

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Menurut Wuyo (2024) dengan judul “*Rancang Bangun Sistem Pengelolaan Data dan Pelayanan Jasa pada Berkah Laundry Karangasem Berbasis Website*”. Penelitian ini menjelaskan bahwa pelibatan teknologi informasi di ranah Usaha Mikro Kecil Menengah (UMKM) sangat krusial untuk menunjang aktivitas operasional harian agar menjadi lebih cepat, efisien, dan dapat diandalkan dibandingkan metode pencatatan konvensional. Konsep yang relevan dengan skripsi ini adalah upaya digitalisasi sistem untuk memfasilitasi alur kerja internal yang terstruktur guna menggantikan metode pembukuan manual yang rentan terhadap kehilangan dokumen. Dalam pengembangan sistemnya, penelitian tersebut menerapkan model *Waterfall* yang linear dan sistematis, dengan teknik pengumpulan data yang meliputi studi pustaka, observasi langsung, serta wawancara mandiri. Sementara itu, untuk mengukur tingkat kepuasan pengguna berdasarkan dimensi kualitas sistem, metode pengujian yang digunakan adalah metode Webqual 4.0 yang dipadukan dengan analisis regresi linear berganda untuk mengukur berdasarkan dimensi kualitas.

Hasil akhir penelitian ini membuktikan bahwa implementasi *website* berbasis PHP dan database MySQL ini mempermudah pengelolaan data pelanggan serta rekapitulasi laporan, dengan hasil pengujian Webqual yang memperoleh nilai determinasi sebesar 0,810 atau 81,0%. Penelitian Wuyo (2024) dengan penelitian saat ini mempunyai kesamaan metodologi pengembangan perangkat lunak yang menggunakan model *Waterfall* serta memanfaatkan diagram UML untuk memetakan spesifikasi kebutuhan sistem. Perbedaannya terletak pada ranah sektor bisnis dan manajemen pelayanan jasa internal *single-channel*, sedangkan penelitian saat ini bergerak di sektor ritel produk elektronik dan menangani integrasi eksternal *multi-channel* secara otomatis.

Penelitian dari Pratama¹ et al., (2025) dengan judul “*Rancang Bangun Sistem Informasi Manajemen Stok dan Penjualan Berbasis Website pada Toko Vanila Aksesoris*” sangat relevan dan menjadi dasar pemikiran penting. Studi Pratama menyebutkan bahwa efisiensi pengelolaan perpindahan stok barang dan transaksi penjualan di era digital sangat krusial bagi kelangsungan operasional dan perkembangan bisnis. Konsep yang relevan dengan tugas akhir ini adalah perancangan sistem informasi manajemen persediaan secara otomatis dan *real-time* yang terpusat pada *back-end website*, bertujuan agar pemilik usaha dapat dengan mudah memantau ketersediaan barang. Metode yang dipakai dalam perancangan sistem ini menggunakan metode pengembangan *Waterfall* yang linear, dibangun menggunakan Bahasa pemrograman PHP berbasis *framework* Laravel serta database MySQL. Metode pengujian yang diterapkan dalam menguji keandalan fungsionalitas aplikasinya sepenuhnya menggunakan *Black Box Testing* dari sudut pandangan masukan dan keluaran sistem pengguna.

Berdasarkan hasil dari pengujian penelitian Pratama membuktikan bahwa sistem berbasis web terbukti sukses berjalan dengan baik sesuai keinginan pengguna, mampu menyajikan laporan stok secara otomatis, serta meningkatkan efisiensi harian operasional toko ritel fashion. Persamaan dengan skripsi ini adalah kesamaan penggunaan *framework* Laravel dan database MySQL, serta kesamaan tujuan fundamental dalam merancang sistem manajemen persediaan toko ritel secara terpusat. Di sisi lain, perbedaan utamanya terletak pada cakupan saluran penjualan, di mana penelitian terdahulu mengasumsikan transaksi penjualan hanya terjadi secara langsung pada satu saluran (*single-channel*), sedangkan penelitian saat ini melangkah lebih jauh dengan mengimplementasikan koneksi otomatis berbasis REST API Shopee Open Platform sebagai media sinkronisasi stok secara otomatis.

