

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Perancangan Sistem

Setiap pengembangan sistem biasanya dimulai dengan membuat rancangan atau desain sistem yang akan dibangun. Tujuan utama pembuatan desain sistem biasanya adalah untuk memberi pengguna pemahaman tentang sistem baru yang akan dibangun (Aziz., 2022). Perancangan sendiri adalah tahap awal dalam proses pengembangan suatu produk atau sistem. Pada tahap ini, berbagai prinsip dan pendekatan digunakan untuk mendefinisikan secara menyeluruh suatu alat, proses, atau sistem agar dapat dibentuk secara fisik (Rahmadani et al., 2022). Menurut Fauzi et al., (2022) menjelaskan bahwa perancangan adalah proses atau tahapan dalam merencanakan sesuatu dengan memanfaatkan teknik tertentu guna merumuskan tujuan yang hendak dicapai.

Berdasarkan definisi di atas dapat disimpulkan perancangan adalah tahap pertama dalam pengembangan sistem. Tahap ini melibatkan merencanakan dan mendefinisikan secara rinci suatu sistem menggunakan prinsip dan teknik tertentu. Perancangan ini bertujuan untuk memberikan gambaran yang jelas kepada pengguna tentang sistem yang akan dibangun, sekaligus menjadi landasan untuk menerapkan sistem secara efektif untuk mencapai tujuan yang telah ditetapkan. Dengan kata lain, agar sistem dapat diimplementasikan secara cepat dan efektif, perancangan sangat penting.

2. Sistem Informasi

Secara luas, sistem informasi terdiri dari kumpulan komponen yang saling terhubung yang digunakan untuk mengumpulkan, mengolah, menyimpan, dan menyebarkan informasi untuk membantu operasi, pengambilan keputusan, dan pengendalian suatu organisasi atau lembaga. Dengan kata lain, sistem informasi terdiri dari perangkat keras (*hardware*), perangkat lunak (*software*), data, prosedur, dan manusia (*brainware*). Semua ini memungkinkan organisasi (Effendi et al., 2023).

Sistem informasi tidak sekadar terkait dengan teknologi komputer, tetapi juga sangat penting untuk mengintegrasikan proses bisnis, mempercepat aliran data, dan menekan kesalahan pengolahan data manual. Oleh karena itu, penerapan sistem informasi berbasis teknologi sangat penting untuk mendukung berbagai aktivitas organisasi, termasuk dalam konteks pemerintahan desa, agar informasi dapat diolah dan disampaikan secara cepat dan akurat.

3. Administrasi Kependudukan

Administrasi kependudukan adalah kumpulan tindakan yang mencakup pendaftaran, pencatatan, pengelolaan data, dan penerbitan dokumen kependudukan yang diperlukan oleh pemerintah dan masyarakat umum (Sihotang et al., 2023). Pelayanan administrasi kependudukan merupakan bagian penting dari pelayanan publik yang harus diberikan pemerintah kepada masyarakat secara cepat, tepat, dan efisien. Pelayanan administrasi kependudukan merupakan bagian penting dalam pelayanan

publik yang harus diberikan oleh pemerintah kepada masyarakat secara cepat, tepat, dan efisien.

Administrasi ini berperan menjamin keteraturan dan akurasi data penduduk, yang menjadi dasar untuk perencanaan pembangunan, pelayanan publik, serta pemenuhan hak dan kewajiban warga negara. Layanan ini mencakup pembuatan berbagai dokumen kependudukan, seperti Kartu Tanda Penduduk (KTP), Kartu Keluarga (KK), akta kelahiran, dan akta kematian, yang menunjukkan identitas dan status hukum seorang warga negara Indonesia (Sihotang et al., 2023)

Studi di berbagai wilayah menunjukkan bahwa inovasi layanan digital dan sistem informasi administrasi kependudukan (SI AK) meningkatkan kinerja layanan dan mengurangi kesalahan pengolahan data manual yang sering terjadi di pemerintahan lokal (Aini & Fitria, 2023). Adanya inovasi teknologi pada administrasi kependudukan menegaskan bahwa pengelolaan data penduduk tidak lagi sebatas pencatatan, tetapi juga menyangkut kemudahan akses dan ketepatan penyajian informasi kepada masyarakat.

4. Website

Website adalah kumpulan halaman-halaman digital yang terhubung melalui link *hypertext* dan dapat diakses melalui peramban (*browser*) di jaringan internet dengan menggunakan alamat unik berupa URL. Sebuah website umumnya tersusun atas elemen seperti teks, gambar, video, dan berbagai fitur interaktif lain yang disusun dalam kerangka tampilan dan struktur logis agar informasi tersampaikan secara sistematis kepada

pengguna. Prayogge & Megawati, (2023) menjelaskan bahwa teknologi dasar seperti HTML, CSS, dan JavaScript sangat penting dalam pembuatan struktur, tampilan, dan interaksi antarmuka web. Website adalah alat yang efektif untuk mengirimkan informasi, berkomunikasi, dan mengelola data bagi pengguna yang berada di seluruh dunia, karena memungkinkan akses lintas waktu dan ruang.

