

BAB II

LANDASAN TEORI

2.1 Kajian Teori

Bab ini berfungsi sebagai landasan konseptual yang dijadikan sebagai landasan dalam pelaksanaan analisis serta perancangan sistem pada penelitian ini. Berbagai teori tersebut digunakan dan dipilih berdasarkan relevansinya dengan permasalahan yang diangkat, serta berasal dari sumber yang telah teruji. Selain itu, bab ini juga memuat penelitian terdahulu yang berkaitan sebagai bahan perbandingan dan penguat dalam penyusunan penelitian ini, khususnya dalam pengembangan sistem informasi berbasis web pada bidang laundry.

2.1.1 Sistem Informasi

Sistem informasi merupakan kesatuan fungsional yang terdiri atas perangkat teknologi, prosedur operasional. Secara lebih spesifik, sistem informasi dapat dipahami sebagai kumpulan subsistem yang terintegrasi dan bekerja secara kolaboratif dalam menyelesaikan persoalan bisnis melalui pemanfaatan komputer sebagai alat pengolah data (Asmarany et al., 2026). Dalam konteks penelitian ini, sistem informasi dirancang sebagai instrumen pendukung operasional Laundry 22 dalam mengelola data pelanggan, transaksi, dan status cucian secara terintegrasi dan efisien.

2.1.2 *Monitoring*

Monitoring adalah pemantauan perkembangan tahapan-tahapan yang telah dilalui. *Monitoring* adalah sebuah proses pemantauan untuk mengetahui sebuah keadaan. Dengan memperhatikan pengertian yang telah dipaparkan, dapat disimpulkan bahwa *monitoring* dapat dipahami sebagai proses pemantauan yang dilakukan untuk memastikan pelaksanaan program sesuai rencana, menilai perkembangan, serta mendeteksi hambatan atau kondisi tertentu (Asmarany et al., 2026).

2.1.3 Transaksi

Transaksi merupakan salah satu proses utama dalam sistem laundry yang mencakup kegiatan pencatatan data pelanggan, jenis layanan, perhitungan biaya, pembayaran, hingga penyimpanan data informasi transaksi secara digital. Penerapan sistem transaksi berbasis web memungkinkan seluruh proses tersebut dilakukan secara terintegrasi sehingga dapat mengurangi kesalahan pencatatan, mempermudah pengelolaan data, serta mendukung penyusunan laporan secara lebih sistematis. Selain itu, sistem juga mampu menghubungkan transaksi dengan proses pelacakan status layanan sehingga informasi mengenai pesanan dapat dipantau secara real-time oleh pengguna (Anwar et al., 2026).

2.1.4 Website

Secara terminologis, *website* didefinisikan sebagai kumpulan halaman situs yang terhimpun dalam satu domain atau subdomain pada *World Wide Web* (WWW) seluruh situs yang dapat diakses publik. Menurut Gregorius, *website* terdiri atas halaman-halaman web yang saling terhubung beserta berkas terkait, dengan *home page* sebagai kumpulan halaman utama. Secara fungsional, *website* menyajikan informasi teks, citra (diam maupun bergerak), animasi, audio, serta kombinasinya, baik statis maupun dinamis, yang membentuk struktur bangunan saling terkait melalui jaringan antarhalaman (Asahan, 2025).

2.1.5 Laundry

Laundry merupakan layanan jasa yang menyediakan pencucian berbagai jenis pakaian dan kain dengan menggunakan air, deterjen, pelembut, dan pewangi. Seiring perkembangan zaman, fungsi laundry tidak hanya terbatas pada pembersihan, tetapi juga mencakup perawatan pakaian untuk menjaga kualitas dan ketahanan bahan (Hasanah et al., 2021). Keberadaan layanan ini menjadi solusi bagi masyarakat dengan tingkat kesibukan tinggi, terutama di kawasan perkotaan, yang mengalami keterbatasan waktu untuk melakukan pencucian secara mandiri. Fenomena ini mendorong pertumbuhan industri laundry sebagai peluang bisnis yang prospektif, seiring dengan meningkatnya permintaan pasar.

