BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Sistem Informasi

Sistem adalah kumpulan langkah atau proses yang saling terkait dan berfungsi secara terpadu untuk menyelesaikan suatu masalah demi tercapainya tujuan tertentu. Sistem dapat diartikan sebagai himpunan komponen atau subsistem yang terorganisir secara sistematis dan saling berinteraksi untuk mewujudkan tujuan bersama. Adapun subsistem adalah bagian dari sistem yang lebih kecil namun berfungsi di dalam kerangka sistem utama (Negara, dkk, 2021:1).

Informasi adalah data yang telah diproses melalui metode tertentu sehingga memiliki makna dan nilai guna bagi pihak yang menerima (Prehanto, 2020:12). Data yang menjadi sumber informasi mencerminkan peristiwa nyata yang telah berlangsung pada waktu tertentu (Prehanto, 2020:12). Sistem informasi merupakan integrasi dari beragam teknologi yang berfungsi untuk menyajikan informasi serta mendukung komunikasi dalam lingkungan organisasi. Unsur-unsur dalam sistem ini terdiri dari *hardware*, *software*, data, tenaga kerja, serta proses yang saling berinteraksi untuk menghasilkan informasi yang berguna dalam proses pengambilan keputusan.

2. Website

Situs *web* adalah kumpulan halaman yang saling terkoneksi dan berisi berbagai bentuk informasi seperti tulisan, gambar, suara, video, dan animasi. yang dapat diakses melalui internet dan dibuat untuk keperluan individu, organisasi, maupun bisnis (Adiwisastra dan Hikmah, 2020:1). Sebuah *website* terdiri dari sejumlah halaman *web* yang saling terkoneksi dan berada dalam satu domain tertentu di jaringan internet (Nurhidayah, dkk, 2020:1).

Situs *web* terdiri dari beberapa halaman yang menampilkan konten informatif dalam berbagai format, seperti teks, visual, audio, animasi, ataupun gabungan dari semuanya, yang bisa bersifat statis maupun interaktif. Halaman-halaman ini saling terhubung membentuk suatu struktur yang utuh dan dapat diakses melalui *browser* oleh pengguna yang terhubung ke internet. *Website* berfungsi sebagai sarana penyebaran informasi secara *online*, dan biasanya disusun dalam file berformat HTM yang dapat diakses menggunakan protokol HTTP (Supendi, dkk, 2024).

a. HTML (*HyperText Markup Language*)

HTML adalah jenis bahasa penanda yang digunakan untuk membangun struktur halaman *web* agar dapat menampilkan beragam konten informasi, seperti teks dan gambar, melalui *browser*. Seiring dengan pertumbuhan pesat jumlah pengguna internet, HTML terus mengalami pembaruan guna mendukung penyajian halaman *web* yang lebih baik. Untuk memastikan pengembangannya berjalan optimal,

dibentuklah organisasi bernama W3C (*World Wide Web Consortium*) yang bertugas mengelola dan memperbarui standar HTML (Supendi, dkk, 2024).

Hypertext Markup Language (HTML) merupakan bahasa standar yang umum digunakan dalam pembuatan halaman *web* di internet (Styawantoro dan Komarudin, 2019:1). Bahasa ini berperan penting dalam menyusun dokumen *web*. HTML memiliki sejumlah fungsi, di antaranya:

a. Mengatur tampilan halaman *web* beserta isi kontennya.
b. Membuat formulir *online* untuk keperluan seperti pendaftaran dan transaksi secara daring.
c. Menyisipkan elemen multimedia seperti gambar, suara, video, animasi, serta komponen seperti Java applet ke dalam halaman *web* (Styawantoro dan Komarudin, 2019:4).

b. PHP *Hypertext Preprocessor*

Rasmus Lerdorf, seorang pengembang perangkat lunak dari Greenland, menciptakan PHP untuk pertama kalinya pada tahun 1995. Awalnya, ia menciptakan PHP untuk mencatat jumlah kunjungan ke situs *web* pribadinya. Karena itulah bahasa ini diberi nama *Personal Home Page* (PHP) *Tools*. Seiring meningkatnya minat dari komunitas pengguna, Rasmus merilis PHP secara terbuka dengan lisensi *open-source*. Kini, PHP menjadi bahasa skrip sisi *server* yang paling banyak digunakan di dunia, dengan versi yang terus berkembang, bahkan telah

mencapai versi 5 dan terus bertambah (www.php.net/usage.php) (Prasetiyo dan Chernovita, 2023).