Penggunaan website dalam sistem informasi bertujuan memungkinkan pengelolaan data dan komunikasi antarunit melalui platform terpusat yang fleksibel dan mudah diakses. Pihak-pihak terkait dapat melihat dan melihat aktivitas pemantauan data, pelaporan, dan pembaruan informasi melalui website. Dalam penelitian ini, situs web dipilih sebagai media utama karena memungkinkan manajemen untuk memantau dan mengelola data yang terintegrasi dari berbagai bagian organisasi. Oleh karena itu, situs web memfasilitasi transparansi, kemudahan akses, dan proses pengambilan keputusan berbasis data yang cepat.

5. *Shortest Job First (SJF)*

Metode penjadwalan proses Algoritma Shortest Job First (SJF) mengatur eksekusi tugas dalam urutan berdasarkan waktu proses (burst time) tersingkat. Cara kerja metode ini adalah dengan mendahulukan proses dengan durasi eksekusi paling singkat untuk dijalankan lebih dulu, sehingga waktu tunggu tiap proses dapat diminimalkan (Rahmawati & Dinamika, 2022). Sejalan dengan hal tersebut, Sundari et al., (2023) menjelaskan bahwa algoritma SJF memilih proses dalam antrian berdasarkan waktu eksekusi

terkecil guna mengurangi rata-rata waktu tunggu secara keseluruhan. Dalam kondisi di mana terdapat dua atau lebih proses dengan waktu eksekusi yang sama, maka digunakan metode *First Come First Served* (FCFS) sebagai mekanisme tambahan untuk menentukan urutan eksekusi proses.

Berikut adalah rumus yang digunakan untuk menghitung kinerja algoritma SJF:

1. Waktu Tunggu (*Waiting Time* – WT)

lama satu pengajuan surat menunggu sejak masuk antrean sampai mulai diproses petugas, dengan Persamaan 2. 1

$$WT = \text{Waktu Mulai Eksekusi} - \text{Waktu kedatangan} \quad (2.1)$$

2. Waktu Penyelesaian (*Turnaround Time* – TAT)

total waktu dari pengajuan masuk sampai benar-benar selesai (bukan cuma mulai dikerjakan), dengan Persamaan 2. 2 dan 2. 3

$$TAT = \text{Waktu Selesai} - \text{Waktu Kedatangan} \quad (2. 2)$$

Atau bisa juga dengan

$$TAT = \text{Waktu tunggu} + \text{waktu eksekusi} \quad (2. 3)$$

3. Rata-rata waktu (*Average Time*)

bukan untuk satu pengajuan saja, tapi untuk menilai kinerja SJF secara keseluruhan dalam satu simulasi/pengujian yang berisi banyak proses, dengan Persamaan 2. 4 atau 2. 5

$$\text{Average Witing Time (AWT)} = \frac{\sum \text{WT}}{\text{Jumlah Proses}} \quad (2.4)$$

Atau bisa juga dengan

$$\text{Average Turnarounf Time (ATAT)} = \frac{\sum \text{TAT}}{\text{Jumlah Proses}} \quad (2.5)$$

Berdasarkan definisi di atas dapat disimpulkan bahwa algoritma *Shortest Job First* (SJF) merupakan metode penjadwalan yang efektif dalam mengoptimalkan waktu tunggu dengan cara memprioritaskan proses dengan durasi eksekusi paling singkat. Pendekatan ini mampu meningkatkan efisiensi sistem, terutama dalam pengelolaan antrian proses.

6. UML

UML (*Unified Modeling Language*) merupakan standar bahasa pemodelan visual yang digunakan untuk memetakan, mendeskripsikan, merancang, serta mendokumentasikan elemen-elemen dalam pengembangan perangkat lunak dengan paradigma *Object-Oriented* (OO) (Saputra et al., 2025). Penggunaan UML dapat mencegah munculnya makna ganda (ambiguitas) serta menghindari penggunaan kata-kata yang terlalu panjang dan detail teknis yang kurang relevan, seperti yang sering terjadi pada pemodelan berbasis kode atau *pseudocode*. Dengan beralih ke UML, risiko kesalahpahaman dalam tim dapat dikurangi secara signifikan.

Hal ini dikarenakan UML merupakan bahasa formal yang setiap simbolnya memiliki arti khusus dan tetap. Selain itu, notasinya yang

sederhana membuat penjelasan sistem menjadi lebih ringkas dan tidak berbelit-belit, namun tetap mampu memotret seluruh bagian penting sistem secara menyeluruh. UML juga sangat fleksibel untuk menangani proyek dalam berbagai skala, serta didukung oleh komunitas pengguna yang besar sehingga mempermudah proses belajar melalui berbagai tutorial yang tersedia. Sebagai bahasa standar internasional, UML memberikan kebebasan bagi pengembang karena penggunaannya tidak terikat pada merek atau produk perangkat lunak tertentu (Sumirat et al., 2023).

