

BAB II

TINJAUAN PUSTAKA

2.1. Teori Terkait

Aplikasi Sistem Informasi berbasis WEB telah banyak dikembangkan oleh perancang sebelumnya. Tujuan adanya *website* ini adalah untuk mempermudah penggunaanya.

Penelitian yang dilakukan Tias Nur Aini & Nurgiyatna, dengan judul (“Sistem Penjualan Motor Bekas di Dealer Sinar Maju Motor Purwodadi”) Metode pengembangan dalam pembuatan sistem informasi penjualan sepeda motor bekas di Dealer Sinar Maju Motor Purwodadi menggunakan *Software Development Life Cycle* (SDLC) dengan model *waterfall*. Pembuatan sistem informasi penjualan sepeda motor bekas di dealer Sinar Maju Motor Purwodadi ditujukan hanya untuk admin dalam membantu mengelola penjualan motor. Dibuat menggunakan bahasa pemrograman *Hypertext Preprocessor* (PHP), *Hypertext Markup Language* (HTML), *Cascading Style Sheet* (CSS), Javascript dan database *MySql* Sistem hanya dipasang untuk satu laptop/pc, dengan *localhost* sebagai servernya dan didukung database *MySql*[4].

Penelitian yang dilakukan Muhammad Iqbal. Azhar, Huzaeni, dengan judul (“Perancangan Aplikasi Penjualan Motor Berbasis WEB”) Metode pengembangan dalam pembuatan aplikasi penjualan sepeda motor bekas di UD. Variasi Motor. Aplikasi penjualan sepeda motor dirancang

menggunakan konteks diagram yang dapat diakses oleh pelanggan dan pengelola sistem (admin). Dibuat menggunakan *Windows 8 Enterprise* sebagai *system* operasi, *Sublime Text* sebagai *Text Editor*, *Microsoft Office Visio 2010* sebagai aplikasi untuk merancang DFD (*Data Flow Diagram*) dan ERD (*Entity Relationship Diagram*), *CodeIgniter Framework 3.1.9*, *Xampp* sebagai *server*, *Google Chrome* sebagai *Browser*[5].

Penelitian yang dilakukan Damara Gilang Muraharami, Andy Darmowinoto, Muhammad Yeza Baihaqi (2019), dengan judul (“*WEB BASED Online Motorbike Dealing System*”) Metode pengembangannya menggunakan konteks RAD. *Website* akan dibuat berdasarkan proses pengembangan perangkat lunak tangkas yang disebut pengembangan aplikasi cepat (RAD). Basis Data akan dibangun menggunakan *MySQL*[6].

Penelitian yang dilakukan Arief Herdiansah, dengan judul (“*Sistem Informasi Penjualan Sepeda Motor Pada Dealer UMKUM Menggunakan PIECES*”) Proses perancangan sistem menggunakan UML dan pengembangan sistem menggunakan model *waterfall*. Proses analisa menggunakan *PIECES*, metode ini dapat dengan mudah menganalisa dari berbagai macam area mulai dari *performance*, *information*, *economic*, *control*, *efficiency*, dan *service*[7].

Penelitian yang dilakukan Andi Kurniawan, dengan judul (“*Sistem Informasi Penjualan Motor Di PT. Aries Putra Motor Berbasis WEB*”) Sistem penjualan ini dibuat menggunakan Bahasa pemrograman *HTML*, dan *PHP*, serta menggunakan database *MySQL*[8].

I Gede Bendesa Subawa, I Nengah Eka Pertayasa Dessy Seri Wahyuni (2022), dengan judul (“Pengembangan *Virtual tour* Program Studi Pendidikan Teknik Informatika Berbasis Fotografi Pendidikan 360 Derajat”) Sistem pembuatan *Virtual tour* itu sendiri menggunakan Aplikasi *Webobook* dengan memanfaatkan *drone* dan dilengkapi kamera 360 untuk mengambil gambarnya[9].

2.2. Landasan Teori

2.2.1 Website

Website adalah salah satu aplikasi yang berisikan dokumen - dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang menggunakan protokol HTTP (*hypertext transfer protokol*) dan untuk mengakses menggunakan perangkat lunak yang disebut *browser*[10].

2.2.2 Promosi

Promosi adalah hal-hal yang berkaitan dengan penjualan dan penyampaian informasi mengenai produk. Promosi merupakan unsur dalam pemasaran yang digunakan untuk memberitahukan, membujuk, dan mengingatkan tentang produk perusahaan Promosi adalah seluruh upaya yang dimulai pihak penjual untuk membangun berbagai saluran informasi dan persuasi untuk menjual barang dan jasa[11].

2.2.3 Sistem Informasi

Informasi adalah data yang telah diolah sedemikian rupa ke dalam suatu bentuk yang lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata, sehingga bermanfaat dan dapat digunakan dalam proses pengambilan keputusan saat ini dan saat mendatang[12]

2.2.4 Virtual Tour

Virtual tour merupakan sebuah simulasi dari sebuah lokasi yang terdiri dari rentetan. Rentetan gambar tersebut akan digabungkan (*stitch*) untuk menghasilkan foto panorama 360 derajat. *Virtual tour* sendiri biasanya digunakan untuk memberi pengalaman pernah berada di suatu tempat hanya dengan melihat layar monitor. Penyajian *virtual tour* dapat dilakukan dengan cara memanfaatkan gambar ataupun video, selain itu dapat menggunakan model 3 dimensi. Untuk penyajian dengan menggunakan gambar, dapat digunakan foto panorama. Jenis foto panorama juga mempengaruhi hasil *virtual tour* yang dihasilkan[13]

2.2.5 MySQL

MySQL adalah sistem manajemen basis data relasional yang menggunakan *SQL* untuk menjalankan fungsinya. Fungsi utama *SQL* adalah membuat *query* dan mengoperasikan sistem *database*. *MySQL*

memungkinkan analisis untuk mengelola, menyimpan, memodifikasi, menghapus, dan mengarsipkan data secara transparan[14].

