

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Konsep Dasar Sistem

Sistem merupakan kumpulan rangkaian – rangkaian dari subsistem yang lebih kecil dan berjalan secara beriringan dengan baik. Sistem terdiri dari gabungan *elemen, komponen, atau variabel* yang saling berinteraksi untuk mencapai tujuan tertentu. Setiap bagian memiliki peran dalam mekanisme kerja yang terstruktur agar sistem berfungsi secara efektif (Maydianto & Ridho, 2021). Keterkaitan antara satu komponen dan komponen lain ini berjalan secara harmonis dan teratur membentuk satu kesatuan yang utuh sehingga disebut dengan sebuah sistem.

Pendapat ini diperkuat oleh (Effendy et al., 2023) dalam penelitiannya yang turut memaparkan bahwa yang dimaksud dengan sistem merupakan sebuah satu kesatuan yang terdiri dari kumpulan komponen atau elemen yang dihubungkan bersama dengan tujuan untuk memudahkan aliran informasi, materi, atau energi untuk mencapai suatu tujuan yang sama.

Dengan adanya pendapat diatas, maka dapat disimpulkan bahwa sistem merupakan kumpulan dari bagian – bagian subsistem yang dihubungkan dan berjalan bersama baik secara fisik maupun non fisik untuk memudahkan aliran informasi dari suatu kondisi yang memiliki satu tujuan yang sama dan harmonis.

2. Sistem Informasi

Menurut (Sitorus & Sakban, 2023) yang mengungkapkan bahwa sistem informasi merupakan serangkaian prosedur - prosedur formal dimana data dikumpulkan dan diproses menjadi informasi untuk mendukung pengambilan keputusan dan pengendalian didalam sebuah organisasi. Prosedur ini tentunya juga saling berkaitan satu sama lain dalam proses berjalannya. Hal ini juga diperjelas dengan pendapat (Maydianto & Ridho, 2021) bahwa sebuah kumpulan dari beberapa komponen yang mengelola data supaya data yang diolah dapat dijadikan sebagai sebuah informasi yang bermakna dan dapat membantu dalam mencapai tujuan dalam suatu organisasi. Menurut (Widarti et al., 2024:28), sistem informasi dirancang untuk mencapai tujuan yang berkaitan dengan efisiensi, pengambilan keputusan, dukungan teknis maupun peningkatan produktifitas tergantung pada integrasi baik antara pengguna, proses, dan teknologi.

Pendapat lain yang dikemukakan oleh (Anika et al., 2023) menjelaskan bahwa sistem informasi dalam konsep teknologi adalah kombinasi dari perangkat keras, perangkat lunak, jaringan dan sumber daya manusia yang bekerja sama untuk mengumpulkan, mengelola, mengolah, dan menyampaikan informasi yang dibutuhkan dalam sebuah organisasi dengan tujuan utama yaitu untuk membantu dalam pengambilan keputusan dan meningkatkan kinerja. Pendapat ini juga diperkuat oleh (Adham, 2024) dalam penelitiannya yang menerangkan bahwa sistem informasi merupakan serangkaian komponen yang bekerja bersama untuk mengumpulkan,

mengelola, menyimpan, memproses, dan menyebarkan informasi yang diperlukan untuk mendukung pengambilan suatu keputusan dalam sebuah organisasi atau entitas.

Dari beberapa pendapat di atas maka dapat disimpulkan bahwa sistem informasi merupakan 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 yang bermakna yang digunakan sebagai pendukung pengambilan keputusan dan kinerja.

3. Penjadwalan Mata Kuliah

Penjadwalan menjadi salah satu bagian penting dalam proses manajemen akademik terutama di institusi pendidikan tinggi. Proses manajemen akademik ini melibatkan penjadwalan Mata Kuliah, pengelolaan sumber daya manusia, dan memastikan pengalaman belajar yang efisien dan bermakna bagi mahasiswa, yang mana penjadwalan yang baik bisa mempengaruhi berbagai aspek yaitu konflik jadwal dan juga optimalisasi pemanfaatan fasilitas yang disediakan (Hartono & Zein, 2023). Dalam penelitian lain yang dilakukan oleh (Wiratna et al., 2023) turut memaparkan permasalahan yang sering terjadi dalam penyusunan jadwal Mata Kuliah adalah bagaimana mengkombinasikan antara dosen, Mata Kuliah, ruang, waktu dan kelas yang baik tanpa adanya tabrakan antar kegiatan.

