

## BAB II

### KAJIAN PUSTAKA

#### A. Kajian Teoritis

##### 1. Sistem Informasi Manajemen

Menurut Fikri, (2025) Sistem merupakan sebuah kumpulan yang saling bergantung yang bekerja sama untuk mencapai tujuan tertentu.

*Sistem* informasi adalah kombinasi aktivitas manusia dan teknologi informasi yang membantu operasi dan manajemen. Mereka adalah alat penting untuk mencapai tujuan organisasi secara efektif dan sistematis (Andika, 2024).

*Sistem* informasi manajemen adalah metode yang dirancang untuk memberikan informasi cepat untuk manajemen tentang kegiatan operasi di dalam dan di luar organisasi. Tujuan dari sistem informasi manajemen adalah untuk mempermudah proses manajemen, meningkatkan perencanaan dan pengawasan, dan mendukung proses pengambilan keputusan (Gede & Bratha, 2022)

Berdasarkan referensi diatas, Dengan demikian, sistem manajemen informasi dapat didefinisikan sebagai sebagai alat logis untuk membantu proses operasional dan pengambilan keputusan dengan mengintegrasikan proses input dan output untuk memenuhi kebutuhan manajemen organisasi.

## 2. *Website*

*Website* adalah kumpulan halaman yang mengandung informasi dalam bentuk teks, gambar, dan video. Saat ini, setiap orang dapat mengaksesnya serta memudahkan pekerjaan manusia, sistem informasi terus dikembangkan (Rahmi, 2023).

Menurut Arief Ichwani, (2021) *Website* adalah media atau sistem yang menyimpan berbagai informasi yang dapat dilihat oleh banyak pengguna yang menggunakan teknologi internet, *website* memiliki peran sebagai media yang menghubungkan penyedia layanan dengan penggunanya.

Didasarkan pada kedua definisi di atas, *website* adalah media berbasis internet yang menyediakan informasi dalam berbagai format untuk mempermudah pekerjaan manusia. Karena itu, *website* terus dikembangkan untuk memenuhi kebutuhan sistem informasi.

## 3. *SMTP (Simple Mail Transfer Protocol)*

*Simple Mail Transfer Protocol (SMTP)* adalah standar untuk transmisi *email*. Akan tetapi, *SMTP* tidak memiliki fitur keamanan dalam desain awalnya. Misalnya, ia mengirim email dalam bentuk teks biasa (tanpa kerahasiaan). *Server SMTP* dapat mengirim perintah *STARTTLS* dalam bentuk teks biasa selama penyiapan koneksi *SMTP* untuk menunjukkan dukungan *TLS*-nya kepada klien. *Enkripsi SMTP* *SMTP* tidak mengenkripsi pesannya sendiri. Dengan demikian, ekstensi

*STARTTLS* pertama kali diperkenalkan untuk enkripsi *email* (Lee, 2022).

*Simple Mail Transfer Protocol (SMTP)*, yang awalnya dirancang pada tahun 1982. Protokol ini mendefinisikan bagaimana *email* dikirim antara server *email* yang berbeda dan awalnya tidak menyertakan dukungan untuk enkripsi. Versi asli *SMTP* hanya menentukan bagaimana server *SMTP* dapat bernegosiasi satu sama lain tentang pengirim, penerima, tanggal, konten, dan informasi lainnya serta melaporkan tentang kemungkinan kesalahan pengiriman. Protokol ini dijalankan melalui *TCP* dan merupakan protokol teks biasa tanpa dukungan autentikasi atau enkripsi, yang membuatnya rentan terhadap penyadapan dan serangan *man-in-the-middle* jika pelaku ancaman dapat memantau koneksi atau menempatkan dirinya di tengah koneksi (Nopanen, 2022).

*Simple Mail Transfer Protocol (SMTP)* adalah pertama kali dikembangkan pada tahun 1982. Pada desain awalnya, *SMTP* tidak dilengkapi dengan fitur keamanan seperti enkripsi atau autentikasi, sehingga pesan dikirim dalam bentuk teks biasa dan rentan terhadap penyadapan serta serangan *man-in-the-middle*. Untuk mengatasi kelemahan ini, ekstensi *STARTTLS* kemudian diperkenalkan guna memungkinkan enkripsi dalam proses transmisi email. Meskipun *SMTP* tetap menjadi protokol utama dalam pengiriman email,

pengembangannya terus dilakukan untuk meningkatkan keamanan komunikasi melalui internet.