Penelitian dari (Utami et al., 2025) dengan judul “*Meningkatkan Manajemen Inventaris dan Transaksi dengan Solusi E-Commerce Terintegrasi: Studi Kasus Desasa Home Decor*”. Penelitian ini menyatakan bahwa penggunaan teknologi informasi sangat penting di sektor bisnis ritel yang kompetitif demi

memastikan manajemen inventaris dan transaksi dapat berjalan cepat serta menghasilkan laporan keuangan yang akurat. Konsep yang relevan dengan tugas akhir ini adalah implementasi *Application Programming Interface* (API) Shopee sebagai jembatan otomatisasi untuk menarik detail produk dan pesanan secara *real-time* langsung dari ekosistem *e-commerce*, sehingga mampu meminimalisir kesalahan input data (*human error*). Berbeda dengan dua penelitian sebelumnya, metode pengembangan sistem yang diterapkan disini menggunakan *Extreme Programming* (XP) yang mengutamakan kolaborasi erat tim dan pengujian berulang (*iterative testing*). Adapun metode pengujian yang digunakan meliputi pengujian fungsionalitas melalui *Black Box Testing* dan evaluasi non-fungsional dengan *User Acceptance Testing* (UAT).

Hasil implementasi menunjukkan seluruh fitur sistem informasi manajemen berbasis *website* dengan *framework* CodeIgniter 4 tersebut berjalan sesuai harapan tanpa kendala pada pengujian *Black Box*, serta mendapatkan skor kepuasan yang sangat tinggi mencapai 97,714% pada pengujian UAT sehingga dikategorikan sangat layak digunakan untuk meningkatkan efisiensi operasional toko. Persamaan utama antara penelitian Utami dengan penelitian ini terletak pada penggunaan arsitektur REST API Shopee Open Platform untuk melakukan integrasi, sinkronisasi data produk, serta pemantauan data transaksi penjualan *marketplace e-commerce*. Perbedaan utamanya pada fokus operasional di mana penelitian terdahulu hanya mengelola data pelayanan internal seperti transaksi toko fisik dan pelaporan sisa laba keuangan, sementara penelitian saat ini berfokus membangun *back-end website* sebagai *Single Source of Truth* (SSOT) data inventaris terpusat dan mengarahkan seluruh proses checkout penjualan melalui tautan rujukan eksternal Shopee.

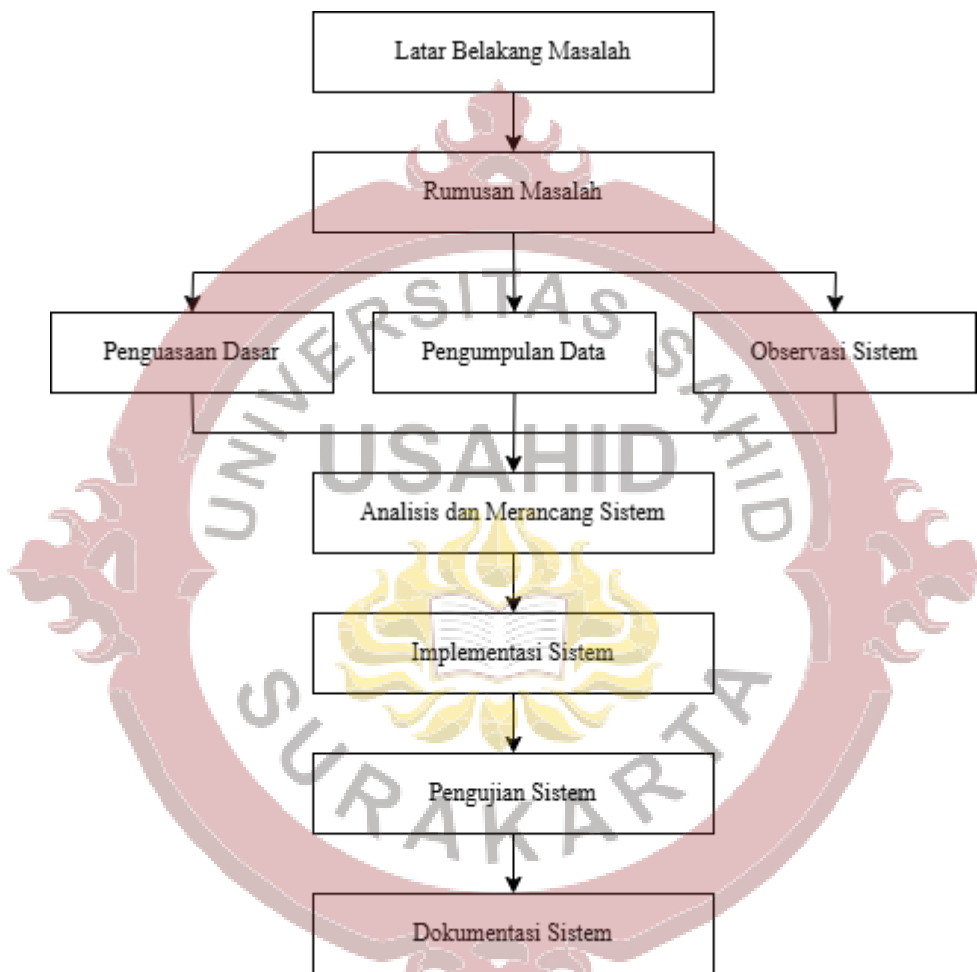
Ringkasan perbedaan jurnal yang direview dengan yang akan dilakukan dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Perbedaan jurnal yang direview dengan yang akan dilakukan