PHP, yang merupakan kependekan dari *Hypertext Preprocessor*, adalah bahasa pemrograman yang digunakan untuk mengembangkan *website* interaktif dan aplikasi berbasis *web*. Tidak seperti HTML yang bersifat statis, PHP mampu berinteraksi dengan *database*, *file*, dan sistem direktori, sehingga memungkinkan pembuatan halaman *web* yang dinamis. Bahasa ini sering digunakan dalam pengembangan *blog*, *e-commerce*, CMS, forum, hingga *platform* media sosial. PHP bersifat lintas *platform* dan dapat dijalankan di berbagai sistem operasi seperti Linux, Windows, dan MacOS. Kode PHP biasanya ditulis dalam file berekstensi ".php" dan menggunakan format teks biasa (Prasetiyo dan Chernovita, 2023).

c. Laravel

Laravel merupakan kerangka kerja yang digunakan dalam pengembangan *web* karena menawarkan kesederhanaan serta fleksibilitas dalam hal desain dan struktur aplikasinya (Saka dan Ratama, 2023). Laravel dikembangkan dengan arsitektur *Model-View-Controller* (MVC), dan menjadi salah satu *framework* favorit di kalangan pengembang PHP untuk membangun aplikasi yang dinamis dan elegan. Laravel bahkan dinobatkan sebagai *framework* terbaik pada tahun 2014 (Saka dan Ratama, 2023).

Laravel memiliki berbagai keunggulan, antara lain ekspresif, sederhana (berkat penggunaan *Eloquent* ORM), dan mudah diakses berkat dokumentasi yang lengkap. Laravel juga dilengkapi dengan banyak fitur, seperti:

a. *Bundles*: sistem modular yang memungkinkan pengemasan fitur secara terpisah dan telah tersedia berbagai bundle siap pakai;
b. *Eloquent* ORM: implementasi pola *Active Record* dalam PHP yang menyederhanakan pengelolaan relasi antar objek *database*;
c. *Application Logic*: logika aplikasi dapat ditulis baik melalui *Controller* maupun langsung di dalam deklarasi *Route*;
d. *Reverse Routing*: menghubungkan URL dan *Route* secara dinamis;
e. *Restful Controllers*: memisahkan logika untuk permintaan HTTP *GET* dan *POST*;
f. *Class Auto Loading*: fitur untuk memuat class secara otomatis termasuk struktur *database*;
g. *Unit Testing*: mendeteksi kesalahan fungsi dan regresi secara otomatis;
h. *Automatic Pagination*: menyederhanakan pembuatan sistem penomoran halaman secara otomatis. (Saka dan Ratama, 2023).

3. Sistem Basis Data

Database Management System (DBMS) merupakan sebuah aplikasi perangkat lunak yang dibuat untuk memfasilitasi pengguna dalam membuat, mengatur, serta mengelola akses terhadap data yang tersimpan di dalam basis data (Budiarto, 2019:14). Sistem basis data adalah sebuah sistem yang mencakup sejumlah *file* atau tabel yang saling terhubung satu sama lain, dan diorganisir dalam lingkungan sistem komputer untuk keperluan pengelolaan data (Dantes, dkk, 2019:12).

DBMS mencakup sekumpulan data yang saling terhubung dan aplikasi perangkat lunak yang memungkinkan pengguna untuk melakukan akses dan pengelolaan data tersebut. Biasanya, data yang terorganisir ini disebut sebagai basis data. Tujuan utama DBMS adalah menyediakan cara yang efisien dan mudah dalam menyimpan serta mengambil data. Sistem ini dirancang untuk menangani informasi dalam jumlah besar, termasuk mendefinisikan struktur penyimpanan dan menyediakan sarana manipulasi data. Selain itu, DBMS juga bertanggung jawab atas perlindungan data dari kerusakan sistem atau akses ilegal, serta menjaga konsistensi data saat digunakan oleh banyak pengguna secara bersamaan (Rumetor, dkk, 2021).

