

## **BAB II. TINJAUAN PUSTAKA**

### **2.1 Sistem**

Sistem merupakan bagian penting dalam mendukung jalannya proses dan pengelolaan informasi. Sistem diartikan sebagai sekumpulan komponen saling berhubungan dan bekerja sama untuk mencapai sebuah tujuan (Candra 2021). Dalam teknologi informasi, sistem berperan sebagai penghubung antara pengguna, komponen perangkat keras, sistem aplikasi, serta tata kerja agar seluruh proses berjalan terstruktur (Efendi et al. 2023). Lebih lanjut, sistem juga membentuk kerangka kerja yang mendukung proses pengambilan keputusan serta pengelolaan informasi dalam organisasi (Manurian et al. 2020).

Sebuah sistem terdiri atas beberapa bagian yang dirancang untuk bekerja sama dalam hal mencapai hasil yang diinginkan. Berdasarkan sifatnya, sistem dikategorikan menjadi dua jenis, yaitu sistem fisik yang menitik beratkan pada struktur, dan sistem dinamis yang berfokus pada proses (Rochman et al. 2019). Karakteristik sistem secara umum meliputi komponen, batasan, masukan, serta tujuan (Purnama et al. 2020). Sistem juga dapat dipahami sebagai kumpulan komponen yang saling berhubungan dan membentuk satu kesatuan utuh yang dirancang secara efisien (To Suli 2023).

Berdasarkan uraian tersebut, sistem dapat dipahami sebagai kesatuan terstruktur yang terdiri atas elemen-elemen saling berinteraksi untuk mencapai tujuan tertentu, sehingga penerapannya diarahkan untuk meningkatkan efektivitas dan efisiensi proses kerja.

### **2.2 Sistem Informasi**

Sistem informasi adalah sistem yang digunakan organisasi untuk mengolah data menjadi informasi yang bermanfaat. Sistem ini menggabungkan teknologi, prosedur, dan peran manusia sehingga dapat membantu kegiatan operasional, pengendalian, dan pengambilan keputusan berjalan lebih efisien (Simare Mare et al. 2022).

Secara umum, sistem informasi merupakan rangkaian prosedur yang dirancang untuk mengumpulkan, mengolah, menyimpan, dan menyajikan data agar dapat digunakan oleh berbagai pihak dalam organisasi (Hendrik Sitorus and Sakban 2021). Komponen utamanya meliputi perangkat keras, perangkat lunak, basis data, dan manusia yang saling terhubung dalam satu sistem kerja.

Struktur sistem informasi mencakup enam bagian utama, yaitu masukan, proses, keluaran, teknologi, basis data, dan kendali, yang saling melengkapi untuk menghasilkan informasi yang relevan dan berkualitas bagi pengguna (Manurian et al. 2020).

Penerapan sistem informasi dapat ditemui di berbagai bidang, seperti penjualan, pendidikan, keuangan, dan layanan publik. Walaupun berbeda konteks, tujuannya sama, yaitu memudahkan akses dan pengelolaan informasi yang dibutuhkan dalam pengambilan keputusan (Hendrik Sitorus and Sakban 2021).

Dengan perancangan yang optimal, sistem informasi berperan tidak hanya sebagai alat pengolah data, tetapi juga sebagai sarana yang memastikan informasi yang dihasilkan memiliki tingkat keakuratan tinggi, relevan dengan kebutuhan pengguna, serta tersaji tepat waktu. Fungsi tersebut menempatkan sistem informasi sebagai elemen penting dalam mendukung strategi organisasi dan proses pengambilan keputusan yang efektif.

### **2.3 Sistem Layanan Laboratorium Lingkungan**

Sistem layanan laboratorium lingkungan merupakan perangkat lunak yang dirancang untuk mengelola alur kerja, mulai dari pengumpulan, pengolahan, hingga pelaporan data uji lingkungan. Dalam literatur internasional, sistem ini dikenal sebagai *Laboratory Information Management System* (LIMS), yang dikembangkan untuk meningkatkan kualitas pengelolaan data laboratorium (Liu et al. 2023).

