

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Teori ini membantu dalam memahami bagaimana *website* dirancang dan diimplementasikan untuk mendukung *website* penjualan *voucher game* dan sistem *top up game online*.

Sebagian besar situs *web* dibuat menggunakan kombinasi berbagai teknologi. Biasanya, kumpulan beberapa teknologi disebut sebagai "tumpukan". diperluas dengan tumpukan LAMP, kombinasi sumber terbuka yang terdiri dari Linux, Apache, MySQL, dan PHP[3].

Framework CodeIgniter digunakan dalam prosedur pengembangan penelitian ini. Untuk memudahkan pembuatan aplikasi berbasis *web*, digunakan bahasa pemrograman PHP untuk membuat *framework CodeIgniter* (Ridwan et al., 2022). *Website* dibuat dengan menggunakan bahasa pemrograman PHP dinamis (Novendri, 2019). *Website* terdiri dari halaman-halaman yang dihubungkan satu sama lain untuk menciptakan jaringan bangunan yang saling berhubungan, dan dapat menampilkan teks, animasi, suara, gambar diam atau bergerak, dan kombinasi dari semua elemen tersebut, baik statis maupun dinamis, menurut kepada Becti (2015)[4].

Penelitian Puspita Dwi Astuti yang mengkaji sistem informasi penjualan obat berbasis desktop merupakan salah satu dari beberapa penelitian terdahulu yang menjadi bahan referensi peneliti untuk mendapatkan informasi. Sebuah aplikasi yang dibuat dengan *Visual Basic* ditampilkan pada penelitian ini. Pembangunan sistem ini bertujuan untuk mengurangi beban kerja asisten apoteker dan meningkatkan efektivitas dan efisiensi operasional apotek[5].

Menurut penelitian Agus Irawan dkk, proses perancangan sistem informasi penjualan pakaian *online* melibatkan pembuatan tujuh *database* yang saling berhubungan melalui penggunaan *form* atau tabel, *flowmap*, diagram aliran data, normalisasi tabel, ERD, struktur *database*, perancangan *input* dan *output*, dan *menu program*. struktural. Hasilnya, *operator*, pemilik, dan pelanggan dapat dengan mudah mempelajari cara memanfaatkan aplikasi sistem penjualan *online* yang mudah digunakan ini[3].

Selain itu Angling Ananditya melakukan penelitian tentang pembuatan sistem informasi penjualan *voucher game online* berbasis *desktop*. Tujuan dari penelitian ini adalah untuk mengembangkan solusi berbasis desktop yang dapat meningkatkan kinerja warung *internet* saat bertransaksi dan efisien. Hasilnya, data *voucher* dan stok *voucher* otomatis tersedia di sistem, memiliki antarmuka yang mudah digunakan dan dapat memperlancar proses transaksi penjualan *voucher* menjadi lebih efisien. Selain itu, proses transaksi telah didigitalkan untuk mempercepat dan menyederhanakan pekerjaan dalam memproses data penjualan *voucher*[5].




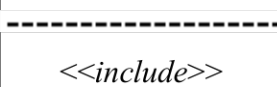

2.2 Landasan Teori

UML merupakan singkatan dari *Unified Modeling Language*, yang merupakan bahasa visual untuk memodelkan, mendokumentasikan, dan merancang sistem perangkat lunak. *UML* digunakan untuk menggambarkan struktur, perilaku, interaksi, dan arsitektur dari sistem perangkat lunak yang akan dibangun. Dalam konteks perancangan *website* penjualan *voucher game*, penggunaan *UML* akan membantu dalam menggambarkan berbagai aspek dari sistem tersebut. Ini termasuk:

1. *Diagram Use Case*

Digunakan untuk menggambarkan fungsionalitas sistem dari sudut pandang pengguna. Ini akan mencakup interaksi antara pengguna sistem misalnya (pelanggan, admin dan sistem pemesanan).