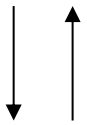
a. MySQL

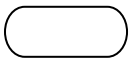
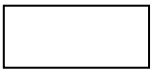
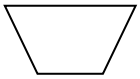
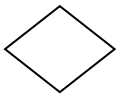
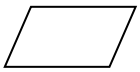
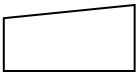
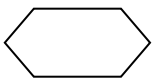
MySQL adalah salah satu RDBMS *open-source* yang tersedia dalam versi gratis maupun versi berbayar. Berkat lisensinya di bawah GNU *General Public License* (GPL), MySQL dapat dimanfaatkan tanpa biaya lisensi, baik untuk penggunaan pribadi maupun komersial (Dewi, 2023). MySQL merupakan RDBMS gratis yang dilisensikan di bawah GNU GPL, yang menyimpan dan mengelola data dalam struktur tabel yang terdiri dari baris dan kolom. Tersedia dalam versi *open-source* dan komersial, MySQL memudahkan pengguna dalam menangani berbagai tabel dalam satu *database* untuk mendukung pengelolaan data secara terstruktur dan efisien (Widyastuti, dkk 2024).

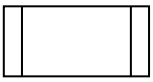
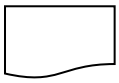
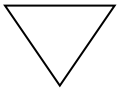
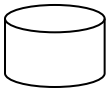
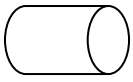
4. *Flowchart*

Flowchart merupakan representasi grafis yang menggunakan simbol tertentu untuk menggambarkan tahapan-tahapan suatu proses secara detail, termasuk hubungan antarproses dalam suatu program atau sistem (Saputro, dkk, 2022).

Tabel 2.1. Simbol-simbol *Flowchart*

Simbol	Nama	Keterangan
	Arus	Simbol dalam *flowchart* digunakan sebagai penghubung antar simbol lainnya atau untuk merepresentasikan kelanjutan dari suatu proses. Simbol ini dikenal juga dengan istilah garis penghubung.

	Terminal	Simbol *flowchart* berfungsi sebagai inisiasi (mulai) atau penutup (berhenti) suatu kegiatan.
	penghubung	Simbol berfungsi untuk memulai dan menyimpulkan atau menghubungkan operasi di dalam lembar atau halaman yang sama.
	baris penghubung	Simbol memberikan tujuan untuk memulai dan mengakhiri atau menghubungkan operasi di beberapa lembar atau halaman.
	Proses	Simbol *flowchart* digunakan untuk menggambarkan operasi komputasi yang dilakukan oleh komputer atau personal computer (PC).
	kegiatan manual	Simbol ini digunakan untuk menandakan aktivitas atau operasi yang dilakukan secara manual, bukan oleh sistem komputer
	keputusan	Simbol tersebut berfungsi untuk memilih suatu prosedur sesuai dengan keadaan saat ini.
	keluar-masuk	Simbol diagram alur berfungsi untuk mewakili proses *input* dan *output*, terlepas dari peralatan spesifik yang digunakan.
	Manual *Input*	Simbol tersebut berfungsi untuk memungkinkan input data secara manual pada *keyboard* virtual.
	persiapan	Simbol ini berperan dalam menunjukkan alokasi area penyimpanan yang tengah atau akan digunakan sebagai bagian dari proses pemrosesan data dalam sistem penyimpanan

