

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Dasar Teori yang Mendukung Produk**

##### **2.1.1. Point of Sales**

*Point of Sales* (POS) adalah sistem yang digunakan untuk melakukan transaksi penjualan barang atau jasa kepada pelanggan. Sistem ini mencakup pencatatan barang yang dibeli, total pembayaran, metode pembayaran, serta pencetakan bukti transaksi (struk). POS modern tidak hanya mencakup fungsi dasar penjualan, tetapi juga integrasi dengan modul manajemen stok, pelanggan (CRM), dan laporan analitik penjualan secara real-time. Dengan POS berbasis web, data transaksi tersentralisasi di server sehingga memudahkan akses multi-user, meminimalkan duplikasi entri, dan memungkinkan pemilik usaha memantau performa penjualan kapan saja melalui dashboard interaktif. Implementasi POS juga sering dilengkapi fitur notifikasi stok minimum, yang secara otomatis memberitahukan staf ketika persediaan mendekati ambang batas, serta modul discount dan promo, yang membantu strategi pemasaran dan meningkatkan loyalitas pelanggan.

##### **2.1.2. Sistem Informasi**

Sebagai sebuah kesatuan, sistem informasi merupakan integrasi dari berbagai elemen yang saling terkait, dirancang untuk mengelola data mulai dari pengumpulan, pemrosesan, penyimpanan, hingga pendistribusian informasi. Komponen-komponen utama yang membentuk sistem informasi ini terdiri dari komponen masukan, komponen model, komponen keluaran, komponen teknologi, dan komponen basis data [7]. Sistem informasi memainkan peran penting dalam mengelola transaksi dan mendukung kegiatan strategis dengan menghasilkan informasi yang berkualitas. Sistem ini terdiri atas subsistem yang saling berhubungan serta beroperasi secara terpadu untuk mengubah data mentah menjadi

informasi yang relevan dan bernilai. Sistem informasi dapat didefinisikan sebagai seperangkat elemen yang terintegrasi, dengan komponen-komponen yang saling terkait untuk memproses, menyimpan, dan menyebarkan informasi. Sistem ini terdiri dari blok masukan, model, keluaran, teknologi, dan basis data, yang semuanya berfungsi mendukung aktivitas operasional dan manajerial dalam sebuah organisasi.

### **2.1.2. Manajemen Persediaan (Inventory Management)**

Manajemen persediaan merupakan serangkaian proses pengendalian terhadap aliran barang, dengan tujuan memastikan ketersediaannya dalam kuantitas dan waktu yang presisi guna memenuhi kebutuhan pelanggan tanpa menimbulkan kelebihan atau kekurangan stok [8]. Tujuan utama manajemen persediaan adalah menjaga keseimbangan antara permintaan dan ketersediaan barang dengan mempertimbangkan efisiensi biaya penyimpanan. Pengelolaan yang efektif dalam manajemen persediaan dapat meminimalisir biaya operasional, mencegah terjadinya stok mati, serta memastikan ketersediaan produk saat dibutuhkan, yang pada gilirannya akan meningkatkan kepuasan pelanggan dan profitabilitas bisnis.

### **2.1.3. Sistem Manajemen Persediaan**

Sistem informasi manajemen persediaan digunakan untuk mempermudah proses pencatatan, pengendalian, serta pelaporan stok barang secara terstruktur [9]. Sistem ini memungkinkan perusahaan untuk melakukan pengambilan keputusan berdasarkan data yang tersaji secara *real-time*. Sistem semacam ini penting dalam mendukung efisiensi operasional dan ketepatan perencanaan stok. Dengan adanya sistem ini, proses manual yang rentan kesalahan dapat diminimalisir, serta memungkinkan analisis data yang lebih mendalam untuk peramalan permintaan dan optimalisasi tingkat persediaan, sehingga mendukung kinerja bisnis yang lebih baik.

