

BAB II

LANDASAN TEORI

2. 1. Tinjauan Pustaka

Pada penelitian (Taufiq dkk., 2023) mengembangkan sistem informasi berbasis web di Pabrik Tahu Pak Gi untuk menggantikan pencatatan manual yang sering menimbulkan kesalahan dan kehilangan data. Sistem dibangun dengan Python, MySQL, dan metode Waterfall, berfungsi mengelola stok, produksi, penjualan, serta pesanan, dan hasil uji blackbox menunjukkan sistem berjalan baik. Penelitian (Widyastuti dkk, 2020) membuat sistem penjualan pakaian berbasis web di toko Syaunah Fashion menggunakan PHP dan MySQL dengan metode Waterfall. Tujuannya mempermudah pelanggan bertransaksi daring serta membantu admin dalam pengelolaan produk dan laporan. Pengujian blackbox membuktikan sistem berfungsi sesuai kebutuhan.

Pada penelitian (Desma Aipina & Harry Witriyono, 2022) mengembangkan aplikasi penjualan hijab berbasis web dengan Laravel dan Bootstrap menggunakan metode incremental. Sistem mampu mengelola produk, pesanan, dan pengguna dengan baik, meski fitur pembayaran belum terintegrasi penuh. Pada penelitian (Nur Sallamah dkk., 2024) berfokus pada pembuatan dan pengembangan *website* penjualan serta layanan manufaktur di PT. Emap Technology guna mendorong transformasi digital dalam kegiatan bisnis perusahaan. Pemanfaatan Laravel sebagai kerangka kerja utama pada sisi backend, serta MySQL untuk pengelolaan basis data.

Metode yang diterapkan adalah prototyping, yang memungkinkan pengembangan sistem dilakukan secara bertahap melalui penyesuaian dari masukan pengguna. Hasil pengujian alpha dan beta menunjukkan sistem bekerja dengan sangat baik, dengan tingkat kepuasan pengguna mencapai 93,33%, yang menandakan bahwa aplikasi ini mampu meningkatkan efisiensi kerja dan mendukung layanan digital perusahaan secara optimal. Perbandingan penelitian terdahulu dengan Oldshine Konveksi disajikan pada Tabel 2.1.

