

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian sebelumnya Fatimatuzzahra, F. (2020). “Sistem Informasi Pengurusan Administrasi Kendaraan Bermotor Pada Dealer Honda Surya Motor Banjarmasin Berbasis *Web*” Metode pengembangan dan pemeliharaan sistem pada sistem ini adalah metode waterfall, dan untuk perancangan sistem menggunakan diagram UML (Unified Modeling Language). Sementara untuk pembuatan sistem menggunakan bahas pemograman PHP dan database MySQL [8].

Penelitian seterusnya yang dilakukan Alfarisi, S., dkk (2020.) berjudul “Perancangan Sistem Aplikasi Penjualan pada Showroom Salman Auto Mobilindo Tugu Depok” Perancangan sistem aplikasi penjualan mobil dibuat dengan menggunakan bahasa pemrograman Java dengan Netbeans serta database Mysql dengan software XAMPP, dimana sistem yang dibuat dapat mempermudah transaksi penjualan mobil, mempercepat pencarian data, dan mempermudah dalam pembuatan laporan. Penggunaan sistem informasi dimulai dengan login kedalam sistem kemudian memillih data pembeli untuk menambahkan data pembeli, lalu memilih data mobil untuk menambahkan data mobil yang tersedia, yang kemudian masuk ke data penjualan untuk melakukan proses transaksi, lalu yang terakhir membuat bukti pembayaran kepada konsumen dan laporan kepada perusahaan [5].

Penelitian selanjutnya juga dilakukan Iskandar, dkk (2020) yang berjudul “Sistem Informasi Katalog Diler Motor di Kota Garut” Perancangan yang dilakukan dengan metode Rational Unified Proses dan bahasa pemrograman PHP serta MySQL sebagai Database. Permasalahan yang ada di dealer dapat diselesaikan dengan bantuan fitur yang ada di media Aplikasi berupa booking service dan ketersediaan unit kendaraan dengan media katalog elektronik yang dapat diakses dimana dan kapan saja dengan berbasis *web* [7].

Penelitian terakhir yang dilakukan Papuangan, dkk (2021) berjudul “Sistem Informasi Penjualan Motor Bekas Berbasis *Web* Pada Bengkel Anca Jaya Monalisa Baru” Berdasarkan hasil perancangan dan pengujian sistem yang dilakukan pada sistem penjualan motor bekas berbasis *web* pada bengkel anca jaya monalisa baru dengan menggunakan pemrograman HTML, PHP, dengan database menggunakan MySQL, sistem menunjukkan bahwa semua fungsi yang terdapat di dalam sistem dapat berjalan dengan baik dan sistem dapat memberikan kemudahan kepada pelaku usaha bengkel anca jaya dalam mempromosikan penjualan motor bekas serta dapat memberikan kemudahan pelanggan dalam melakukan transaksi pembelian motor bekas secara online [10].

2.2 Landasan Teori

Perancangan dan pembuatan Laporan Tugas Akhir ini, memerlukan beberapa teori-teori yang dapat menunjang pada penelitian ini.

2.2.1 Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat material, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan[13].

2.2.2 Website

Website dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara dan video atau gabungannya yang disediakan melalui jalur koneksi *internet* sehingga dapat diakses dan dilihat oleh semua orang. Halaman *website* dibuat menggunakan bahasa standal yaitu HTML. Skrip HTML ini akan diterjemahkan oleh *web* browser sehingga dapat ditampilkan dalam bentuk informasi yang dapat dibaca oleh semua orang. Secara umum, *website* dibagi menjadi 3 jenis, yaitu *website* statis, dinamis, dan interaktif[14].

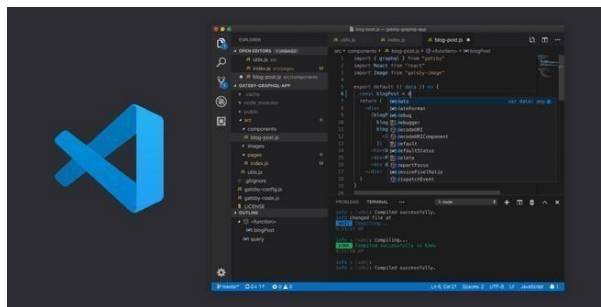
2.2.3 Bootstrap



Gambar 2. 1 Bootstrap

Bootstrap adalah kerangka kerja CSS yang sumber terbuka dan bebas untuk merancang situs *web* dan aplikasi *web*. Kerangka kerja ini berisi templat desain berbasis HTML dan CSS untuk tipografi, formulir, tombol, navigasi, dan komponen antarmuka lainnya, serta juga ekstensi opsional JavaScript. Tidak seperti kebanyakan kerangka kerja *web* lainnya, kerangka kerja ini hanya fokus pada pengembangan front-end saja[15].