### **2.1.4. Website**

*Website* adalah sebuah sistem informasi global yang terbentuk dari koneksi antarjaringan komputer menggunakan protokol internet. Peran krusial *website*

dalam internet adalah memungkinkan penggunanya untuk mengakses, mencari, dan bertukar aneka informasi, data, gambar, video, dan konten lain menggunakan peramban *website* [10]. Perkembangan *website* merepresentasikan perjalanan evolusi jaringan global yang memungkinkan ketersediaan informasi dan sumber daya secara daring. *Website* ini memiliki beberapa komponen utama adalah sebagai berikut:

1. *Protokol*

*Website* beroperasi di atas protokol *HTTP (Hypertext Transfer Protocol)* yang merupakan bahasa komunikasi antara browser *website* dan server untuk mengambil halaman *website*.

2. *Hypertext*

Informasi di *website* disajikan dalam bentuk dokumen *website* yang biasanya ditulis dengan bahasa markup seperti *HTML (Hypertext Markup Language)*. Dokumen ini bisa memuat teks, gambar, video, audio, formulir, dan elemen lainnya.

3. *Dokumen Website*

Informasi di *website* disajikan dalam bentuk dokumen *website* yang biasanya ditulis dengan bahasa markup seperti *Hypertext Markup Language (HTML)*. Dokumen ini bisa memuat teks, gambar, video, audio, formulir, dan elemen lainnya.

4. *URL*

Setiap dokumen *website* memiliki alamat unik yang disebut *URL*, yang digunakan untuk mengidentifikasi dan mengakses halaman *website* secara akurat.

5. *Browser Website*

Pengguna mengakses *website* melalui perangkat lunak yang disebut browser *website*, seperti *Chrome, Firefox, Safari*, atau *Edge*. Browser bertugas menginterpretasikan dokumen *website*, mengunduh sumber daya seperti gambar dan video, serta menampilkan halaman *website* kepada pengguna.

6. *Interaksi Pengguna*

*Website* memungkinkan interaksi dua arah antara pengguna dan konten, seperti mengklik tautan, mengisi formulir, memberikan komentar, dan berinteraksi dengan aplikasi *website* lainnya.

#### 7. Aplikasi *Website*

Selain sebagai sumber informasi, *website* juga digunakan untuk menyediakan aplikasi berbasis *website* yang berjalan di browser, seperti email online, penyimpanan berbasis *cloud*, media sosial, *e-commerce*, dan lain-lain.

#### 8. Keamanan *Website*

Keamanan adalah aspek penting dari *website* yang mencakup enkripsi data pada protokol komunikasi *Hypertext Transfer Protocol Secure (HTTPS)*, perlindungan dari serangan siber, pengelolaan hak akses, dan perlindungan privasi pengguna.

### 2.1.5. Pengertian *MySQL*

*MySQL* merupakan *software* database *Relational Database Management System (RDBMS)* yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user (multi-user)* [11]. Dalam pengembangan aplikasi, *MySQL* berperan sebagai basis data yang dapat menjalankan proses secara *multi-threaded* (sinkron atau berbarengan). Basis data ini sering dihubungkan dengan pemrograman *PHP* untuk mendukung fungsionalitas aplikasi.

Dengan demikian, dapat ditarik kesimpulan bahwa *MySQL* merupakan basis data relasional yang berperan penting dalam menyimpan data-data esensial yang diperlukan oleh suatu perangkat lunak.

### 2.1.6. Pengertian *PHP*

*Hypertext Preprocessor (PHP)* merupakan Bahasa pemrograman *server-side*, pada *PHP* prosesnya akan dilakukan di dalam server beberapa server yang sering digunakan seperti *Apache*, *MySQL*, dan lain – lain [12]. *PHP* juga merupakan Bahasa *open source* yang memungkinkan untuk pembuatan aplikasi secara dinamis

dan melakukan pengolahan serta pemrosesan data. Data di proses pada server dan hasilnya ditampilkan kepada *user* dengan menggunakan browser.