UML terdiri dari sekelompok diagram atau bagan sistem. Diagram atau bagan tersebut menggambarkan permasalahan dan solusinya. Terdapat sembilan diagram pada UML yaitu *use-case*, *class*, *object*, *state*, *sequence*, *collaboration*, *activity*, *component*, dan *deployment diagram*. namun dalam penelitian ini hanya akan digambarkan empat diagram sebagai berikut:

a. Use Case

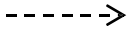
Use case diagram adalah salah satu komponen penting dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan hubungan antara aktor, kumpulan *use case*, serta interaksi di antaranya. Diagram ini berperan untuk menunjukkan secara jelas bagaimana pengguna berinteraksi dengan sistem. Tujuan utamanya yaitu mengidentifikasi dan menjelaskan kebutuhan pengguna sehingga dapat memberikan gambaran yang pasti mengenai fungsi-fungsi utama yang harus ada dalam sistem.

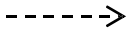
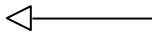
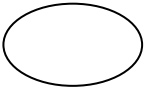
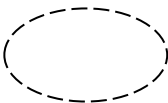
Dengan memahami hubungan antara aktor dan sistem, pengembang akan lebih mudah dalam menentukan fitur maupun tampilan yang sesuai dengan kebutuhan pengguna. Tidak hanya sebagai alat untuk memetakan fungsi, *use case* diagram juga memiliki manfaat yang lebih luas, seperti membantu dalam menyusun struktur organisasi serta membangun model awal arsitektur sistem (Saputra et al., 2025).

Selain itu, *use case* diagram juga berperan penting dalam mendukung kelancaran proses pengembangan. Diagram ini dapat menjadi media komunikasi yang efektif antara tim teknis dan pemilik proyek. Melalui visualisasi yang jelas, kedua pihak dapat menghindari kesalahpahaman dan memiliki pemahaman yang sama mengenai sistem yang sedang dikembangkan (Widodo et al., 2026).

.Simbol dalam *use case* diagram dapat dilihat pada table 2.1

Tabel 2. 1 Komponen use case diagram

No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Digunakan untuk menentukan posisi yang digunakan oleh <i>user</i> saat berinteraksi dengan <i>use case</i> .
2.		<i>Dependency</i>	Penghubung antara perubahan pada elemen mandiri yang akan mempengaruhi pada elemen lain yang bergantung pada elemen tersebut.
3.		<i>Generalization</i>	Penghubung objek yang mewarisi perilaku dan struktur dari objek orang tua.

4.		<i>Include</i>	Menjelaskan secara eksplisit <i>use case</i> sebagai permulaan.
5.		<i>Extend</i>	Menyatakan <i>use case</i> yang bersifat sebagai target untuk memperbesar bentuk perilaku dari <i>use case</i> utama pada bagian tertentu.
6.		<i>Associatiom</i>	Mengidentifikasi paket yang menampilkan sistem secara terbatas
7		<i>Sistem</i>	Menentukan bagian yang menampilkan sistem dengan cakupan terbatas.
8		<i>Use case</i>	Penjelasan mengenai rangkaian aktivitas yang dilakukan sistem untuk menciptakan sebuah capaian yang dapat diukur oleh subjek..
9		<i>Collaboration</i>	Kolaborasi antar aturan dan komponen lain yang berfungsi secara bersamaan guna menciptakan perilaku yang signifikan pada kuantitas dan bagian-bagian tersebut.
10		<i>Note</i>	Komponen nyata yang ada pada saat aplikasi beroperasi dan memnggambarkan sebuah sumber daya komputasi.

Sumber : (Rasiban et al., 2024)

b. Activity Diagram

Activity Diagram merupakan salah satu instrumen dalam UML yang digunakan untuk memvisualisasikan alur kerja atau rangkaian aktivitas dalam suatu proses sistem. Diagram ini menggambarkan secara sistematis tahapan kegiatan, tindakan operasional, serta titik-titik pengambilan

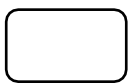
keputusan yang terjadi dalam urutan waktu tertentu, sehingga logika proses dalam sistem dapat dipahami secara lebih mendalam dan terstruktur (Rasiban et al., 2024) .

Secara lebih rinci, *activity diagram* berfungsi untuk menampilkan alur kerja yang berlangsung di dalam sistem, baik pada proses bisnis maupun pada fitur atau menu dalam perangkat lunak. Diagram ini menitikberatkan pada penggambaran operasional internal sistem secara mandiri, tanpa berfokus pada tindakan langsung dari aktor. Dengan demikian, *activity diagram* memberikan gambaran yang jelas mengenai bagaimana sistem merespons dan menjalankan fungsinya secara terstruktur (Musthofa & Adiguna, 2022).

Berdasarkan uraian di atas dapat disimpulkan *Activity diagram* digunakan untuk menggambarkan alur kerja dan proses dalam sistem secara terstruktur, sehingga memudahkan pemahaman terhadap logika dan mekanisme operasional sistem.