2.1.6 Basis Data

Menurut penelitian (Putri et al., 2025) yang ada, sistem database merupakan kumpulan data yang disusun dan dikelola sedemikian rupa sehingga dapat menyimpan, memproses, dan mengambil data secara efisien dalam suatu sistem informasi. Database digunakan untuk mendukung operasi pada berbagai aplikasi informasi seperti sistem akademik, manajemen sumber daya, dan sistem informasi besar lainnya. Perancangan basis data meliputi tahapan konseptual, logikal, serta implementasi dengan tujuan mendapatkan struktur data yang memenuhi kebutuhan pengguna serta menjamin integritas dan performa sistem informasi.

2.1.7 *Unified Modelling Language* (UML)

Unified Modelling Language (UML) adalah bahasa pemodelan yang menjadi standar untuk menciptakan gambaran visual dari sistem perangkat lunak. UML berperan sebagai 'peta' atau 'cetak biru' yang memudahkan pengembang dan analis untuk memahami, merancang, serta mengomunikasikan arsitektur sistem dengan baik (Ramadani, 2025).

a. *Use Case Diagram*

Use Case Diagram adalah gambaran grafis dari beberapa atau aktor, *use case*, dan interaksi diantaranya yang memperkenalkan suatu sistem, untuk dikembangkan dalam *Use Case* yang disajikan terdapat empat proses utama yang dilakukan dalam sistem informasi *Monitoring* yang akan dibuat (Ramadani, 2025).

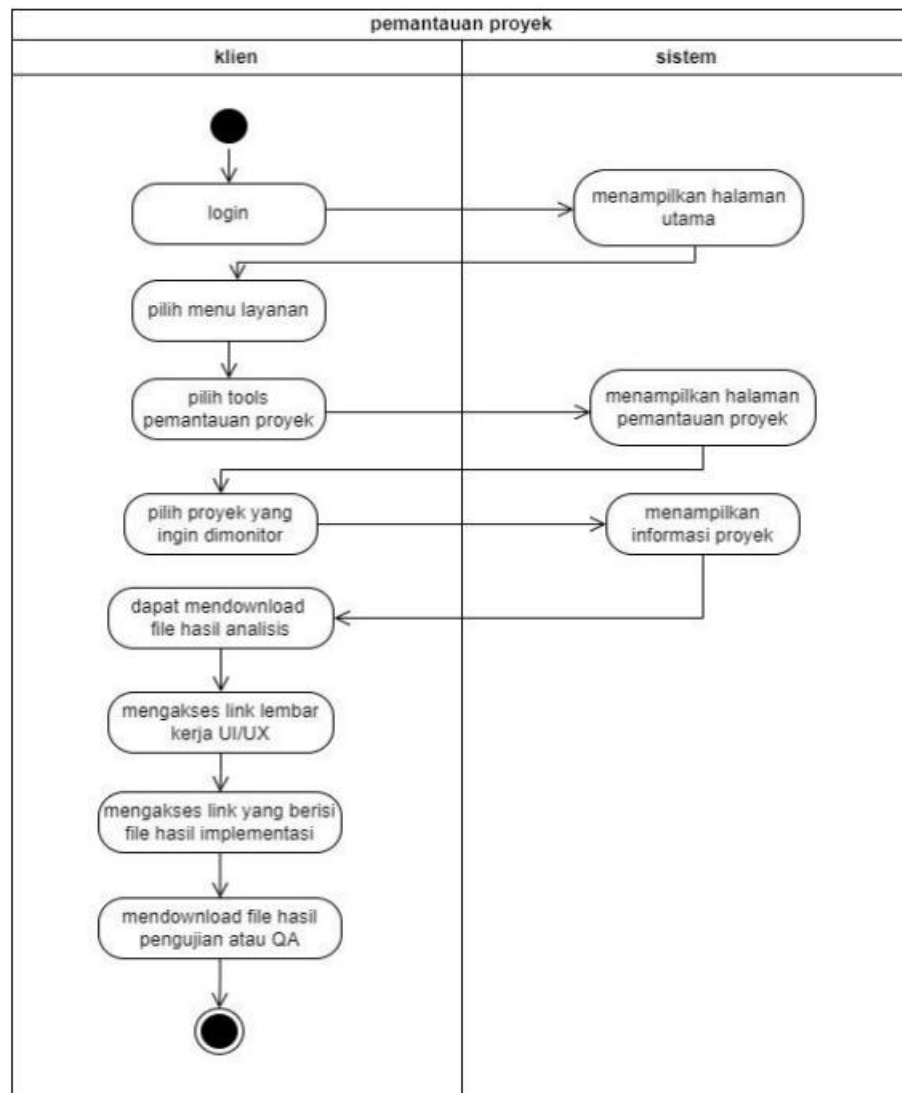
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
3		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
4		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
5		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