### 2.1.7. Pengertian *Laravel*

*Laravel* merupakan *framework* atau kerangka kerja dari bahasa pemrograman *PHP*, didalam *framework laravel* terdapat konsep *Model View Controller* (MVC) yang berarti komponen utama pengembangan sistem dibuat secara terpisah, adalah sebagai berikut antara *user* interface, data dan bagian kontrol [13]. Komponen MVC terdiri dari adalah sebagai berikut:

#### a. Model

Model berperan dalam mengelola basis data, memungkinkan berbagai operasi seperti mengambil, menyimpan, memperbarui, dan menghapus data dari tabel. Selain itu, Model juga bertanggung jawab untuk menangani validasi yang berasal dari *Controller*.

#### b. *View*

*View* dalam MVC berfungsi untuk menampilkan informasi kepada pelanggan berdasarkan data yang diproses oleh *Controller*.

#### c. *Controller*

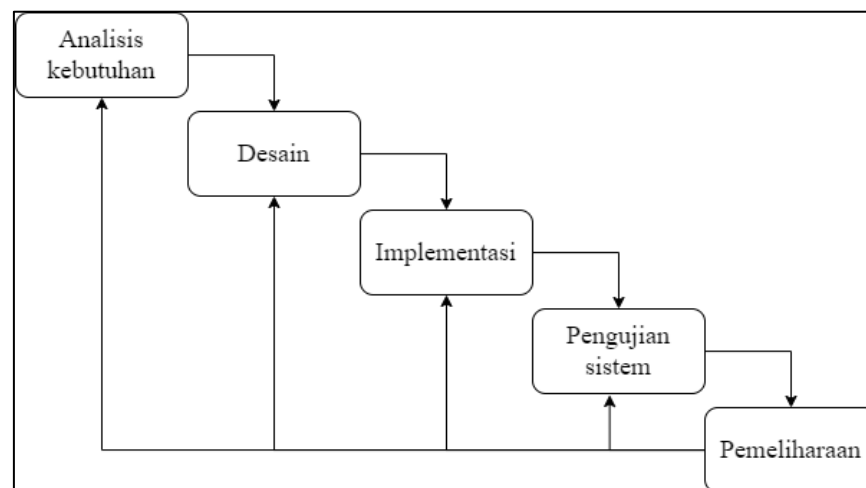
*Controller* berfungsi untuk mengatur dan sebagai perantara antara model (yang mengelola data) dan *View* (tampilan yang diberikan kepada pelanggan). Tugas utamanya adalah mengatur logika aplikasi dan menangani permintaan dari pelanggan.

Dalam pengembangan situs *website* dengan *framework Laravel*, dua *tool* utama yang sering digunakan adalah *Composer* dan *Artisan*. *Composer* berfungsi sebagai manajer dependensi, menyediakan kumpulan dependencies dan libraries esensial yang diperlukan saat memulai proyek *Laravel* atau ketika menambahkan library baru. Semua dependencies dan library yang digunakan akan tercatat dalam berkas *composer.json* yang terletak di direktori utama aplikasi. Sementara itu, *Artisan* adalah *command-line interface* (CLI) bawaan *Laravel* yang dirancang

untuk membantu pengembang menjelajahi dan memanfaatkan fitur-fitur *Laravel* secara lebih mendalam.

### 2.1.8. Pengertian *Waterfall*

Dikenal sebagai salah satu metode pengembangan sistem atau *Software Development Life Cycle (SDLC)*, Model *Waterfall* mengusung pendekatan berurutan secara linier. Metode ini melibatkan serangkaian fase yang berlangsung secara berurutan, dimulai dari analisis, perancangan, implementasi kode program, pengujian, hingga tahap pemeliharaan [14]. Gambar *Flowchart* Metode Pengembangan Sistem *Waterfall* ditunjukkan pada Gambar 2.1 sebagai berikut:



Gambar 2. 1 Flowchart Metode Pengembangan Sistem *Waterfall*

Metode pengembangan sistem, seperti yang digambarkan dalam Gambar 2.1 dengan flowchart *Waterfall*, mendefinisikan kerangka kerja berurutan yang diikuti dalam proyek ini. Alur ini dimulai dari identifikasi kebutuhan dan berlanjut melalui desain, implementasi, pengujian, hingga pemeliharaan, mencerminkan pendekatan yang disiplin untuk memastikan setiap aspek pengembangan tertangani secara sistematis.