Penelitian	Persamaan	Perbedaan	Masalah Utama
Wuyo (2024)	Menggunakan diagram UML dan model <i>Waterfall</i> untuk memetekan spesifikasi sistem.	Fokus pada sistem internal toko (<i>single-channel</i>).	Rekapitulasi laporan harian yang masih manual menggunakan buku.
Pratama et al., (2025)	Memiliki tujuan untuk merancang sistem manajemen inventaris ritel yang terpusat.	Proses transaksi hanya berjalan pada satu saluran internal toko.	Pengelolaan Data & Pelayanan jasa yang masih manual.
Utami et al., (2025)	Mengimplementasikan arsitektur REST API Shopee Open Platform.	Menggunakan metode <i>Extreme Programming</i> (EP), bukan <i>Waterfall</i> .	Pengelolaan transaksi <i>e-commerce</i> secara manual yang memakan waktu.
Penelitian Sekarang	Mentransformasikan sistem pencatatan manual/konvensional yang rentan kesalahan menjadi sistem informasi berbasis <i>website</i> yang lebih terstruktur dan akurat.	Fokus memecahkan masalah <i>multi-channel</i> dengan solusi otomatis (SSOT).	Ketidak akuratan data stok akibat pembaruan <i>multi-channel</i> yang terpisah.

2.2 Kerangka Pemikiran

Kerangka pemikiran merupakan gambaran alur berpikir peneliti yang menjelaskan keterkaitan antar variable yang diteliti, sehingga memudahkan dalam perumusan, dalam penelitian ini kerangka pemikiran bisa dilihat Gambar 2.1.



Gambar 2. 1 Bagan Alur Kerangka Pemikiran

Uraian dari kerangka pemikiran sebagai berikut:

1. Latar Belakang

Toko Bengawan Komputer saat ini masih menggunakan pendataan manual berbasis *Microsoft Excel* dan menghadapi tantangan dalam mengelola stok yang tersebar di berbagai saluran seperti toko fisik dan Shopee.

2. Rumusan Masalah

Permasalahan utama dalam penelitian ini adalah bagaimana merancang *website* e-katalog sebagai etalase digital yang mengarahkan transaksi ke *marketplace* dan Bagaimana membangun sistem manajemen persediaan terpusat pada *back-end* untuk memantau data stok dari berbagai saluran penjualan.

3. Penguasaan Dasar

Pemahaman mendasar mengenai teknologi yang digunakan untuk membangun sistem, yaitu PHP, Laravel, Tailwind CSS, Rest API.

4. Pengumpulan Data

Pengumpulan data menggunakan metode observasi terhadap kegiatan yang berlangsung di Toko Bengawan Komputer dan melakukan metode wawancara tanya jawab langsung dengan pemilik toko, serta mengambil landasan teori dari buku, jurnal.

5. Observasi Sistem

Berdasarkan pengumpulan data yang telah dilakukan akan memberikan gambaran fitur-fitur apa saja yang dibutuhkan dalam perancangan dan pengembangan sistem agar sesuai dengan kebutuhan pengguna.

6. Analisis dan Merancang Sistem

Dari hasil analisis, dilakukan proses perancangan sistem yang mencakup desain arsitektur sistem menggunakan metode *Unified Modeling Language* (UML) untuk memetakan alur kerja sistem e-katalog.