#### **4. Framework**

Menurut Ikhsan, (2023) *Framework*, juga dikenal sebagai "*framework*" adalah kumpulan fungsi, prosedur, dan kelas yang dirancang untuk mencapai tujuan tertentu. Mereka dapat digunakan untuk mempercepat dan mempermudah pekerjaan programmer tanpa harus membuat fungsi atau kelas sendiri.

. *Framework* dirancang untuk menjadi *set tools* frontend standar untuk pengembang aplikasi *web* di berbagai perusahaan. Kerangka kerja adalah kumpulan komponen program yang memiliki tujuan khusus untuk melaksanakan perintah tertentu, sehingga penulisan kode program lebih efisien serta efektif. (Apriliando, 2021).

*Framework* merupakan kerangka kerja yang terdiri dari kumpulan fungsi, prosedur, kelas yang dirancang untuk mempermudah dan mempercepat pekerjaan *programmer* tanpa perlu membuat semuanya dari awal. *Framework* berfungsi sebagai standar alat bantu dalam pengembangan aplikasi, khususnya di sisi *frontend*, dan diciptakan untuk meningkatkan efisiensi serta efektivitas dalam penulisan kode program. Salah satu contoh *framework open-source* adalah buatan Mark Otto dan Jacob Thornton dari *Twitter*, yang digunakan secara luas oleh pengembang *web* di berbagai perusahaan.

## 5. *Laravel*

Menurut Subecz, (2021) *Laravel* adalah *framework PHP* yang paling dapat digunakan untuk *programmer* pemula dan tingkat lanjut. *Laravel* dapat mengurangi waktu pengembangan aplikasi *web* dan memasarkannya dengan metode *PHP* berorientasi objek *modern*. Sintaksisnya yang ekspresif dan fungsi-fungsinya yang *modern* menarik dan tangguh. *Laravel* menyediakan fitur-fitur canggih seperti lapisan abstrak basis data ekspresif dan injeksi ketergantungan dan sangat *scalable*.

*Laravel* adalah *framework* lain yang membantu pengembang memaksimalkan penggunaan *PHP* selama pengembangan *website*. Selain itu, banyak programmer di seluruh dunia menggunakan *Laravel*, sebuah kerangka kerja pemrograman yang berbasis *open source*, yang memiliki fitur seperti *modularitas*, *routing*, dan *template engine*. *Laravel* telah menjadi populer dalam beberapa tahun terakhir karena kemudahan penggunaan dan dokumentasi yang lengkap. (Aipina & Witriyono, 2022).

Karena dokumentasi yang lengkap dan kemudahan penggunaannya, *Laravel* adalah *framework PHP* berbasis *Open Source*, baik pemula maupun tingkat lanjut. Ini dapat ditarik kesimpulan dari definisi kedua referensi di atas. Dengan sintaksis yang ekspresif dan fitur modern seperti *template engine*, *routing*, *modularitas*, abstraksi basis data, dan injeksi ketergantungan, *framework* ini dimaksudkan

untuk mempercepat proses pengembangan aplikasi web. Pengembang dapat membuat aplikasi yang tangguh dan dapat diskalakan dengan *Laravel*, yang memanfaatkan pendekatan berorientasi objek yang efektif untuk meningkatkan penggunaan *PHP*.

## 6. *PHP*

*PHP* adalah bahasa pemrograman yang dibuat Rasmus Ler-dorf. Biasanya digunakan untuk scripting di sisi *server*. Arsitektur *LAMP* adalah metode yang sangat populer di industri web untuk mengimplementasikan sistem informasi yang murah, kuat, skalabel, dan aman. Banyak *data-base* relasional memanfaatkan *PHP*. Salah satunya adalah database *MySQL* (Pradana & Hardi, 2021).

*PHP* banyak digunakan untuk membuat aplikasi web karena kodenya dieksekusi di sisi server sehingga kode aslinya tidak terlihat di sisi klien (*browser*), memungkinkan pengembang untuk menyisipkan kode ke dalam *HTML* menggunakan bahasa yang sama seperti perl dan shell *UNIX* (Hakim & Zailani, 2022).