Dapat disimpulkan dari beberapa pendapat diatas, sejatinya penjadwalan Mata Kuliah yang baik merupakan rangkaian proses penyusunan Mata Kuliah yang memastikan pengelolaan yang efektif antara, dosen, Mata Kuliah, ruang, waku dan kelas yang optimal sehingga dapat mengurangi konflik antar jadwal dan penggunaan fasilitas yang maksimal.

4. *Website*

Website merupakan suatu hal yang sudah menjadi bagian dari kemajuan teknologi, *website* juga menjadi salah satu metode untuk menampilkan informasi, berupa gambar, teks, suara, ataupun video interaktif dan memiliki kelebihan untuk menghubungkan satu dokumen dengan dokumen lainnya. *Website* secara umum memiliki beberapa fungsi yang meliputi media komunikasi yang bersifat dinamis, media informasi yang menekankan kualitas konten isi dari tulisan, media hiburan, serta media transaksi (Ahmad et al., 2024).

4.1. *Framework*

Dalam konsep pembuatan sebuah sistem, *framework* menjadi salah satu tumpuan yang bisa membantu dan memberikan kemudahan dalam implementasinya. Menurut (Muthia Kansha et al., 2023) *framework* merupakan kumpulan fungsi, class, dan aturan – aturan yang ada, berbeda 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 kerangka kerja tersebut dikembangkan untuk mempermudah proses desain sebuah situs berbasis web.

Dari pendapat diatas, maka 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. *Framework* juga merupakan komponen pemrograman yang siap digunakan kapan saja sehingga programmer tidak perlu membangun script yang sama dari awal.

4.2. *Laravel*

Salah satu *framework* sumber terbuka yang digemari untuk membangun sebuah sistem situs web adalah *Laravel*. *Framework* ini berbasis bahasa pemrograman *Hypertext Preprocessor (PHP)* dengan menerapkan pola *MVC (Model View Controller)* dalam implementasinya. *Framework Laravel* ini sendiri memiliki fitur routing yang digunakan untuk menghubungkan antara request user dan controller untuk penerimanya (Yuniarti et al., 2022). Pendapat diatas juga diperjelas oleh (Rahayu et al., 2023) yang menerangkan bahwa *Laravel* adalah sebuah *framework PHP* 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 (*business logic*), penyimpanan data (*data logic*), dan tampilan antarmuka (*presentation logic*) (Siswanto, 2023:14).

Dari pendapat diatas, dapat disimpulkan bahwa *Laravel* merupakan suatu *framework* populer yang bersifat *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.

4.3. Laragon

Laragon merupakan suatu *server virtual* yang bisa digunakan untuk menjalankan sebuah program. Menurut (Andarsyah et al., 2022) menerangkan bahwa *laragon* merupakan sebuah perangkat lunak yang bersifat *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 (Handoyo & Anwar, 2023).

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 di dalamnya yang tentunya dapat memperkecil kebutuhan instalasi sistem.

4.4. *Hypertext Preprocessor (PHP)*

PHP (Hypertext Preprocessor) merupakan sebuah bahasa pemrograman yang populer digunakan dalam sebuah *framework* yang bersifat *open source*. Menurut Hermiati, (Hermiati et al., 2021) menerangkan bahwa *PHP* dikenal sebagai sebuah bahasa scripting yang menyatu dengan tag-tag *Hypertext Markup Language (HTML)*, dieksekusi di *server*, dan digunakan untuk membuat halaman web yang dinamis. (Sudaria et al., 2021) dalam penelitiannya menjelaskan bahwa *PHP* adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan kode program menjadi kode mesin yang dapat dimengerti dan dijalankan oleh komputer yang berbasis *server-side* yang juga dapat ditambahkan pada *HTML*. *PHP* juga bisa digunakan dalam berbagai sistem operasi seperti *linux*, *windows*, dan *unix* dengan salah satu fungsi utamanya adalah menangani kebutuhan standar pembuatan aplikasi website (Setiawan et al., 2022).

Dari pendapat diatas, maka dapat disimpulkan bahwa *PHP* merupakan bahasa pemrograman *open source* yang populer dan sering digunakan dalam pengembangan aplikasi web. Berfungsi untuk menerjemahkan kode program menjadi bahasa mesin yang dapat dijalankan oleh komputer.