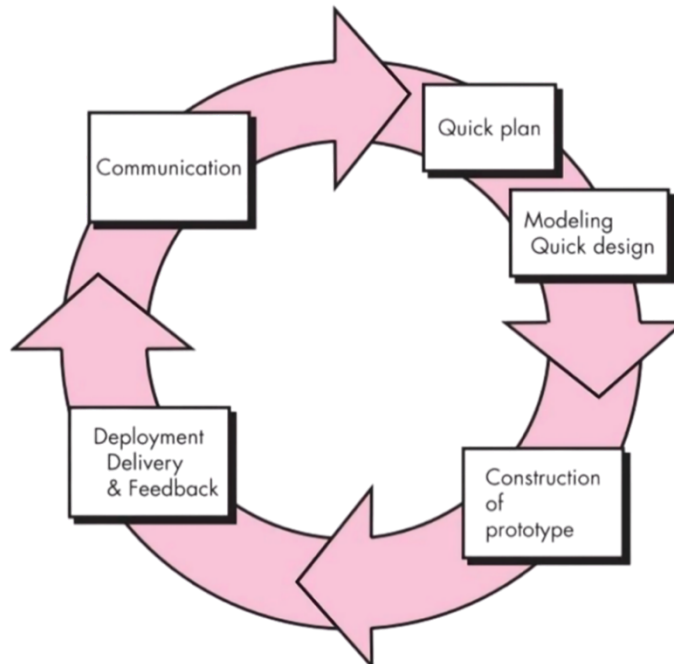
LIMS didefinisikan sebagai sistem yang mengatur data laboratorium secara terstruktur, mencakup pengelolaan sampel, alur kerja, peralatan, serta pencatatan hasil uji secara digital dan terintegrasi (Oluwole et al. 2022). Dalam laboratorium lingkungan, LIMS mencatat metadata sampel seperti waktu, lokasi pengambilan, serta parameter yang diuji, sekaligus memastikan kontrol kualitas berjalan otomatis sesuai standar modern. Sebagai contoh, pada pemantauan kualitas air, LIMS dapat mengotomasi pengumpulan hingga 150 parameter sekaligus, sehingga hasil uji lebih akurat dan risiko kesalahan input manual berkurang (Liu et al. 2023).

LIMS tidak hanya berfungsi sebagai alat pencatat data, tetapi juga menjamin transparansi dan kesesuaian dengan standar laboratorium. Dengan begitu, laboratorium lingkungan dapat memberikan hasil uji yang lebih valid dan dapat dipercaya, sekaligus meningkatkan kepercayaan pengguna.

## 2.4 Prototyping

Metode *prototyping* merupakan pendekatan dalam pengembangan perangkat lunak yang berfokus pada pembuatan rancangan awal sistem secara cepat untuk menggambarkan fungsionalitas utama yang akan dikembangkan. Tujuan utamanya adalah memudahkan komunikasi antara pengembang dan pengguna, terutama ketika kebutuhan sulit dijelaskan secara detail sejak awal (Meisak et al. 2022).

Prototyping dilakukan secara berulang dalam tiga tahap utama: desain, pembuatan model, dan evaluasi. Pada tahap desain, dibuat rancangan awal antarmuka dan fungsi utama program. Selanjutnya, model dikembangkan dan diuji oleh pengguna untuk memperoleh masukan, yang kemudian digunakan dalam penyempurnaan sistem. Melalui pendekatan ini, pengembangan dilakukan dengan lebih cepat, fleksibel, dan secara bertahap disesuaikan dengan kebutuhan pengguna (Putri et al. 2020).



**Gambar 2.1** Metode Prototyping

Metode prototyping yang dikembangkan oleh Pressman terdiri dari beberapa tahap, yaitu:

1. *Communication*, tahapan awal ini bertujuan untuk mengetahui kebutuhan pengguna dengan melakukan kegiatan observasi dan wawancara. Informasi yang terkumpul menjadi dasar perancangan sistem, misalnya terkait pencatatan sampel, proses

- pengujian laboratorium, dan pembuatan laporan (Kurniawan et al. 2024).
2. *Quick Plan, Modelling, and Quick Design*, setelah kebutuhan diperoleh, dilakukan perencanaan cepat dan perancangan sistem secara visual. Alur sistem digambarkan dengan diagram seperti flowchart, use case, serta desain antarmuka awal. Desain ini memberi gambaran nyata kepada pengguna, bahkan dapat dibuat dengan alat bantu seperti Figma (Kurniawan et al. 2024).
  3. *Construction of Prototype*, pada tahap ini, prototipe sistem mulai dibangun dengan mengimplementasikan fitur inti terlebih dahulu. Pembuatan dilakukan secara cepat menggunakan teknologi antarmuka dan fungsi dasar sistem. Tujuannya adalah agar pengguna dapat mencoba langsung sistem dan memberikan masukan yang relevan (Kurniawan et al. 2024).
  4. *Deployment, Delivery, and Feedback*, setelah prototipe awal diuji, sistem dikembangkan lebih lanjut hingga menjadi versi fungsional penuh. Proses ini mencakup penerapan basis data, pemrograman backend, dan integrasi antarmuka. Sistem kemudian diuji kembali, dan umpan balik dari pengguna dijadikan dasar penyempurnaan hingga sistem memenuhi kebutuhan dan siap digunakan (Kurniawan et al. 2024).

Dengan metode prototyping, pengembangan dilakukan secara bertahap melalui komunikasi, perancangan cepat, pembuatan model, dan evaluasi berulang. Pendekatan ini memungkinkan keterlibatan langsung pengguna, sehingga masukan dapat segera direspons. Dengan demikian, sistem yang dihasilkan lebih adaptif, sesuai kebutuhan nyata, dan tepat sasaran.