7. Implementasi Sistem

Tahap ini merupakan proses penerapan dari perancangan ke dalam bentuk *website* menggunakan PHP, Laravel yang mengintegrasikannya secara langsung dengan API Shopee Open Platform.

8. Pengujian Sistem

Tahap ini dilakukan untuk memastikan seluruh fitur fungsional e-katalog dan sinkronisasi stok otomatis berjalan dengan baik menggunakan *Black Box Testing* serta menguji pengalaman pengguna melalui kuesioner *System Usability Scale* (SUS).

9. Dokumentasi Sistem

Tahap terakhir Menyusun dokumentasi lengkap mengenai hasil pengembangan sistem, mulai dari analisis kebutuhan, desain sistem, implementasi hingga hasil pengujian. Dokumentasi ini berguna sebagai laporan akhir penelitian dan referensi bagi pengembang selanjutnya.

2.3 Landasan Teori

2.3.1. Sistem

Terdapat beberapa definisi sistem, diantaranya:

1. Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan.
2. Sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional (dengan satuan fungsi dan tugas khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses tertentu.
3. Sistem merupakan sekumpulan atau group dari subsistem/bagian/komponen baik fisik maupun nonfisik yang saling berhubungan satu dengan yang lain dan bekerja bersama secara harmonis dalam mencapai satu tujuan (Pendidikan & Konseling, 2023).

2.3.2. E-Katalog

Sistem informasi e-katalog produk berbasis *website* merupakan platform digital terintegrasi yang dirancang untuk menyajikan informasi produk kepada pelanggan secara cepat, akurat, dinamis, dan interaktif guna menggantikan proses pengelolaan katalog manual atau cetak yang kurang efisien (Manulang, 2025).

2.3.3. Shopee Open Platform

Shopee *open* platform merupakan sebuah platform yang disediakan oleh Shopee bagi para pengembang (*developers*) untuk mengintegrasikan sistem internal toko dengan ekosistem *e-commerce*. Integrasi menggunakan API Shopee ini sangat berguna untuk melakukan sinkronisasi data secara otomatis dan meminimalisir

ketidaksesuaian informasi data barang antara toko fisik (*offline*) dan *marketplace* (*online*) (Gusdevi et al., 2024).

2.3.4. Website

Situs *website* menjadi salah satu wadah yang dicari-cari oleh khalayak banyak dalam mendapatkan informasi yang dibutuhkan. *Website* disebut juga situs web atau portal yang merupakan kumpulan halaman *website* yang berhubungan antara satu dengan lainnya dan dapat diakses dan dilihat oleh para pengguna internet (Adani, 2020).

2.3.5. REST API

Representational State Transfer (REST) merupakan salah satu gaya arsitektur perangkat lunak yang memanfaatkan protokol HTTP untuk pertukaran data. Implementasi REST API memungkinkan sistem untuk berkomunikasi secara efisien melalui metode standar seperti GET, POST, PUT, dan DELETE, dengan format pertukaran data yang ringan seperti JSON guna mendukung integrasi antar platform yang fleksibel (Polgan et al., 2023).

2.3.6. JavaScript Objek Notation (JSON)

Javascript Object Notation (JSON) merupakan format pertukaran data yang sering digunakan dalam implementasi sistem berbasis web dan web service saat ini. JSON memiliki karakter yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah dibaca oleh sistem komputer (Guna et al., 2023).

2.3.7. UML

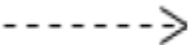
Menurut Narulita et al., (2024), *Unifield Modeling Language* adalah bahasa standar untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak. *Unified Modeling Language* (UML) merupakan sebuah standar penulisan atau semacam *blue print*

dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik.

2.3.7.1. Use Case Diagram

Use case diagram merupakan permodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Dengan demikian, *use case* diagram menjadi dasar yang penting dalam proses analisis dan perancangan sistem karena mampu memberikan gambaran umum. Simbol-simbol yang digunakan dalam *use case* diagram ditunjukkan pada Tabel 2.2.

