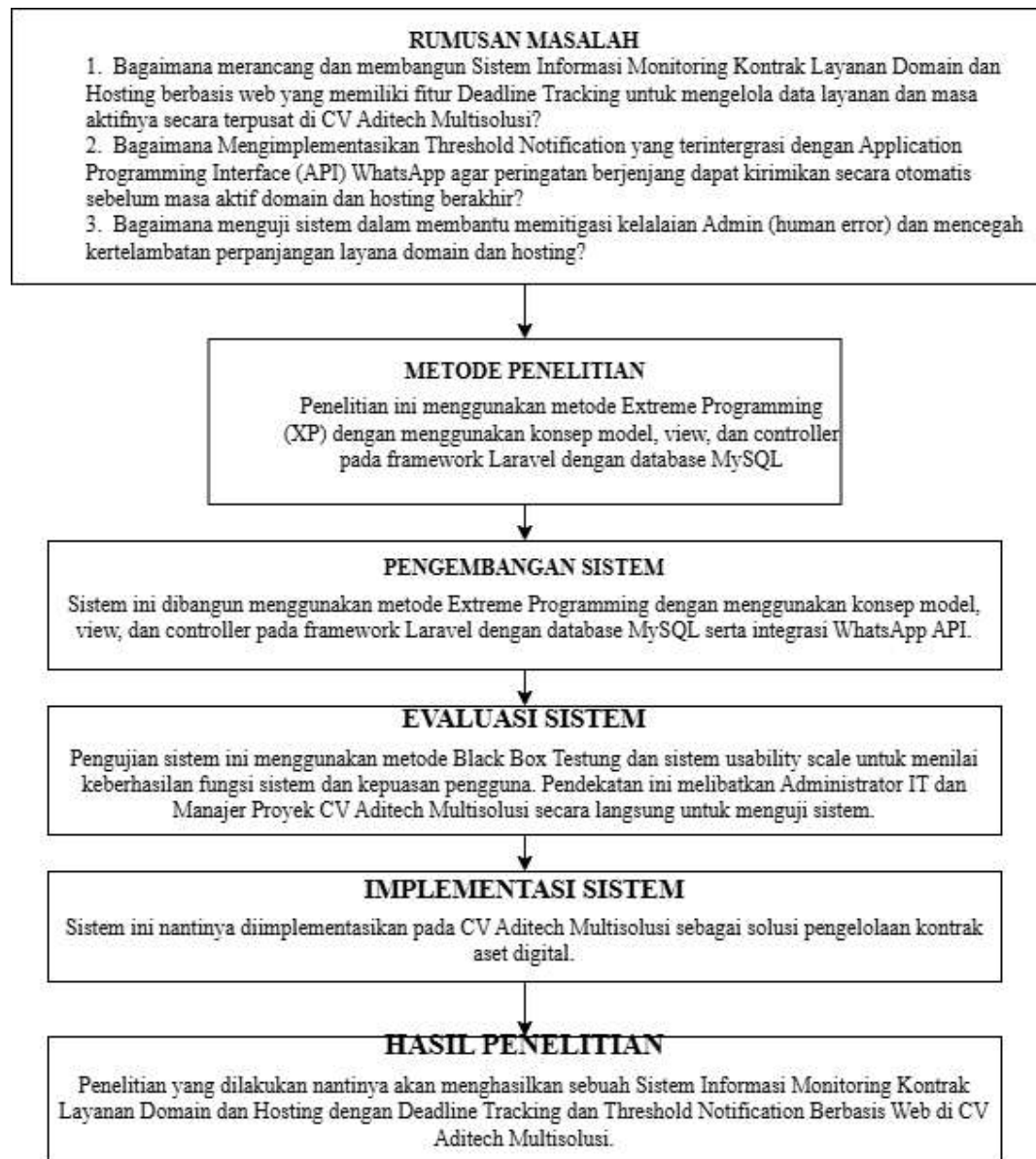
Penelitian terdahulu menunjukkan bahwa sistem informasi berbasis web dapat meningkatkan efisiensi pencatatan dan monitoring data dibandingkan spreadsheet manual. Berbagai studi mengenai manajemen kontrak dan pengingat masa aktif menegaskan perlunya data terpusat, pencarian yang cepat, serta penyajian status yang mudah dipahami. Namun, penerapan khusus untuk monitoring kontrak layanan *domain* dan *hosting* dengan pengingat berjenjang masih perlu dikembangkan.

Penelitian lain juga menunjukkan bahwa notifikasi otomatis melalui aplikasi pesan instan dapat membantu pengguna menerima informasi penting secara lebih cepat. Mekanisme pengingat tersebut relevan untuk masa aktif *domain* dan *hosting* karena keterlambatan perpanjangan dapat menyebabkan layanan *website* atau aplikasi tidak tersedia.

Berdasarkan kajian tersebut, penelitian ini berfokus pada integrasi pengelolaan kontrak layanan *domain* dan *hosting*, *Deadline Tracking* untuk menghitung sisa masa aktif, *Threshold Notification* berjenjang yang dapat dikonfigurasi, serta pengiriman notifikasi otomatis melalui WhatsApp API.

Sistem dikembangkan berbasis web untuk digunakan di CV Aditech Multisolusi.

M. Kerangka Berpikir



Gambar 2.1 Kerangka Berfikir