Simbol dalam *Activity diagram* dapat dilihat pada table 2.2

Tabel 2. 2 Komponen activity diagram

No	Gambar	Nama	Keterangan
1.		<i>Activity</i>	Mengindikasikan cara setiap objek antarmuka saling berinteraksi.
2.		<i>Action</i>	Mengindikasikan keadaan sistem yang mencerminkan pelaksanaan satu operasi.

3.		<i>Initial node</i>	Menunjukkan cara kerja komponen dibuat ataupun dimulai.
4.		<i>Activity final node</i>	Menunjukkan cara kerja sebuah komponen yang dibuat dan dihancurkan.
5.		<i>Fork node</i>	Menunjukkan sebuah arus yang pada fase tertentu berkembang menjadi beberapa bagian.

Sumber : (Rasiban et al., 2024)

c. *Sequence Diagram*

Sequence diagram merupakan salah satu model dalam UML yang digunakan untuk memvisualisasikan interaksi antarobjek dalam suatu sistem berdasarkan urutan waktu. Diagram ini menggambarkan secara rinci bagaimana objek-objek saling berkomunikasi serta menunjukkan urutan langkah yang terjadi ketika suatu proses atau skenario dijalankan. Melalui *sequence diagram*, alur pesan dan pertukaran informasi antarobjek dapat dipahami secara kronologis sehingga memudahkan pengembang dalam menganalisis jalannya sistem (Rasiban et al., 2024).

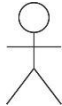
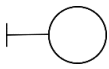
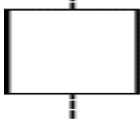
Selain itu, *sequence diagram* juga merepresentasikan kolaborasi dinamis antar berbagai objek dalam sistem. Diagram ini menampilkan rangkaian pesan (*message*) yang dikirim antarobjek serta memperlihatkan bagaimana interaksi tersebut berlangsung selama proses eksekusi. Dengan adanya pemetaan tersebut, pengembang dapat memahami alur komunikasi

dan urutan kerja setiap objek secara lebih jelas dan (Musthofa & Adiguna, 2022)..

Sehingga dapat disimpulkan *sequence diagram* digunakan untuk menggambarkan interaksi dan pertukaran pesan antarobjek dalam sistem secara berurutan berdasarkan waktu, sehingga membantu memahami alur komunikasi dan proses kerja sistem secara detail.

Simbol dalam *sequence diagram* dapat dilihat pada table 2.3

Tabel 2. 3 Komponen sequence diagram

No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Menunjukkan cara setiap kelas antarmuka saling berinteraksi.
2.		<i>Entity class</i>	Menunjukkan cara setiap kelas antarmuka saling berinteraksi.
3.		<i>Boundary class</i>	Menjelaskan bentuk atau tampilan dari foem.
4.		<i>Controll class</i>	Menjelaskan hubungan antara batas dengan tabel.
5.		<i>A focus of control & a life line</i>	Menjelaskan tempat awal dan akhir dari message.
6.		<i>A message</i>	Menjelaskan proses pengiriman pesan.

Sumber : (Rasiban et al., 2024)

d. Class Diagram

Class diagram merupakan salah satu bagian dari UML yang digunakan untuk memvisualisasikan struktur statis pada suatu sistem atau

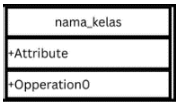
aplikasi berbasis objek. Berbeda dengan diagram lain yang bersifat dinamis, class diagram berfokus pada penggambaran komponen utama penyusun sistem, seperti kelas, atribut, metode, serta hubungan antar kelas secara tetap. Dengan adanya diagram ini, pengembang dapat memperoleh gambaran menyeluruh mengenai kerangka dasar sistem sebelum diimplementasikan ke dalam kode program (Rasiban et al., 2024).

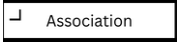
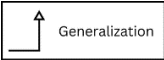
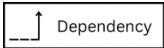
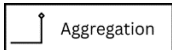
Selain itu, *class diagram* juga berfungsi sebagai representasi visual yang menggambarkan struktur penyusun suatu program, termasuk dalam memetakan kerangka dasar basis data (*database*) yang akan digunakan. Melalui diagram ini, pengembang dapat memahami arsitektur data yang mendukung sistem secara lebih jelas dan terstruktur (Sandfreni et al., 2021)

Berdasarkan teori di atas dapat disimpulkan *class diagram* digunakan untuk menggambarkan struktur statis sistem yang meliputi kelas, atribut, metode, serta hubungan antar kelas, sehingga membantu pengembang memahami arsitektur sistem dan perancangan basis data secara menyeluruh

Simbol dalam *class diagram* dapat dilihat pada table 2.4

Tabel 2. 4 Komponen class diagram

No	Simbol	Nama	Keterangan
1.		Kelas	Sebuah kategori dalam susunan sebuah sistem.
2.		Antarmuka (<i>interface</i>)	Serupa dengan gagasan antarmuka dalam OOP.