Tabel 2. 1 Tabel Perbandingan

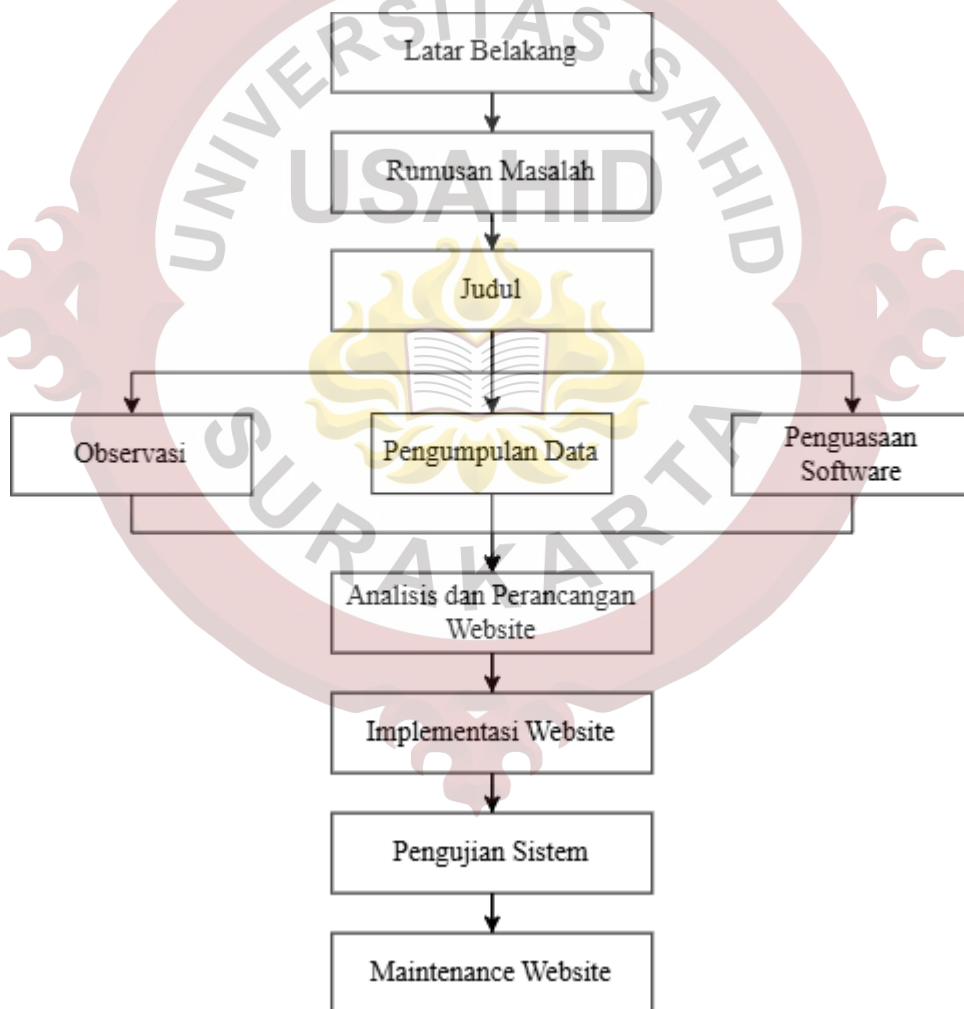
Peneliti	Tujuan Penelitian	Software	Perbandingan dengan Oldshine Konveksi
(Nasional dkk., 2023)	Membangun sistem informasi untuk pabrik tahu agar proses manual menjadi otomatis	Python, MySQL	Oldshine Konveksi menggunakan Laravel, PHP, dan Bootstrap yang lebih fleksibel dan menarik secara visual.
(Widyastuti, Handini., Siregar, Juarni., dan Ishak, 2020)	Mengembangkan sistem penjualan pakaian berbasis web	PHP, MySQL	Oldshine Konveksi menggunakan Laravel dan Bootstrap.
(Desma Aipina & Harry Witriyono, 2022)	Membuat sistem berbasis Laravel untuk manajemen penjualan	Laravel, Bootstrap, MySQL	Sama – sama menggunakan Laravel dan Bootstrap, namun Oldshine menggunakan metode Waterfall serta menambahkan uji Blackbox dan Beta untuk pengujian sistem.
(Nur Sallamah dkk., 2024)	Membangun sistem manufaktur berbasis web	Laravel, MySQL	Sama - sama memakai Laravel dan MySQL, tetapi Oldshine berfokus pada penjualan dan pemesanan konveksi

Berdasarkan Tabel 2.1, penelitian Oldshine Konveksi memiliki kedudukan yang berbeda dari penelitian – penelitian sebelumnya karena berfokus pada pengembangan *website* promosi dan pemesanan konveksi berbasis Laravel yang bertujuan memperluas jangkauan pasar serta memudahkan pelanggan dalam melakukan pemesanan secara

daring. Sistem dirancang menggunakan metode Waterfall agar proses pengembangan berlangsung terstruktur mulai dari tahap analisis hingga implementasi. Pengujian dilakukan menggunakan Blackbox dan *Beta Testing* guna menilai fungsionalitas dan pengalaman pengguna.

2. 2. Kerangka Berpikir

Menurut (Syahputri dkk., 2023) kerangka pemikiran adalah dasar pemikiran dari penelitian yang disintesiskan dari fakta-fakta, observasi dan kajian kepustakaan penelitian. Kerangka berpikir pada penelitian ini dapat disajikan pada Gambar 2.1.



Gambar 2. 1 Kerangka Berpikir

Berikut penjelasan dari Gambar 2.1.

a. Latar Belakang

Bagian ini menjelaskan bahwa proses pengelolaan transaksi dan pencatatan penjualan di Oldshine Konveksi masih dilakukan secara manual, sehingga proses transaksi, pencatatan data, dan pembuatan laporan penjualan belum berjalan secara efektif. Untuk mengatasi permasalahan tersebut, diperlukan sebuah *website* yang dapat mempermudah pengelolaan transaksi, pencatatan penjualan, dan penyusunan laporan penjualan secara terintegrasi.