Tabel 2. 2 Simbol-simbol *Use Case* Diagram

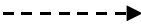
Simbol	Keterangan
	<i>Actor</i> : Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i> : Hubungan ini terjadi ketika perubahan pada elemen independen memengaruhi elemen dependen yang bergantung padanya.
	<i>Include</i> : Hubungan di mana objek anak mewarisi perilaku dan struktur data dari objek induk.
	<i>Generalization</i> : Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i> : Menunjukkan bahwa <i>use case</i> target memperluas perilaku <i>use case</i> sumber pada titik tertentu.
	<i>Association</i> : Apa yang menghubungkan antara objek satu dengan objek lainnya
	<i>System</i> : Menspesifikasikan paket yang menampilkan sistem secara terbatas

	<i>Use Case</i> : Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
	<i>Collaboration</i> : Interaksi aturan dan elemen yang menghasilkan perilaku lebih besar dari jumlah bagiannya (sinergi).
	<i>Note</i> : Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

2.3.7.2. Class Diagram

Class diagram dalam UML berfungsi sebagai menggambarkan struktur dari sistem dengan menampilkan kelas-kelas, atribut, metode, dan hubungan antar kelas, yang menunjukkan hubungan antar objek dan struktur data sistem. Simbol *Class* Diagram bisa dilihat pada Tabel 2.3.

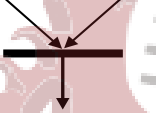
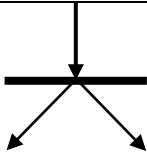
Tabel 2. 3 Simbol-simbol *Class* Diagram

Simbol	Keterangan
	<i>Generalization</i> : Relasi antar kelas dengan makna generalisasi – spesialisasi (umum – khusus).
Nama Kelas + Atribut + Operasi ()	<i>Class</i> : Himpunan dari objek - objek yang berbagi atribut serta operasi yang sama.
	<i>Depedency</i> : Relasi antarkelas dengan makna kebergantungan antar kelas.
	<i>Association</i> : Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai <i>multiplicity</i> .
	<i>Directed Association</i> : Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain biasanya juga disertai dengan <i>multiplicity</i> .

2.3.7.3. Activity Diagram

Activity Diagram adalah jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan aliran kerja atau aktivitas dalam suatu sistem atau proses. Diagram ini memodelkan proses bisnis dan urutan langkah operasional. Simbol-simbol yang digunakan dalam *activity diagram* ditunjukkan pada Tabel 2.4.

Tabel 2. 4 Simbol-simbol *Activity Diagram*

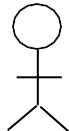
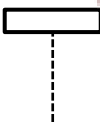
Simbol	Keterangan
	<i>Start point</i> : Diletakkan pada pojok kiri atas dan merupakan awal dari aktivitas.
	<i>Activities</i> : Menggambarkan sebuah proses dari suatu kegiatan.
	<i>Join</i> (penggabungan): Digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision</i> : Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	<i>Final</i> : Status akhir yang dilakukan sistem.
	<i>Swimlane</i> : Pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.
	<i>Fork</i> : Percabangan digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel.

2.3.7.4. Sequence Diagram

Sequence diagram menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antar objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. Membuat diagram *sequence* juga dibutuhkan untuk melihat skenario yang

ada pada *use case*. Simbol-simbol yang digunakan dalam *sequence diagram* ditunjukkan pada Tabel 2.5.

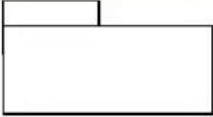
Tabel 2. 5 Simbol-simbol *Sequence Diagram*

Simbol	Keterangan
	Aktor: orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem.
	<i>Message</i> : Simbol mengirim pesan antar <i>class</i> .
	<i>Self Message</i> : Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> : Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
	<i>Lifeline</i> : Garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

2.3.7.5. Component Diagram

Component Diagram adalah visualisasi yang memetakan struktur bagian-bagian utama dari sebuah sistem perangkat lunak serta hubungan antar modul tersebut. Representasi ini menampilkan bagaimana unit-unit perangkat lunak terhubung dan berkolaborasi melalui antarmuka dan ketergantungan, sehingga memudahkan pemodelan arsitektur sistem secara modular dan terorganisir. Simbol *Component Diagram* bisa dilihat pada Tabel 2.6.

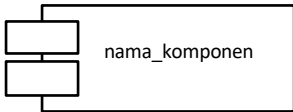
Tabel 2. 6 Simbol *Component Diagram*

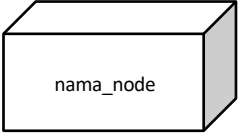
Simbol	Keterangan
	<i>Package</i> : Sebuah bungkus dari satu atau lebih komponen.
	<i>Component</i> : Semua komponen-komponen yang ada diletakkan didalam <i>node</i> untuk memastikan keberadaan posisi mereka.
	<i>Depedency</i> : Simbol yang menjelaskan sebuah keterkaitan antara komponen, satu komponen dengan yang lain.
	<i>Communication Path</i> : Simbol ini digunakan untuk mengarahkan relasi antar komponen, jika suatu komponen memiliki relasi atau keterkaitan dengan komponen lainnya maka dipakailah simbol <i>link</i> ini.