5. Basis Data

Basis data merupakan salah satu komponen penting dalam penyusunan program *website*. Menurut (Jamaluddin et al., 2022:2) basis data merupakan kumpulan informasi yang disimpan dalam komputer secara sistematis sehingga dapat diperiksa menggunakan program komputer untuk memperoleh informasi dari dalamnya. Sedangkan menurut (Wijaya et al., 2021) dalam penelitiannya menerangkan bahwa basis data adalah kumpulan dari beberapa data yang saling berelasi antara satu dengan lainnya sehingga data tersebut dapat dimanipulasi, ditampilkan, dan dicari dengan cepat. *Database Management System (DBMS)* merupakan kumpulan data yang terkait dan seperangkat program yang digunakan untuk mengakses data tersebut dengan tujuan menyediakan cara menyimpan dan mengakses data yang dikumpulkan (R. A. Putri, 2022:4).

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.

5.1. *MySQL*

Menurut (Joshua et al., 2021) sebenarnya *MySQL* merupakan salah satu konsep utama dalam database yang merupakan turunan dari *SQL* (*Structured Query Language*), sedangkan *SQL* sendiri merupakan konsep operasi *database* terutama untuk memilih dan memasukkan data dengan mudah dan tepat serta operasi data secara otomatis. Pendapat diatas didukung oleh (Kalsum Siregar et al., 2024) dalam penelitiannya yang mengungkapkan bahwa *SQL* merupakan bahasa pemrograman paling populer untuk menangani basis data yang digunakan untuk mengakses basis data relasional dan memudahkan praktisi data dalam mengolah data, dan *MySQL* merupakan salah satu sistem basis data yang mendukung *SQL*.

Dari pendapat diatas, dapat disimpulkan bahwa *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.




6. *Flowchart*

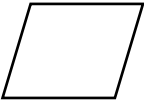
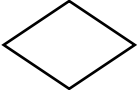




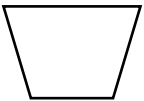


Flowchart dapat diartikan sebagai bagan alur suatu program. Dalam penjelasannya, menurut (Rasiban et al., 2024) mengungkapkan bahwa *flowchart* adalah diagram yang menampilkan langkah – langkah dan keputusan untuk melakukan suatu proses dari suatu program yang digambar

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 untuk menggambarkan alur dari suatu proses. Flowchart juga dapat diartikan sebagai diagram alir grafis yang menggambarkan langkah-langkah sistematis suatu proses atau logika program secara berurutan yang membantu analisis pemecahan masalah dalam pengembangan aplikasi atau perangkat lunak (M. P. Putri et al., 2022:19).

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*

Bagan	Nama	Fungsi
	<i>Flow</i>	Arah aliran program
	<i>Terminator</i>	Awal atau akhir program
	<i>Proces</i>	Menyatakan suatu proses

	<i>Input / Output</i>	Input / Output data
	<i>Decision</i>	Menyatakan suatu kondisi
	<i>Predefine Proses</i>	Menyatakan sub-program
	<i>Preparation</i>	Inisialisasi / pemberian nilai awal
	<i>Display</i>	Menyatakan alat output
	<i>Document</i>	Input / Output Berbentuk Dokumen
	<i>Manual Operation</i>	Proses Manual
	<i>On Page Connector</i>	Penghubung bagian dalam satu halaman
	<i>Off Page Connector</i>	Penghubung bagian berbeda halaman

Sumber : (Sari & Sari siregar, 2021)

7. *Extreme Programming (EP)*

Metode ini merupakan model pengembang perangkat lunak yang mencoba menyederhanakan berbagai tahapan dalam proses pengembangan sehingga lebih efektif dan fleksibel (Rudianto, 2023), yang mana model pengembangan ini tidak hanya berfokus pada pengkodean namun juga mencakup semua lini pengembangan perangkat lunak. Model ini juga merupakan salah satu model dari model yang dapat digunakan dalam

melakukan pembangunan maupun pengembangan suatu sistem informasi yang dikerjakan secara efisien dan pada setiap tahapannya dilakukan perubahan yang cepat serta fleksibel dibandingkan dengan model lainnya (Taufiq et al., 2022). Salah satu manfaat pendekatan *extreme programming* adalah menyediakan tahapan dengan cepat dan mengulang proses untuk berbagai bagian berdasarkan fokus yang dicapai. Proses tersebut dilakukan lagi berdasarkan kebutuhan (Fazrin et al., 2021). Metode ini memiliki beberapa tahapan pelaksanaan yaitu perencanaan (*planning*), pemodelan (*designing*), pengkodean (*coding*), dan pengujian (*testing*).

Dari pendapat yang ada, maka dapat disimpulkan bahwa *extreme programming* merupakan model pengembangan perangkat lunak dengan tujuan menyederhanakan tahapan pengembangan sehingga lebih efektif dan 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.