Gambar 2.1 Simbol *Use Case*

Sumber : Hafsari, Arribe, Andria, & Miransya (2024)

b. *Activity Diagram*

Activity Diagram menunjukkan aktivitas sistem dalam bentuk kumpulan aksi-aksi, bagaimana masing-masing aksi tersebut dimulai, keputusan yang mungkin terjadi hingga berakhirnya aksi. *Activity Diagram* juga dapat menggambarkan proses lebih dari satu aksi dalam waktu bersamaan “*Diagram Activity* adalah aktifitas-aktifitas, objek, *state*, transisi *state*, dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas” (Ramadani, 2025).

Gambar 2.2 Simbol *Activity Diagram*

Sumber : Hafsari, Arribe, Andria, & Miransya (2024)

c. *Class diagram*

Menurut (Nurmiati, 2022) *Class diagram* ialah mendeskripsikan struktur sistem asal segi pendefinisian kelas-kelas yang akan dirancang untuk menciptakan sistem. Diagram ini menampilkan kelas-kelas atau kelompok, yang Dimana didalamnya memiliki atribut, metode (operasi), dan hubungan antar objek.

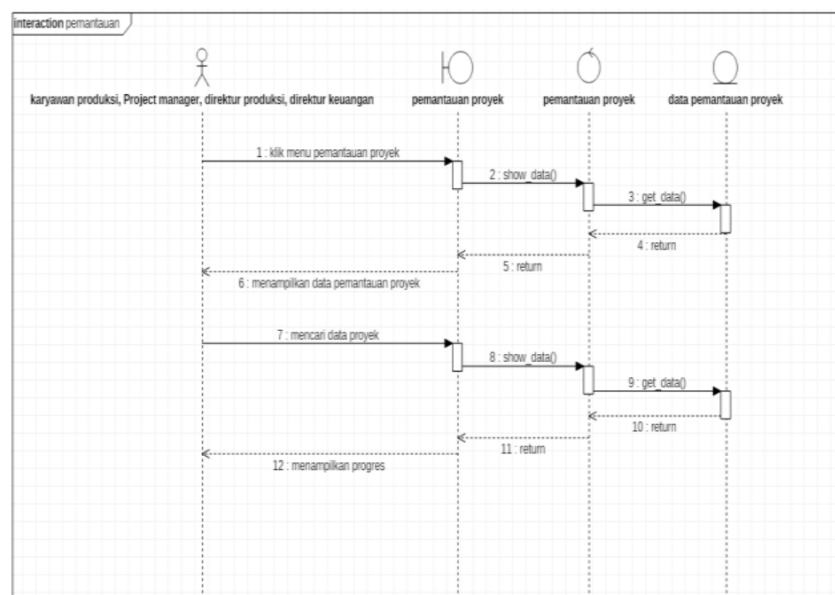
Simbol	Keterangan
Generalization 	relasi antara <i>object</i> anak (<i>descendent</i>) dalam berbagi perilaku serta struktur data dari <i>object</i> induk (<i>ancestor</i>)
Nary Association 	Upaya untuk memisahkan asosiasi dari 2 <i>object</i> atau lebih
Class 	Himpunan dari <i>object</i> yang memiliki atribut dan operasi yang sama
Colaboration 	penjelasan urutan aksi yang ditampilkan sistem dengan satuan terukur bagi suatu aktor
Realization 	Operasi yang dilakukan oleh <i>object</i>
Defendancy 	Perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) dengan elemen yang tidak mandiri
Association 	hubungan antara <i>object</i> satu dengan <i>object</i> lainnya.