2.2.6 *Laravel 10*

Aplikasi ini merupakan *framework* PHP *open-source* yang sangat populer untuk pengembangan aplikasi web. Dibuat oleh Taylor Otwell, *Laravel* menawarkan cara menarik untuk membangun aplikasi dengan berbagai fitur yang membantu dalam mengelola sesi, autentikasi pengguna, routing, dan banyak lagi. Selain itu, berbagai paket dan alat bantu pengembangan yang disediakan oleh *framework* ini merupakan pola pengembangan yang luas dan konsisten[15].

2.2.7 *JavaScript*

Yaitu bahasa pemrograman yang dinamis. *JavaScript* populer di Internet dan dapat bekerja di sebagian besar *browser* web seperti *Google Chrome*, *Internet Explorer*, *Mozilla Firefox*, *Netscape*, dan *Opera*. Kode *JavaScript* dapat disisipkan ke halaman web menggunakan tag skrip[16].

2.2.8 UML


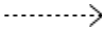
Bahasa pemodelan visual untuk merancang dan mendeskripsikan sistem perangkat lunak. UML adalah standar industri yang digunakan untuk mendeskripsikan, merancang, dan memodelkan sistem perangkat lunak secara visual untuk kemudahan pemahaman dan implementasi. UML menggunakan notasi grafis dan model yang terdiri dari beberapa elemen yang mewakili berbagai aspek sistem perangkat lunak. Contoh: kelas,

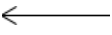
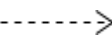
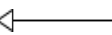




objek, relasi, aktivitas, *use case*, dll. UML memungkinkan pengembang perangkat lunak untuk secara efektif memahami, merencanakan, dan mengkomunikasikan desain sistem perangkat lunak. UML juga dapat digunakan untuk membuat dokumentasi yang lengkap dan terstruktur untuk sistem perangkat lunak yang Anda buat. Dokumen tersebut dapat mencakup ikhtisar sistem perangkat lunak, spesifikasi persyaratan, rencana arsitektur, dan dokumen lain yang diperlukan untuk membangun dan memelihara sistem perangkat lunak[17]. Terdapat ada beberapa diagram UML yang sering digunakan dalam pengembangan sebuah sistem, yaitu.


1. *Use Case Diagram*

Use Case Diagram digunakan untuk mengkomunikasikan interaksi manusia (*actor*) dengan apa yang bisa dilakukan oleh *system*. Sebuah *Use Case* dapat mewakili beberapa jalur interaksi manusia dengan sistem dan setiap jalur disebut sebagai *scenario*. Tabel 2. 1 dapat dilihat dibawah ini.

Tabel 2. 1 *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang dimainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya






NO	GAMBAR	NAMA	KETERANGAN
			elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemennya (sinergi).

NO	GAMBAR	NAMA	KETERANGAN
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

2. Activity Diagram

Activity Diagram merupakan *diagram* yang menjelaskan alur kerja aktivitas pengguna atau sistem yang berbeda, individu yang melakukan tindakan dan urutan terjadinya aktivitas. Tabel 2.2 dapat dilihat dibawah ini.

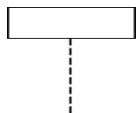


Tabel 2. 2 *Activity Diagram* 1

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

3. *Sequence Diagram*

Sequence Diagram mengilustrasikan objek-objek yang terdapat pada *Use Case* dan menggambarkan arus pesan antara satu sama lain pada Objek *Use Case*. *Sequence Diagram* bersifat dinamis dan lebih banyak menampilkan aktifitas objek berdasarkan urutan waktu. Tabel 2. 3 dapat dilihat dibawah ini.



Tabel 2. 3 *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

4. *Class Diagram*

Class diagram merupakan gambaran diagram yang menjelaskan hubungan antar *class* dalam sebuah sistem yang sedang dibuat dan menjelaskan bagaimana caranya agar mereka saling berkolaborasi. Tabel 2. 4 dapat dilihat dibawah ini.

Tabel 2. 4 *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 induk (<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 aktor
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 (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

2.2.9 Database

“*Database* atau basis data adalah sekumpulan informasi atau data secara sistematis sehingga dapat diperiksa oleh program komputer untuk memperoleh informasi dari basis data tersebut”[18].

2.2.10 Framework

Framework adalah kumpulan intruksi-intruksi yang dikumpulkan dalam *class* dan *function-function* dengan fungsi masing-masing untuk memudahkan *developer* dalam memanggilnya tanpa harus menuliskan *syntax* program yang sama berulang-ulang serta dapat menghemat waktu[19].

2.2.11 Laragon

Laragon adalah *stack* pengembangan yang kuat dan ringan yang membantu pengembangan mengatur *server* lokal, seperti *Apache*, *Nginx*, *MySQL*, *MariaDB*, *PHP*, dan *Node.js*. Ini menawarkan antarmuka yang ramah pengguna dan membuat proses instalasi dan pengelola lingkungan pengembangan lebih efisien dan praktis.[20].