	proses terdifinisi	Simbol diagram alur melayani tujuan pelaksanaan suatu bagian (subprogram)/prosedur.
	penyimpanan *online*	Simbol ini menunjukkan bahwa data masuk berasal dari media penyimpanan seperti diska, atau data keluar disimpan ke media tersebut.
	unit pita magnetic	Simbol ini digunakan untuk menyatakan bahwa data masuk diambil dari pita magnetik atau bahwa hasil keluaran ditulis ke media pita magnetik
	kartu plong	Simbol ini digunakan untuk menunjukkan bahwa *input* berasal dari kartu data atau *output* akan dicetak ke kartu tersebut
	dokumen	Simbol ini menandakan bahwa data *input* berasal dari dokumen fisik, atau bahwa *output* diproduksi dalam format cetak
	penyimpanan *offline*	Simbol ini menunjukkan bahwa data atau informasi yang tercantum akan disimpan dalam media penyimpanan
	magnetik *Disk*	Simbol ini digunakan saat data masuk atau keluar menggunakan media penyimpanan berbasis cakram magnetik
	magnetik drum	Simbol ini menunjukkan proses *input* atau *output* data melalui perangkat penyimpanan drum magnetik

.

5. Unified Modeling Language (UML)

UML juga digunakan dalam pemodelan proses bisnis serta sistem *non-software*. Bahasa ini dikembangkan oleh James Rumbaugh, Ivar Jacobson, dan Grady Booch melalui *Rational Software Corporation* dan berlandaskan pendekatan berorientasi objek. UML menyediakan berbagai notasi visual yang memungkinkan perancang sistem melihat dan memahami sistem dari berbagai perspektif. Penggunaannya pun tidak terbatas pada rekayasa perangkat lunak saja, tetapi meluas ke berbagai bidang yang memerlukan pemodelan (Irfan, dkk, 2023).

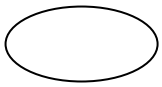
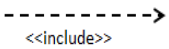
UML, atau *Unified Modeling Language*, merupakan bahasa pemodelan yang digunakan secara global untuk membuat rancangan sistem perangkat lunak. Dengan UML, pengembang dapat menggambarkan, merancang, membangun, dan mencatat berbagai elemen sistem secara terstruktur. Sama seperti arsitek merancang bangunan sebelum konstruksi, pengembang sistem menggunakan diagram UML untuk merancang struktur dan alur kerja dalam proses pengembangan *software*. Pemahaman terhadap istilah dan simbol dalam UML akan memudahkan pengembang dalam memahami spesifikasi sistem secara menyeluruh (Irfan, dkk, 2023).

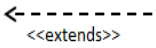
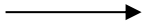
a. *Use case diagram*

Use case diagram adalah bentuk pemodelan yang bertujuan untuk menunjukkan hubungan atau interaksi antara aktor (pengguna) dengan sistem yang sedang dirancang. Diagram ini digunakan untuk

mengenali peran-peran penting dalam sistem serta menentukan pihak-pihak yang terlibat dalam proses tersebut memiliki otoritas untuk menggunakan fungsi-fungsi tertentu dalam sistem tersebut (Irfan, dkk, 2023).

Tabel 2.2. Tabel *Use Case*

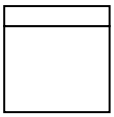
Gambar	Nama	Keterangan
	*Use Case*	Fungsi-fungsi sistem tergambar ketika terjadi pertukaran pesan antar komponen, baik antar unit maupun antara unit dengan *actor*.
	Aktor	Entitas yang terlibat dalam interaksi dengan sistem informasi, seperti orang, proses, atau sistem eksternal, berada di luar cakupan sistem tersebut. Karena itu, aktor tidak selalu merujuk pada manusia meskipun sering digambarkan dengan simbol orang; umumnya, penamaan aktor diawali dengan kata benda untuk merepresentasikan fungsinya
	Asosiasi	Aktor dapat berinteraksi dengan *use case*, begitu juga *use case* dapat berinteraksi dengan aktor yang terlibat.
	*Include*	Relasi *include* menunjukkan keterkaitan antara *use case* utama dengan fungsi tambahan yang hanya bisa dijalankan jika dipanggil oleh *use case* lainnya sebagai bagian dari proses. Terdapat dua interpretasi umum dalam konsep *include* pada *use case*: 1. *Use case* tambahan secara otomatis dipanggil setiap kali *use case* utama dijalankan.

