

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang sebelumnya dilakukan oleh Soma Setiawan, et al., (2022), yang dipublikasikan dalam jurnal berjudul *Implementasi QR Code untuk Monitoring Proses Produksi Bagi Konsumen Pada Usaha Sablon*, membahas tentang penerapan *QR Code* sebagai solusi untuk memudahkan pelanggan dalam melakukan pengecekan mandiri terkait proses produksi pesanan mereka. Dengan adanya *QR Code* ini, pelanggan tidak perlu mengandalkan proses manual yang sering kali memakan waktu. Cukup dengan memindai kode yang terdapat pada bukti pemesanan menggunakan *smartphone*, mereka dapat langsung mengetahui status produksi pesanan mereka. Dengan menerapkan sistem ini, diharapkan operasional bagi pelaku usaha sablon dapat lebih efisien, sekaligus memberikan kenyamanan tambahan bagi pelanggan, karena mereka dapat memantau status pesanan secara *real-time* [3].

Penelitian lain dilakukan oleh Avivul Ardian Sahid, et al., (2023), yang dipublikasikan dalam jurnal berjudul *Rancang Bangun SmartBox Menggunakan Metode Prototype untuk Penerimaan Paket Berbasis Internet of Things*. Penelitian ini bertujuan untuk mengatasi masalah penerimaan paket ketika pemilik rumah tidak berada di tempat. Dalam upaya itu, mereka merancang sebuah *SmartBox* yang terhubung dengan internet. Sistem ini

memanfaatkan teknologi *Internet of Things* (IoT) dan dilengkapi dengan mekanisme penguncian otomatis yang hanya dapat diakses oleh penerima yang sah. Metode yang digunakan dalam penelitian ini adalah metode prototype, dan hasil uji menunjukkan bahwa sistem berfungsi dengan baik. *SmartBox* ini mampu memberikan notifikasi kepada pengguna saat paket tiba, serta menawarkan solusi praktis dan aman untuk proses penerimaan paket secara mandiri. Inovasi ini diharapkan dapat mengatasi tantangan dalam sistem pengiriman modern dan memberikan kenyamanan lebih bagi para pengguna [4].

Penelitian lain yang dilakukan oleh Farah Dila Faza, et al.. (2023), yang dipublikasikan dalam jurnal Berjudul *Smart Box* Penerima Paket Berbasis *Website* Menggunakan Esp32-Cam Dan Notifikasi Telegram. Pengujian telah menunjukkan bahwa sistem ini mampu berfungsi dengan baik dalam menerima paket dan memberikan notifikasi secara *real-time*. Diharapkan, adopsi teknologi ini dapat meningkatkan efisiensi dalam proses penerimaan paket, mengurangi risiko kehilangan barang, dan mengoptimalkan layanan logistik. Dengan integrasi IoT dan sistem pemantauan berbasis *website*, *Smart Box* ini menawarkan solusi inovatif untuk pengiriman barang yang lebih aman dan efektif [5].

2.2 Landasan Teori

2.2.1. QR Code

Kode QR (*Quick Response*) adalah jenis kode dua dimensi yang dirancang untuk menyimpan beragam informasi, seperti data biner, numerik, dan alfanumerik, dalam bentuk gambar digital. Contoh informasi yang bisa disimpan dalam kode QR meliputi nomor telepon, alamat situs web, dan data pribadi. Tujuan utama kode ini adalah untuk menyampaikan informasi secara efisien serta memberikan respons yang cepat. Sebagai perkembangan dari barcode atau kode batang, kode QR memiliki keunggulan dalam menyimpan data baik secara horizontal maupun vertikal, sehingga menawarkan kapasitas penyimpanan yang jauh lebih besar dibandingkan barcode konvensional [6].

2.2.2. Website

Website merupakan kumpulan halaman yang dirancang untuk menampilkan berbagai jenis informasi, seperti teks, gambar statis atau bergerak, animasi, dan suara. Halaman-halaman ini, baik yang bersifat statis maupun dinamis, saling terhubung membentuk rangkaian yang terintegrasi, di mana setiap halaman dihubungkan melalui jaringan yang ada [7].

2.2.3. Database

Database dapat diartikan sebagai kumpulan data yang saling berhubungan dan disimpan secara terorganisir, sehingga

menghindari pengulangan (redundansi) yang tidak perlu. Tujuannya adalah untuk memenuhi berbagai kebutuhan [8].