#### 1. Analisa kebutuhan perangkat lunak

Dalam rekayasa perangkat lunak, kebutuhan fungsional merinci apa saja proses, fitur, dan layanan yang akan ditawarkan sistem kepada penggunanya. Di sisi lain,

kebutuhan non-fungsional adalah spesifikasi yang berkaitan dengan kinerja, keamanan, kemudahan penggunaan, serta aspek operasional dan kebutuhan spesifik dari sisi pelanggan.

## **2. Desain**

Desain adalah tahapan mentransformasikan kebutuhan *software* menjadi konsep rancangan sebelum proses pembuatan kode program. Mulai dari struktur data, basis data, desain antarmuka, desain menu hingga arsitektur *software*.

## **3. Implementasi**

Tahapan Implementasi merupakan proses menerjemahkan desain sistem menggunakan kode program, sehingga hasil yang didapatkan berupa sistem berbasis *Website*.

## **4. Pengujian**

Pengujian sistem dilakukan untuk memeriksa terhadap *software* yang dibuat untuk memastikan semua proses yang ada dan tidak terjadi *error* serta memastikan *output* yang dihasilkan sesuai dengan kebutuhan pelanggan.

## **5. Pemeliharaan**

Tahapan ini dilakukan jika kondisi *software* sudah dikirimkan dan digunakan oleh pelanggan. Pendukung dilakukan jika terjadi malafungsi pada *software* sehingga pembuat *software* harus melakukan perubahan atau pengulangan proses dimulai dari analisa spesifikasi kebutuhan hingga pengujian.

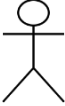



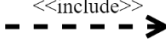
### **2.1.9. Pengertian Unified Modeling Language (UML)**

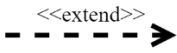
*Unified Modeling Language* (UML) dapat diartikan sebagai bahasa pemodelan berorientasi objek yang berfungsi untuk merepresentasikan sistem. Sistem tersebut didominasi oleh objek wddan dinotasikan menggunakan simbol-simbol spesifik [15]. Dalam pemodelan UML, terdapat beragam diagram yang berfungsi menjelaskan proses dan alur kerja sistem yang akan dirancang adalah sebagai berikut:

#### **1. Use case diagram**

Sebagai salah satu komponen UML, *Use case diagram* berfungsi untuk memvisualisasikan interaksi antara aktor dengan sistem. Pemanfaatan diagram ini sangat penting dalam proses identifikasi dan memahami berbagai fungsi yang tersedia dalam sistem, sekaligus menentukan aktor-aktor yang memiliki otorisasi untuk mengakses dan memanfaatkannya [16]. Tabel Simbol *Use case diagram* ditunjukkan pada Tabel 2.1 sebagai berikut:

Tabel 2. 1 Simbol *Use case diagram*

Simbol	Nama Simbol	Keterangan
	<i>Actor</i>	Mewakili peran pengguna (orang atau pelanggan), sistem, dan alat yang berinteraksi dengan use case.
	<i>Use Case</i>	Ini berfungsi untuk merepresentasikan koneksi antara aktor dengan use case..
	<i>Association</i>	Ini berfungsi sebagai penghubung yang mengaitkan aktor dengan use case.
	<i>Generalization</i>	Ini berfungsi untuk menggambarkan peran spesifik aktor dalam berinteraksi dengan use case.
	<i>Include</i>	Ini mengindikasikan bahwa suatu use case merupakan bagian integral yang harus ada dari use case lain. Use case ini akan selalu dijalankan sebagai bagian dari alur use case utama.



	<i>Extend</i>	Sebuah use case dapat menunjukkan bahwa ia sepenuhnya merupakan ketentuan tambahan (opsional) dari use case lain, hanya jika kondisi tertentu terpenuhi. Ini berarti implementasinya tidak wajib dilakukan setiap kali use case utama dijalankan.
---	---------------	---

Berbagai simbol yang membentuk *Use case diagram* dijelaskan secara rinci dalam Tabel 2.1. Tabel ini berfungsi sebagai referensi penting untuk memahami dan mengkonstruksi diagram kasus penggunaan, yang krusial dalam memetakan interaksi fungsional antara aktor dan sistem.

