

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Penelitian ini berperan dalam memberikan pemahaman mengenai cara perancangan dan penerapan *website* guna menunjang fungsi *e-commerce* pada situs Rafli Garden.

Penerapan *website* pemesanan online dan informasi stok barang telah menjadi solusi efektif bagi pelaku usaha kecil dan menengah (UMKM), khususnya di sektor tanaman hias. Sistem ini memungkinkan penyajian informasi produk yang lebih terstruktur melalui katalog digital, di mana pelanggan dapat melihat ketersediaan stok secara *real-time* serta melakukan pemesanan tanpa harus datang langsung ke lokasi. Dengan demikian, pelaku usaha dapat meningkatkan efisiensi dalam pengelolaan persediaan dan transaksi, serta memperluas jangkauan pasar melalui saluran digital. Pemanfaatan teknologi ini tidak hanya mendorong profesionalisme dalam pengelolaan bisnis, tetapi juga menjadi langkah awal dalam transformasi digital UMKM di bidang agrikultur urban.

Penelitian yang dilakukan oleh Dewa Gede et al. (2024) dalam jurnal Rancang Bangun Web Ecommerce Pada Dewa Bike Berbasis Framework Laravel membahas tentang pengembangan sistem *e-commerce* menggunakan *framework* Laravel pada toko sepeda Dewa Bike. Penelitian ini menunjukkan bahwa pemanfaatan Laravel mampu mempercepat proses pengembangan berkat fitur bawaan seperti *routing*,

Eloquent ORM, dan *Blade template engine*, sehingga menghasilkan aplikasi penjualan online yang lebih terstruktur dan mudah dikelola oleh pengelola usaha [3].

Penelitian yang dilakukan oleh Aditya et al. (2022) dalam jurnal Pengembangan Sistem Informasi E-Commerce dengan Pemanfaatan API Midtrans menggunakan Framework Laravel (Studi Kasus: Byboot.id) mengembangkan sistem *e-commerce* dengan integrasi *API* Midtrans untuk mempermudah proses pembayaran *online*. Hasil penelitian ini menunjukkan bahwa pemanfaatan Laravel mempermudah pengelolaan transaksi dan integrasi metode pembayaran, sehingga meningkatkan efisiensi proses penjualan serta memberikan pengalaman berbelanja yang lebih aman dan nyaman bagi pelanggan [4].

Penelitian yang dilakukan oleh Ratino et al. (2023) dalam jurnal Implementasi Framework Laravel Dalam Pengembangan Aplikasi E-Commerce Untuk Toko Jago Software membuktikan bahwa *framework* Laravel mampu mempercepat proses pengembangan sistem *e-commerce* melalui manajemen *database* yang lebih terstruktur dan penggunaan fitur-fitur modular. Penelitian ini menegaskan bahwa Laravel menjadi solusi tepat bagi pengembangan aplikasi penjualan *online* karena mendukung fleksibilitas, keamanan, serta kemudahan pemeliharaan sistem [5].

2.2. Landasan Teori

2.2.1. HTML

HTML merupakan bahasa markup yang digunakan untuk membangun halaman *web*. *HTML* sendiri merupakan singkatan dari *Hypertext Markup Language*, yang berarti *HTML* menggunakan simbol-simbol khusus berupa *tag* yang kemudian akan diterjemahkan oleh *browser* menjadi tampilan halaman *web* [6].

2.2.2. JavaScript

JavaScript merupakan bahasa pemrograman yang digunakan dalam pengembangan *website* untuk menciptakan tampilan yang lebih dinamis. Dengan kata lain, *JavaScript* berperan dalam menghadirkan interaktivitas pada suatu situs *web*, sehingga memungkinkan pengguna berinteraksi langsung dengan elemen-elemen yang ada. *Website* dinamis yang dimaksud adalah situs yang kontennya dapat berubah atau bergerak tanpa perlu melakukan pemuatan ulang secara manual. Contohnya meliputi animasi gambar, peta interaktif (*maps*), tayangan *slide* (*slideshow*), jajak pendapat (*polling*), dan lain sebagainya [7].

2.2.3. MySQL

MySQL merupakan salah satu sistem manajemen basis data (*Database Management System/DBMS*) yang populer dan

berfungsi sebagai sistem manajemen basis data relasional (*Relational Database Management System/RDBMS*). Selain itu, *MySQL* merupakan perangkat lunak bersifat *open source* yang dikenal memiliki kinerja tinggi, andal, serta mudah digunakan. *MySQL* bekerja dengan arsitektur *client-server* maupun *embedded systems*. Sebagai program *database server*, *MySQL* mampu menerima dan mengirim data dengan sangat cepat, mendukung penggunaan oleh banyak pengguna (*multi-user*), serta menggunakan perintah standar *SQL (Structured Query Language)*, sehingga dapat dioperasikan baik sebagai klien maupun sebagai server [8].