3.		Assosiasi (<i>Association</i>)	Hubungan antar tiap kategori yang memiliki arti yang umum, asosiasi umumnya juga disertai dengan multiplikitas.
4.		Asosiasi berarah (<i>Directed Association</i>)	Hubungan antar kategori dimana satu kategori dimanfaatkan oleh kategori lain, yang umumnya disertai dengan multiplikitas.
5.		Generalisasi (<i>Generalization</i>)	Hubungan antar kategori dengan makan untuk menyamaratakan secara khusus.
6.		Kebergantuaan (<i>Dependency</i>)	Hubungan antar kategori yang bermakna ketergantungan antar kategori.
7.		Aggregasi (<i>Aggregation</i>)	Hubungan antar kategori yang dengan arti keseluruhan atau sebagian.

Sumber : (Rasiban et al., 2024)

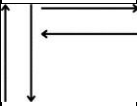
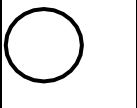
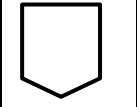
7. Flowchart

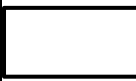
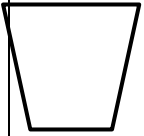
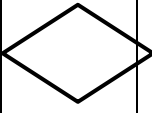
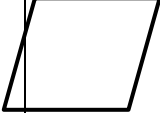
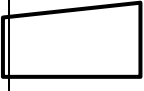
Flowchart merupakan representasi visual yang digunakan untuk menggambarkan alur proses atau mekanisme kerja secara terstruktur dan sistematis sehingga mudah dipahami. Dalam *flowchart*, digunakan simbol-simbol standar seperti persegi panjang untuk menunjukkan langkah proses, panah sebagai arah aliran, serta simbol berlian yang merepresentasikan pengambilan keputusan.

Selain itu, *flowchart* memiliki berbagai manfaat, antara lain untuk memodelkan proses bisnis, mendukung analisis sistem, merancang solusi yang lebih efektif dan efisien, serta sebagai sarana dokumentasi yang mempermudah pemahaman dan komunikasi alur kerja (Ramadhan & Gunawan, 2024).

Berdasarkan definisi di atas *flowchart* adalah alat visual terstruktur yang menggunakan simbol standar untuk menjelaskan alur proses secara jelas, serta berfungsi dalam analisis, pemodelan, perancangan, dan dokumentasi sistem secara efektif. Simbol dalam *flowchart* dapat dilihat pada

Tabel 2.5 Komponen *Flowchart*

No	Gambar	Nama	Keterangan
		<i>Flow direction symbol</i>	Simbol yang dipergunakan untuk mengaitkan satu simbol dengan simbol lainnya.
		<i>Termin Symbol</i>	Simbol yang menunjukkan awal tahapan awal ataupun akhir dari sebuah aktivitas.
3.		<i>Connector Symbol</i>	Simbol yang dipakai untuk masuk atau keluar dari proses dalam halaman yang serupa.
4.		<i>Connector Symbol</i>	Simbol yang digunakan untuk memasuki atau keluar dari proses dalam halaman yang berbeda.

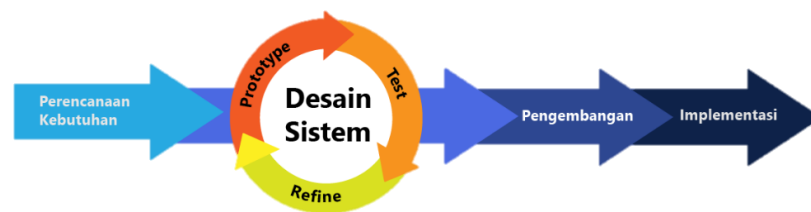
5.		<i>Processing Symbol</i>	Simbol yang menandakan bahwa pengelolaan dilakukan secara manual tanpa menggunakan komputer.
6.		<i>Manual Operating Symbol</i>	Simbol yang menunjukkan pengelolaan yang tidak dilakukan oleh komputer.
7.		<i>Decision Symbol</i>	Simbol yang digunakan untuk memilih prosedur sesuai dengan keadaan yang berlaku.
8.		<i>Input-Output Symbol</i>	Simbol yang menunjukkan proses masuk dan keluarnya data tanpa bergantung pada jenis perangkat yang dipakai.
9.		<i>Manual Input Symbol</i>	Simbol input data secara manual melalui keyboard secara langsung

Sumber : (Rasiban et al., 2024)

8. RAD

Rapid Application Development (RAD) merupakan metode pengembangan perangkat lunak yang menitikberatkan pada percepatan proses perancangan dan pembuatan aplikasi. Pendekatan ini dijalankan melalui siklus kerja berulang (iteratif) disertai evaluasi berupa umpan balik secara terus-menerus, sehingga proses pengembangan menjadi lebih cepat dan terarah (Alam et al., 2023).