		2. Sistem akan terlebih dahulu memverifikasi apakah *use case* yang disertakan telah dieksekusi sebelum melanjutkan ke fungsi tambahan tersebut.
	*Extend*	Interaksi antar *use case* dapat mencerminkan fungsi yang diperluas, di mana *use case* pelengkap tetap memiliki kemampuan untuk beroperasi secara mandiri tanpa kehadiran *use case* utama-konsep ini menyerupai pewarisan (*inheritance*) dalam paradigma pemrograman berbasis objek. Biasanya, nama *use case* tetap konsisten, dengan panah menunjukkan arah perluasan; *use case* yang diperluas biasanya masih dalam kategori fungsi yang sama dengan induknya.
	*Generalization*	Relasi antara generalisasi dan spesialisasi merepresentasikan struktur hierarkis yang menghubungkan dua *use case* dalam sistem, di mana salah satunya merupakan bentuk umum dari yang lain. Panah biasanya diarahkan ke *use case* yang bersifat generik untuk menunjukkan struktur pewarisan fungsi.

b. *Activity diagram*

Activity diagram merupakan jenis diagram yang digunakan untuk merepresentasikan alur aktivitas dan proses kerja dalam suatu sistem, baik dalam konteks bisnis maupun perangkat lunak. Fokus utama dari diagram ini adalah pada aktivitas-aktivitas yang dilakukan oleh sistem, bukan pada tindakan pengguna (aktor) (Irfan, dkk, 2023).

Tabel 2.3. Simbol *Actvity Diagram*

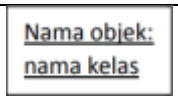
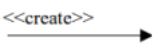
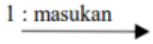
Gambar	Nama	Keterangan
	Status awal	Diagram aktivitas memiliki titik awal yang menunjukkan kondisi awal dari proses sistem yang akan dijalankan.
	Aktivitas	Aktivitas yang dilakukan oleh sistem biasanya diawali dengan kata kerja yang menunjukkan tindakan tertentu.
	Percabangan	Cabang asosiasi digunakan ketika terdapat beberapa pilihan aktivitas yang dapat dijalankan.
	Penggabungan	Penggabungan asosiasi dilakukan untuk menyatukan beberapa aktivitas menjadi satu kesatuan proses.
	Status akhir	Diagram aktivitas juga mencerminkan titik akhir eksekusi sistem, yang menunjukkan bahwa seluruh aktivitas telah selesai dilakukan.
	Swimlane	Pembagian tanggung jawab antar unit dalam organisasi yang mengelola pelaksanaan setiap aktivitas dalam proses bisnis.

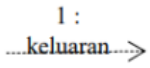
c. *Sequence diagram*

Sequence diagram adalah representasi visual yang menunjukkan bagaimana objek-objek dalam suatu *use case* berinteraksi satu sama lain melalui pesan yang dikirim dan diterima, serta bagaimana urutan aktivitas tersebut berlangsung sesuai dengan siklus hidup objek. Untuk menyusun diagram ini, penting untuk memahami objek-objek yang

terlibat dalam skenario serta peran yang dimainkan masing-masing objek dalam kelas (Irfan, dkk, 2023).