b. Rumusan Masalah

Masalah utama yang dikaji yaitu bagaimana membuat *website* Oldshine Konveksi menggunakan *Framework* Laravel yang dapat mempermudah proses transaksi, pengelolaan data penjualan, dan pembuatan laporan penjualan?

c. Judul

Pembuatan *Website* Oldshine Konveksi Menggunakan *Framework* Laravel.

d. Pengumpulan Data

Pengumpulan data dilakukan dengan melakukan wawancara kepada Bapak Aryo Wiranjoyo selaku pemilik usaha guna memahami kebutuhan sistem, mengamati secara langsung proses kerja di konveksi, serta menelaah berbagai sumber pustaka yang relevan untuk memperkuat dasar teori dan mendukung perancangan sistem.

e. Observasi

Observasi dilakukan dengan mengamati secara langsung kegiatan di Oldshine Konveksi untuk memahami kebutuhan dan kendala yang ada. Hasil pengamatan ini menjadi dasar dalam menyesuaikan konsep serta metode pengembangan *website* serupa agar sesuai dengan kebutuhan.

f. Penguasaan Software

Berbagai perangkat lunak digunakan selama proses pengembangan *website* untuk mendukung setiap tahap pengerjaan. Di antaranya, Visual Studio Code berfungsi sebagai editor kode, Laravel digunakan sebagai kerangka kerja utama, dan XAMPP membantu menjalankan server lokal. MySQL dimanfaatkan untuk pengelolaan basis

data, sedangkan PHP, HTML, dan CSS digunakan dalam pembuatan struktur serta tampilan *website*. Bootstrap berperan dalam pengaturan antarmuka agar lebih responsif, draw.io digunakan untuk merancang diagram sistem, dan Canva membantu dalam pembuatan elemen visual seperti desain grafis *website*.

g. Analisis dan perancangan *website*

Tahap ini berfokus pada identifikasi kebutuhan sistem, baik fungsional maupun nonfungsional, yang mencakup perancangan basis data, diagram alur, *use case*, dan antarmuka agar sesuai dengan kebutuhan Oldshine Konveksi. Proses analisis melibatkan *Use Case Diagram* untuk menggambarkan interaksi pengguna dengan sistem dan *Activity Diagram* untuk menunjukkan alur proses. Sementara itu, tahap perancangan mencakup pembuatan model teknis seperti *Sequence Diagram*, *Class Diagram*, *Component Diagram*, dan *Deployment Diagram* untuk menjelaskan struktur, hubungan, serta penerapan sistem pada lingkungan fisik.

h. Implementasi *Website*

Tahap ini merupakan proses menerjemahkan rancangan yang telah dibuat ke dalam bentuk kode program menggunakan *framework* Laravel. Pada bagian ini, pengembangan *website* dilakukan dengan menambahkan fitur utama seperti pengelolaan produk, sistem pemesanan, serta penyusunan laporan penjualan.

i. Pengujian

Tahap ini bertujuan memastikan bahwa *website* berfungsi sesuai dengan kebutuhan yang telah ditetapkan. Proses pengujian dilakukan dengan metode blackbox testing dan *beta testing* untuk menilai kinerja setiap fitur, serta uji beta guna memperoleh umpan balik langsung dari pengguna.

j. *Maintenance*

Tahap pemeliharaan dilakukan setelah sistem selesai dikembangkan dan mulai digunakan. Kegiatannya mencakup perbaikan kesalahan atau bug, penambahan serta pembaruan fitur, dan memastikan *website* tetap berfungsi dengan baik sesuai dengan kebutuhan pengguna yang terus berkembang.

2.3. Teori Pendukung

2.3.1. Website

Pada penelitian (Apandi & Syalis Ibnih Melati Istini, 2023) *website* merupakan sekumpulan halaman yang dapat diakses melalui jaringan internet dan menampilkan berbagai informasi dalam bentuk teks, gambar, video, maupun elemen multimedia lain yang dapat diakses melalui komputer maupun perangkat seluler. Pembuatan *website* memiliki tujuan yang beragam, mulai dari memberikan informasi terkait produk dan jasa hingga dimanfaatkan untuk kepentingan edukasi maupun hiburan. Dalam bidang bisnis, *website* berfungsi sebagai media promosi sekaligus sarana transaksi secara daring, seperti pada e-commerce yang memfasilitasi aktivitas jual beli produk secara online.

