

BAB II

TINJAUAN PUSTAKA

2.1 Peneliti Terkait

Penelitian yang dilakukan oleh Reni Sumarni, dkk. (2023) dengan judul penelitian Perancangan Aplikasi Persediaan Barang Dan *Sparepart* Di Bengkel Nazwa Motor Kota Tasikmalaya. Penelitian ini menggunakan Metode pengembangan sistem yang menggunakan metode *System Development Life Cycle* (SDLC), sedangkan alat bantu analisis yang digunakan yaitu Flowmap, Diagram Context, Data Flow Diagram (DFD), dan *Entity Relationship* Diagram (ERD). Perancangan sistemnya menggunakan Microsoft Visual Studio 2008, dan rancangan *database* nya menggunakan Microsoft Office Access 2013 [3].

Penelitian lain yang dilakukan Rafki Deshramadhan, dkk.(2023) berjudul Aplikasi Penjualan *Sparepart* Pada Bengkel Agung Motor Sukabumi Berbasis *Java*. Metode yang digunakan dalam penelitian ini adalah pendekatan kualitatif karena data yang diperoleh hanyalah data non-naratif. Sedangkan metode pengumpulan data meliputi observasi, angket, dan dokumen. Aplikasi ini dibuat menggunakan java netbeans dengan database MYSQL yang sangat memudahkan pengguna dalam mengelola data [4].

Penelitian yang hampir sama juga dilakukan oleh Ade Ferry Qadafi, dkk.(2020) penelitian yang berjudul Sistem Informasi Inventory Gudang Dalam Ketersediaan Stok barang Menggunakan Metode Buffer Stok.

Penelitian ini menggunakan metode UML (*Unified Modeling Language*). bahasa pemrograman yang digunakan yaitu PHP dengan *database* MYSQL, dan kualitas perangkat lunak yang dihasilkan dari hasil pengujian juga memiliki tingkat yang Sangat Layak [5].

Selanjutnya penelitian lain yang di lakukan oleh Afridus Luga Tukan, dkk.(2023) penelitian yang berjudul Aplikasi Penjualan Suku Cadang Dan Jasa Perbaikan Sepeda Motor Pada Bengkel Cmm Motor menggunakan Metode Sdlc berbasis Desktop. Tujuan dari penelitian ini adalah mengembangkan sistem informasi penjualan dan servis suku cadang kendaraan bermotor dengan menggunakan Visual Basic.Net sebagai bahasa pemrograman dan MYSQL sebagai data basenya. Metode penelitian ini menggunakan metode deskriptif kualitatif. Pengujian yang di gunakan penulis yaitu dengan pengujian Blackbox dan metode pengujian penulis menggunakan blackbox setting. Penerapan sistem ini diharapkan dapat memberikan efisiensi dan efektifitas kerja pada bengkel CMM motor [6].

Selain itu penelitian lain yang dilakukan Oleh Bahrain Erekat Budiono, dkk.(2024) yang berjudul Aplikasi Penjualan *Sparepart* Motor Pada Bengkel Motor Stage Tuner Garage Berbasis *Java*. Metode penelitian ini menggunakan metode Waterfall kualitatif dengan mengamati secara langsung, observasi, wawancara pengelola bengkel dan dokumentasi informasi yang relevan seperti, analisis kebutuhan, kajian kepustakaan, eksperimen, implementasi dan evaluasi. Hasil dari penelitian ini adalah memperbarui sistem secara konvensional menjadi sistem yang

terkomputerisasi agar dapat mempermudah proses penjualan bagi pemilik bengkel stage tuner garage [7].

2.2 Landasan Teori

2.2.1 Aplikasi Dekstop

Aplikasi berbasis desktop merupakan aplikasi yang dijalankan pada masing-masing komputer. Aplikasi berbasis desktop harus diinstall terlebih dahulu ke dalam komputer agar dapat digunakan. Menurut Stefano (2014) berpendapat bahwa, “aplikasi desktop adalah suatu aplikasi yang mampu beroperasi secara offline, tetapi kita harus menginstalnya sendiri pada laptop atau komputer”. Salah satu perbedaan yang mendasar dari web based dan desktop based adalah bahasa pemrogramannya kalau pada desktop based didukung oleh beberapa macam program seperti Microsoft Office, Borland Delphi, dan lain lain.

Sedangkan untuk web based menggunakan bahasa pemrograman HTML, PHP, CSS, *Java Script*, dan lain lain. Fungsionalitas dari aplikasi berbasis desktop tersebut yaitu aplikasi berbasis desktop dapat berjalan secara mandiri dan umumnya tidak menggunakan jaringan internet untuk menggunakannya. Aplikasi jenis ini juga umumnya telah ditentukan penggunaannya pada sistem operasi tertentu. Dari pengertain diatas dapat disimpulkan bahwa model pembelajaran berbasis desktop adalah model pembelajaran

aplikasi program dikomputer yang dapat beroperasi secara offline dengan menginstalnya terlebih dahulu.