Dari kedua referensi di atas maka dapat ditarik kesimpulan bahwa *PHP* (*Hypertext Preprocessor*) adalah bahasa pemrograman yang dirancang untuk membuat halaman *web* dinamis dan dijalankan di sisi *server*, sehingga kode aslinya tidak terlihat oleh pengguna melalui *browser*. Diciptakan oleh Rasmus Lerdorf dengan dasar bahasa *C*, *PHP* dikenal sebagai bahasa yang mudah dipelajari dan digunakan, bahkan dengan editor sederhana seperti *notepad*. *PHP*

mendukung penyisipan kode ke dalam *HTML* dan memiliki sintaks yang tidak sensitif terhadap huruf besar atau kecil. Umumnya digunakan dalam arsitektur *LAMP*, *PHP* karena tangguh, skalabel, aman, serta mendukung berbagai jenis database relasional, seperti *MySQL*.

## 7. *MySQL*

*MySQL* adalah sistem manajemen basis data relasional (RDBMS) adalah yang bersifat *open-source* dan sumber terbuka. Fungsi *MySQL* adalah untuk mengatur dan menyimpan data dalam bentuk tabel yang berhubungan satu sama lain, sehingga pengguna dapat dengan mudah menyimpan, mengakses, dan mengelola data. *MySQL* juga menyediakan bahasa kueri yang kuat, yaitu *SQL*, yang digunakan untuk mengambil dan mengubah data di dalam basis data. Dengan banyak fitur canggihnya, *MySQL* dapat digunakan dalam berbagai jenis aplikasi dan lingkungan, mulai dari aplikasi web hingga sistem bisnis yang kompak.

(Muntasir, 2023).

Menurut Yudianto, (2023) *MySQL* adalah salah satu opsi *RDBMS* yang dapat digunakan untuk menyimpan, memanipulasi keterkaitan antara tabel. Selain itu, *MySQL* memiliki fungsi untuk data warehousing (gudang data), yaitu pengumpulan data dari berbagai sumber, e-commerce, dan aplikasi logging.

Dari kedua referensi didapatkan kesimpulan *MySQL* adalah sistem manajemen basis data relasional (*RDBMS*) *open-source* yang secara luas digunakan untuk menyimpan, mengelola, serta memanipulasi data dalam bentuk tabel yang saling terkait. Dengan menggunakan bahasa kueri *SQL* (*Structured Query Language*), *MySQL* memungkinkan pengguna untuk mengakses dan mengelola data secara efisien. *MySQL* tidak hanya cocok untuk aplikasi *web*, menjadikannya pilihan fleksibel dan andal dalam pengelolaan basis data.

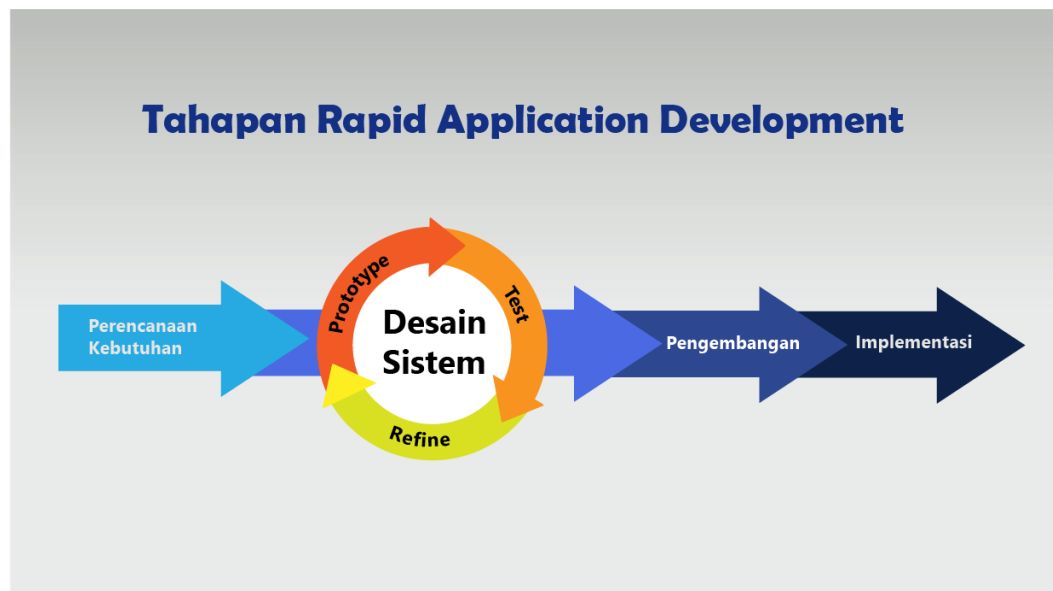
#### **8. *RAD* (*Rapid Application Development*)**