Gambar 2.3 Simbol *Class Diagram*

Sumber : Hafsari, Arribe, Andria, & Miransya (2024)

d. *Sequence Diagram*

Sequence Diagram merupakan representasi visual yang menggambarkan bagaimana komponen suatu sistem saling berinteraksi seiring waktu, dengan menunjukan alur proses secara kronologis. Diagram ini memperlihatkan pertukaran pesan antar objek beserta responnya, sehingga membantu memahami alur komunikasi dalam sistem secara jelas (Cahyo et al., 2025).

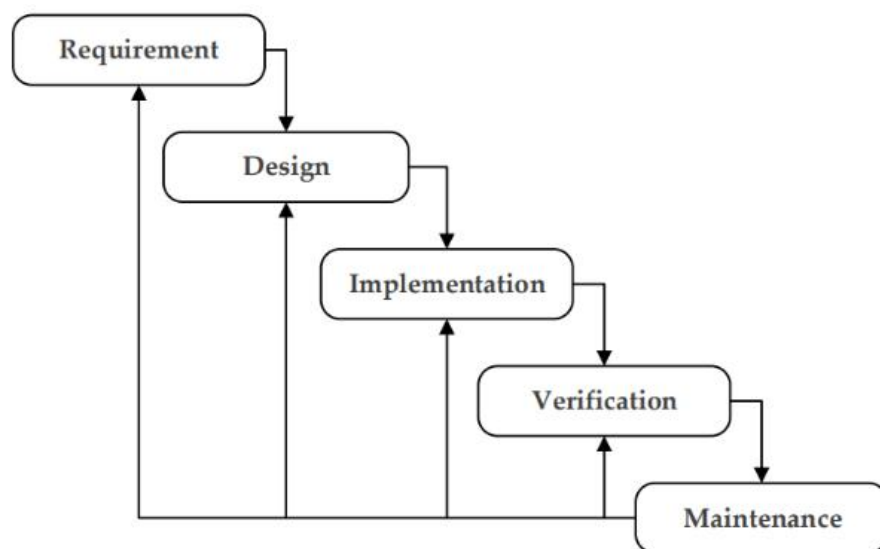
Gambar 2.4 Simbol *Sequence Diagram*

Sumber : Hafsari, Arribe, Andria, & Miransya (2024)

2.1.8 Metode *Waterfall*

Metode *Waterfall* adalah salah satu model peningkatan *software* untuk diterapkan pada proses pengembangan sistem informasi. Kerangka ini dipilih karena menekankan pendekatan terstruktur serta dijalankan secara berurutan, dengan penyelesaian setiap tahap sebelum melanjutkan ke tahapan berikutnya.

Kelebihan metode ini terletak pada dokumentasi yang lengkap dan alur kerja yang jelas, sehingga cocok diterapkan pada proyek-proyek dengan kebutuhan yang stabil dan tidak banyak berubah. Model *waterfall* ini menyediakan pendekatan alur hidup perangkat lunak secara sekuensial terurut dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung.