2.3.2. PHP (*Hypertext Preprocessor*)

Menurut (Apandi & Syalis Ibnih Melati Istini, 2023) PHP adalah salah satu bahasa pemrograman *opensource* yang banyak digunakan untuk pengembangan aplikasi berbasis web. Bahasa ini dapat dijalankan di berbagai sistem operasi, termasuk Windows, Linux, maupun MacOS. Keunggulan utama PHP terletak pada kemampuannya dalam membangun aplikasi web secara lebih cepat dan praktis, serta kompatibilitasnya dengan beragam sistem basis data, seperti MySQL, Oracle, dan PostgreSQL.

2.3.3. HTML (*Hypertext Markup Language*)

Pada penelitian (Rina Noviana, 2022) menjelaskan bahwa *Hypertext Markup Language* (HTML) merupakan bahasa markup yang digunakan untuk mengatur penyajian informasi di internet serta memungkinkan pengguna berpindah dari satu halaman ke halaman lain melalui tautan.

2.3.4. CSS (*Cascading Style Sheet*)

Menurut (Rina Noviana, 2022) *CSS (Cascading Style Sheet)* adalah bahasa pemrograman yang berfungsi untuk mengatur gaya visual sekaligus mempercantik tampilan halaman web agar terlihat lebih menarik dan terstruktur. Dokumen CSS dapat berdiri sendiri, disisipkan langsung pada HTML, atau dijadikan file terpisah yang

dipanggil oleh HTML untuk mendefinisikan *style*. Melalui CSS, pengembang dapat menyesuaikan berbagai elemen, mulai dari warna teks, jenis huruf, spasi antarparagraf, ukuran kolom, hingga latar belakang halaman. Tidak hanya itu, CSS juga berperan dalam pengaturan tata letak, penyesuaian tampilan lintas perangkat, serta penerapan efek visual pada *website*. Sifatnya yang sederhana namun fleksibel menjadikan CSS mudah dipelajari, tetapi sangat efektif dalam mengendalikan penyajian dokumen HTML, baik untuk desain sederhana maupun kompleks.

2.3.5. Visual Studio Code

Menurut (Solihin dkk., 2024) *Visual Studio Code* merupakan editor kode sumber buatan Microsoft yang dapat digunakan pada sistem operasi *Windows*, *Linux*, dan *macOS*. Aplikasi ini memiliki fitur seperti *debugging*, integrasi dengan Git dan GitHub, penyorotan sintaks, penyelesaian kode otomatis, *snippet*, serta kemampuan refactoring kode. Visual Studio Code tergolong ringan namun memiliki kemampuan tinggi dalam menulis dan mengelola berbagai bahasa pemrograman seperti *JavaScript*, *Node.js*, *C++*, *C#*, *Python*, dan *PHP*, sehingga sangat membantu pengembang dalam proses pembuatan program.

2.3.6. XAMPP

Pada penelitian (Rina Noviana, 2022) menjelaskan bahwa XAMPP merupakan singkatan dari X-platform, Apache, MySQL, PHP, dan Perl, yaitu perangkat lunak web server *opensource* yang dapat dijalankan di berbagai sistem operasi seperti *Windows*, *Linux*, dan *MacOS*. Aplikasi ini berfungsi sebagai server lokal (*localhost*) yang memudahkan pengguna dalam proses perancangan, pengeditan, dan pengembangan aplikasi. XAMPP membantu pembuatan *website* dengan memberikan kemudahan, kecepatan, dan struktur kerja yang teratur, melalui tiga komponen utamanya yaitu *htdocs*, *Control Panel*, dan *PhpMyAdmin*.

2.3.7. Basis Data

Menurut (Fahzirah dkk., 2024) Basis data merupakan sekumpulan data yang tersusun secara terstruktur sehingga memudahkan proses penyimpanan, pengelolaan, dan pencarian informasi. Penggunaan basis data didukung oleh model data, skema,

serta bahasa kueri yang memungkinkan pengguna melakukan pengolahan dan akses terhadap data yang tersimpan. Pengelolaan basis data dilakukan melalui Sistem Manajemen Basis Data atau Database Management System (DBMS), yaitu perangkat lunak yang berfungsi mengatur penyimpanan, keamanan, dan akses data agar tetap terjaga konsistensi serta keandalannya.