2.2.4 Visual Studio Code



Gambar 2. 2 Visual Studio Code

Visual Studio Code (Vc Code) ini adalah sebuah teks editor ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *Linux*, *Mac* dan *Windows*. Teks editor ini secara langsung mendukung bahasa

pemrograman *JavaScript*, *Typescript* dan *Node.js*, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang melalui *marketplace VS Code* (seperti *C++*, *C#*, *Python*, *Go*, *Java* dan beberapa bahasa pemrograman lainnya).

Banyak sekali fitur-fitur yang disediakan oleh *VS Code*, diantaranya *Intellisense*, *Git Intregation*, *Debugging* dan fitur ekstensi yang menambah kemampuan teks editor. Fitur-fitur tersebut akan terus bertambah seiring dengan bertambahnya versi *VS Code*. Pembaruan versi *VS Code* ini juga dilakukan berkala setiap bulan, dan inilah yang membedakan *VS Code* dengan teks editor yang lain.

Teks editor *VS Code* juga bersifat *open source*, yang mana kode sumbernya dapat dilihat dan dapat ikut berkontribusi untuk pengembangannya. Kode sumber dari *VS Code* ini pun dapat dilihat pada link *Github*. Hal ini juga yang membuat *VS Code* menjadi favorit para pengembang aplikasi, karena para pengembang aplikasi bisa ikut serta dalam proses pengembangan *VS Code* ke depannya[16].

2.2.5 *MySQL*

MySQL adalah *DBMS* yang *open source* dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi *MySQL* adalah *database server* yang gratis dengan lisensi *GNU General Public License (GPL)* sehingga dapat dipakai untuk

keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada[17].

2.2.6 PHP



Gambar 2. 3 PHP

PHP (Hypertext Preprocessor) Pertama kali ditemukan pada 1995 oleh seorang Software Developer bernama Rasmus Lerdorf. Ide awal PHP adalah ketika itu Rasmus ingin mengetahui jumlah pengunjung yang membaca resume online. Script yang dikembangkan baru dapat melakukan dua pekerjaan, yakni merekam informasi pengunjung dan menampilkan jumlah pengunjung dari suatu website. Dan sampai sekarang kedua tugas tersebut masih tetap populer digunakan di dunia web saat ini. Kemudian, dari situ banyak orang di milis mendiskusikan script buatan Rasmus Lerdorf, hingga akhirnya rasmus mulai membuat sebuah tool/script bernama PHP, PHP dibangun dari scripts yang ditulis secara plaintext. PHP Interpreter adalah bagian dari perangkat lunak yang ada pada Web Server, yang membaca file tersebut dan mengartikannya, memberikan keluaran HTML dan petunjuk mengenai bagaimana perilaku yang ada maupun menginterpretasikan masukan dari pengguna[18].

2.2.7 HTML

HTML (Hypertext Markup Language) adalah suatu cara memberikan tanda yang memberikan perintah kepada *browser* bagaimana suatu teks terstruktur. *HTML* memberikan perintah kepada *browser* bagaimana struktur dari dokumen, bagaimana *heading*-nya, bagaimana paragrafnya, bagaimana suatu teks akan ditampilkan dan lainnya. Dengan informasi yang diberikan, *browser* dibangun dengan perintah dasar bagaimana menampilkan setiap elemen yang ada[19].

2.2.8 CSS

CSS (Cascading Style Sheets) adalah aturan untuk mengatur beberapa *komponen* dalam sebuah *web* sehingga akan lebih terstruktur dan seragam. *CSS* bukan merupakan bahasa pemrograman. Sama halnya *styles*, misalnya *heading*, *subbab*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (file). Pada umumnya *CSS* dipakai untuk memformat tampilan halaman *web* yang dibuat dengan bahasa *HTML* dan *XHTML*[19].

2.2.9 Database

Basisdata (database) adalah suatu alat yang digunakan untuk menyimpan *informasi*, mengambil informasi kapanpun dibutuhkan,

dan mengatur informasi yang tersimpan. Jika menggambarkan lemari file merupakan suatu *basisdata*.

DBMS (Database Management System) merupakan suatu perangkat lunak yang digunakan untuk membuat, memelihara, mengontrol dan mengakses *basisdata* secara praktis dan efisien. Sedangkan *RDBMS* merupakan salah satu *DBMS* yang mendukung adanya relasi atau hubungan[20].