Gambar 2.5 Tahapan Metode *Waterfall*

Sumber : Hafsari, Arribe, Andria, & Miransya (2024)

a. Analisis Kebutuhan

Tahap awal dilakukan analisis *requirement* sistem atas dasar informasi hasil analisis, ditetapkan kebutuhan sistem *monitoring* sasaran menggunakan platform website dengan PHP dan MySQL sebagai solusi permasalahan pencatatan konvensional.

b. Perancangan Sistem

Pada fase desain, sistem *monitoring* sasaran dirancang melalui beberapa langkah, yaitu pembuatan diagram use case dimanfaatkan guna memodelkan hubungan interaksi aktor dan sistem, sedangkan *Activity Diagram* disusun untuk memodelkan alur aktivitas, perancangan basis data sebagai pengelolaan data sasaran, serta perancangan antarmuka (*interface*) meliputi tampilan *input* dan *output* yang *user-friendly*.

c. Implementasi

Tahap implementasi dilakukan dengan membangun pengembangan sistem aplikasi berbasis web memanfaatkan PHP digunakan sebagai bahasa pemrograman serta MySQL dimanfaatkan sebagai basis data untuk sistem sesuai dengan hasil analisis dan perancangan sebelumnya.

d. Pengujian

Setelah implementasi, sistem dilakukan pengujian guna memastikan kesesuaian sesuai kebutuhan dan desain. Pengujian dilakukan untuk mengidentifikasi kesalahan serta mengevaluasi untuk memastikan sistem beroperasi dengan efektif sesuai spesifikasi yang diharapkan (Supiyandi et al., 2022).

2.1.9 Laravel

Laravel merupakan *framework* php yang memiliki desain sederhana dan fleksibel. Dan seperti pada *framework* lainnya. Laravel memiliki konsep MVC (*Model View Controller*) dengan command line tool yang disebut ‘Artisan’. Laravel bersifat *open source* yang biasanya digunakan untuk perancangan aplikasi web dengan berbagai fitur yang mempermudah developer dalam menulis kode program lebih singkat dan mudah dipahami. Selain itu Laravel memiliki beberapa keunggulan seperti fitur routing, query builder, dan ORM (Febriyanti et al., 2025).

2.1.10 PHP

PHP merupakan Bahasa pemrograman *server-side* yang banyak digunakan untuk mengembangkan aplikasi web dinamis. Dengan keunggulan seperti integrasi yang mudah dengan berbagai database, dukungan komunitas yang luas, serta sumber daya pengembang yang melimpah, PHP menjadi pilihan ideal dalam pengembangan sistem informasi (Zainudin et al., 2024)

2.1.11 MySQL

MySQL adalah sebuah database atau media penyimpanan data yang mendukung script PHP. MySQL juga mempunyai query atau Bahasa SQL (*Structured Query Language*) yang simple dan menggunakan escape character yang sama dengan PHP, selain itu MySQL adalah database tercepat saat ini. (Setiawan, 2024).

2.1.12 XAMPP

XAMPP merupakan paket PHP berbasis *open source* yang dikembangkan oleh sebuah komunitas aplikasi *Open Source*. Dengan menggunakan XAMPP kita tidak perlu lagi melakukan penginstalan program yang lain karena semua kebutuhan telah disediakan oleh XAMPP. Beberapa paket yang telah disediakan adalah Apache, MySql, PHP, Filezilla, dan PhpMyAdmin (Setiawan, 2024).

2.1.13 Visual Studio Code

Visual studio code adalah kode editor yang dikembangkan oleh Microsoft untuk windows, Linux dan mac OS. Visual studio code memudahkan dalam penulisan code yang mendukung beberapa jenis Bahasa pemrograman yang digunakan dan memberi variasi warna sesuai dengan fungsi dalam rangkaian code tersebut. Selain itu, fitur lainnya adalah kemampuan untuk menambah ekstensi dimana para pengembang dapat menambah ekstensi untuk menambah fitur yang tidak ada di visual studio code (Firnando et al., 2023).

2.1.14 *Black Box Testing*

Pengujian *Black Box Testing* pada penelitian ini menggunakan teknik *Functional Testing*, yaitu pengujian yang dilakukan dengan memeriksa fungsi sistem berdasarkan skenario penggunaan tanpa memperhatikan struktur kode.