## 2. Activity Diagram

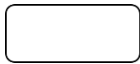
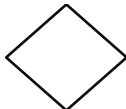
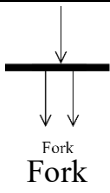
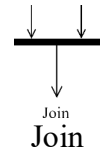
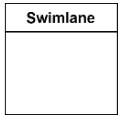
Diagram aktivitas adalah representasi visual yang menggambarkan aktivitas atau aliran kerja dalam suatu sistem. Diagram ini memvisualisasikan setiap langkah atau tindakan yang terjadi dalam sebuah proses, serta bagaimana tindakan-tindakan tersebut saling terhubung [17]. Tabel Simbol *Activity Diagram* ditunjukkan pada Tabel 2.2 sebagai berikut:

Tabel 2. 2 Simbol *Activity Diagram*

Simbol	Keterangan
Start Point 	<i>Start point</i> atau <i>initial</i> berfungsi untuk menggambarkan proses paling awal dari sebuah aktivitas dan menunjukkan di mana aliran kerja dimulai
End Point 	<i>End point</i> atau <i>final</i> berfungsi untuk menggambarkan proses paling akhir dari sebuah aktivitas dan menunjukkan di mana aliran kerja berhenti atau selesai.

Tabel 2.2 menyajikan simbol-simbol yang digunakan dalam *Activity Diagram*. Simbol-simbol ini memungkinkan pemodelan alur kerja atau proses bisnis secara step-by-step, mengilustrasikan urutan kegiatan dan keputusan dalam sistem atau suatu proses. Tabel Simbol *Activity Diagram* (lanjutan) ditunjukkan pada Tabel 2.3 sebagai berikut:

Tabel 2. 3 Simbol *Activity Diagram* (lanjutan)


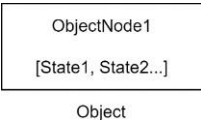
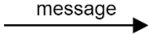
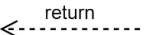

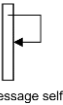
Simbol	Keterangan
 Action	<i>Action</i> berfungsi untuk menggambarkan suatu proses, langkah-langkah, atau kegiatan dalam sebuah aktivitas.
 Decision	<i>Decision</i> berfungsi untuk menggambarkan keputusan yang akan dibuat.
 Fork	<i>Fork</i> berfungsi untuk membagi suatu aktivitas menjadi beberapa aktivitas
 Join	<i>Join</i> berfungsi untuk menggabungkan dua atau beberapa aktivitas menjadi satu aktivitas
 Swimlane	<i>Swimlane</i> berfungsi untuk mengelompokkan diagram activity berdasarkan aktor


Lanjutan dari pembahasan sebelumnya, Tabel 2.3 melengkapi daftar simbol yang krusial untuk membuat *Activity Diagram* yang komprehensif. Simbol-simbol ini membantu dalam menggambarkan detail proses, seperti percabangan dan penggabungan aktivitas, memastikan alur kerja sistem terpetakan secara menyeluruh.

### 3. *Sequence Diagram*

Sebagai model dinamis, *Sequence Diagram* atau diagram urutan memvisualisasikan kronologi pesan yang dikirimkan antara berbagai objek selama berlangsungnya suatu interaksi tertentu [18]. Diagram ini berperan penting dalam memahami spesifikasi sistem waktu nyata, terutama dalam menggambarkan alur komunikasi antar objek berdasarkan waktu. Pada *Sequence Diagram*, terdapat beberapa komponen atau simbol dengan fungsi berbeda. Tabel Simbol *Sequence Diagram* ditunjukkan pada Tabel 2.4 sebagai berikut:

Tabel 2. 4 Simbol *Sequence Diagram*

Simbol	Keterangan
	Aktor yang berinteraksi dengan sistem informasi yang dikembangkan.
	<i>Object</i> menggambarkan objek dari sistem, berkolaborasi dengan mengirim atau menerima pesan.
	<i>Message</i> merupakan simbol mengirim pesan atau interaksi antara objek yang menunjukkan urutan peristiwa yang terjadi.
	<i>Return</i> adalah simbol yang menggambarkan penyampaian informasi atau pesan balik dari sebuah objek ke aktor.
	Berfungsi sebagai landasan dalam menyusun basis data atau mewakili tabel didalam basis data.
	Ini berfungsi untuk menggambarkan pesan yang dikirimkan oleh objek itu sendiri, yang sekaligus menunjukkan kronologi peristiwa yang berlangsung.