RAD juga mengutamakan efisiensi waktu dengan memanfaatkan prototype atau model awal sistem yang dikembangkan secara fleksibel berdasarkan masukan pengguna atau klien. Dengan cara ini, sistem yang dibangun dapat lebih sesuai kebutuhan pengguna serta mampu mencapai fungsi yang diharapkan dalam waktu yang relatif singkat (Ramadhan et al., 2024) .



Gambar 2. 1 Alur Metode RAD
Sumber : (Sondang, 2024)

Terdapat 4 fase utama dalam RAD yaitu

a. Perencanaan Kebutuhan (*Requirements Planning*)

Pada tahap ini, pengembang dan pengguna berkolaborasi dalam sebuah lokakarya untuk mendefinisikan ruang lingkup dan kebutuhan dasar sistem secara umum, tanpa terlalu mendetail. Tujuannya adalah menyepakati tujuan utama dan fitur-fitur paling krusial

b. Desain Pengguna (*User Design*)

Tim pengembang secara cepat membuat prototipe fungsional dari sistem. Prototipe ini kemudian diuji coba langsung oleh pengguna untuk mendapatkan umpan balik. Proses pembuatan prototipe, pengujian, dan penyempurnaan (*refine*) ini dilakukan secara berulang (iteratif) hingga

desain sistem sepenuhnya sesuai dengan harapan dan kebutuhan pengguna.

c. Konstruksi Cepat (*Rapid Construction*)

Setelah desain disetujui, prototipe tersebut diubah menjadi produk akhir. Fase ini berfokus pada kecepatan pengembangan dengan memanfaatkan komponen yang dapat digunakan kembali (*reusable components*) dan *tools* pengembangan modern untuk mengubah model kerja menjadi sistem yang solid dan fungsional.

d. Implementasi

Tahap akhir ini mencakup proses finalisasi sistem, pengujian akhir, pelatihan pengguna, dan peluncuran sistem ke lingkungan operasional. Karena pengguna telah terlibat secara intensif sejak awal, proses transisi ini dapat berjalan dengan lebih lancar.

Berdasarkan teori di atas dapat disimpulkan RAD merupakan metode pengembangan sistem yang mengedepankan kecepatan melalui proses iteratif dan pembuatan model awal sistem berbasis umpan balik agar sistem cepat selesai dan sesuai kebutuhan pengguna.

9. MySQL

MySQL merupakan sistem manajemen basis data relasional (*Relational Database Management System*) bersifat sumber terbuka (*open source*) yang menggunakan *Structured Query Language* (SQL) untuk mengelola dan mengakses data secara efisien. MySQL banyak digunakan dalam pengembangan aplikasi berbasis web karena memiliki kecepatan

tinggi, kestabilan performa, serta dukungan luas terhadap berbagai sistem operasi. Selain itu, sifatnya yang gratis, mudah diintegrasikan, dan kompatibel dengan berbagai bahasa pemrograman menjadikan MySQL sebagai pilihan utama bagi pengembang sistem informasi modern. Menurut Aziz (2025), keunggulan MySQL terletak pada kemampuannya dalam menangani data dalam jumlah besar dengan performa tinggi, keamanan yang baik, serta kemudahan implementasi pada aplikasi berbasis PHP.

Berdasarkan uraian di atas dapat disimpulkan MySQL adalah sistem manajemen basis data relasional *open source* yang andal untuk mengelola data menggunakan SQL. Dengan kecepatan tinggi, performa stabil, serta kemudahan integrasi dengan berbagai bahasa pemrograman khususnya PHP, MySQL menjadi pilihan utama dalam pengembangan aplikasi web karena mampu menangani data dalam jumlah besar secara efisien dan aman.

10. Laravel

Laravel adalah framework open source berbasis PHP yang menerapkan pola arsitektur *Model-View-Controller* (MVC) untuk memisahkan lapisan logika aplikasi, data, dan tampilan secara terstruktur. *Framework* ini menyediakan banyak fitur bawaan seperti *routing*, *middleware*, *authentication*, *migration*, dan ORM (*Eloquent*) yang memudahkan pengembang dalam menangani operasi basis data, validasi, dan keamanan aplikasi. Menurut Sinlae et al., (2024) Laravel membantu menyederhanakan logika pengelolaan data dan memudahkan penanganan

CRUD (*Create, Read, Update, Delete*) sehingga proses pengembangan menjadi lebih konsisten dan cepat.

Berdasarkan teori di atas dapat disimpulkan Laravel merupakan framework PHP berbasis MVC yang memudahkan pengembangan aplikasi dengan memisahkan struktur sistem secara terorganisir. Dengan berbagai fitur bawaan seperti routing, authentication, dan ORM, Laravel mampu menyederhanakan pengelolaan data serta proses CRUD, sehingga pengembangan aplikasi menjadi lebih cepat, efisien, dan konsisten.