2.2.4. *Dbeaver*

Dbeaver adalah aplikasi *database management* berbasis *open source* yang mendukung berbagai sistem manajemen basis data seperti *MySQL*, *PostgreSQL*, *Oracle*, dan *SQL Server*. Aplikasi ini memudahkan pengelolaan *database* melalui antarmuka grafis yang intuitif. *Dbeaver* menyediakan fitur visualisasi struktur *database*, *editor SQL* dengan *auto-complete*, serta kemampuan untuk mengelola data secara langsung. Selain itu, aplikasi ini mendukung proses *impor* dan *ekspor* data dalam berbagai format dan dapat dijalankan di berbagai sistem operasi. [9].


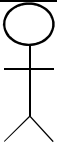
2.2.5. *UML*


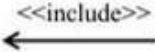

UML (Unified Modeling Language) merupakan sebuah bahasa visual yang digunakan untuk memodelkan, mendokumentasikan, serta merancang sistem perangkat lunak. *UML* berfungsi untuk merepresentasikan struktur, perilaku, interaksi, dan arsitektur dari sistem perangkat lunak yang akan dikembangkan. Dalam konteks perancangan *website e-commerce*, pemanfaatan *UML* dapat membantu dalam memvisualisasikan berbagai komponen dan aspek dari sistem yang dirancang [10]. Berikut macam-macam diagram yang ada di *UML*:

1) *Diagram Use Case*

Digunakan untuk memvisualisasikan fungsionalitas sistem berdasarkan perspektif pengguna. Diagram ini mencakup interaksi antara pengguna sistem, seperti pelanggan, admin, dan sistem pemesanan..

Tabel 2.1 Simbol *UseCase*






No	Gambar	Keterangan
1		<i>UseCase</i> : Interaksi antara sistem dan aktor
2		<i>Actor</i> : Menggambarkan pengguna, sistem yang lain, atau alat ketika berkomunikasi dengan <i>usecase</i> .



No	Gambar	Keterangan
3		Asosiasi antar faktor dan <i>usecase</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
4		<i>Include</i> , Menunjukkan spesialisasi <i>actor</i> untuk dapat berpartisipasi dengan <i>usecase</i>
5		<i>Extend</i> , Merupakan perluasan dari <i>usecase</i> lain jika kondisi atau syarat terpenuhi.

2) *Diagram Class*

Digunakan untuk menggambarkan statis dari sistem, termasuk kelas-kelas yang ada, atribut-atributnya, serta hubungan antara kelas-kelas tersebut (misalnya, hubungan pewarisan dan asosiasi).

Tabel 2.2 Simbol *Class*

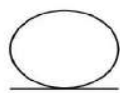
No	Gambar	Nama	Keterangan
1		<i>Generatio n</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Associatio n</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
4		<i>Collaborat ion</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem menghasilkan suatu hasil terukur bagi suatu <i>actor</i> .
5		<i>Realizatio n</i>	Operasi yang benar-benar dilakukan suatu objek

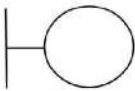

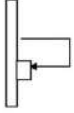


No	Gambar	Nama	Keterangan
6		<i>Dependen cy</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7		<i>Associatio n</i>	Upaya untuk menghubungkan objek satu dengan objek yang lainnya.

3) *Diagram Sequence*

Berfungsi untuk menggambarkan alur pesan atau pemanggilan yang terjadi antar objek dalam sistem, dengan menekankan pola interaksi antar objek tersebut guna mencapai suatu tujuan tertentu.

Tabel 2.3 Simbol *Sequence*

No	Gambar	Nama	Keterangan
1		<i>Entity Class</i>	Gambaran sistem sebagai landasan dalam menyusun basis data



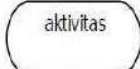
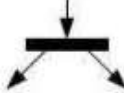

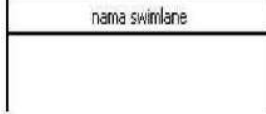
No	Gambar	Nama	Keterangan
2		<i>Boundary Class</i>	Menangani komunikasi antar lingkungan sistem
3		<i>Control Class</i>	Bertanggung jawab terhadap kelas-kelas objek yang berisi logika
4		<i>Recursive Class</i>	Pesan untuk dirinya
5		<i>Activation</i>	Mewakili proses durasi aktivasi sebuah operasi
6		<i>Life Line</i>	Komponen yang digambarkan garis putus terhubung dengan objek

4) *Diagram Activity*

Digunakan untuk merepresentasikan struktur fisik dari suatu sistem, mencakup komponen-komponen perangkat lunak

yang terlibat serta mekanisme interaksi antar komponen tersebut.

Tabel 2.4 Simbol Activity

No	Simbol	Keterangan
1		<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas
2		<i>End Point</i> , digunakan untuk mengakhiri aktivitas
3		<i>Activities</i> , menggambarkan suatu proses/kegiatan.
4		<i>Fork</i> /percabangan digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
5		<i>DecisionPoints</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i> .
6		<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.