8. *Constraint Satisfaction Problem (CSP)*

Metode *constraint satisfaction problem* merupakan salah satu metode pemecahan masalah dengan melibatkan sejumlah *variable* yang harus memenuhi serangkaian batasan atau *constraint*. Metode ini memerlukan *constraint programming* dalam menyelesaikan permasalahan yang ada. *Constraint programming* adalah metode komputasi yang menyelesaikan masalah dengan menetapkan kondisi atau batasan yang harus dipenuhi.

Pendekatan ini memodelkan masalah sebagai kumpulan *constraint*, lalu mencari solusi yang memenuhinya. Alternatif pemrograman ini banyak digunakan dalam berbagai bidang untuk optimasi dan pemecahan masalah kompleks (W. A. Putri, 2021). Dalam pengertian lain dijelaskan bahwa metode ini adalah teknik untuk mendapatkan suatu penyelesaian atau solusi dari suatu permasalahan melalui pencarian sebuah objek atau kondisi yang memenuhi kriteria tertentu (Harfani, 2022). *Constraint Satisfaction Problem (CSP)* adalah pendekatan matematis untuk mencari solusi yang memenuhi serangkaian batasan (*constraints*). *Constraint* merupakan pembatas solusi yang valid dalam suatu masalah optimasi. Sistem berbasis aturan adalah program yang mengemas pengetahuan pakar dalam bentuk aturan-aturan domain tertentu untuk menyelesaikan masalah (Lubis, 2023).

Dari beberapa pendapat diatas maka dapat disimpulkan bahwa *Constraint Satisfaction Problem (CSP)* adalah metode pemecahan masalah yang mengandalkan variabel dan batasan (*constraints*) untuk mencari solusi valid yang didalamnya terdapat tiga komponen utama yaitu *variable*, *domain*, dan *constraint*. Pendekatan ini menggunakan *constraint programming* untuk memodelkan masalah dan menemukan solusi yang memenuhi semua syarat. *CSP* bersifat matematis, sistematis, dan banyak diterapkan dalam optimasi serta penyelesaian masalah kompleks. Selain itu, sistem berbasis aturan dalam *CSP* memanfaatkan pengetahuan pakar berupa aturan domain tertentu untuk efisiensi solusi. Dengan demikian, *CSP*

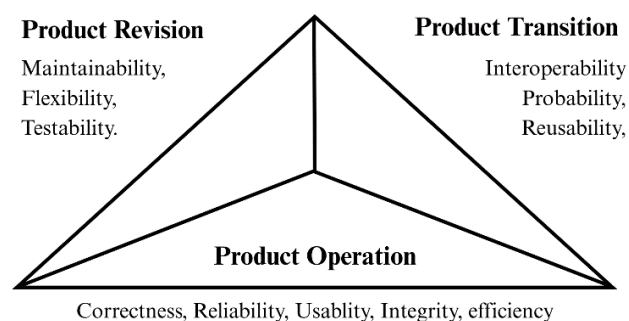
menjadi alat efektif dalam berbagai bidang untuk menangani masalah yang memerlukan kepatuhan terhadap sejumlah batasan.

9. *Black-box testing*

Black-box testing merupakan serangkaian metode pengujian yang memfokuskan pengujian di fungsionalitas perangkat lunak. *Black-box testing* memungkinkan seorang *software engineer* dan tester mendapat serangkaian kondisi input yang seluruhnya menggunakan persyaratan kondisi sebuah program tanpa perlu melihat dan menguji *source code system* (Pratama et al., 2023). Dengan kata lain, *black-box testing* memposisikan penguji hanya memperhatikan *input* yang di masukkan dan *output* yang didapatkan sebagai timbal balik dari sistem yang berjalan tanpa tau proses didalamnya.

10. *Matriks McCall*

Pengujian menggunakan matriks *McCall* merupakan sebuah metode pengujian dengan fokus untuk menilai kualitas suatu sistem dengan tiga faktor berbeda yaitu *product operation*, *product revision*, dan *product transition* (Fahmi Sulaiman et al., 2022).



Gambar 2 .1 Siklus McCall

Sumber : (Malahayati & Nadeak, 2023)

Tahapan yang dilakukan dalam pengujian *McCall* ini dimulai dari rentang kategori kualitas yang dijadikan dasar pengujian, selanjutnya dilanjutkan dengan pembobotan dari Setiap kategori untuk mengetahui nilai kepentingan dari sebuah aspek variabel dan indikator (Malahayati & Nadeak, 2023).