11. System Usability Scale (SUS)

System Usability Scale (SUS) adalah salah satu metode yang umum digunakan untuk mengevaluasi tingkat kemudahan penggunaan suatu sistem. Metode ini mencakup sepuluh pernyataan yang menggambarkan pengalaman pengguna saat berinteraksi dengan sistem, dengan menggunakan skala *Likert* lima tingkat, mulai dari “Sangat Tidak Setuju” hingga “Sangat Setuju”. Keunggulan SUS terletak pada proses pengisiannya yang sederhana, cepat, serta mudah dipahami oleh responden dengan berbagai latar belakang. Hasil dari pengisian tersebut kemudian dikonversi menjadi satu nilai yang merepresentasikan tingkat usability sistem secara keseluruhan. Nilai ini dapat dijadikan indikator awal untuk menentukan apakah sistem sudah cukup mudah digunakan atau masih membutuhkan pengembangan lebih lanjut. Dalam penerapannya SUS berperan menilai sejauh mana pengguna merasa nyaman serta terbantu ketika memanfaatkan fitur tersebut (Oktria & Novandya, 2026).

Metode *System Usability Scale* (SUS) merupakan cara yang efektif dan praktis untuk mengukur tingkat kemudahan penggunaan suatu sistem. Dengan proses yang sederhana dan hasil berupa skor, SUS dapat memberikan gambaran awal apakah sistem sudah mudah digunakan atau masih perlu diperbaiki. Dalam konteks fitur, metode ini membantu mengetahui tingkat kenyamanan dan kemudahan yang dirasakan oleh pengguna.

B. Kajian Empiris

Pertama, penelitian yang dilakukan oleh Saputra et al., (2024) yang mengembangkan Sistem Informasi Kependudukan Desa (SIKADES) berbasis web. Penelitian ini berhasil merancang sistem yang meningkatkan efisiensi dan efektivitas pelayanan administrasi di tingkat desa dengan menggantikan metode manual yang belum optimal. Sistem ini menawarkan solusi inovatif bagi masyarakat untuk melakukan input data secara online guna mengurangi penggunaan kertas. Proses administrasi menjadi lebih terstruktur dengan adanya fitur verifikasi oleh Lurah dan pemantauan real-time oleh Camat. Hasil pengujian menunjukkan bahwa sistem ini mampu meningkatkan akurasi data serta memberikan kenyamanan akses bagi masyarakat.

Kedua, penelitian yang dilakukan oleh Amalia et al., (2025) dengan judul "*Digitizing Village Administration in Sleman with a Web-Based System using RAD Method*". Penelitian ini menekankan pentingnya teknologi modern seperti cloud computing dan keamanan data untuk meningkatkan transparansi manajemen desa di era Revolusi Industri 4.0. Selain otomatisasi surat-menyurat, sistem ini dirancang agar masyarakat dapat mengawasi penggunaan dana desa

dan program pembangunan. Implementasi ini mendukung transformasi digital desa menuju tata kelola yang lebih akuntabel, dengan rencana pengembangan aplikasi mobile di masa depan untuk meningkatkan aksesibilitas warga.

Ketiga, penelitian dari Ardila et al., (2024) mengenai sistem informasi pelayanan dokumen kependudukan pada Disdukcapil Kabupaten Langkat. Dengan menerapkan metode *Waterfall*, penelitian ini membangun sistem terstruktur melalui lima tahap pengembangan untuk mengatasi kendala waktu dan biaya pada layanan manual. Sistem ini memungkinkan pengajuan permohonan secara online yang terintegrasi dengan SIAK, sehingga validitas data terjaga. Keberhasilan proyek ini diharapkan menjadi referensi bagi daerah lain dalam mengadopsi solusi modern untuk pengelolaan dokumen kependudukan yang lebih efektif.

Keempat, penelitian yang dilakukan oleh Alajuri et al., (2025) dengan judul "*Pengembangan Website Pelayanan Administrasi bagi Masyarakat Kelurahan Sei Enam, Tanjungpinang*". Penelitian ini berfokus pada digitalisasi pelayanan administrasi kependudukan di tingkat kelurahan/desa menggunakan metode *Rapid Application Development* (RAD). Sistem berbasis website ini memfasilitasi pembuatan berbagai dokumen penting secara daring. Fokus utama penelitian tersebut adalah meningkatkan efisiensi waktu pemrosesan dokumen dibandingkan dengan metode manual yang lambat. Meskipun demikian, pengajuan surat pada sistem ini masih bersifat kumulatif dan konvensional, sehingga belum menyentuh aspek optimasi antrean ketika terjadi lonjakan volume pengajuan dokumen oleh masyarakat.