2.3.8. MySQL

Menurut (Rina Noviana, 2022) MySQL adalah salah satu perangkat lunak *Database Management System* (DBMS) bersifat *open source* yang banyak digunakan karena mendukung akses multiuser, memiliki kemampuan *multithreaded*, serta tersedia secara gratis. Dari penjelasan tersebut dapat dipahami bahwa SQL merupakan bahasa khusus yang digunakan untuk berinteraksi dengan database, di mana fungsinya mencakup pembuatan serta pengelolaan data di dalamnya. SQL sendiri dimanfaatkan untuk berbagai operasi, termasuk melakukan pembaruan data pada basis data, sesuai dengan prinsip yang diterapkan dalam Relational Database Management System (RDBMS).

2.3.9. Bootstrap

Penelitian (Data Ardian Dirgantara & Rian Andrian, 2023) menjelaskan bahwa *bootstrap* merupakan salah satu *framework* CSS yang banyak digunakan dalam pengembangan *website* responsif. *Framework* ini menyediakan beragam fitur dan komponen siap pakai, seperti sistem grid, tipografi, formulir, tombol, dan elemen lainnya yang membantu mempermudah proses perancangan tampilan situs agar dapat menyesuaikan dengan berbagai perangkat.

2.3.10. Laravel

Pada penelitian (Saefudin dkk., 2023) menjelaskan bahwa Laravel merupakan *framework* berbasis *Model-View-Controller* (MVC) yang ditulis menggunakan PHP dan dikembangkan untuk meningkatkan kualitas perangkat lunak sekaligus menekan biaya pemeliharaan. *Framework* ini dibuat oleh Taylor Otwell dan dikenal sebagai salah satu *framework* PHP terbaik. Sebagai platform *open source*, Laravel menawarkan

sintaks yang rapi dan ekspresif, sehingga memudahkan pengembang dalam menyederhanakan sekaligus mempercepat proses pembangunan aplikasi web.

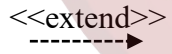
2.3.11. UML (Unified Modeling Language)

Unified Modeling Language (UML) adalah standar pemodelan visual dalam perancangan sistem perangkat lunak (Ramdany, 2024). Diagram UML yang digunakan meliputi:

a. Use Case Diagram

Use case diagram merupakan diagram yang menggambarkan interaksi antara pengguna dan sistem (Ramdany, 2024)). Simbol-simbol *use case diagram* disajikan pada Tabel 2.2.

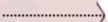
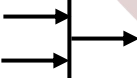
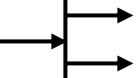
Tabel 2. 2 Simbol – simbol *use case diagram*

No	Simbol	Nama	Keterangan
1		Aktor	Aktor adalah entitas yang berinteraksi dengan sistem, yang bisa berupa orang, proses, atau sistem lain.
2		<i>Include</i>	<i>Include</i> menunjukkan penambahan fungsionalitas yang diperlukan agar <i>use case</i> utama dapat berjalan.
3		<i>Extend</i>	<i>Use case</i> yang ditambahkan tetap dapat berfungsi atau berdiri sendiri meskipun tanpa adanya <i>use case</i> tambahan tersebut.
4		Asosiasi	Interaksi antara aktor dengan <i>use case</i> , di mana aktor berperan serta dalam <i>use case</i> atau <i>use case</i> tersebut menjalin hubungan langsung dengan aktor.
5		<i>Use case</i>	digambarkan sebagai unit yang berinteraksi dengan aktor atau unit lain, dan biasanya diawali dengan kata kerja.
6		Generalisasi	menggambarkan keterkaitan antara dua fungsi, di mana salah satunya bersifat lebih umum sementara yang lain lebih khusus.

b. Activity Diagram

Activity diagram menurut (Ramdany, 2024) adalah gambaran alur aktivitas dalam sistem yang memperlihatkan urutan tindakan dan hubungan antar aktivitas, serta memvisualisasikan alur kerja dari awal hingga akhir proses. Simbol – simbol *activity* diagram dapat disajikan pada Tabel 2.3.