## 2.5 Alat Bantu Perancangan Sistem

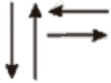








### 2.5.1 Flowchart








Menentukan alur kerja yang jelas adalah bagian penting dalam membangun sistem, karena membantu menunjukkan cara kerja proses secara teratur. Salah satu metode yang dapat digunakan untuk memvisualisasikan alur tersebut ialah *flowchart*, diagram yang menggambarkan tahapan dalam suatu sistem secara berurutan. Flowchart menggambarkan alur proses dari awal sampai akhir dengan simbol-simbol yang telah ditetapkan, serta menunjukkan keterkaitan antara aktivitas, input, output, dan penyimpanan data yang dipakai (Zalukhu et al. 2023).

Flowchart berguna dalam menganalisis, merancang, dan menerapkan program karena memungkinkan masalah-masalah dipecah menjadi bagian-bagian yang lebih kecil dan lebih mudah dipahami (Sutanti et al. 2020). Selain itu, flowchart memberikan gambaran yang rapi tentang urutan aktivitas operasional, sehingga membantu menjelaskan cara kerja sistem dan memudahkan proses pengecekan kembali (Listyoningrum et al. 2023). Biasanya, flowchart disusun dari atas ke bawah atau dari kiri ke kanan, dengan langkah-langkah yang jelas dan teratur (Febryana et al. 2024).

Oleh karena itu, flowchart menjadi alat penting dalam merancang sistem informasi karena menampilkan informasi secara visual, sederhana, dan terorganisir, sehingga memudahkan pemahaman terhadap proses yang sedang dikembangkan.

**Tabel 2.1** Tabel Flowchart

Simbol	Nama Simbol	Keterangan
	<i>Flow Direction symbol</i>	Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.
	<i>Terminator Symbol</i>	Yaitu simbol untuk permulaan ( <i>start</i> ) atau akhir ( <i>stop</i> ) dari suatu kegiatan.
	<i>Connector Symbol</i>	Yaitu simbol untuk keluar masuk atau penyambungan proses dalam lembar / halaman yang sama.
	<i>Connector Symbol</i>	Yaitu simbol untuk keluar masuk atau penyambungan proses pada lembar / halaman yang berbeda.
	<i>Processing Symbol</i>	Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.
	<i>Symbol Manual Operation</i>	Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer.
	<i>Symbol Decision</i>	Simbol pemilihan proses berdasarkan kondisi yang ada.
	<i>Symbol Input-Output</i>	Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya.
	<i>Symbol Manual Input</i>	Simbol untuk pemasukan data secara manual online keyboard.

Simbol	Nama Simbol	Keterangan
	<i>Simbol Preparation</i>	Simbol untuk mempersiapkan penanganan yang akan digunakan sebagai tempat pengolahan di dalam simbol.
	<i>Simbol Predefine Proses</i>	Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure.
	<i>Simbol Display</i>	Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	<i>Simbol Display and On-line Storage</i>	Simbol yang menyatakan input yang berasal dari pita <i>magnetic</i> atau output disimpan ke pita <i>magnetic</i>
	<i>Simbol magnetic tape Unit</i>	Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	<i>Simbol Punch Card</i>	Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu.
	<i>Simbol Document</i>	Simbol yang menyatakan input atau output dalam bentuk kertas atau output dicetak ke kertas.

## 2.5.2 Data Flow Diagram (DFD)


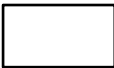


*Data Flow Diagram* (DFD) adalah teknik pemodelan guna memvisualisasikan aliran data yang terjadi di dalam suatu proses pengolahan informasi. Melalui simbol-simbol, DFD memperlihatkan bagaimana data diolah, disimpan, dan berpindah antarbagian dalam sistem. Dengan cara ini, DFD tidak hanya memberikan gambaran visual, tetapi juga mendukung analisis serta perancangan perangkat lunak secara terstruktur.

DFD membantu memahami logika kerja sistem secara lebih jelas sehingga memudahkan identifikasi permasalahan yang perlu diperbaiki (Irfan et al. 2024). Diagram ini juga dapat menggambarkan sistem sebagai jaringan komponen yang saling terhubung, baik secara manual maupun terkomputerisasi. Keunggulan lain dari DFD adalah kemampuannya menampilkan sistem dari tingkat paling umum hingga detail melalui proses dekomposisi (Satyaninggrat et al. 2023). Empat elemen utama menjadi dasar penyusunan DFD, yaitu:

1. Data terminator berbentuk kotak sebagai entitas eksternal yang berinteraksi dengan sistem.
2. Data store berupa dua garis sejajar sebagai tempat penyimpanan data.
3. Data flow berupa panah untuk menunjukkan arah aliran data.
4. Proses berbentuk lingkaran atau oval untuk merepresentasikan aktivitas pengolahan data (Agus Darmadi and Aprillia Darmadi 2020).

Dengan demikian, DFD dapat dipahami sebagai sarana yang membantu menghubungkan pengembang dengan para pemangku kepentingan, serta bertindak sebagai dasar yang kuat dalam merancang sistem informasi yang efisien, terstruktur, dan mudah dinilai selama proses pengembangannya.