Tabel 2.3. Simbol *Sequence diagram*

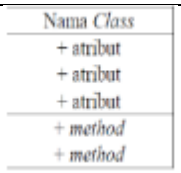
Gambar	Nama	Keterangan
	Aktor	Entitas seperti orang, proses, atau sistem eksternal yang terlibat dalam interaksi dengan sistem informasi berada di luar lingkup sistem utama yang dibangun. Karena itu, aktor tidak selalu merepresentasikan manusia walaupun disimbolkan dengan ikon manusia. Nama aktor umumnya diawali dengan kata benda untuk menunjukkan peran atau fungsinya.
	Lifeline	Menggambarkan tahapan atau fase dari keberadaan suatu objek selama masa hidupnya.
	Objek	Menandai objek yang terlibat dalam pertukaran pesan selama proses interaksi.
	Waktu aktif	Menunjukkan bahwa objek dalam kondisi aktif dan sedang melakukan aktivitas tertentu, di mana setiap tindakan yang terjadi selama periode aktif ini direpresentasikan sebagai bagian dari proses.
	Pesan tipe *Create*	Mengilustrasikan bahwa suatu objek bertanggung jawab dalam pembuatan objek baru, dengan panah diarahkan ke objek hasil ciptaan. Panah juga menunjukkan objek yang menjalankan metode tertentu, yang harus sesuai dengan metode yang terdapat dalam diagram kelas dari objek yang dimaksud.
	Pesan tipe *Send*	Sebuah objek yang berfungsi untuk mengirimkan data, masukan, atau informasi kepada objek lain, ditunjukkan dengan panah yang

		menunjuk ke arah objek penerima informasi
	Pesan tipe *return*	Menggambarkan proses pengembalian dari satu objek ke objek lain setelah menyelesaikan eksekusi metode tertentu, di mana panah diarahkan ke penerima hasil pengembalian.
	Pesan tipe *destroy*()	Menunjukkan bahwa suatu objek mengambil tindakan untuk menghapus atau mengakhiri objek lain, ditandai dengan panah yang mengarah ke objek yang dihilangkan. Idealnya, setiap objek yang dibuat memiliki proses penghancuran (*destroy*) yang menyertainya.

d. *Class diagram*

Class diagram digunakan untuk menampilkan struktur internal dari sistem dengan menjelaskan kelas-kelas utama yang menyusun sistem tersebut. Setiap kelas mencakup atribut (properti) dan operasi (metode) yang dapat dijalankan dalam konteks sistem (Irfan, dkk, 2023). Diagram kelas digunakan untuk merepresentasikan deskripsi kelas, atribut, objek, serta keterkaitannya satu sama lain. Diagram ini menyajikan gambaran menyeluruh terhadap struktur sebuah sistem melalui penggambaran hubungan antar kelas. Umumnya, dalam satu sistem terdapat lebih dari satu *class diagram* yang menunjukkan berbagai aspek struktur sistem. Diagram ini sangat berguna untuk memvisualisasikan struktur internal sistem berbasis objek. Selain itu, *class diagram* berperan dalam menjelaskan jenis-jenis objek yang digunakan dalam sistem beserta relasi yang terjadi antar objek tersebut (Suharni, dkk, 2023).

Tabel 2.3. Simbol *Class diagram*

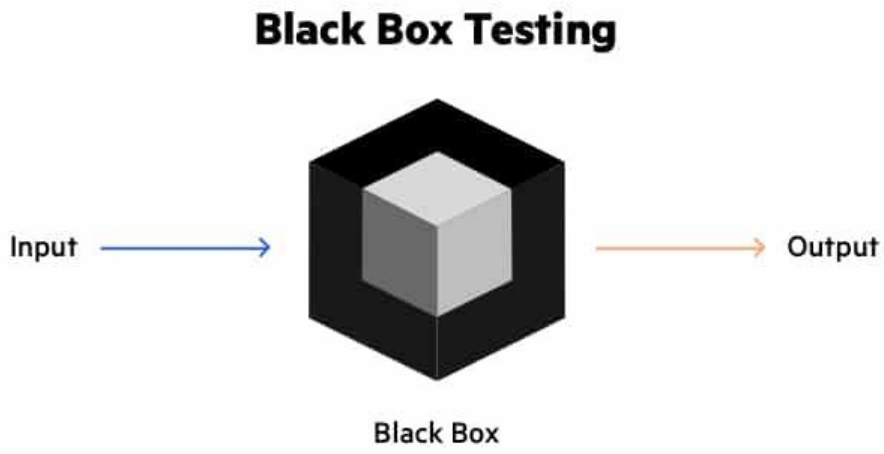
Gambar	Nama	Keterangan
	*Class*	Himpunan objek-objekdari berbagai atributyang me-miliki operasi yang sama.
	Association	Hubungan antara dua kelas yang bersifat umum, dan umumnya mencantumkan jumlah (*multiplicity*) objek yang terlibat dalam relasi tersebut.
	Directed Association	Hubungan antara kelas yang menunjukkan bahwa satu kelas memanfaatkan atau menggunakan fungsi dari kelas lainnya.
	*Aggregation*	Hubungan yang menggambarkan keterkaitan antara bagian dan keseluruhan suatu objek dikenal sebagai relasi bagian-keseluruhan (*whole-part relationship*).
	*Composition*	Relasi *composition* menunjukkan ketergantungan penuh suatu kelas terhadap kelas induknya, di mana bagian tidak dapat berdiri sendiri tanpa keseluruhannya.
	*Dependency*	Menggambarkan metode atau fungsi dalam sebuah kelas yang melibatkan pemanfaatan atau interaksi dengan kelas lainnya.