*Rapid Application Development* (*RAD*) adalah adaptasi "kecepatan tinggi" dari model sekuensial linier yang mencapai perkembangan cepat dengan menggunakan pendekatan konstruksi berbasis komponen. *RAD* bertujuan untuk memenuhi tuntutan bisnis yang berubah secara cepat dengan mengurangi waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara perancangan dan penerapan sistem informasi (Titania Pricilia & Zulfachmi, 2021)

Menurut Parhusip, (2022) *Rapid Application Development* (*RAD*) adalah sebuah strategi siklus hidup yang menggabungkan berbagai teknik terstruktur bersama dengan teknik prototyping dan pengembangan aplikasi kolaboratif untuk tujuan mempercepat pengembangan sistem atau aplikasi.

Dari penjelasan kedua referensi di atas maka *RAD* adalah model pengembangan perangkat lunak yang menekankan kecepatan dan

efisiensi dengan siklus pengembangan yang singkat, biasanya antara 60 hingga 90 hari. Model ini merupakan adaptasi dari pendekatan *sekuensial linier* dengan fokus pada konstruksi berbasis komponen untuk mempercepat proses perancangan hingga penerapan sistem. *RAD* dirancang untuk merespons kebutuhan bisnis yang berubah dengan cepat dan menggabungkan berbagai teknik guna menghasilkan sistem berkualitas tinggi dalam waktu yang lebih singkat dibandingkan metode tradisional. Untuk Ilustrasinya bisa dilihat pada Gambar dibawah ini.



Gambar 2.1 Tahapan Metode *RAD*

Pada gambar 2.1 Tahapan Metode *RAD* menjelaskan Proses *Rapid Application Development (RAD)* diawali dengan tahap *requirements Planning/ Perencanaan Kebutuhan*, yang melibatkan pengembang, klien, dan pengguna untuk menetapkan tujuan sistem, strategi penyelesaian masalah, serta batasan waktu dan anggaran Tahap ini penting untuk menyamakan persepsi sejak awal agar risiko kesalahan

dapat diminimalkan. Selanjutnya, Tahap desain sistem pengembang membuat *prototype* secara cepat sesuai kebutuhan pengguna dan menyerahkannya kepada klien untuk memperoleh umpan balik. *Prototype* yang dihasilkan masih bersifat sementara dan dapat mengalami perubahan. Pada tahap *Pengembangan/Construction* dilakukan evaluasi terhadap umpan balik pengguna yang mencakup aspek fitur, fungsi, tampilan, dan antarmuka sistem. Proses pembuatan dan evaluasi *prototype* ini dilakukan secara berulang hingga sesuai dengan harapan klien. Tahap terakhir adalah *implementasi/cutover* umpan balik dan pengujian sistem untuk menghasilkan produk akhir yang stabil, mudah digunakan, dan siap diterapkan (Hapsari & Azis, 2024).

Menurut (Utami, 2020) *Metode RAD (Rapid Application Development)* memberikan beberapa keuntungan dalam pemanfaatannya dalam membangun sebuah sistem yang dapat dilihat pada Tabel dibawah ini :