**Tabel 2.2** Tabel DFD

Simbol	Nama Simbol	Keterangan
	Aliran Data	Menunjukkan arus data dalam proses
	Kesatuan Luar	Sistem yang dapat berupa orang, organisasi atau lainnya yang berada di luar lingkungan luar yang akan memberikan input atau menerima output sistem
	Proses	Proses pengolahan data yang menunjukkan suatu kegiatan yang mengubah aliran data yang masuk menjadi keluaran
	Penyimpanan Data	Tempat penyimpanan dokumen atau file yang dibutuhkan

### 2.5.3 Entity Relationship Diagram


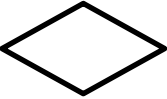
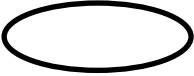

*Entity Relationship Diagram* (ERD) adalah teknik pemodelan yang digunakan dalam proses perancangan sistem informasi untuk menggambarkan struktur serta hubungan antar data (Sutanti et al. 2020). ERD berfungsi membantu analis dan perancang sistem dalam memahami keterkaitan antar data sebelum diimplementasikan ke dalam basis data.

ERD mencakup tiga komponen utama, yaitu entitas yang merepresentasikan objek, atribut yang menjelaskan karakteristik dari

entitas, dan relasi yang menunjukkan keterhubungan antara satu entitas dengan entitas lainnya. Relasi ini bisa berupa hubungan dua entitas, tiga entitas, atau lebih dari tiga entitas (Syaqila et al. 2024).

Secara umum, ERD memudahkan proses membuat database karena bisa mengatur, menyimpan, serta menghubungkan data secara terstruktur. Dengan menggunakan simbol-simbol, ERD memberikan gambaran visual yang jelas, sehingga proses analisis dan penerapan database bisa berjalan lebih efisien.

**Tabel 2.3** Tabel ERD

Simbol	Nama Simbol	Keterangan
	Entitas	Objek yang bisa dikenali atau dibedakan di dalam suatu sistem.
	Relasi	Hubungan yang terjadi antara beberapa entitas.
	Atribut	Ciri atau sifat yang dimiliki oleh entitas; atribut yang menjadi kunci biasanya diberi garis bawah.
	Garis	Penghubung antara entitas, relasi, dan atribut.

#### 2.5.4 Unified Modeling Language

Unified Modeling Language (UML) berperan sebagai media visual untuk menjelaskan rancangan dan cara kerja sistem. UML membantu dalam proses pemodelan, penjabaran, pembangunan, dan pendokumentasian perangkat lunak sehingga komunikasi antara pengembang dan pengguna menjadi lebih efektif (Ramdany et al. 2024).

Dalam proses perancangan sistem terdapat beberapa jenis diagram yang sering digunakan seperti, *Use Case Diagram* untuk menjelaskan bagaimana actor berinteraksi dengan sistem, *Activity Diagram* untuk menampilkan alur aktivitas, *Sequence Diagram* untuk menggambarkan aliran pesan antar objek pertukaran pesan antar objek dan *Class Diagram* untuk memaparkan susunan dan keterkaitan antar kelas (To Suli 2023). Oleh sebab itu, UML memiliki peranan penting sebagai standar dalam pengembangan sistem

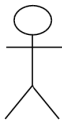
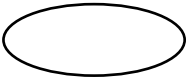
berbasis objek karena mampu memberikan representasi rancangan yang jelas, terstruktur, serta mudah dipahami oleh seluruh pihak yang terlibat.

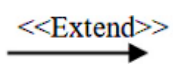
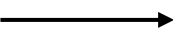

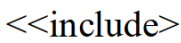
#### 2.5.4.1 Use Case Diagram

*Use Case Diagram* dianggap sebagai alat visual dalam UML yang digunakan untuk menunjukkan cara aktor berinteraksi dengan sistem, sehingga menjelaskan kebutuhan, fungsi, serta batasan dari sistem yang sedang dirancang (Narulita et al. 2024). Diagram ini membantu menjelaskan berbagai pola interaksi pengguna dengan sistem melalui gambar simbol seperti stick-man untuk mewakili aktor dan elips untuk mewakili use case, serta hubungan include dan extend yang menunjukkan keterkaitan antar fungsi (Ramdany et al. 2024). Dengan demikian, *Use Case Diagram* tidak hanya sekadar alat teknis, tetapi juga menjadi sarana komunikasi yang memudahkan pemahaman mengenai kebutuhan sistem (Aditya and Abduh 2022).

Secara keseluruhan, *use case diagram* dapat disimpulkan sebagai representasi visual yang membantu pengembang dan pengguna memahami kebutuhan serta fungsionalitas sistem. Melalui pendekatan ini, batasan sistem terlihat lebih jelas, interaksi antar aktor dapat digambarkan, dan rancangan sistem menjadi lebih sesuai dengan kebutuhan pengguna.