6. BlackBox

Pengujian *black-box* adalah metode pengujian perangkat lunak yang menitikberatkan pada evaluasi fungsi-fungsi sesuai kebutuhan sistem. Proses ini dilakukan dengan cara mengisi data pada form yang tersedia untuk mengamati apakah hasil yang diberikan sesuai dengan ekspektasi

(Arsa, dkk, 2024). Metode *black box* digunakan untuk menguji fungsi perangkat lunak tanpa perlu memahami struktur internal atau kode program yang mendasarinya (Utami, dkk, 2022).

Metode *black box* digunakan untuk mengevaluasi apakah *input* dan *output* sistem bisa dijalankan berdasarkan dengan spesifikasi yang telah ditentukan dalam tahap perancangan. Kelebihan dalam pengunaan uji *black box* adalah tidak diperlukannya akses kode, lebih mudah dalam pemahaman hasil ujinya karena pengujian pengguna dan pengembang dilakukan secara terpisah, serta dapat membuat kumpulan kode besar menjadi lebih efisien (Mubarok dan Hidayat, 2023).



Gambar 2.1. *Blackbox Testing*

Sumber: Wulandari, dkk (2025)

B. Kajian Empiris

Kajian empiris dalam penelitian ini adalah sebagai berikut:

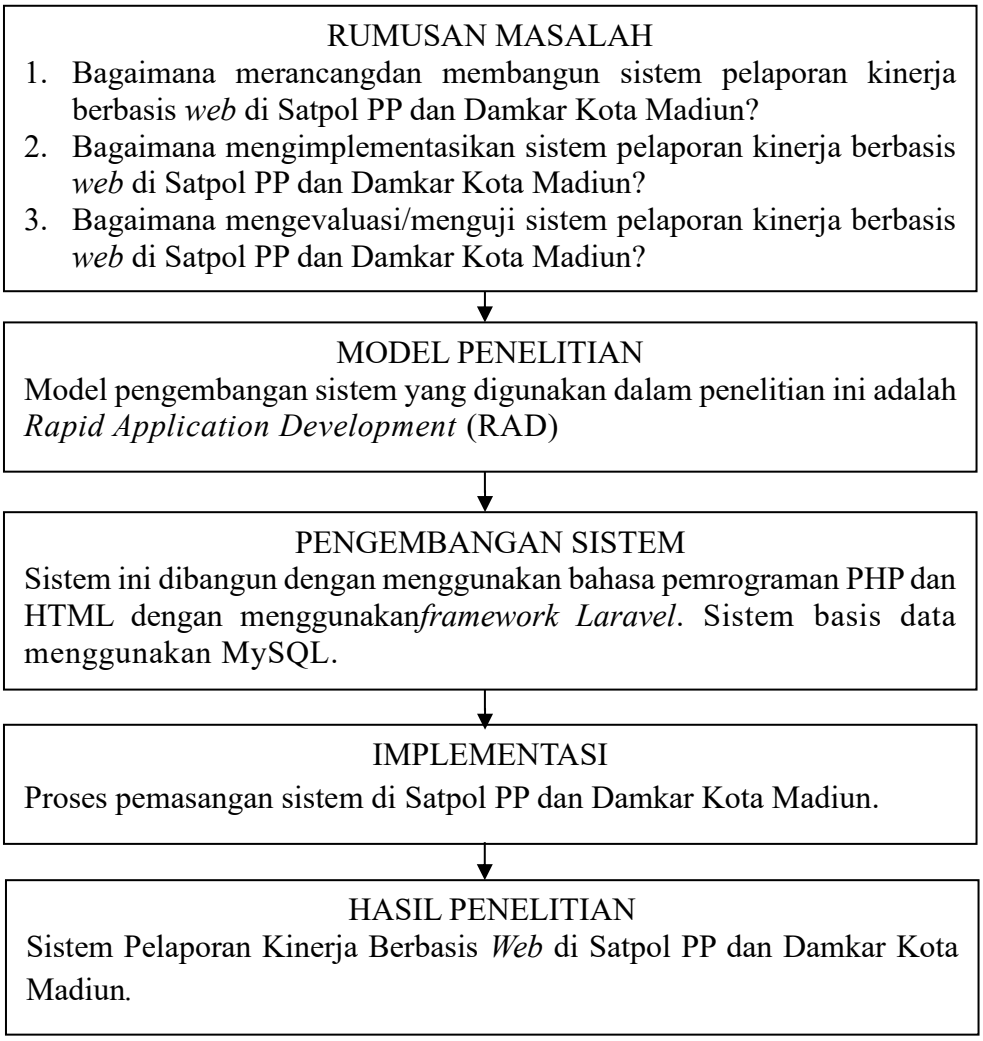
1. Berdasarkan hasil penelitian yang berjudul "Metode Scrum Pada Rancang Bangun Sistem Informasi Pelaporan Harian Pegawai BPS Provinsi Jambi" menunjukan bahwa sistem yang dibangun telah memenuhi seluruh kebutuhan fungsional (Arsa, dkk, 2024).