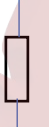
Tabel 2. 3 Simbol – simbol *activity* diagram

No	Simbol	Nama	Keterangan
1		<i>Activity</i>	Menggambarkan langkah atau tindakan dalam suatu proses untuk mencapai tujuan dalam sistem.
2		<i>Control flow</i>	Menggambarkan urutan langkah dan alur kompleks dengan percabangan, penggabungan, serta pembagian aliran.
3		<i>Object flow</i>	Aliran data antar aktivitas dalam sistem menggunakan panah yang menunjukkan perpindahan data.
4		<i>Start Point</i>	Menandakan titik awal dari alur aktivitas dalam sebuah proses.
5		<i>End point</i>	Menandakan titik akhir dari alur aktivitas dalam sebuah proses.
6		<i>Join</i>	menggabungkan beberapa alur paralel menjadi satu alur. Proses selanjutnya baru berjalan setelah semua alur sebelumnya selesai.
7		<i>Fork</i>	pemisahan alur menjadi aktivitas yang berjalan secara paralel, yang kemudian disatukan kembali menggunakan simbol join.
8		<i>Decision</i>	percabangan alur proses berdasarkan kondisi tertentu untuk menentukan langkah selanjutnya.

c. *Sequence Diagram*

Sequence diagram pada penelitian (Ramdany, 2024) menunjukkan komunikasi antara objek-objek dalam dan luar sistem melalui pengiriman pesan yang tersusun berdasarkan urutan waktu. Simbol – simbol *sequence diagram* dapat disajikan pada Tabel 2.4.

Tabel 2. 4 Simbol – simbol *sequence diagram*

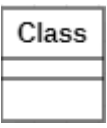
No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Aktor berfungsi memulai interaksi dengan sistem dengan mengirimkan pesan ke objek dalam diagram.
2		<i>A focus of control & a life line</i>	Lifeline menunjukkan keberadaan objek selama proses berlangsung, sedangkan focus of control menandai saat objek menjalankan suatu aktivitas.
3		<i>A message</i>	Simbol ini digunakan untuk menggambarkan urutan dan alur pertukaran informasi dalam sistem.
4		<i>Object</i>	Objek yang berinteraksi dalam pesan.

d. *Class Diagram*

Menurut (Ramdany, 2024) *class diagram* digunakan untuk menggambarkan struktur sistem melalui hubungan antar kelas yang terdapat di dalamnya. Diagram ini menampilkan atribut, metode, serta keterkaitan antar kelas yang membentuk suatu sistem. Setiap kelas memiliki fungsi dan tanggung jawab tertentu yang mendukung jalannya proses dalam sistem. Keberadaan *class diagram* membantu pengembang memahami alur interaksi antar objek sehingga proses perancangan dan pengembangan sistem dapat dilakukan secara lebih terstruktur dan sistematis. Selain itu, diagram ini

dapat digunakan sebagai acuan dalam pembuatan basis data dan implementasi kode program agar sesuai dengan kebutuhan sistem yang dirancang. Simbol – simbol *class* diagram dapat disajikan pada Tabel 2.5.

Tabel 2. 5 simbol – simbol *class* diagram

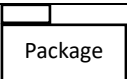
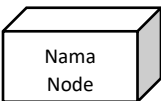
No	Simbol	Nama	Keterangan
1		<i>Class</i>	Menggambarkan struktur kelas dalam pemrograman berorientasi objek, terdiri dari nama kelas, atribut, dan metode, untuk memperlihatkan fungsi dan organisasi sistem secara jelas.
2		<i>Association</i>	menunjukkan hubungan antar kelas, digambarkan dengan garis yang menghubungkan kelas-kelas tersebut dan panah terbuka di ujungnya yang menandakan aliran pesan satu arah.
3		<i>Aggregation</i>	Menggambarkan hubungan keseluruhan dan bagian dalam UML, di mana bagian bisa berdiri sendiri, ditandai dengan garis yang berujung berlian terbuka.
4		<i>Composition</i>	Hubungan di mana satu kelas sangat bergantung pada kelas lain dan tidak bisa berdiri sendiri
5		<i>Generalization</i>	Menggambarkan hubungan pewarisan antara kelas spesifik dan kelas umum, di mana kelas spesifik mewarisi atribut dan metode kelas umum.
6		<i>Dependency</i>	Menunjukkan bahwa sebuah kelas membutuhkan kelas lain untuk menjalankan operasinya.