**Tabel 2.4** Simbol Simbol Use Case Diagram

Simbol	Nama Simbol	Keterangan
	<i>Actor</i>	<i>Actor</i> mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan use case.
	<i>Use Case</i>	Use case merupakan abstraksi dan interaksi antara sistem dan aktor.

Simbol	Nama Simbol	Keterangan
	Extend	Ekstend menunjukkan bahwa suatu <i>use case</i> merupakan suatu tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.
	Asosiasi	Asosiasi merupakan abstraksi dari penghubung antara aktor dan <i>use case</i> .
	Generalisasi	Menunjukkan spesialis aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Include	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.

#### 2.5.4.2 Class Diagram


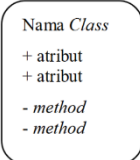
*Class diagram* merupakan salah satu bentuk pemodelan dalam *Unified Modeling Language* (UML) yang berfungsi untuk menampilkan struktur utama suatu sistem. Diagram ini memperlihatkan kelas-kelas penyusun beserta atribut dan fungsinya, sehingga memberikan gambaran umum mengenai komponen penting yang membentuk sistem (Putra and Andriani 2019).

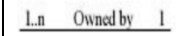

*Class diagram* memperlihatkan hubungan antar kelas beserta aturan dan interaksi antar elemen sistem. Melalui pemodelan ini, pengembang dapat merancang komponen secara lebih terstruktur sesuai kebutuhan fungsional (Sutrisno and Karnadi 2020)(Aldi 2025). Diagram ini juga digunakan untuk menggambarkan

keterkaitan antar kelas dan alur data, sehingga menunjukkan bagaimana sistem bekerja berdasarkan desain yang telah direncanakan (Sandfreni et al. 2021).

Berdasarkan uraian tersebut, class diagram dapat dipahami sebagai model yang menggambarkan struktur sistem sekaligus menjadi penghubung antara tahap perancangan dan implementasi perangkat lunak, sehingga proses pengembangan dapat berlangsung lebih terarah.

**Tabel 2.5** Simbol Class Diagram

Simbol	Nama Simbol	Keterangan
	<i>Dependency</i>	<i>Dependency</i> digunakan untuk menunjukkan suatu operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain.
	<i>Class</i>	<i>Class</i> adalah blok-blok yang membangun pada pemrograman berorientasi objek. <i>Class</i> digambarkan seperti sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah nama dari <i>class</i> . Bagian tengah adalah atribut <i>class</i> . Bagian akhir adalah method-method.

Simbol	Nama Simbol	Keterangan
	Association	Asosiasi adalah sebuah <i>relationship</i> paling umum antara 2 <i>class</i> dan dapat dilambangkan dengan sebuah garis yang menghubungkan antara 2 <i>class</i> . Garis ini juga bisa melambangkan tipe-tipe <i>relationship</i> dan menampilkan hukum-hukum multiplisitas pada <i>relationship</i> .
	Composition	Menunjukkan bahwa suatu <i>class</i> tidak dapat berdiri sendiri, melainkan bergantung pada <i>class</i> lain melalui relasi Composition.

#### 2.5.4.3 Activity Diagram



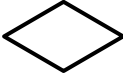

*Activity Diagram* merupakan jenis diagram dalam *Unified Modeling Language* (UML) yang berfungsi untuk menggambarkan alur atau tahapan suatu proses maupun aktivitas yang terjadi di dalam sistem. Diagram ini menggambarkan langkah-langkah proses dari awal hingga akhir, termasuk keputusan atau percabangan yang mungkin terjadi di dalamnya (Wayahdi and Ruziq 2023).

*Activity diagram* menampilkan urutan tindakan atau aktivitas yang saling terkait, baik secara satu per satu maupun secara bersamaan. Diagram ini bisa digunakan dalam berbagai tahap perancangan, mulai dari pemodelan proses bisnis hingga perancangan algoritma yang lebih teknis (Ramdany et al. 2024). *Activity diagram* menampilkan aliran yang terstruktur, di mana suatu aktivitas dapat mencakup subaktivitas dan objek yang saling berhubungan. Diagram ini memperlihatkan aliran kerja (*workflow*) secara

menyeluruh dengan menekankan pada hubungan antara input, proses, output, serta mekanisme perpindahan antaraktivitas dalam sistem (Aldi 2025).

Dengan demikian, *activity diagram* adalah bentuk grafis yang menampilkan aliran dalam sistem, yang berfokus pada urutan tindakan, keputusan yang terjadi, dan proses yang berlangsung secara bersamaan. Diagram ini berperan penting dalam membantu para pengembang dan analis memahami bagaimana sistem beroperasi secara terstruktur.

**Tabel 2.6** Simbol Activity Diagram

Simbol	Nama Simbol	Keterangan
	Status Awal	Menunjukkan status awal aktivitas sistem dalam diagram aktivitas.
	Aktivitas	Aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja.
	Percabangan	Menunjukkan percabangan jika ada lebih dari satu pilihan aktivitas.
	Status Akhir	Menunjukkan status akhir dari aktivitas sistem.