Tabel 2.1 Keuntungan dan Kerugian Metode *RAD*

| Keuntungan  | Kerugian   |
|---|--|
| Pengembangan Cepat <i>RAD</i> memungkinkan aplikasi dikembangkan dengan cepat, mengurangi waktu pemasaran produk. | Untuk proyek yang besar namun terukur, <i>RAD</i> membutuhkan sumber daya manusia yang cukup untuk membuat jumlah tim <i>RAD</i> yang tepat. |

|  |  |
|--|--|
| <p>Keterlibatan Pengguna: <i>RAD</i> membutuhkan pengembang Pengguna terlibat dalam proses dan pelanggan yang pengembangan, sehingga berkomitmen untuk kegiatan aplikasi lebih sesuai dengan yang diperlukan untuk kebutuhan mereka. menyelesaikan sistem dalam Iterasi Cepat: <i>RAD</i> mendukung waktu yang lebih singkat perubahan cepat berdasarkan kerangka kerja. Jika komitmen umpan balik, memungkinkan kurang dari salah satu konstituen, perbaikan yang lebih baik. proyek <i>RAD</i> akan gagal.</p> |  |
| <p>Fleksibilitas: Kemampuan untuk merespons perubahan kebutuhan dan perubahan pasar.</p> <p>Peningkatan Produktivitas: Memungkinkan pengembang untuk fokus pada fitur utama aplikasi.</p>  | <p>Tidak semua jenis aplikasi cocok untuk <i>RAD</i>. Jika sebuah sistem tidak dapat dimodulasi dengan benar, membangun komponen yang diperlukan untuk <i>RAD</i> akan bermasalah. Jika kinerja tinggi adalah masalah dan kinerja harus dicapai melalui penyetelan antarmuka ke komponen sistem,</p> |

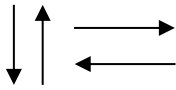

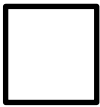




## 9. Flowchart

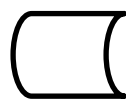
Menurut Fatwa, (2025) *Flowchart* merupakan gambaran urutan sebuah proses dalam bentuk grafik yang sistematis. Pada umumnya *flowchart* dibuat untuk memudahkan dalam penyelesaian masalah pada proses evaluasi. *Flowchart* memiliki fungsi penting dalam

menggambarkan alur sebuah proses, terutama dalam menjelaskan urutan langkah-langkah dari satu proses ke proses lainnya.

Menurut Maulana, (2025) *Flowchart* bisa disebut sebagai alat visualisasi yang menggambarkan alur proses sebuah sistem. *Flowchart* mampu mengidentifikasi titik lemah dalam proses dan merancang alur yang lebih efisien. *Flowchart* menggunakan simbol-simbol standar untuk menggambarkan alur seperti pada Tabel di bawah ini:

Tabel 2. 2 Simbol *Flowchart*

| Simbol  | Nama                         | Fungsi  |
|---|------------------------------|---|
|   | <i>Flow Direction Symbol</i> | Yaitu Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga <i>Connecting Line</i> |
|  | <i>Terminator</i>            | Yaitu Simbol untuk Permulaan ( <i>Start</i> ) atau akhir ( <i>Stop</i> ) dari suatu Kegiatan  |
|  | <i>Process</i>               | Simbol yang menyatakan Suatu Proses dilakukan Komputer  |
|  | <i>Document</i>              | Simbol yang menyatakan bahwa <i>input</i> berasal dari document dalam bentuk fisik dan output bisa dicetak                                      |
|  | <i>Input/output Data</i>     | Proses <i>input/output</i> data parameter dan Informasi   |
|  | <i>On Page connector</i>     | Penghubung Bagian <i>Flowchart</i> yang berada pada satu halaman  |
|  | <i>Off Page Connector</i>    | Penghubung Bagian <i>Flowchart</i> yang berada pada halaman Berbeda   |




---

*Storage Data*      Sebagai Penyimpan Data

---

Berdasarkan kedua definisi di atas, dapat diketahui bahwa *flowchart* atau yang biasa disebut diagram alir, merupakan sebuah diagram yang menggambarkan proses sebuah sistem secara rinci. *Flowchart* berfungsi sebagai alat bantu dalam menggambarkan alur sebuah sistem yang mampu mengidentifikasi titik lemah dari sistem yang bersangkutan.

#### **10. UML (*Unified Modelling Language*)**

Menurut Hidayati, (2023) *UML* adalah standar untuk pembangunan software berbasis objek dan memiliki banyak diagram, seperti *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*. Kelebihan *UML* adalah memungkinkan pengembang menggunakan bahasa pemodelan visual atau gambar untuk berbagai jenis proses umum pemrograman dan rekayasa.