2.3.7.6. Deployment Diagram

Deployment diagram adalah representasi visual dalam UML yang menggambarkan konfigurasi fisik perangkat keras dan lunak. Diagram ini memetakan hubungan antar mode serta penempatan modul-modul sistem pada infrastruktur nyata, sehingga memberikan gambaran menyeluruh tentang arsitektur dan penyebaran elemen-elemen sistem dalam ekosistem perangkat keras. Simbol Deployment Diagram bisa dilihat pada Tabel 2.7.

Tabel 2. 7 Simbol *Deployment Diagram*

Simbol	Keterangan
	<i>Component</i> : semua komponen- komponen yang ada diletakkan didalam <i>node</i> untuk memastikan keberadaan posisi mereka.

	<i>Node</i>: menggambarkan bagian <i>hardware</i> dalam sebuah sistem. Notasi untuk <i>node</i> digambarkan sebagai sebuah kubus tiga dimensi.
	<i>Association</i>: Menghubungkan dua <i>node</i> yang mengindikasikan jalur komunikasi antara elemen-elemen <i>hardware</i>.

2.3.8. PHP (*Perl Hypertext Preprocessor*)

PHP merupakan bahasa pemrograman scripting open-source yang terutama digunakan untuk pengembangan web. Skript PHP dijalankan di server, menghasilkan HTML yang dikirimkan ke client (browser pengguna). Yang membedakan PHP dengan JavaScript dari sisi klien adalah bahwa kode dieksekusi di server, menghasilkan HTML yang kemudian dikirim ke klien (*The PHP Documentation Group*, 2026).

2.3.9. Laravel

Laravel adalah *framework* berbasis PHP dengan arsitektur Model-View-Controller (MVC) yang dirancang untuk membangun aplikasi *website* secara efisien, terorganisir, dan aman melalui fitur-fitur unggulan seperti Eloquent ORM, Blade templating engine, serta sistem migrasi database yang terbukti dapat mempercepat waktu pengembangan hingga 30% (Sinlae et al., 2024).

2.3.10. TailwindCSS

TailwindCSS adalah *framework utility-first* yang dirancang untuk mempercepat pengembangan antarmuka pengguna. Dengan kelas utilitas seperti *grid*, *flex*, dan *breakpoints*, TailwindCSS memungkinkan pengembang untuk menciptakan desain *responsif* secara efisien tanpa perlu menulis CSS kustom. *Framework* ini mendukung perubahan tata letak berdasarkan ukuran layar secara langsung, sehingga menjadi solusi ideal untuk implementasi desain. Keunggulan

TailwindCSS termasuk fleksibilitas tinggi, dokumentasi yang lengkap, dan kemudahan integrasi dengan teknologi *front-end* lainnya (*Tailwind CSS. "Utility-First Fundamentals."* 2026).

2.3.11. Pengujian Black Box

Pengujian *Black Box* adalah metode pengujian perangkat lunak yang berfokus untuk menguji fungsionalitas perangkat lunak tanpa perlu memahami detail kode sumber yang digunakan untuk membuatnya. Dalam pengujian *blackbox*, pengujian difokuskan pada fitur-fitur dan fungsi-fungsi yang seharusnya bekerja sesuai dengan spesifikasi tanpa perlu menyelidiki struktur internal atau kode program yang digunakan. Pada tahap pengujian perangkat lunak, penting untuk memilih teknik yang sesuai, efektif, serta mampu mengidentifikasi kesalahan yang mungkin terlewatkan, sehingga dapat meningkatkan kualitas perangkat lunak (Irham Soetomo Handayanto, 2024).

2.3.12. System Usability Scale (SUS)

System Usability Scale (SUS) merupakan metode evaluasi *usability* yang terdiri dari 10 pernyataan dengan skala Likert 1-5, mulai dari sangat tidak setuju hingga sangat setuju. SUS digunakan untuk mengukur persepsi pengguna terhadap kemudahan penggunaan antarmuka sistem secara subjektif, namun hasil akhirnya disajikan dalam bentuk skor kuantitatif pada rentang 0-100 (Hasudungan et al., 2024).