## 2.6 Teknologi Pengembang Sistem

### 2.6.1 PHP

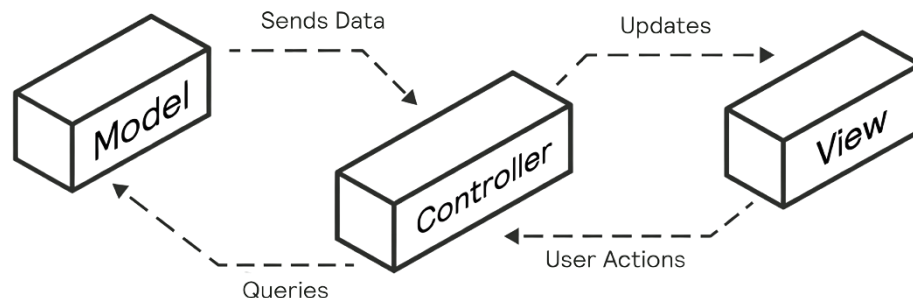
PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman dinamis yang digunakan untuk membangun dan menjalankan aplikasi web pada sisi server (*server-side scripting*) dan dapat diintegrasikan langsung ke dalam kode *Hyper Text Markup Language* (HTML). Bahasa ini bersifat *open source*, mudah dipelajari, serta membutuhkan web server untuk menampilkan

hasilnya ke browser pengguna. PHP memungkinkan pembuatan halaman web dinamis, pengolahan data secara real-time, serta interaksi langsung antara sistem dan pengguna (Kansha 2023). Selain itu, PHP memiliki kemampuan untuk terintegrasi dengan beragam sistem basis data, sehingga menjadikannya salah satu bahasa pemrograman yang banyak dimanfaatkan dalam pembuatan aplikasi dan sistem informasi berbasis web (Utami 2022).

Dibandingkan dengan bahasa skrip lain seperti *JavaScript* dan *Python*, PHP lebih umum digunakan untuk pemrosesan sisi server, sedangkan JavaScript lebih banyak digunakan pada bagian frontend maupun backend.

Fleksibilitas dan kemudahan penggunaannya membuat PHP cocok untuk membangun berbagai aplikasi web dinamis, mulai dari sistem layanan daring hingga e-commerce dan sistem informasi organisasi (To Suli 2023). Karakteristik ini menjadikan PHP relevan bagi pengembang yang ingin membuat aplikasi web interaktif dan mudah dikembangkan lebih lanjut.

### 2.6.2 Framework Laravel



**Gambar 2.2** Laravel

Laravel merupakan framework PHP yang digunakan untuk pengembangan aplikasi web dengan cara yang lebih terstruktur dan efisien. Laravel bersifat *open source* dan dibuat untuk membantu pengembang merancang aplikasi web dengan lebih mudah. Framework ini diperkenalkan oleh Taylor Otwell pada tahun 2011 hingga saat ini, tetap menjadi salah satu *framework* PHP yang banyak digunakan karena sintaksnya sederhana dan kelengkapan fiturnya (Sir and Efmi Maiyana 2025). Pada pengembangan *backend*, *Laravel* memberi pendekatan yang mendukung pembuatan sistem yang tertata, mudah dikelola, dan efektif (Fauziah et al. 2022).

Secara teknis, Laravel menerapkan pola arsitektur *Model View Controller* (MVC). Pendekatan ini dibagi menjadi tiga bagian utama: *Model* yang mengelola data, *View* yang menampilkan antarmuka kepada pengguna, dan *Controller* sebagai penghubung antara *Model* dan *View*. (Sir and Efmi Maiyana 2025).

Laravel juga menyediakan berbagai fitur canggih seperti *Eloquent Object-Relational Mapper* (ORM) yang memungkinkan pengelolaan database tanpa harus menulis perintah SQL secara manual (Fauziah et al. 2022). Selain itu, *Laravel* menyertakan *Artisan Console* untuk menjalankan tugas pengembangan secara otomatis, sistem routing yang fleksibel, dan fitur keamanan bawaan seperti perlindungan dari serangan *Cross-Site Request Forgery* (CSRF), enkripsi data, serta sistem autentikasi pengguna (Herson, Utomo, & Sari, 2025). Popularitas *Laravel* terus meningkat karena kemudahan penggunaan, dokumentasi yang lengkap, serta dukungan komunitas yang aktif, baik untuk para pemula maupun profesional (Fauziah et al. 2022).