Pengujian Black Box dapat diartikan sebagai pendekatan uji fungsionalitas atau struktural. Pada pengujian black box tidak memerlukan pengetahuan tentang logika internal sistem yang digunakan untuk mengembangkan kasus pengujian. Sebagai contoh pengujian *black box* adalah menguji fungsi tombol, tes fungsional akan memvalidasi harapan dan fungsi tombol tersebut dengan luaran yang dihasilkan pada saat menekan tombol tersebut. Kegunaan dilakukannya pengujian fungsional antara lain untuk mensimulasikan penggunaan sistem yang sebenarnya dan tidak membuat asumsi struktur sistem. Kelemahan pengujian fungsional adalah adanya potensi kesalahan logis di dalam perangkat lunak yang dikembangkan (Kustiawan et al., 2022).

Dalam penelitian ini, teknik *Black Box Testing* yang digunakan adalah *Functional Testing* (Pengujian testing). Teknik ini dilakukan dengan menguji setiap fungsi sistem berdasarkan skenario penggunaan tanpa melihat struktur kode program secara internal. Pengujian difokuskan pada kesesuaian antara data masukan (*input*) yang diberikan dengan keluaran (*output*) yang dihasilkan sistem. Oleh karena itu, penyusunan skenario uji pada Bab IV dilakukan dalam bentuk tabel yang memuat fitur yang diuji, skenario pengujian, data masukan, hasil yang diharapkan, hasil aktual, dan status pengujian untuk memastikan setiap fungsi sistem berjalan sesuai kebutuhan pengguna.

2.1.15 *User Acceptance Testing (UAT)*

User Acceptance Testing (UAT) merupakan salah satu metode pengujian *alpha* yang dilakukan oleh *end-user* dimana *user* tersebut adalah staff/karyawan perusahaan yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan/fungsinya (Ariandi et al., 2021). Setelah dilakukan sistem testing, acceptance testing menyatakan

bahwa sistem telah memenuhi persyaratan. Meskipun pengguna terlibat aktif dalam pengujian ini, pengembang yang relevan terus memberikan bantuan penuh. Setelah itu, kriteria persetujuan akan dibuat oleh tim pengujian dan pengguna. (Hartono et al., 2025) Dalam penelitian ini, pengujian *User Acceptance Testing* (UAT) dilakukan menggunakan instrumen kuesioner dengan Skala Likert untuk mengukur tingkat penerimaan pengguna terhadap sistem yang telah dikembangkan.

2.2 Perbandingan Metodologi Pengembangan Sistem

Metodologi adalah fondasi utama yang sangat krusial dalam merancang dan mengembangkan perangkat lunak profesional. Memilih metodologi yang tepat bertujuan untuk menghasilkan sistem informasi yang dapat menjawab kebutuhan bisnis secara efektif. Kesuksesan proyek pengembangan perangkat lunak sangat tergantung pada manajemen proyek secara menyeluruh, termasuk penentuan metodologi yang mencakup dinamika tahapan perancangan model yang menggambarkan aktivitas dan siklus hidup sistem (Juli et al., 2024). Pada sub bab ini akan dijelaskan analisis perbandingan tiga metode pengembangan perangkat lunak yang umum digunakan, yaitu *Waterfall*, *Prototype*, dan *Rapid Application Development* (RAD).

2.2.1 Metode *Waterfall*

Metode air terjun atau yang disebut metode *waterfall* sering dinamakan siklus hidup klasik (*Classic life cycle*), nama model ini sebenarnya adalah “Linear Sequential Model” Dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), pemodelan (*modelling*), konstruksi (*contruction*), serta penyerahan sistem ke para pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan. (Juli et al., 2024)