2.2.10 XAMPP



Gambar 2. 4 *Xampp*

XAMPP dikembangkan dari sebuah tim proyek bernama Apache Friends, yang terdiri dari Tim Inti (*Core Team*), Tim Pengembang (*Development Team*) & Tim Dukungan (*Support Team*). *XAMPP* adalah singkatan yang masing- masing hurufnya adalah: X=Program ini dapat dijalankan dibanyak sistem operasi seperti: Windows, Linux, Mac Os, dan Solaris. A=Apache, merupakan aplikasi *WebServer*. Tugas utama Apache adalah menghasilkan halaman *web* yang benar kepada *user* berdasarkan kode PHP yang dituliskan oleh pembuat halaman *web*. Jika diperlukan juga berdasarkan kode PHP yang dituliskan, makadapat saja suatu Database diakses terlebih dahulu (misalnya dalam MySql)

untuk mendukung halaman *web* yang dihasilkan. *M=MySQL*, merupakan aplikasi *database server*. Perkembangannya disebut *SQL* yang merupakan kepanjangan dari *Structured Query Language*. *SQL* merupakan bahasa terstruktur yang digunakan untuk mengolah database. *MySQL* dapat digunakan untuk membuat dan mengelola database beserta isinya. Kita dapat memanfaatkan *MySQL* untuk menambahkan, mengubah, dan menghapus data yang berada dalam database. *P= PHP*, bahasa pemrograman *website*[21].

2.2.11 UML (Unified modeling language)

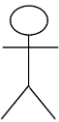


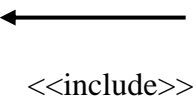

Menurut (Pressman, 2010:841) Unified modeling language (UML) adalah bahasa standar untuk menulis denah perangkat lunak. UML dapat digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak. Dengan kata lain, seperti arsitek bangunan membuat denah yang akan digunakan oleh sebuah perusahaan konstruksi, arsitek software membuat diagram UML untuk membantu pengembang perangkat lunak membangun perangkat lunak. Jika memahami kosakata UML, dapat lebih mudah memahami dan menentukan sistem dan menjelaskan desain sistem kepada orang lain.



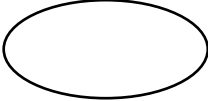


Unified Modeling Language merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah software yang berorientasikan pada objek. UML merupakan sebuah standar penulisan atau semacam blue print

dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik[8]. Terdapat beberapa diagram UML yang sering digunakan dalam pengembangan sebuah sistem yaitu:

1. *Use case*: Merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem dan mempresentasikan sebuah interaksi antara aktor dan sistem. Didalam *use case* terdapat aktor 6 yang merupakan sebuah gambaran entitas dari manusia atau sebuah sistem yang melakukan pekerjaan di sistem[22].


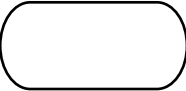


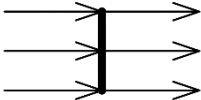
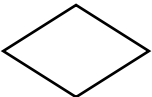
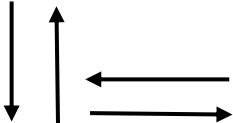
Tabel 2. 1 Simbol Use Case Diagram

Gambar	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancertor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisitis.
		Menspesifikasikan bahwa <i>use case</i> target

Gambar	Nama	Keterangan
<<extend>>	<i>Extend</i>	memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<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 aktor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang <i>eksis</i> saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.



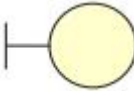




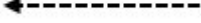
2. *Activity Diagram*: Merupakan gambaran alur dari aktivitas-aktivitas didalam sistem yang berjalan.

Tabel 2. 2 Simbol Activity Diagram

Gambar	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.
	<i>Decision</i>	Pilihan untuk mengambil keputusan.
	<i>Fork/Join</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

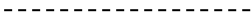
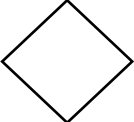


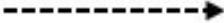
3. *Sequence Diagram*: Menggambarkan interaksi antar objek didalam dan disekitar yang berupa message yang digambarkan terhadap waktu.



Tabel 2. 3 Simbol Sequence Diagram

Gambar	Nama	Keterangan
	<i>Lifeline</i>	Objek <i>entity</i> , antar muka yang saling berinteraksi.
	<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
	<i>Boundary Class</i>	Menggambarkan penggambaran dari form.
	<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan.
	<i>Control Class</i>	Menggambarkan penghubung antara <i>Boundary</i> dengan tabel.
	<i>Activation</i>	Sebagai sebuah objek yang akan melakukan sebuah aksi.
	<i>Message</i>	Mengindikasikan komunikasi antara objek dengan objek.
	<i>Self Mesagge</i>	Mengindikasikan komunikasi kembali kedalam sebuah objek itu <i>sendiri</i> .

4. *Class Diagram*: Merupakan gambaran struktur dan deskripsi dari *class*, *package* dan objek yang saling berhubungan seperti diantaranya pewarisan, asosiasi dan lainnya.

Tabel 2. 4 Simbol Class Diagram

Gambar	Nama	Keterangan
	<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>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Dependency</i>	Operasi yang benar-benar dilakukan oleh suatu objek.

	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.