### 2.6.3 XAMPP

XAMPP adalah perangkat lunak gratis dan *open-source* yang menyediakan paket server lengkap untuk pengembangan aplikasi web. Perangkat lunak ini mencakup komponen utama seperti Apache Web Server sebagai server, *MariaDB* (sebelumnya MySQL) sebagai basis data, PHP sebagai bahasa pemrograman server-side, serta Perl untuk kebutuhan pemrograman yang lebih kompleks. XAMPP dapat dijalankan di berbagai sistem operasi seperti *Windows*, *Linux*, dan *macOS*, sehingga memudahkan pengembangan web lintas platform (Putra and Nita 2019). Dengan instalasi yang sederhana, pengguna langsung mendapatkan server *Apache*, basis data *MariaDB* atau *MySQL*, serta dukungan penuh untuk PHP. Kemudahan konfigurasi, sifat *open-source*, serta ketersediaan berbagai modul menjadikan XAMPP pilihan populer untuk membangun dan menguji aplikasi web secara lokal sebelum dipublikasikan ke server produksi (Putra and Nita 2019).

### 2.6.4 MySQL

MySQL adalah perangkat lunak Sistem Manajemen Basis Data (DBMS) berbasis *open-source* yang secara luas digunakan untuk mengelola dan menyimpan data dalam aplikasi web. Selain itu, MySQL bukan hanya sekadar sistem basis data relasional (RDBMS), karena data disimpan dalam tabel yang saling terhubung melalui baris dan kolom. Struktur ini memungkinkan pengaturan data secara

logis dan akses yang efisien melalui bahasa perintah SQL (*Structured Query Language*). Keunggulan seperti kemudahan penggunaan, performa yang cepat, serta fleksibilitas menjadikan MySQL sebagai pilihan populer untuk berbagai jenis aplikasi, mulai dari situs web kecil hingga sistem perusahaan skala besar (Sir and Efmi Maiyana 2025).

## 2.7 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses sistematis yang bertujuan memastikan bahwa perangkat lunak yang dikembangkan beroperasi sesuai dengan kebutuhan dan fungsionalitas yang telah ditentukan sebelumnya. Dalam pengujian ini, berbagai kekurangan atau kesalahan dalam sistem dapat terdeteksi sejak dini, sehingga memungkinkan perbaikan sebelum perangkat lunak diimplementasikan. Pengujian merupakan bagian integral dari siklus hidup pengembangan perangkat lunak, bersama dengan analisis, desain, dan pengkodean, karena berperan penting dalam menjaga kualitas serta keandalan aplikasi yang dibuat (Azhari et al. 2023).

Dalam praktiknya, pengujian perangkat lunak dapat dilakukan dengan berbagai pendekatan, seperti *White Box Testing* yang memeriksa detail struktur internal dan alur logika program secara prosedural, serta *Black Box Testing* yang fokus pada spesifikasi fungsional perangkat lunak tanpa memperhatikan struktur internal kode. Dalam penelitian ini, metode *Black Box Testing* digunakan karena mampu memvalidasi setiap fungsi dalam perangkat lunak apakah sudah berjalan sesuai dengan kebutuhan dan desain yang telah ditetapkan, sekaligus membantu dalam menemukan kesalahan tanpa perlu memeriksa kode secara langsung (Salsabila et al. 2022).

### 2.7.1 Black Box Testing

*Black Box Testing* merupakan metode pengujian perangkat lunak yang fokus pada verifikasi hasil eksekusi aplikasi berdasarkan input yang diberikan, dengan tujuan memastikan fungsi sistem berjalan sesuai dengan persyaratan yang telah ditentukan. Metode ini tidak hanya mencakup orang, tetapi juga objek dan benda-benda alam serta tidak memerlukan analisis source code program. Populasi yang diuji mencakup fungsi-fungsi aplikasi, bukan hanya jumlah yang terlihat (Mintarsih 2023).

Dalam penerapannya, *Black Box Testing* dilakukan dengan menyusun *test case* yang mengecek setiap fungsi aplikasi, mengevaluasi kesesuaian alur kerja dengan kebutuhan pengguna, serta mendeteksi kesalahan atau bug yang terlihat pada antarmuka.

Pendekatan ini berperan dalam meningkatkan kualitas perangkat lunak dengan menemukan kesalahan yang belum terdeteksi selama tahap pengembangan, sehingga sistem menjadi lebih andal saat digunakan (Mintarsih 2023).

## 2.8 Peneliti Terdahulu

**Tabel 2.7** Penelitian Terdahulu

No	Nama Peneliti (Tahun)	Judul Penelitian	Hasil Penelitian	Hasil Pembahasan
1.	(Yasir 2025)	Rancang Bangun Aplikasi Sistem Pendaftaran Uji Sampel pada Laboratorium Dinas Lingkungan Hidup Kabupaten Luwu Berbasis Website	Aplikasi berhasil dibangun menggunakan <i>Laravel</i> , PHP, <i>MySQL</i> dengan metode waterfall. Pengujian <i>Black Box</i> menunjukkan semua fitur berjalan sesuai, validasi ahli memperoleh skor rata-rata 3,98 (Sangat Layak), dan uji SUS menghasilkan skor 94,3 (Best Imaginable).	Sistem mampu mengatasi kelemahan layanan manual, menyediakan pendaftaran online, pelacakan status, pengunduhan hasil, serta manajemen data yang lebih efisien. Namun, sistem belum mendukung pembayaran digital, sehingga disarankan adanya integrasi pembayaran otomatis untuk pengembangan selanjutnya.
2.	(Wardiah et al. 2024)	Implementasi Sistem Informasi Pengelolaan Laboratorium untuk Meningkatkan	Sistem informasi laboratorium berhasil dikembangkan dan dihosting menggunakan metode <i>waterfall</i> ,	Sistem mampu meningkatkan efisiensi administrasi pengujian, pengelolaan alat dan bahan,