2.2.2 Metode *Prototype*

Model *Prototype* merupakan suatu Teknik untuk mengumpulkan informasi tertentu mengenai kebutuhan-kebutuhan informasi pengguna secara cepat. Berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan/pemakai dan dipakai untuk menyaring kebutuhan pengembangan perangkat lunak. *Prototype* didefinisikan sebagai alat yang memberikan ide bagi pembuat maupun pemakai potensial tentang cara sistem berfungsi dalam bentuk lengkapnya dan proses untuk menghasilkan sebuah *Prototype* disebut prototyping. (Titania Pricillia, 2021)

2.2.3 Metode *Rapid Application Development (RAD)*

Rapid Application Development (RAD) adalah model proses pengembangan perangkat lunak yang bersifat inkremental terutama untuk waktu pengerjaan yang pendek. Model RAD adalah adaptasi dari air terjun versi kecepatan tinggi dengan menggunakan model air terjun untuk pengembangan setiap komponen perangkat lunak. (Deni Murdiani, 2022)

2.2.4 Analisis Perbandingan Metode Pengembangan Sistem

Berdasarkan kajian terhadap metode *waterfall*, *Prototype*, dan RAD, masing-masing metode memiliki karakteristik kelebihan, dan kekurangan yang berbeda. Metode *waterfall* menerapkan tahapan pengembangan yang dilakukan secara sistematis dan berurutan mulai dari analisis kebutuhan, perancangan, implementasi, hingga pengujian. Metode ini sangat sesuai digunakan apabila kebutuhan sistem telah diketahui dengan jelas sejak awal sehingga kebutuhan selama proses pengembangan relative kecil.

Sementara itu, metode *Prototype* lebih berfokus pada pembuatan model awal sistem yang digunakan untuk memperoleh umpan balik dari pengguna. Metode ini cocok diterapkan pada kondisi ketika kebutuhan pengguna belum terdefinisi secara rinci dan masih memungkinkan terjadinya perubahan selama proses pengembangan. Namun ketergantungan terhadap umpan balik pengguna

dapat menyebabkan perubahan berulang yang berpotensi memperpanjang waktu pengembangan.