11. *Unified Modeling Language (UML)*

Pada sebuah sistem, *UML* memiliki fungsi sebagai alat bantu untuk melakukan pendeskripsian serta desain sistem yang dibangun menggunakan konsep pemrograman berorientasi obyek (Nistrina & Sahidah, 2022). Dalam penelitian lain yang dilakukan oleh (Suharni et al., 2023) juga menjelaskan bahwa *UML* merupakan pemodelan berorientasi objek yang menganalogikan sebuah sistem seperti kehidupan nyata yang didominasi oleh object yang digambarkan atau dinotasikan dalam simbol yang spesifik.

Menurut (Narulita et al., 2024) turut mengungkapkan bahwa *UML* memberikan sebuah standar pembuatan *blue print* dari sebuah sistem yang dapat terdiri dari konsep proses bisnis, *class* yang dapat dibuat di pemrograman tertentu, rancangan basis data, serta komponen yang diperlukan untuk pengembangan sistem. Dalam *UML* terdapat beberapa *class* yang menjadi kerangka penyusun didalamnya yang meliputi *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.




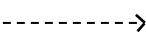
Dari pendapat diatas dapat disimpulkan bahwa *UML* merupakan suatu alat pemodelan yang digunakan untuk membantu pengembang aplikasi dalam merancang, memvisualisasi, serta mendokumentasikan tahapan


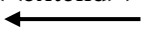
pengembangan sistem perangkat lunak sebelum dilakukan proses implementasi.

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 (Sutrisno & Karnadi, 2020). *Use case diagram* juga berfungsi sebagai dasar dalam merancang suatu perilaku sistem dan juga menjadi landasan penting dalam tahap analisis kebutuhan sistem yang merepresentasikan sistem dari pengguna dan menjadi panduan untuk masuk ke dalam tahapan implementasi sistem.

Tabel 2 .2 Simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
1		Aktor	Mewakili peran orang
2		<i>Use Case</i>	Abstraksi antara sistem dan aktor
3		<i>Association</i>	Penghubung <i>use case</i> dengan aktor
4		Generalisasi	Spesialisasi aktor untuk berhubungan dengan <i>use case</i>


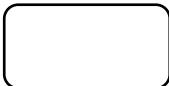
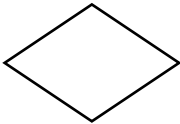


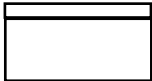
<p>5 <<include>></p>  <p style="text-align: right;"><i>Include</i></p>	<p>Simbol yang menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.</p>
<p>6 <<extend>></p>  <p style="text-align: right;"><i>Extend</i></p>	<p>Simbol yang menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.</p>

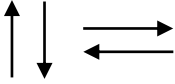
Sumber : (Rasiban et al., 2024)

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, 2020). Diagram ini juga memberikan gambaran tentang bagaimana setiap aliran berawal, kondisi yang mungkin terjadi dan bagaimana aktivitas tersebut berakhir. Sehingga *activity diagram* menjadi alat bantu dalam pemetaan alur kerja atau proses sistem secara menyeluruh dan memungkinkan pengembang untuk memahami interaksi sistem sebelum masuk ke dalam tahapan implementasi sistem.

Tabel 2 .3 Simbol *Activity Diagram*

No	Simbol	Nama	Keterangan
1		Status Awal	menggambarkan suatu <i>diagram activity</i> memiliki sebuah status awal.
2		Aktivitas	Simbol aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3		<i>Decision</i>	Simbol percabangan yang dimana ada pilihan aktivitas yang lebih dari satu.
4		<i>Join</i>	Simbol penggabungan yang dimana lebih dari satu aktivitas lalu digabungkan menjadi satu.
5		Status Akhir	menggambarkan suatu <i>diagram activity</i> memiliki status akhir yang dilakukan sistem
6		<i>Swimlane</i>	memisahkan organisasi bisnis yang bertanggung

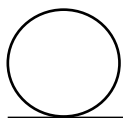
		jawab terhadap aktivitas yang terjadi.
7		<i>Line connector</i> Untuk menghubungkan satu simbol dengan simbol yang lain.