2. Berdasarkan hasil penelitian yang berjudul "Perancangan Sistem Pelaporan Kinerja Pegawai Bagian KebersihanBerbasis *Website* di Fakultas Teknik Universitas Sebelas Maret Surakarta" menunjukan bahwa Sistem informasi ini berfungsi mendukung bagian kepegawaian dalam memantau progres dan performa staf kebersihan melalui laporan aktivitas. Selain itu, sistem ini juga mempermudah pembagian tugas, validasi hasil pekerjaan, serta penyusunan laporan kinerja staf kebersihan (Sutarni, dkk, 2022).
3. Berdasarkan hasil penelitian yang berjudul "Pengembangan Sistem Informasi Pelaporan Kinerja Guru pada Keterlaksanaan Pembelajaran Berbasis *Dashboard* di SMKN 3 Takalar" menunjukan bahwa Pihak sekolah dapat memanfaatkan sistem ini untuk memperlancar proses evaluasi, penyusunan laporan, verifikasi dokumen, serta penyajian informasi yang dapat diakses secara cepat melalui tampilan dashboard interaktif (Nurafni, dkk, 2022).
4. Berdasarkan hasil penelitian yang berjudul "Pengembangan *Web Daily Report* untuk Efisiensi Pelaporan Kegiatan Pegawai" menunjukan bahwa sistem berfungsi dengan baik dan memenuhi tujuan perancangan.

Implementasi dan pengujian yang dilakukan menunjukkan bahwa sistem ini siap digunakan dan dapat diandalkan untuk manajemen laporan harian pegawai (Firmansyah, dkk, 2024).

5. Berdasarkan hasil penelitian yang berjudul "Pengembangan Aplikasi Laporan Kinerja Harian Berbasis *Website* Terhadap Kinerja Dosen STAHN MPU Kuturan Singaraja" menunjukan bahwa Aplikasi pelaporan harian kinerja dosen berjalan dengan efektif dan memberikan kontribusi positif bagi para dosen di STAH Negeri MPU Kuturan Singaraja (Heriawan, dkk, 2023).

C. Kerangka Berpikir

Kerangka berpikir dalam penelitian iniadalah sebagai berikut:



Gambar 2.2. Kerangka Berpikir