Menurut (Dilova, 2023) *UML* adalah sebuah bahasa model yang berdasarkan grafik dan gambar yang digunakan untuk memvisualisasikan, menspesifikasikan, membangun, dan mendokumentasikan sistem pengembangan *software* yang berbasis *OO* (*Object Oriented*).

Sesuai dengan kedua definisi di atas, *UML* adalah bahasa pemodelan standar yang digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan sistem informasi dalam pengembangan perangkat lunak berbasis objek. *UML*, yang

berasal dari kombinasi metode *Booch*, *OMT*, dan *OOSE*, adalah alat penting dalam proses analisis dan desain berorientasi objek karena mampu mempermudah komunikasi antar tim pengembang dan meningkatkan pemahaman struktur dan alur sistem.

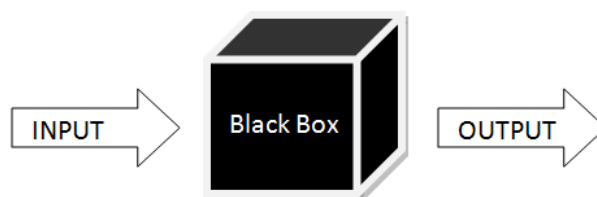
### **11. BlackBox**

Metode *blackbox* menguji sistem aplikasi untuk menunjukkan masalah seperti kegagalan fungsi dan menu aplikasi yang hilang. Dengan kata lain, metode *blackbox* menguji kinerja sistem aplikasi. (Uminingsih, 2022).

*Partitioning ekuivalensi (partisi ekuivalensi)*, analisis nilai batas (analisis nilai batas), ketahanan (pengujian acak), dan pengujian kebutuhan adalah beberapa teknik pengujian *blackbox*. Metode *blackbox* berfokus pada kebutuhan fungsional aplikasi untuk menemukan kesalahan dalam beberapa kategori, seperti kesalahan fungsi yang tidak benar, kesalahan antarmuka, kesalahan struktur data, kesalahan performasi, dan kasus lainnya. (Admojo, 2022).

Menurut kedua referensi bahwa pengujian Blackbox adalah metode pengujian perangkat lunak yang tidak melihat struktur kode dalam sistem aplikasi. Kesalahan seperti kesalahan inisialisasi dan performa, menu yang hilang, kesalahan antarmuka, kesalahan struktur data, dan fungsi yang tidak berjalan dengan benar adalah tujuan utamanya. Metode ini menggunakan berbagai pendekatan untuk memastikan bahwa aplikasi memenuhi persyaratan dan spesifikasi. Pembagian

equivalence, analisis nilai batas, pengujian ketahanan, dan pengujian persyaratan adalah beberapa teknik ini. seperti yang ditunjukkan pada gambar di bawah ini.



Gambar 2.2 Pengujian Blacbox

## B. Kajian Empiris

Berdasarkan penelitian yang dilakukan oleh Andriall & Nasir, (2023) untuk mengelola segala Sistem Informasi Manajemen Kejaksaan Republik Indonesia (SIMKARI) diciptakan untuk memungkinkan seluruh unit kerja di Kejaksaan Agung, Kejaksaan Tinggi, dan Kejaksaan Negeri seluruh Indonesia bekerja secara komputerisasi dan terintegrasi yang dibangun berbasis *web* tanpa membutuhkan instalasi pada masing-masing pengguna. Penelitian tersebut meneliti tentang pengujian SIMKARI menggunakan *System Usability Scale* dan mendapatkan hasil pengujian rata-rata/mean untuk kriteria *learnability* sebesar 4,08, kriteria *efficiency* sebesar 4,11, kriteria *memorability* sebesar 4,00, kriteria eror 4,25, dan untuk kriteria *satisfaction* sebesar 4,20. Dengan hasil ini menunjukkan bahwa pengujian menggunakan *usability testing* masuk dalam kategori baik.