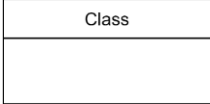

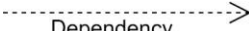
	Berfungsi untuk menunjukkan proses yang terjadi dari hasil objek yang diterima.
---	---

Tabel 2.4 menguraikan simbol-simbol yang dipakai dalam *Sequence Diagram*. Diagram ini berfokus pada urutan pesan yang dipertukarkan antar objek dalam suatu skenario tertentu, dan simbol-simbol ini vital untuk memvisualisasikan interaksi temporal antar komponen sistem.

#### 4. Class Diagram

Sebagai salah satu diagram struktur statis dalam *UML*, diagram kelas (*Class Diagram*) berfungsi untuk memodelkan struktur sistem. Diagram ini secara eksplisit menampilkan Ini mencakup kelas-kelas yang ada dalam sistem, atribut yang dimiliki oleh setiap kelas, serta metode yang dapat dijalankan, serta relasi atau hubungan antar objek [19]. Tabel Simbol *Class Diagram* ditunjukkan pada Tabel 2.5 sebagai berikut:

Tabel 2. 5 Simbol *Class Diagram*

Simbol	Keterangan
 Association	<i>Association</i> menggambarkan bagaimana sebuah objek terhubung dengan objek lainnya.
 Class	<i>Class</i> merepresentasikan sekelompok objek yang berbagi atribut dan operasi yang identik..
 Agregation	<i>Agregation</i> menunjukkan hubungan satu ke banyak (one-to-many) antara satu kelas dengan kelas lainnya.
 Dependency	<i>Depedency</i> menggambarkan hubungan antar <i>class</i> yang memiliki makna saling bergantung antara sebuah <i>class</i> dan <i>class</i> yang berbeda.


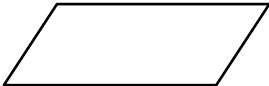

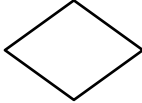

Pada Tabel 2.5, dijelaskan berbagai simbol yang digunakan dalam *Class Diagram*. Simbol-simbol ini merepresentasikan struktur statis sistem, termasuk

kelas-kelas, atributnya, metodenya, serta hubungan (relasi) antar kelas, yang fundamental dalam perancangan arsitektur sistem.

### 2.1.11. Pengertian *Flowchart*

*Flowchart* adalah penggambaran langkah-langkah dari suatu program dalam bentuk simbol yang saling berhubungan urut sesuai proses dari sistem yang dibangun [21]. Kegunaan utama flowchart adalah membantu analis dan programmer menyederhanakan masalah dengan membaginya ke dalam bagian-bagian yang lebih kecil. Tabel Simbol *Flowchart* ditunjukkan pada Tabel 2.7 sebagai berikut:

Tabel 2. 6 Simbol *Flowchart*

Simbol	Keterangan
	Simbol mulai/ selesai berfungsi menandakan suatu proses telah dimulai atau selesai
	Simbol <i>Input / Output</i> menggambarkan proses data yang dimasukkan oleh sistem atau melalui perangkat lunak.
	Simbol proses berfungsi sebagai indikator kegiatan operasional yang terjadi.
	Percabangan ( <i>Decision</i> ) dimanfaatkan dalam program untuk melakukan penyeleksian kondisi tertentu.
	Ini digunakan untuk menandai dokumen yang menjadi masukan atau keluaran, khususnya yang berasal dari pencatatan atau proses yang dilakukan secara manual.