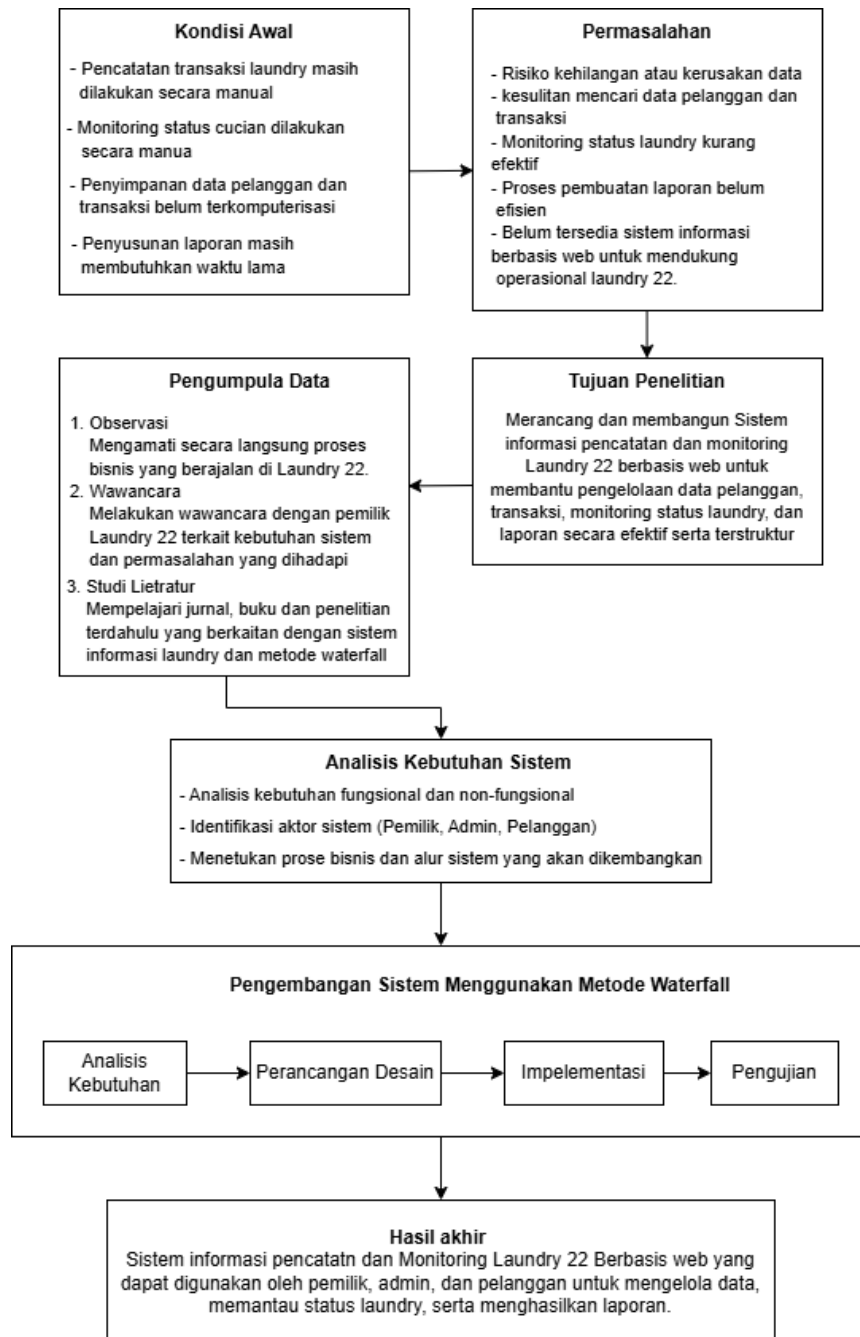
Adapun metode *Rapid Application Development* (RAD) menekankan kecepatan dalam proses pengembangan melalui pendekatan inkremental dan penggunaan komponen yang dapat digunakan kembali. Metode ini efektif digunakan pada proyek yang memiliki batasan waktu yang ketat dan didukung oleh tim pengembang yang berpengalaman. Tapi metode RAD membutuhkan keterlibatan pengguna yang intensif dan sumber daya yang memadai selama proses pengembangan.

Metode *waterfall* pada penelitian ini dipilih karena dinilai paling sesuai dengan kondisi objek penelitian, yaitu Laundry 22. Berdasarkan observasi dan wawancara, kebutuhan sistem yang akan dikembangkan telah dapat diidentifikasi dengan cukup jelas sehingga tidak memerlukan perubahan kebutuhan secara berulang seperti metode *Prototype*. Selain itu, ruang lingkup sistem yang dikembangkan relatif sederhana dan tidak menuntut penyelesaian dalam waktu yang sangat singkat seperti metode RAD. Oleh karena itu, metode *waterfall* dipilih karena memiliki tahapan yang terstruktur, mudah dipahami, terdokumentasi dengan baik, serta sesuai untuk pengembangan sistem informasi laundry berbasis web pada penelitian ini.

2.3 Kerangka Berpikir

Kerangka berpikir penelitian ini diawali dengan identifikasi masalah yang terjadi pada proses pencatatan dan *monitoring* Laundry 22. Selanjutnya dilakukan analisis kebutuhan sistem dan studi literatur sebagai dasar dalam memahami permasalahan serta menentukan solusi yang sesuai. Berdasarkan hasil analisis tersebut, diusulkan pengembangan sistem informasi berbasis web dengan menggunakan metode *waterfall*. Metode ini terdiri dari tahapan analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Melalui tahapan tersebut diharapkan dapat menghasilkan sistem informasi pencatatan dan *monitoring* Laundry 22 berbasis web yang mampu mendukung pengelolaan data transaksi,

monitoring status laundry serta penyajian laporan secara lebih efektif dan terstruktur. Alur kerangka berpikir ini dapat dilihat pada Gambar 2.6.



Gambar 2.6 Kerangka Berpikir