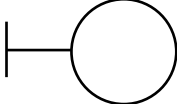
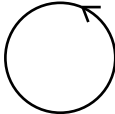
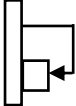


Sumber : (Rasiban et al., 2024)

c. *Sequence diagram*

Sequence diagram merupakan jenis diagram yang menggambarkan interaksi antara objek satu dengan lainnya dalam sebuah sistem secara kronologis dan juga menunjukkan bagaimana objek dalam suatu sistem berkomunikasi satu sama lain dan secara berurutan dalam eksekusi sebuah kondisi. Menurut (Sutrisno & Karnadi, 2020), diagram ini merupakan sebuah diagram yang menampilkan interaksi yang terjadi dari respon antar objek yang saling berkomunikasi. Komunikasi ini terjadi melalui pertukaran pesan dengan skenario tertentu termasuk urutan waktu terjadinya interaksi. Manfaat yang bisa didapat dengan adanya diagram ini yaitu pemahaman alur eksekusi sistem dapat dipahami secara detail, analisis hubungan antar komponen, serta dapat memastikan komunikasi yang dirancang sudah tepat.

Tabel 2 .4 Simbol Sequence Diagram

No	Simbol	Nama	Keterangan
1		<i>Entity Class</i>	Simbol yang menggambarkan sistem

			sebagai landasan dalam menyusun basis data.
2		<i>Boundary Class</i>	Simbol yang menangani komunikasi antar lingkungan sistem.
3		<i>Control Class</i>	Simbol yang bertanggung jawab terhadap class objek yang berisi logika.
4		<i>Recursive</i>	Simbol yang melambangkan pesan untuk dirinya.
5		<i>Activation</i>	Simbol yang mewakili proses durasi aktivasi sebuah sistem.
6		<i>Life Line</i>	Simbol yang menggambarkan aktivitas dari objek.

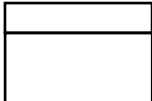
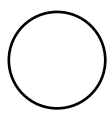



Sumber : (Rasiban et al., 2024)

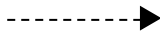
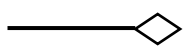
d. *Class diagram*

Class diagram adalah penggambaran struktur suatu sistem yang mendefinisikan kelas-kelas yang akan digunakan untuk merancang suatu sistem. Bisa juga di artikan sebagai diagram yang digunakan untuk menggambarkan struktur statis dari suatu sistem aplikasi yang berorientasi objek. *Class diagram* mendefinisikan kerangka dasar yang menunjukkan interaksi antar kelas.

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 dalam struktur sistem
2		<i>Interface</i>	menggambarkan konsep interface pada pemrograman berorientasi objek
3		<i>Association</i>	Relasi antar kelas yang berarti umum, asosiasi biasanya disertai dengan <i>Multiplicity</i> .
4		<i>Direct Association</i>	Relasi antar kelas yang berarti kelas yang atau digunakan oleh kelas yang lain, asosiasi langsung biasanya juga disertai dengan <i>multiplicity</i> .
5		<i>Generalisasi</i>	Relasi antar kelas dengan arti generalisasi sampai spesialisasi (umum ke khusus).

6		<i>Dependency</i>	Relasi antar kelas dengan arti kebergantungan antar kelas'
7		<i>Aggregation</i>	Relasi antar kelas dengan arti keseluruhan sampai bagian (whole - part)

Sumber : (Rasiban et al., 2024)

B. Kajian Empiris

Penelitian ini dalam penyusunannya menggunakan beberapa referensi jurnal yang berhubungan dengan topik yang diangkat yaitu sistem informasi penjadwalan mata kuliah di lingkungan Program Studi Teknik Informatika Universitas PGRI Madiun yang digunakan sebagai acuan dan disusun secara sistematis dengan harapan dapat membantu dalam membangun sistem ini. Studi literatur dari penelitian terdahulu digunakan sebagai media untuk memperkuat dan mematangkan rancangan pembangunan sistem dengan melakukan perbandingan dan evaluasi dari sistem yang sudah dibuat pada penelitian terdahulu.

Penelitian sebelumnya yang dilakukan oleh (Wulandari et al., 2022) mengenai penjadwalan mata kuliah pada *Program Studi Matematika FMIPA Untan* yang juga menggunakan metode *constraint satisfaction problem* namun dengan menggunakan modifikasi algoritma *backtracking*. Algoritma ini dimodifikasi untuk menghindari *trashing* yaitu penelusuran yang tidak mengarah pada solusi. Hasilnya kombinasi jadwal yang cukup optimal berhasil diperoleh dengan gabungan antara mata kuliah, dosen, waktu, dan ruangan.

Penelitian lain dengan judul *Implementation of the Constraint Satisfaction Problems Method in Genetic Algorithms for Course Scheduling Systems* yang dilakukan oleh (Martinus et al., 2023) yang juga mengimplementasikan pendekatan *constraint satisfaction problem* pada penelitiannya yang membahas tentang sistem penjadwalan menghasilkan temuan yang efisien untuk penjadwalan mata kuliah dengan menggunakan metode *constraint satisfaction problem* dengan fungsi metode ini adalah untuk memastikan semua batasan (ruang, dosen, dan waktu) yang ada terpenuhi sebelum melakukan evaluasi solusi pembagian yang tepat.