Tabel 2.7 menampilkan berbagai simbol yang dipakai dalam *Flowchart*. Simbol-simbol ini esensial untuk memvisualisasikan alur logis dari suatu proses atau algoritma, dengan jelas menunjukkan langkah-langkah, keputusan, dan arah aliran kontrol dalam sistem.

### 2.1.12. Pengujian *Blackbox*

Pengujian *black box* adalah pengujian sistem tanpa memperdulikan bagaimana bentuk dari struktur *code* dibelakangnya. Pengujian ini fokus terhadap

bagaimana sistem tersebut mencapai tujuan dari suatu proses yang sudah ditetapkan dan mencari celah program yang tidak sesuai dengan prosedur atau spesifikasi yang sudah ditetapkan [22]. Pengujian *Blackbox* berpusat pada validasi kebutuhan fungsional perangkat lunak, dengan mengacu pada spesifikasi yang telah ditentukan. Kategori kesalahan (error) yang dapat teridentifikasi melalui pengujian *Blackbox* ini adalah sebagai berikut:

1. Kesalahan fungsi yang hilang atau tidak benar.
2. Error pada antarmuka.
3. Kesalahan dalam struktur data atau akses database *eksternal*.
4. Kesalahan operasional atau kinerja.
5. Kesalahan inisialisasi dan terminasi.

Untuk mendapatkan hasil nilai efektifitas pada pengujian *black box*, maka dilakukan perhitungan untuk menentukan jumlah nilai efektifitas dari sistem dapat dihitung menggunakan Persamaan. Rumus untuk mengitung hasil nilai efektifitas dapat dilihat pada Persamaan 2.1 adalah sebagai berikut:

$$\text{Presentase Keberhasilan} = \left( \frac{\sum \text{Skenario Berhasil}}{\sum \text{Total Skenario}} \right) \times 100\% \quad (2.1)$$

$\sum$  Skenario Berhasil adalah jumlah skenario pengujian *Black Box* yang berhasil dilakukan berdasarkan *input-process-output* yang diharapkan, sedangkan  $\sum$  Total Skenario adalah jumlah sampel data yang digunakan [23].

## 2.2. Produk Sejenis atau Terkait

Penelitian terdahulu yang relevan dan mendukung penelitian ini adalah sebagai berikut:

1. Penelitian ini [24] membahas “Pengembangan Sistem Informasi Persediaan Barang pada Toko Sumber Rejeki Berbasis *Website*”. Penelitian ini menunjukkan bagaimana sistem informasi persediaan dapat digunakan untuk memonitor keluar-masuk barang secara real-time, serta

menyediakan laporan stok yang akurat dan dapat diakses oleh admin toko. Dengan pendekatan berbasis *website*, pengelolaan barang menjadi lebih terstruktur dan risiko kehabisan stok dapat diminimalisir.

2. Penelitian ini [25] berjudul “Sistem Informasi Bengkel Motor Berbasis *Website* di Bengkel Speed Garage”. Penelitian ini relevan karena membahas secara langsung proses digitalisasi manajemen suku cadang di sektor otomotif, termasuk fitur pencatatan stok, histori pemakaian barang, serta laporan penggunaan harian. Sistem yang dikembangkan mampu meningkatkan efisiensi dan transparansi dalam pengelolaan barang pada skala bengkel.
3. Penelitian ini [26] menjelaskan tentang "Perancangan Sistem Informasi Persediaan Barang pada Toko Elektronik Berbasis *Website*" – Jurnal Teknologi dan Sistem Informasi. Penelitian ini menekankan pentingnya akurasi data stok dan efisiensi pengelolaan barang dengan bantuan sistem informasi berbasis *website*, yang sangat relevan dengan topik penelitian ini.
4. Penelitian ini [27] dikaji pengembangan sistem informasi persediaan barang berbasis *website* di CV. Makmur Sejahtera Paropo, sebuah perusahaan dagang yang memerlukan sistem pencatatan yang akurat dan cepat. Penelitian ini menampilkan penerapan sistem informasi yang mampu mencatat proses keluar-masuk barang secara otomatis, mengurangi potensi kesalahan pencatatan manual, serta meningkatkan kontrol terhadap stok. Kesesuaian dengan penelitian ini terlihat pada kebutuhan pengelolaan persediaan yang lebih sistematis, terutama untuk toko-toko dengan frekuensi transaksi tinggi seperti Alfa Jaya Motor.
5. Penelitian ini [28] dilakukan penelitian berjudul “Pengembangan Rancang Bangun Sistem pada Bengkel Aries Star Motor Berbasis *Websitesiste*” yang bertujuan untuk mendigitalisasi pencatatan stok suku cadang serta mempermudah proses pengawasan ketersediaan barang. Penelitian ini tidak hanya menghadirkan fitur pencatatan barang, tetapi juga dilengkapi dengan notifikasi stok minimum, laporan stok bulanan, dan manajemen pengguna,