No	Nama Peneliti (Tahun)	Judul Penelitian	Hasil Penelitian	Hasil Pembahasan
		n Efisiensi Kinerja UPTD Laboratorium Lingkungan	diuji dengan <i>black box</i> (semua fitur berjalan baik) dan usability testing dengan skor 81,25% (kategori sangat bagus).	serta mendukung konsep paperless. Pendampingan pengguna memastikan staf UPTD dapat mengoperasikan sistem dengan baik.
3.	(Modestus La'a and Vip Pramarta 2023)	Pengembangan Sistem Informasi Laboratorium untuk Meningkatkan Efisiensi dan Akurasi	Sistem dikembangkan melalui analisis kebutuhan, desain, pengembangan, implementasi, dan evaluasi. Menggunakan teknologi seperti RFID untuk meningkatkan akurasi data dan integrasi antar modul. Hasil evaluasi menunjukkan peningkatan efisiensi, pengurangan kesalahan manual, serta laporan yang lebih cepat dan akurat.	Sistem informasi laboratorium terintegrasi dapat mengurangi pekerjaan manual, meningkatkan akurasi, memperlancar aliran informasi, serta meningkatkan produktivitas dan kualitas layanan laboratorium.
4.	(Ahmadiyah 2015)	Rancang Bangun Sistem Layanan Laboratorium Lingkungan	Sistem layanan laboratorium berbasis web berhasil dibangun menggunakan Laravel dengan	Sistem mampu mengintegrasikan seluruh alur layanan pengujian sampel secara

No	Nama Peneliti (Tahun)	Judul Penelitian	Hasil Penelitian	Hasil Pembahasan
		pada UPT DLH Provinsi Jawa Timur	arsitektur MVC. Fitur utama meliputi pendaftaran sampel, pembagian kerja analisis otomatis, validasi berlapis, cetak sertifikat, serta laporan harian/bulanan/tahunan. Pengujian <i>black box</i> menunjukkan semua fungsi berjalan sesuai kebutuhan.	digital, memusatkan data, mempercepat prosedur, dan mengurangi risiko kesalahan manual.
5.	(Kusumawaty, n.d.)	Perancangan Sistem Informasi Laboratorium Berbasis Web untuk Mempercepat Proses Administrasi Pelayanan Jasa Pengujian	Sistem informasi laboratorium berbasis web diterapkan di Baristand Industri Medan. Hasil uji coba menunjukkan penurunan signifikan waktu proses administrasi serta menghilangkan kesalahan pengetikan sertifikat.	Sistem mampu mempercepat pelayanan jasa pengujian, meminimalkan kesalahan administrasi, menyediakan laporan cepat, dan mendukung evaluasi berkala oleh pimpinan.

Hasil dari penelitian sebelumnya menunjukkan bahwa penerapan sistem informasi yang terintegrasi dalam layanan laboratorium dapat mempercepat pengelolaan data uji dan mengurangi risiko kesalahan yang terjadi akibat proses pencatatan manual. Proses kerja laboratorium yang sistematis terbukti mampu meningkatkan efisiensi waktu, ketepatan hasil pengujian, serta memudahkan pemantauan status sampel secara real time. Selain itu, metode pengembangan sistem berbasis *prototyping* banyak

digunakan karena memungkinkan proses peningkatan sistem dilakukan secara bertahap melalui evaluasi langsung terhadap rancangan awal, sehingga sistem yang dihasilkan bisa disesuaikan dengan kebutuhan operasional di lapangan.

Berdasarkan teori sistem informasi yang menekankan integrasi data dan efisiensi kerja, konsep *Laboratory Information Management System* (LIMS) yang berfokus pada pengelolaan data laboratorium secara elektronik, serta metode pengembangan *prototyping* yang memungkinkan penyempurnaan sistem melalui umpan balik dari dosen pembimbing selaku peneliti utama dan praktisi di PT Citra Lestari Enviro, penelitian ini bertujuan untuk merancang sistem layanan laboratorium berbasis web sesuai dengan kebutuhan operasional laboratorium tersebut.

*Framework Laravel* digunakan sebagai dasar pengembangan karena kemampuannya dalam membangun aplikasi web yang terstruktur, modular, dan efisien. Melalui landasan teori dan pengalaman empiris dari dosen pembimbing di lingkungan kerja laboratorium, sistem yang dikembangkan diharapkan dapat meningkatkan efisiensi kerja, akurasi data, serta kualitas layanan laboratorium lingkungan.