2.2.2 Netbeans



Gambar 2.1 Logo NetBeans

Netbeans merupakan salah satu *software* yang sering digunakan dalam dunia programmer atau developer. Bukanlah sebagai teks editor biasa, *Netbeans* adalah suatu aplikasi IDE atau *Integrated Development Environment* yang berbasis bahasa *Java* dan berjalan diatas *Swing*. Maksudnya *Swing* disini adalah suatu teknologi yang memungkinkan pengembangan aplikasi desktop dan dapat berjalan di berbagai macam platform seperti *Windows*, *Mac OS*, *Linux* dan *Solaris* [8].

2.2.3 XAMPP



Gambar 2.2 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, maka dapat 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.

2.2.4 Data Base

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.

2.2.5 *Java*



Gambar 2.3 Java

Java adalah sekumpulan perangkat lunak komputer dan spesifikasi yang menyediakan sistem untuk mengembangkan perangkat lunak aplikasi dan menerapkannya dalam lingkungan komputasi lintas platform. *Java* bisa digunakan untuk membuat aplikasi berbasis *Dekstop, Website, Mobile*, hingga aplikasi *embedded device* seperti perangkat pintar atau *microprosesor*. Banyak aplikasi android dibuat menggunakan bahasa pemrograman *Java* [11].

2.2.6 UML (*Unified Modeling Language*)

Usecase Unified Modeling Language (UML) adalah sebuah bahasa untuk menentukan, visualisasi, kontruksi, dan

mendokumentasikan artifact (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak). UML merupakan notasi yang lengkap untuk membuat visualisasi model suatu sistem. Sistem berisi informasi dan fungsi, namun secara normal digunakan untuk pemodelan sistem komputer. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. (Rosa, 2016).

1. *Usecase* diagram yaitu salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, *Usecase* diagram juga dapat mendeskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya.

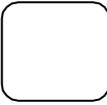
Tabel 2.1 Simbol *Usecase* Diagram

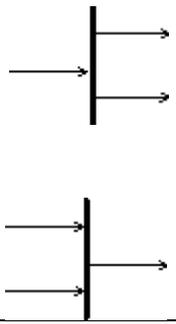
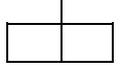
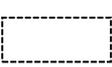
No	Gambar	Nama	Keterangan
1.		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2.		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).

No	Gambar	Nama	Keterangan
3.		Generalization	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancertor</i>).
4.		Include	Menspesifikasikan bahwa <i>software</i> sumber secara eksplisit.
5.		Extend	Menspesifikasikan bahwa use case target memperluas perilaku dari <i>software</i>

2. *Activity Diagram* merupakan rancangan aliran aktivitas atau aliran kerja dalam sebuah sistem yang akan dijalankan.

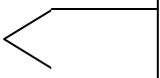
Tabel 2.2 Simbol *Activity Diagram*

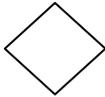
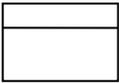
No	Gambar	Nama	Keterangan
1.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antar-muka saling berinteraksi satu sama lain.
2.		Action	State dari sistem yang Mencerminkan eksekusi suatu aksi.
3.		Initial Node	Bagaimana objek dibentuk atau diawali.
4.		Final Node	Bagaimana objek dibentuk dan dihancurkan.

No	Gambar	Nama	Keterangan
5.		Fork Node	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.
6.		Decision	Pilihan untuk mengambil keputusan.
7.		Fork/Join	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
8.		Rake	Menunjukkan adanya dekomposisi.
9.		Time	Tanda waktu
10.		Send	Tanda pengiriman

3. *Class Diagram* dapat dikatakan bersifat statis, alasannya karena diagram kelas tidak menggambarkan apa yang terjadi jika mereka berhubungan melainkan menggambar hubungan apa yang terjadi.

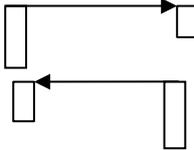
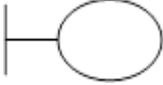
Tabel 2.3 Simbol *Class Diagram*

No	Gambar	Nama	Keterangan
1.		Generalization	Hubungan dimana objek anak (descendent) berbagi perilaku struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)

No	Gambar	Nama	Keterangan
2.		Nary Association	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3.		Class	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4.		Collaboration	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5.		Dependency	Operasi yang benar - benar dilakukan oleh suatu objek.
6.		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.

4. *Sequence Diagram* menggambarkan interaksi antar di sekitar (pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi *horizontal* (objek-objek yang terkait).

Tabel 2. 4 Simbol *Sequence Diagram*

No	Gambar	Nama	Keterangan
1.		Life Line	Objek <i>entity</i> , antar muka yang saling berinteraksi.
2.		Message	Spesifikasi dari komunikasi antar objek yang memuat informasi -informasi tentang aktifitas yang terjadi.
3		Actor	Menggambarkan orang yang sedang berinteraksi dengan <i>system</i> .
4		Boundary <i>Class</i>	Menggambarkan penggambaran dari form.
5		Entity <i>Class</i>	Menggambarkan hubungan Kegiatan yang akan dilakukan.
6.		Control <i>Class</i>	Menggambarkan penghubung antara Boundary dengan tabel.