Penelitian serupa juga dilakukan oleh (Putra et al., 2024) yang juga membahas penggunaan metode *constraint satisfaction problem* untuk optimalisasi penjadwalan. Dalam penelitian ini, metode *Constraint Satisfaction* digunakan untuk memastikan bahwa semua kendala penjadwalan, seperti ketersediaan dosen, ruang kelas, dan waktu, terpenuhi. Hasilnya melalui simulasi komputer didapatkan bahwa tidak ada konflik jadwal serta proses penjadwalan otomatis berjalan lebih cepat dari pada proses manual.

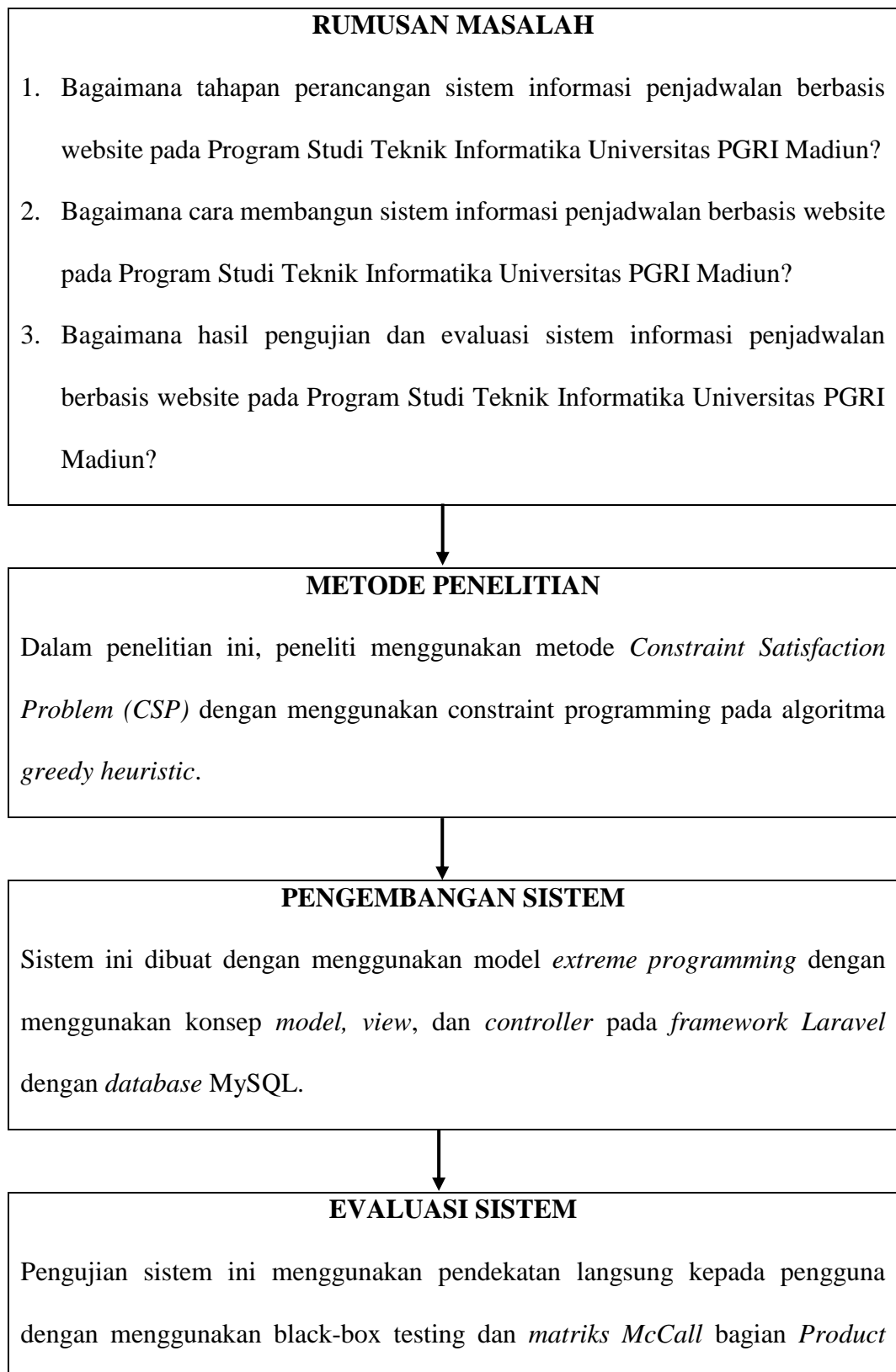
Dalam penelitian lain yang membahas mengenai sistem otomatisasi penjadwalan yang dilakukan oleh (Larasati, 2024) turut menjelaskan bahwa otomatisasi penjadwalan dengan menggunakan *website* dapat memudahkan penyampaian informasi terkait pembagian jadwal serta efisiensi waktu kerja.