Tabel 2. 1 Simbol UseCase

No	Simbol	Keterangan
1		<i>Use Case</i> : Abstraksi dan interaksi antara sistem dan aktor.
2		<i>Actor</i> : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .
3		Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
4		<i>Include</i> , Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .
5		<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

2. Diagram Class

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

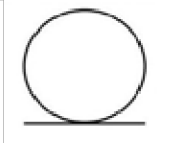
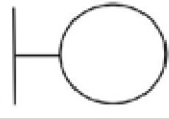
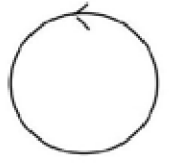
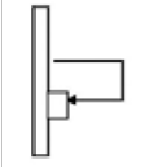
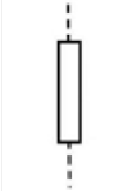

Tabel 2. 2 Simbol Class

No	Gambar	Nama	Keterangan
1		<i>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 terukur bagi suatu <i>actor</i> .
5		<i>Realization</i>	Operasi yang benar-benar dilakukan suatu objek.
6		<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.
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

3. *Diagram Sequence*

Digunakan untuk menggambarkan urutan pesan atau panggilan yang terjadi di antara objek-objek dalam sistem, menyoroti interaksi antara objek-objek tersebut dalam rangka mencapai suatu tujuan.




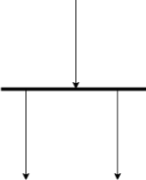
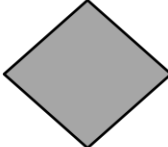
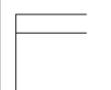
Tabel 2. 3 Simbol Sequence

No	Gambar	Nama	Keterangan
1		<i>Entity Class</i>	Gambaran sistem sebagai landasan dalam menyusun basis data
2		<i>Boundary Class</i>	Menangani komunikasi antar lingkungan sistem
3		<i>Control Class</i>	Bertanggung jawab terhadap kelas – kelas terhadap objek yang berisi logika
4		<i>Recursive</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 menggambarkan struktur fisik dari sistem, termasuk komponen-komponen perangkat lunak yang digunakan, serta cara komponen-komponen tersebut berinteraksi satu sama lain.

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> , akhir aktivitas.
3		<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis.
4		<i>Fork</i> /percabangan digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
5		<i>Decision Points</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.

Berikut *tools* yang akan dipakai :

1. HTML

HTML adalah bahasa markup untuk membuat halaman *web*. Kepanjangan dari *HTML* adalah *Hypertext Markup Language* artinya, *HTML* menggunakan *symbol* tertentu (*tag*) yang nantinya akan diterjemahkan oleh *browser* ke halaman *web*.

2. Java Script

JavaScript adalah bahasa pemrograman yang digunakan untuk pengembangan *website* agar lebih dinamis. Ibarat kata, *JavaScript* memberikan “kehidupan” dalam *website* sehingga terciptanya interaksi antara pengunjung dengan situs tersebut. *Website* dinamis yang dimaksud berarti konten di dalamnya dapat bergerak atau mengubah apapun yang tampak di layar tanpa harus dimuat ulang secara manual. Misalnya seperti konten gambar animasi, *maps*, *slideshow*, *polling*, dan sebagainya.

3. XAMPP

XAMPP adalah singkatan dari *Cross-platform*, *Apache*, *MySQL*, *PHP*, dan *Perl*. Ini adalah sebuah perangkat lunak yang menyediakan lingkungan *server* yang lengkap dan siap pakai untuk pengembangan dan pengujian aplikasi *web* secara lokal pada komputer pribadi. *XAMPP* memungkinkan pengguna untuk menjalankan server web *Apache*, basis data *MySQL*, serta bahasa pemrograman *PHP* dan *Perl* pada komputer lokal tanpa perlu menginstalnya secara terpisah. Ini sangat berguna bagi para pengembang *web* yang ingin mengembangkan dan menguji situs atau aplikasi *web* sebelum mempublikasikannya secara *online*.

4. Mysql

MySQL adalah sebuah *database* manajemen *system (DBMS)* populer yang memiliki fungsi sebagai *relational database* manajemen *system (RDBMS)*. Selain itu *MySQL software* merupakan suatu aplikasi yang sifatnya *open source* serta server basis data *MySQL* memiliki kinerja sangat cepat,

reliable, dan mudah untuk digunakan serta bekerja dengan arsitektur *client server* atau *embedded systems*. *MySQL* adalah program *database server* yang mampu menerima dan mengirimkan data dengan sangat cepat, multi *user* serta menggunakan perintah standar *SQL (Structured Query Language)* dan baik digunakan sebagai *client* maupun *server*. Alat yang dirancang terintegrasi dengan *database* kehadiran mahasiswa pada suatu matakuliah sehingga dapat berperan sebagai pengganti sistem absensi manual. *Database* kehadiran dibuat dengan *MySQL XAMPP*.