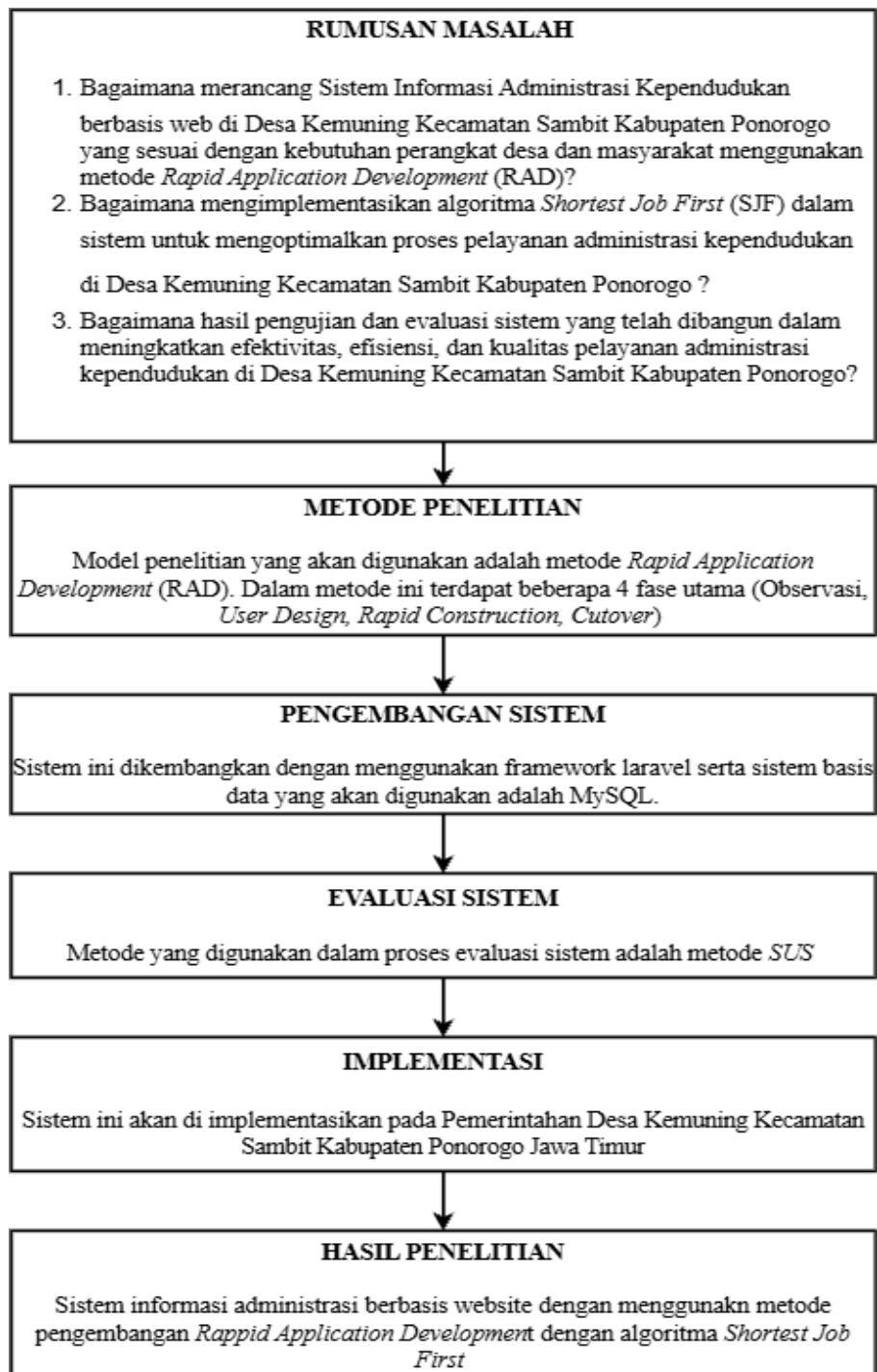
Kelima, penelitian yang dilakukan oleh) Hutabarat et al., (2025) dengan judul "Pengembangan Sistem Informasi Pelayanan Administrasi Kependudukan Berbasis Web di Kantor Desa Helvetia". Penelitian ini berfokus pada pengembangan sistem informasi pelayanan guna mempermudah proses pencarian data warga dan mempercepat pembuatan berbagai surat. Sistem ini berhasil meningkatkan transparansi, akurasi, dan efisiensi pelayanan desa secara keseluruhan,

Berdasarkan paparan penelitian terdahulu di atas, terdapat perbedaan mendasar dengan penelitian yang penulis lakukan dengan judul "Sistem Informasi Administrasi Kependudukan Berbasis Web Algoritma *Shortest Job First* (SJF)".

Adapun perbedaan tersebut terletak pada integrasi *Algoritma Shortest Job First* (SJF). Jika penelitian terdahulu umumnya berfokus pada digitalisasi data, transparansi, dan otomatisasi surat-menyurat, penelitian ini menambahkan dimensi optimasi pada manajemen antrean pelayanan. Algoritma SJF digunakan untuk memprioritaskan proses administrasi berdasarkan waktu pengerjaan terpendek, sehingga dapat meminimalkan rata-rata waktu tunggu (*average waiting time*) warga di Desa Kemuning. Selain itu, konteks implementasi pada Desa Kemuning, Ponorogo memberikan karakteristik kebutuhan data dan regulasi lokal yang berbeda dari lokasi penelitian-penelitian sebelumnya.

C. Kerangka Berfikir

Kerangka berfikir penelitian ini dapat dilihat pada gambar 2.2 berikut



Gambar 2. 2 Kerangka Berfikir