Kuesioner *System Usability Scale* (SUS) terdiri dari pernyataan positif dan negatif. Pernyataan bernomor ganjil merupakan pernyataan positif, sedangkan pernyataan bernomor genap merupakan pernyataan negatif. Skor dari masing-masing pernyataan dikonversi terlebih dahulu sesuai dengan ketentuan perhitungan SUS sebelum dilakukan perhitungan skor akhir.

$$\begin{aligned} \text{Skor SUS} = & ((Q1 - 1) + (5 - Q2) + (Q3 - 1) + (5 - Q4) + (Q5 - 1) + (5 - Q6) \\ & + (Q7 - 1) + (5 - Q8) + (Q9 - 1) + (5 - Q10)) \times 2,5 \\ & \dots\dots\dots \text{Persamaan 2.1} \end{aligned}$$

Nilai rata-rata skor SUS dari seluruh responden dihitung dengan membagi jumlah total skor SUS individu dengan jumlah responden:

$$\bar{x} = \frac{\sum x}{n} \dots\dots\dots \text{Persamaan 2.2}$$

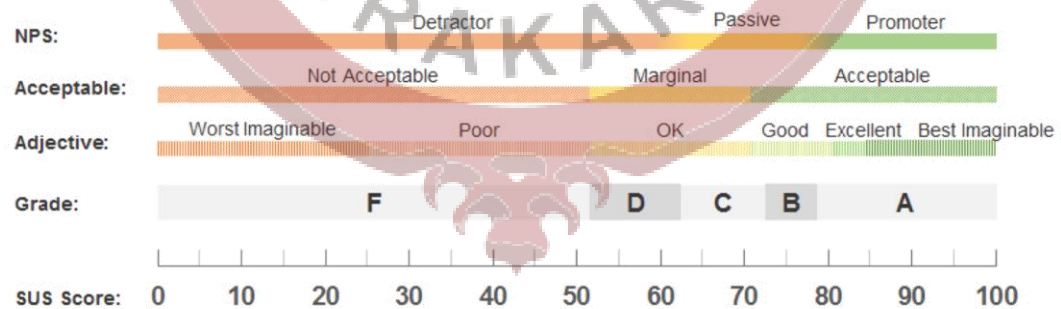
Dimana:

$\bar{x}$: Nilai rata-rata

$\sum x$: Jumlah skor SUS

n : Jumlah responden

Skor SUS yang diperoleh selanjutnya diinterpretasikan menggunakan skala interpretasi standar yang mencakup *acceptability ranges*, *grade scale*, dan *adjective rating*. Skala interpretasi ini digunakan untuk memberikan gambaran tingkat *usability* dan kelayakan sistem berdasarkan skor SUS yang diperoleh. Skala interpretasi hasil skor SUS dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Skala Interpretasi Hasil Skor SUS

Dalam sistem e-katalog, *usability* memiliki peran penting karena antarmuka pengguna menjadi media utama interaksi antara pengguna dan sistem, baik pada sisi pembeli maupun penjual (Salsabilla et al., 2024). Rincian skor SUS beserta

klasifikasi tingkat kelayakan meliputi *grade*, *percentile range*, *adjective rating*, *acceptability*, dan *Net Promoter Score* (NPS) disajikan pada Tabel 2.8.

Tabel 2. 8 Rincian Skor dan Skala Grade SUS

Grade	SUS	Percentile range	Adjective	Acceptable	NPS
A+	84.1 – 100	96 – 100	Best Imaginable	Acceptable	Promoter
A	80.8 – 84.0	90 – 95	Excellent	Acceptable	Promoter
A-	78.9 – 80.7	85 – 89	Good	Acceptable	Promoter
B+	77.2 – 78.8	80 - 84		Acceptable	Passive
B	74.1 – 77.1	70 - 79		Acceptable	Passive
B-	72.6 – 74.0	65 - 69		Acceptable	Passive
C+	71.1 – 72.5	60 - 64		Acceptable	Passive
C	65.0 – 71.0	41 - 59	OK	Marginal	Passive
C-	62.7 – 64.9	35 - 40		Marginal	Passive
D	51.7 – 62.6	15 - 24		Marginal	Detractor
F	0 – 51.6	0 - 14	Worst Imaginable / Poor	Not Acceptable	Detractor