yang membuatnya sangat relevan dijadikan acuan dalam pengembangan sistem informasi persediaan pada bengkel atau toko sejenis. Pendekatan berbasis *website* memberikan fleksibilitas dalam akses dan pengelolaan data, yang menjadi nilai tambah dalam penelitian ini.

### **2.2.1. Perbedaan Penelitian Ini dengan Penelitian Terdahulu**

Penelitian ini memiliki beberapa perbedaan mendasar dibandingkan penelitian-penelitian sebelumnya adalah sebagai berikut:

1. Penelitian ini [29] secara spesifik menyoroti pengembangan sistem informasi untuk toko suku cadang kendaraan bermotor sebagai bagian dari sektor UMKM. Berbeda dengan sebagian besar penelitian sebelumnya yang cenderung berfokus pada konteks retail umum, seperti toko elektronik atau restoran, penelitian ini mengangkat kebutuhan khas dari bisnis suku cadang, seperti tingginya frekuensi transaksi dan perputaran barang yang cepat, yang mengembangkan sistem informasi berbasis *website* untuk mendukung pencatatan stok suku cadang di bengkel "Bangkit Jaya Motor".
2. Penelitian ini [30] diimplementasikan fitur notifikasi otomatis untuk memantau stok minimum secara *real-time*. Fitur ini dirancang untuk memberikan peringatan kepada pengguna ketika jumlah barang mendekati batas minimal, sehingga pengadaan dapat segera dilakukan sebelum stok benar-benar habis. Penambahan fitur ini belum banyak dijumpai pada penelitian terdahulu, di mana sistem berbasis *website* dirancang untuk mendeteksi kebutuhan restok secara otomatis berdasarkan parameter tertentu.
3. Penelitian ini [31] menggunakan *framework Laravel* sebagai landasan utama pengembangan sistem. Pemilihan *Laravel* didasarkan pada kemampuannya dalam menyediakan struktur pengembangan yang modular, keamanan yang baik, serta kemudahan pengelolaan database menggunakan *ORM (Eloquent)*. *Laravel* terbukti mampu meningkatkan efisiensi proses pengembangan dan mempermudah pengelolaan data stok secara sistematis.

Pendekatan ini berbeda dari penelitian terdahulu yang masih banyak menggunakan *PHP native*.

4. Penelitian ini [32] mendukung penggunaan dengan dua tingkatan akses, adalah sebagai berikut sebagai admin dan sebagai kasir. Penggunaan hak akses berbasis peran ini memberikan kontrol keamanan yang lebih tinggi, di mana setiap pengguna hanya dapat mengakses fitur sesuai wewenangnya. Hal ini belum banyak dijumpai dalam penelitian lain yang hanya menyediakan akses tunggal untuk semua pengguna. yang menegaskan pentingnya pembatasan akses dalam pengembangan sistem informasi manajemen.
5. Penelitian ini [33] juga berbeda karena melibatkan proses evaluasi sistem terhadap cara kerja manual yang sebelumnya digunakan. Evaluasi dilakukan untuk mengukur peningkatan efisiensi dari sisi waktu kerja, akurasi pencatatan data, serta kemudahan penggunaan sistem oleh staf toko. pendekatan serupa digunakan untuk menilai sejauh mana sistem informasi berbasis *website* mampu memperbaiki kinerja operasional dibanding metode manual, terutama dalam sektor penjualan dan persediaan barang.