2.2.4. Visual Studio Code

Visual Studio Code adalah editor kode sumber yang dikembangkan oleh Microsoft dan tersedia untuk Windows, Linux, serta macOS. Aplikasi ini mempermudah proses penulisan kode dengan mendukung berbagai bahasa pemrograman, serta memberikan variasi warna yang sesuai dengan fungsi masing-masing bagian kode. Selain itu, salah satu fitur menarik dari Visual Studio Code adalah kemampuan untuk menambahkan ekstensi, yang memungkinkan para pengembang untuk memperluas fungsionalitas aplikasi dengan fitur-fitur tambahan. Visual Studio Code bersifat open source, artinya siapa pun dapat mengakses kode sumbernya dan berkontribusi pada pengembangan aplikasi ini. Kode sumber juga dapat ditemukan melalui tautan di GitHub, yang menjadikan Visual Studio Code sangat populer di kalangan para pengembang dan mendorong pertumbuhan aplikasi ini di masa depan [9].



Gambar 2.1 logo Virtual Studio Code

2.2.5. PHPMyAdmin

Sebuah aplikasi yang dirancang untuk menyederhanakan manajemen MySQL. Memanfaatkan PHPMyAdmin untuk membuat database, membuat table, memasukan , mengekspor, dan memperbaharui data dengan GUI dan jauh lebih mudah tanpa memerlukan tambahan query SQL secara manual.



Gambar 2.2 logo phpMyAdmin

2.2.6. Laragon

Laragon adalah perangkat lunak sumber terbuka yang dapat digunakan di berbagai sistem operasi. Berfungsi sebagai server virtual atau localhost, Laragon mendukung sejumlah sistem operasi yang berbeda. Salah satu keunggulan Laragon adalah kemampuannya untuk memungkinkan pengguna memilih domain sesuai dengan keinginan mereka, sering kali disebut dengan *pretty URLs*. Aplikasi ini sangat bermanfaat dalam pengelolaan aplikasi berbasis web [10].

2.2.7. CodeIgneter

CodeIgneter adalah sebuah *Web Application Framework* (WAF) yang dirancang khusus untuk mempermudah pengembang

web dalam menciptakan aplikasi web. Codeigniter terdiri dari kumpulan pustaka kode dan alat yang diintegrasikan untuk membentuk suatu kerangka kerja (*framework*) yang efisien [11].



Gambar 2.3 Logo *CodeIgniter*

2.2.8. MySQL

MySQL adalah sebuah server basis data yang sangat bermanfaat. Dengan menggunakan MySQL, kita dapat memenuhi kebutuhan pribadi atau bisnis tanpa harus membeli atau membayar lisensi. Selain fungsinya sebagai server basis data, MySQL juga merupakan perangkat lunak yang memungkinkan akses ke database yang telah dikonfigurasi sebagai server. Hal ini berarti bahwa program kita berfungsi sebagai klien dalam sistem tersebut.

2.2.9. API Key

API Key adalah kode unik yang digunakan untuk mengidentifikasi dan mengautentikasi aplikasi atau pengguna saat mengakses layanan tertentu. Dalam penelitian ini, API Key

digunakan untuk menghubungkan sistem *Smart Box* dengan layanan Wablas, sehingga sistem dapat mengirim notifikasi WhatsApp secara otomatis kepada penerima paket. Penggunaan API Key memastikan bahwa hanya aplikasi yang sah yang dapat mengakses layanan tersebut, sehingga keamanan dan kontrol akses tetap terjaga.

2.2.10. UML (*Unified Modeling Language*)

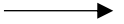
Menurut Pressman (2010:841), *Unified Modeling Language* (UML) adalah bahasa standar yang digunakan untuk membuat diagram perangkat lunak. UML berfungsi untuk memvisualisasikan, mendefinisikan, mengembangkan, dan mendokumentasikan artefak dari sistem perangkat lunak. Secara analogi, seperti seorang arsitek yang menyusun denah untuk sebuah proyek konstruksi, arsitek perangkat lunak juga menggunakan diagram UML untuk membantu pengembang dalam membangun perangkat lunak [12].

Unified Modeling Language merupakan metode pemodelan visual yang umum digunakan dalam perancangan dan pengembangan perangkat lunak berorientasi objek. UML berperan sebagai standar penulisan atau cetak biru yang mencakup proses bisnis dan penulisan kelas-kelas dalam bahasa yang spesifik.

Ada beberapa diagram UML yang sering digunakan dalam pengembangan sistem, di antaranya adalah:

1. *Use Case*: Diagram ini menggambarkan fungsionalitas yang diharapkan dari suatu sistem dan merepresentasikan interaksi antara aktor dan sistem. Dalam *Use Case*, aktor diartikan sebagai entitas yang bisa berupa manusia atau sistem lain yang melakukan tugas dalam konteks sistem yang sedang dikembangkan.