e. *Deployment Diagram*

Pada penelitian (Kristianingrum & Al-Fadillah, 2022) *deployment diagram* merupakan salah satu jenis diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan konfigurasi fisik sistem ketika aplikasi dijalankan. Informasi yang ditampilkan menjelaskan bagaimana setiap komponen atau artefak perangkat lunak ditempatkan dan diimplementasikan pada node tertentu sehingga seluruh bagian sistem dapat saling berkomunikasi dan bekerja sesuai fungsinya. Penyusunan Deployment Diagram dilakukan pada tahap perancangan sebagai acuan implementasi sebelum proses pengkodean (coding), sehingga struktur penyebaran

aplikasi pada lingkungan operasional dapat dipahami dengan lebih jelas. Simbol – simbol *deployment* diagram dapat disajikan pada Tabel 2.6.

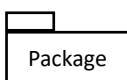
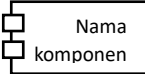
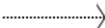
Tabel 2. 6 simbol – simbol *deployment* diagram

No	Simbol	Nama	Keterangan
1		<i>Package</i>	Merupakan sebuah bungkus dari satu atau lebih node.
2		<i>Node</i>	Node berfungsi sebagai representasi dari elemen fisik berupa perangkat keras atau lingkungan tempat perangkat lunak dijalankan.
3		<i>Dependence</i>	Hubungan ini menunjukkan bahwa suatu elemen bergantung pada elemen lain untuk menjalankan fungsinya.
4		<i>Link</i>	Relasi antar node.

f. *Component* Diagram

Menurut (Bhatt & Nandu, 2021) *Component* Diagram adalah jenis diagram dalam UML yang digunakan untuk menunjukkan bagian fisik dari sebuah sistem, seperti file, pustaka, atau dokumen yang disimpan dalam node. Simbol – simbol *component* diagram dapat disajikan pada Tabel 2.7.

Tabel 2. 7 Simbol – simbol *component* diagram

No	Simbol	Nama	Keterangan
1		<i>Package</i>	Merupakan sebuah bungkus dari satu atau lebih node.
2		<i>component</i>	Komponen sistem.
3		<i>Dependence</i>	Hubungan ketergantungan antar komponen ditunjukkan oleh arah panah menuju komponen yang digunakan.
4		<i>Interface</i>	Komponen berkomunikasi melalui jalur perantara yang telah ditentukan.
		<i>Link</i>	Relasi antar komponen.

2.3.12. Canva

Menurut (Syahputra dkk., 2024), Canva merupakan aplikasi berbasis web yang digunakan untuk membuat berbagai desain visual seperti poster, video, presentasi, dan konten media sosial. Pengguna dapat memilih dari berbagai template yang tersedia, baik yang gratis maupun berbayar. Layanan ini bisa digunakan melalui *website* atau perangkat mobile seperti Android dan Iphone sehingga proses pembuatan desain seperti brosur dan cover eBook menjadi lebih mudah dan fleksibel. Dalam pembuatan *website*, Canva dimanfaatkan untuk membuat mockup *website* yaitu rancangan tampilan awal yang menggambarkan susunan elemen antarmuka secara visual sebelum tahap pengkodean dilakukan.

2.3.13. Metode PIECES

Pada penelitian (Natalia dkk., 2024) Metode PIECES merupakan salah satu pendekatan yang digunakan untuk mengevaluasi dan mengidentifikasi permasalahan dalam sistem informasi sehingga dapat menjadi dasar dalam pengambilan keputusan untuk pengembangan maupun perbaikan sistem. Pendekatan ini membantu menemukan permasalahan secara lebih terperinci dengan meninjau beberapa aspek penting yang terdapat dalam sistem. Aspek yang dianalisis meliputi kinerja sistem, kualitas informasi, aspek ekonomi, pengendalian atau keamanan, tingkat efisiensi, serta kualitas layanan yang diberikan kepada pengguna. Analisis terhadap keenam aspek tersebut dikenal sebagai analisis PIECES, yang terdiri dari *Performance, Information, Economy, Control, Efficiency, dan Service*.