Selanjutnya pada penelitian yang dilakukan oleh Taslia, (2023) yaitu tentang perancangan sistem informasi pengarsipan surat pada kantor Kejaksaan Kabupaten Mamuju Utara. Pada penelitian tersebut membahas mengenai Kejaksaan Negeri Mamuju Utara pada pelaksanaan tugas dan aktivitasnya dalam kegiatan pengarsipan yaitu mengelola surat masuk dan surat keluar masih bersifat manual yaitu dengan cara mencatat setiap surat di buku *register* dan pengarsipannya disimpan di lemari arsip yang telah disediakan sehingga proses kerjanya pun membutuhkan waktu yang lebih lama. Dari aktivitas pekerjaan yang dilakukan dalam pengelolaan surat masuk dan keluar maka penulis mencoba memberikan solusi dengan menggunakan teknologi informasi yaitu menggunakan sistem informasi pengarsipan surat sehingga dalam kegiatan kerja pegawai akan lebih cepat dan efisien. Sistem informasi pengarsipan berbasis *web* dikembangkan dengan menggunakan *framework laravel* dapat mengatasi masalah pengarsipan dokumen secara manual di Kejaksaan Mamuju Utara karena telah memiliki fitur-fitur yang sesuai dengan kebutuhan pengguna. Hasil dari penelitian ini adalah Sistem informasi pengarsipan berbasis *web* dikembangkan dengan menggunakan *framework laravel* dapat mengatasi masalah pengarsipan dokumen secara manual di Kejaksaan Mamuju Utara karena telah memiliki fitur-fitur yang sesuai dengan kebutuhan pengguna, yaitu mengelola surat masuk, mengelola surat keluar, permintaan nomor surat, disposisi online dan penyusunan laporan.

Penelitian berikutnya yang dilakukan oleh Rohman & Yulianingsih, (2024) yang mengembangkan sistem penjadwalan sidang berbasis *Android* yang menggunakan aplikasi *online Firebase dan Android Studio*. Sistem ini memiliki kemampuan untuk memberikan notifikasi mengenai jadwal sidang di Kantor DPRD Provinsi Sumatera Selatan. Anggota dewan lebih mudah menerima notifikasi dari sistem ini. karena ini memungkinkan mereka untuk menerima informasi tentang jadwal sidang secara langsung melalui notifikasi yang ditampilkan pada *smartphone* masing-masing anggota DPRD. Aplikasi email ini dibuat untuk memberikan informasi yang akurat dan tepat waktu.