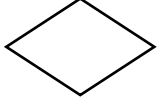
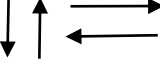
Tabel 2.1 Simbol *Use Case Diagram*

Simbol	Keterangan
	Aktor: Berperan sebagai perwakilan orang, sistem lain, atau alat yang terlibat dalam komunikasi dengan <i>Use Case</i> .
	<i>Use Case</i> : Merupakan representasi dari interaksi dan hubungan antara sistem dan aktor.
	Asosiasi: Merupakan gambaran mengenai hubungan antara aktor dan <i>Use Case</i> .
	Generalisasi: Menunjukkan spesialisasi dari aktor yang memungkinkan mereka berpartisipasi dalam <i>Use Case</i> tertentu.
	Menunjukkan bahwa suatu <i>Use Case</i> seluruhnya merupakan fungsionalitas dari <i>Use Case</i> lainnya
	Menunjukkan bahwa suatu <i>Use Case</i> merupakan tambahan fungsional dari <i>Use Case</i> lainnya jika suatu kondisi terpenuhi

2. *Activity Diagram*: Sebuah diagram aktivitas dalam UML menggambarkan perilaku dinamis suatu sistem atau bagian dari

sistem dengan menampilkan aliran kontrol antara berbagai tindakan yang dilakukan oleh sistem.

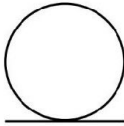
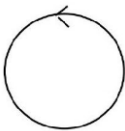
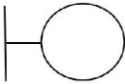
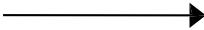
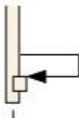
Tabel 2.2 Simbol *Activity Diagram*

No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperhatikan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
3		<i>Activity final Node</i>	Bagaimana objek dibentuk dan diakhiri
4		<i>Decision</i>	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus di ambil pada kondisi tertentu.
5		<i>Line connector</i>	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

3. *Sequence Diagram*: menggambarkan interaksi antara objek-objek di dalam dan sekitar sistem termasuk pengguna, tampilan, dan lain-lain dalam bentuk pesan yang diatur sesuai waktu. Diagram ini biasanya digunakan untuk memvisualisasikan skenario atau langkah-langkah yang diambil sebagai respons

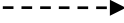
terhadap suatu peristiwa, dengan tujuan untuk menghasilkan output tertentu.

Tabel 2.3 Simbol *Sequence Diagram*

No	Simbol	Keterangan
1.		<i>Entity Class</i> merupakan bagian dari sistem yang berfungsi sebagai deskripsi awal sistem, berisi kumpulan kelas berbentuk entitas yang menjadi dasar dalam pembuatan database.
2.		<i>ControlClass</i> berfungsi sebagai penghubung antara Boundary Class dan tabel dalam sistem.
3.		<i>Boundary Class</i> adalah gambaran dari penggambaran sebuah tabel
4.		Pesen atau <i>message</i> menunjukkan proses pengiriman pesan antar <i>class</i>
5.		<i>Self message</i> menunjukkan proses pengiriman suatu pesan yang akan dikirim ke objek itu sendiri.
6.		<i>Activation</i> menggambarkan suatu objek yang melakukan sebuah aksi/eksekusi operasi
7.		<i>Lifeline</i> merupakan garis titik yang terhubung ke objek sepanjang garis lifeline memiliki aktivitas

4. *Class Diagram*: Diagram ini menggambarkan struktur serta mendeskripsikan kelas, paket, dan objek yang saling berhubungan. Didalamnya, Anda dapat melihat hubungan seperti pewarisan dan asosiasi lainnya.

Tabel 2.4 Simbol *Class Diagram*

No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek individu (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
5		<i>Realization</i>	Operasi yang benar benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan Dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen tidak mandiri.

No	Gambar	Nama	Keterangan
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dan objek lainnya

2.2.10. Black Box Testing

Black Box Testing atau yang dikenal sebagai pengujian perilaku, merupakan metode pengujian di mana struktur internal dan logika perangkat lunak yang diuji tidak diketahui oleh penguji. Pengujian ini berdasarkan pada spesifikasi kebutuhan, sehingga tidak memerlukan analisis kode. Pengujian ini dilakukan dari sudut pandang pengguna akhir. Beberapa jenis pengujian yang termasuk dalam *black box testing* antara lain partisi, analisis nilai batas, grafik penyebab dan efek, pengujian orthogonal array, pengujian transisi negara, serta *fuzzing*.

2.2.11. White Box Testing

White Box Testing atau pengujian struktural adalah metode di mana perangkat lunak yang diuji bersifat transparan bagi penguji. Dalam pengujian ini, pengujian dirancang dari perspektif pengembang, karena struktur internal kode dikenali. Penguji melakukan pengujian pada seluruh bagian kode yang dapat diuji, dengan tujuan untuk mengidentifikasi kesalahan logis dalam kode sumber perangkat lunak.