2. 4. Pengujian Sistem

Pada penelitian ini akan dilakukan pengujian eksternal dan internal untuk menilai kegunaan *website*. Pengujian ini menggunakan dilakukan dengan 2 cara yaitu:

a. Blackbox Testing

Menurut (Samdono dkk., 2024) Black Box Testing merupakan metode pengujian yang menilai fungsi aplikasi dari sudut pandang pengguna tanpa menguji kode program. Pengujian ini bertujuan memastikan setiap fitur menghasilkan keluaran yang sesuai dengan masukan yang diberikan.

Pada penelitian (Samdono dkk., 2024) menjelaskan bahwa tingkat efektivitas digunakan untuk mengukur kelayakan aplikasi berdasarkan kesesuaian fungsi dengan kebutuhan perusahaan. Nilai efektivitas dinyatakan dalam bentuk persentase, sehingga semakin tinggi hasil yang diperoleh maka semakin baik tingkat kelayakan aplikasi. Perhitungan efektivitas dilakukan menggunakan rumus yang diterapkan pada setiap skenario pengujian sebagaimana ditunjukkan pada persamaan berikut:

$$\text{Nilai Efektivitas Tabel} = \left(\frac{\sum \text{Pengujian Sesuai}}{\sum \text{Jumlah Skenario Pengujian}} \right) \times 100\% \quad (1)$$

$$\text{Nilai Efektivitas Website} = \left(\frac{\sum \text{Nilai Tabel A} + \text{Nilai Tabel B} + \dots + \text{Nilai Tabel n}}{\sum \text{Jumlah Tabel Diuji}} \right) \quad (2)$$

b. Beta Testing

Pada penelitian (Kresensia dkk., 2025) Pengujian beta dilakukan secara langsung pada lingkungan penggunaan sebenarnya dengan melibatkan responden untuk memberikan penilaian terhadap aplikasi yang dikembangkan melalui kuesioner. Proses pengujian ini bertujuan memperoleh gambaran tingkat penerimaan dan kualitas aplikasi berdasarkan hasil isian pengguna. Setiap pertanyaan dalam kuesioner dinilai menggunakan *Skala Likert* 1–5, di mana nilai 1 menunjukkan Sangat Tidak Setuju dan nilai 5 menunjukkan Sangat Setuju.

Tabel 2. 8 Bobot Nilai

No	Skala	Tingkat Kepuasan
1	5	Sangat Setuju
2	4	Setuju
3	3	Netral
4	2	Tidak Setuju
5	1	Sangat Tidak Setuju

Skor dihitung menggunakan rumus berikut.

$$Y = \left(\frac{\sum \text{Skor Diperoleh}}{\sum \text{Skor Maksimal}} \right) \times 100\% \quad (3)$$

Y = Persentase Hasil Uji

Kriteria kelayakan *website* disajikan pada Tabel 2.9.

Tabel 2. 9 Kriteria Skor

Persentase (%)	Interpretasi
80-100	Sangat Layak
60 – 79,99	Layak
40 – 59,99	Cukup Layak
20 – 39,99	Kurang Layak
0 – 19,99	Tidak Layak

Tabel 2. 10 Tabel Pertanyaan

No	Pertanyaan
1	Apakah <i>Website</i> Oldshine Konveksi dapat diakses dan berfungsi dengan baik?
2	Apakah <i>Website</i> Oldshine Konveksi mudah digunakan oleh pengguna?
3	Apakah Menu dan navigasi pada <i>website</i> Oldshine Konveksi mudah dipahami?
4	Apakah Informasi produk dan layanan yang tersedia pada <i>website</i> Oldshine Konveksi sudah lengkap dan jelas?
5	Apakah Fitur-fitur pada <i>website</i> Oldshine Konveksi berjalan dengan baik tanpa mengalami kendala atau error?
6	Apakah Proses pemesanan produk melalui <i>website</i> Oldshine Konveksi berlangsung dengan cepat dan lancar?
7	Apakah Penggunaan <i>website</i> Oldshine Konveksi membantu memudahkan dalam melakukan pemesanan?
8	Apakah Pelanggan puas menggunakan <i>website</i> Oldshine Konveksi untuk melakukan pemesanan produk?