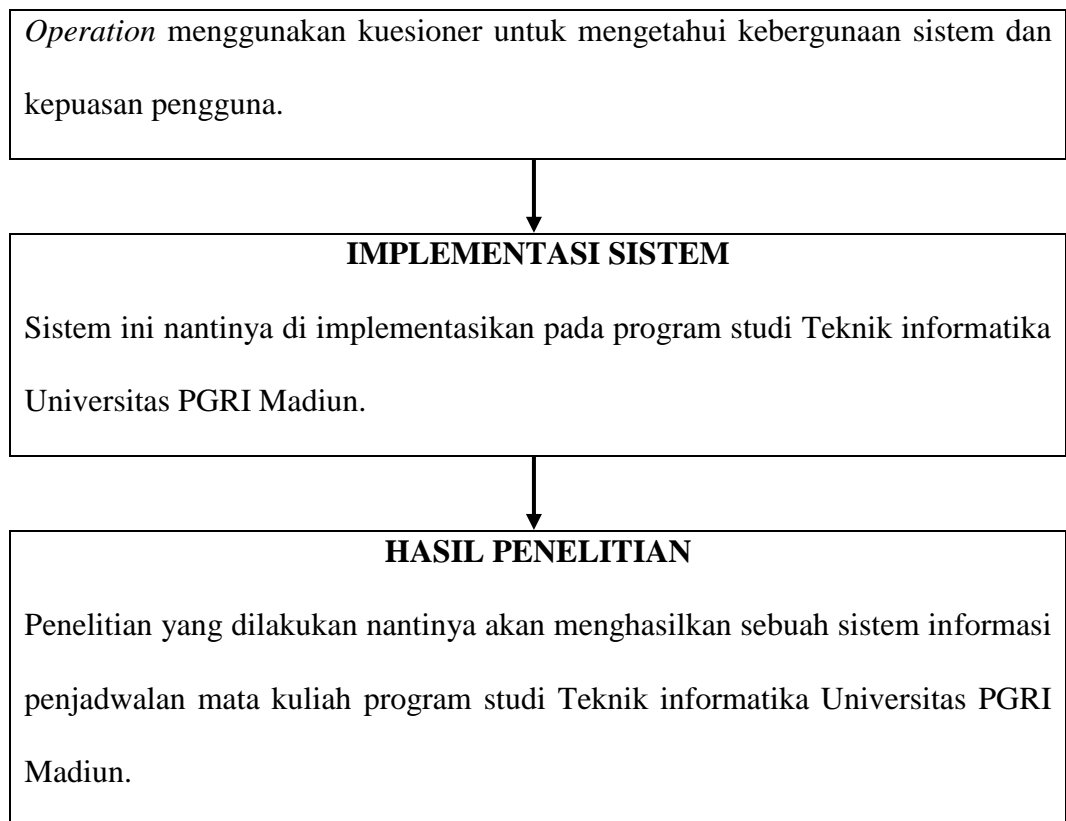
Penelitian yang dilakukan oleh (Taufiq et al., 2022) turut menjelaskan bahwa penerapan teknik *extreme programming* dalam penelitiannya meningkatkan efisiensi dan kemampuan beradaptasi dari sistem informasi

penjadwalan yang dikembangkan.. Dalam penelitian ini juga diungkapkan bahwa sistem yang dikembangkan ini berhasil memuat sebuah sistem yang bebas dari kesalahan dalam implementasi pembagian mata kuliah dalam sistem penjadwalannya.

Dari beberapa sumber literatur diatas, maka penggunaan metode yang tepat dalam proses pembangunan sistem sangat diperlukan. Pada proses pembangunan ini, peneliti memilih menggunakan metode extreme programming dengan menggunakan arsitektur *Model, View, dan Controller* dengan *framework laravel* serta penggunaan algoritma metode *Constraint Satisfaction Problem (CSP)* yang dinilai cocok untuk penentuan jadwal dengan batasan – batasan yang ditetapkan.

C. Kerangka Berfikir





Gambar 2 .2 Kerangka Berfikir