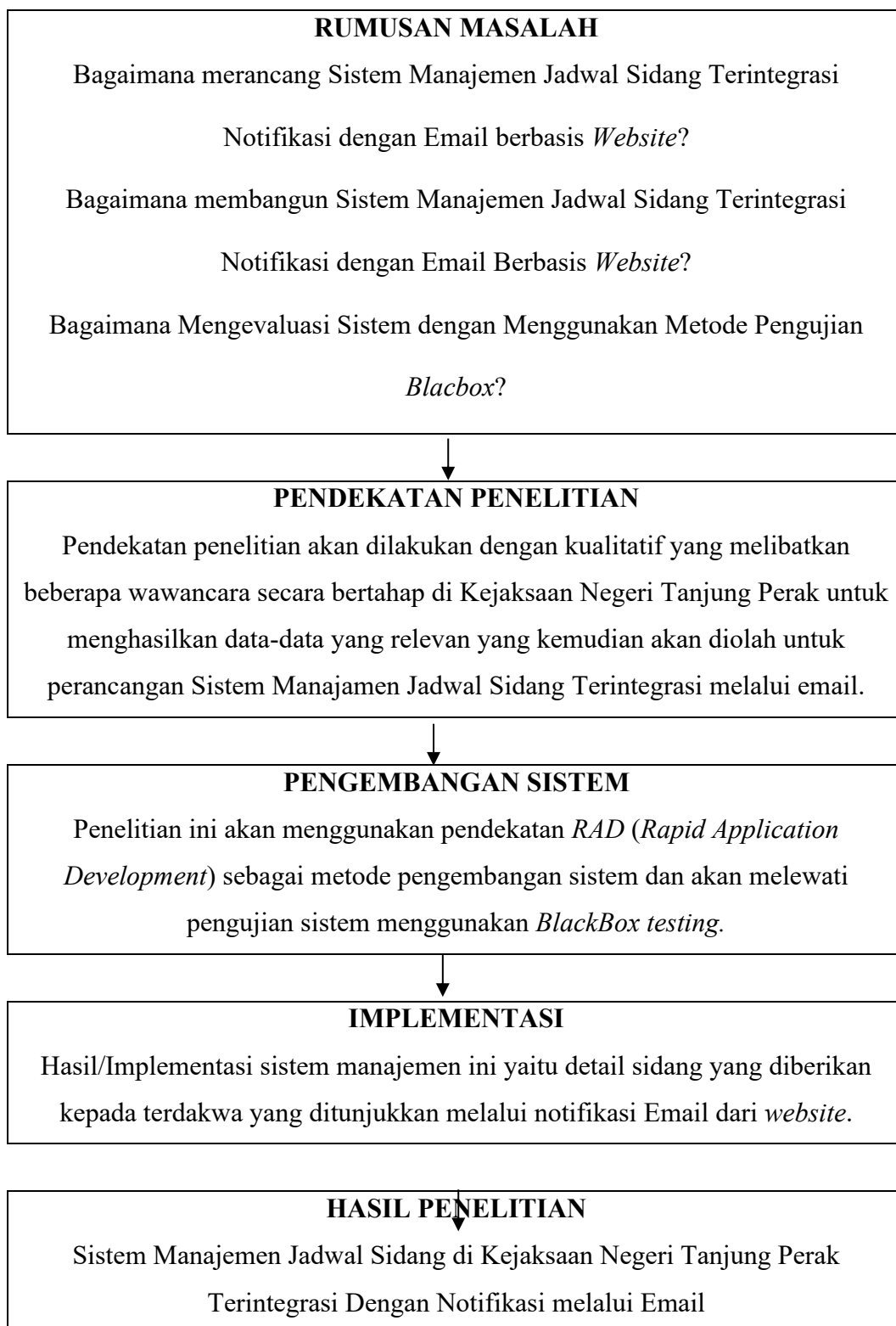
### **C. Kerangka Berfikir**

Jadwal sidang Kejaksaan Negeri Tanjung Perak masih dibuat secara manual atau tidak terintegrasi. Hal ini dapat menyebabkan miskomunikasi, keterlambatan informasi, dan ketidaktepatan jadwal sidang. Sistem informasi yang mampu mengelola dan menyampaikan data secara cepat sangat penting dalam lingkungan kerja yang dinamis dan berorientasi pada waktu. Oleh karena itu, sistem manajemen jadwal sidang diperlukan yang tidak hanya terorganisir dengan baik tetapi juga dapat memberikan notifikasi kepada jaksa, panitera.

Sistem Manajemen Jadwal Sidang yang terintegrasi dengan notifikasi *Email* berbasis *Website* dibuat untuk memenuhi kebutuhan dalam sistem ini. Sistem ini bertujuan untuk meningkatkan efisiensi koordinasi dan mengurangi kemungkinan keterlambatan atau ketidakhadiran dalam proses

persidangan. Untuk pengembangan sistem ini, metode *Rapid Application Development (RAD)* dipilih karena sesuai dengan kebutuhan sistem dan waktu yang mendesak. Selain itu pendekatan ini juga fleksibel terhadap perubahan dan memerlukan validasi cepat dari pengguna. *RAD* menekankan kecepatan pengembangan melalui *iterasi prototipe* dan keterlibatan aktif pengguna akhir.

Diharapkan sistem ini akan meningkatkan kualitas manajemen sidang di Kejaksaan Negeri Tanjung Perak, terutama dalam hal ketepatan waktu, akurasi informasi, dan kemampuan untuk berkomunikasi dengan *stakeholder* dengan lebih baik. Sebuah notifikasi yang dikirim *Email* akan dikirim memastikan bahwa semua pihak selalu diberi tahu tentang perubahan jadwal sidang atau pengingat waktu sidang. Pada akhirnya, sistem ini dapat digunakan untuk mendukung transparansi, digitalisasi proses hukum, dan pembangunan layanan publik yang lebih responsif. Alur kerangka berpikir penelitian ini yang menggambarkan hubungan antara permasalahan, metode penelitian, pengembangan sistem, hingga hasil yang diharapkan. ditunjukkan pada Gambar dibawah ini.



Gambar 2.8. Kerangka Berpikir