

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Studi oleh Maulia Usnain¹, dkk (2022) dengan judul “Perancangan Sistem Informasi Inventaris Aset Berbasis Web Menggunakan Metode *Waterfall*” merupakan rancangan aplikasi yang digunakan untuk mencatat, menghitung aset yang ada pada instansi, manajemen aset, dan pelaporan aset. Pembuatan aplikasi ini menggunakan metode *Waterfall*. Aplikasi ini diimplementasikan menggunakan bahasa pemrograman *PHP*, penyimpanan *database MySQL*[3].

Yusril Habibullah, dkk (2023), melakukan penelitian bejulul “Aplikasi Toko *Online* dengan Penerapan Sistem *Payment Gateway* untuk Transaksi Pembayaran Berbasis Web” yang menjelaskan penggunaan *payment gateway* untuk mempermudah transaksi pembayaran. Hasil dari penelitian ini adalah analisis aplikasi toko *online* yang telah terintegrasi dengan layanan pembayaran *online* menggunakan *payment gateway* Midtrans. Sistem ini memfasilitasi transaksi bagi pelanggan dan memudahkan pembayaran *online* saat membeli produk karena dilengkapi dengan *payment gateway*. Ada lebih dari satu metode pembayaran yang tersedia, seperti transfer bank, kartu kredit, dompet digital dan lainnya[4].

2.2 Landasan Teori

2.2.1 Website

Situs web ini dapat diakses oleh pengguna internet di seluruh dunia. Pengguna Internet dapat mengakses semua informasi melalui web. Situs web adalah fasilitas internet yang dapat menghubungkan dokumen lokal atau jarak jauh[5].

2.2.2 *E-commerce*

E-commerce adalah bisnis di mana barang atau jasa dijual, dibeli, dan diiklankan menggunakan sistem elektronik seperti internet, televisi, dan jaringan teknologi lainnya. Dengan teknologi yang lebih progresif dan perkembangan inovasi tinggi oleh pengusaha, bisnis *e-commerce* dikenal lebih cepat dan lebih dikenal secara luas[6].

2.2.3 *MySQL*

MySQL adalah semacam *server* basis data yang merupakan sumber terbuka. Perangkat lunak ini dilengkapi dengan kode sumber, yang dapat digunakan untuk menulis *MySQL*, karena bersifat *open source*. dan formulir atau kode yang dapat dirancang dengan sistem operasi[7].

2.2.4 *PHP*

PHP adalah bahasa tambahan *HTML* yang memungkinkan aplikasi dinamis untuk diatur dan diproses. Meskipun *sintax* yang dipilih akan berjalan sepenuhnya di *server*, hanya hasilnya yang akan dikirim ke *browser* [8].

2.2.5 *Bootstrap*

Bootstrap adalah kerangka kerja *CSS* yang berfokus pada pengembangan situs web *front-end*. Termasuk *HTML*, *CSS*, dan *JavaScript* untuk tampilan *website* yang *modern*, responsif, dan *mobile friendly*[9].

2.2.6 *CodeIgniter*

CodeIgniter adalah *framework PHP* berbasis *model view controller (MVC)*. *CodeIgniter* memiliki pustaka yang luas untuk menangani fungsi standar yang dibutuhkan oleh aplikasi berbasis web, seperti memvalidasi *form* dan mengakses *database*. Ini membuat proses pembuatan sistem lebih mudah[10].

2.2.7 *Payment Gateway*

Payment gateway adalah platform untuk transaksi *online* yang membantu menjelaskan dan memverifikasi data transaksi sesuai dengan persyaratan penyedia layanan[11].

2.2.8 *Unified Modeling Language (UML)*

Merupakan *tool* untuk merancang pengembangan sistem informasi yang berdasarkan gambar dan grafik untuk menggambarkan. *UML* juga merupakan standar untuk membuat analisis dan desain dalam sistem informasi[12].

1. *Use case Diagram*

Use case diagram adalah rangkaian atau uraian kelompok yang saling terkait. *Use case* diagram tidak menjelaskan detail

tentang penggunaannya, tetapi hanya memberi gambaran hubungan tiap aktor dengan *usecase*[13]. Tabel *usecase* diagram dapat dilihat seperti pada Tabel 2.1 berikut.

Tabel 2.1 Tabel *Usecase* Diagram

No	Simbol	Nama Simbol	Keterangan
1.		<i>Use Case</i>	Fungsi sistem sebagai unit yang menukar pesan antara unit atau aktor.
2.		<i>Actor/Aktor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang diproduksi di luar lebih umum daripada yang lain.
3.		<i>Association/ Assosiasi</i>	Komunikasi antara aktor dan <i>usecase</i> atau <i>usecase</i> yang terkait dengan aktor.
4.		<i>Generalization/ Generalisasi</i>	Hubungan antara generalisasi dan spesialisasi antara dua <i>usecase</i> dimana fungsi lebih umum daripada <i>usecase</i> lain.
5.		<i>Include/Uses</i>	<i>Include</i> berarti <i>use case</i> yang selalu dipanggil saat <i>use case</i> tambahan dijalankan
6.		<i>Extend/Ekstensi</i>	Hubungan aplikasi tambahan dengan aplikasi dengan aplikasi yang diakui ditambahkan

2. Acitvity Diagram

Diagram aktivitas menggambarkan aktivitas sistem, alur kerja, atau proses bisnis, atau menu yang ditemukan dalam perangkat lunak. Apa yang perlu dipertimbangkan di sini adalah bahwa diagram aktivitas menjelaskan bahwa aktivitas sistem bukanlah sesuatu yang dilakukan seorang aktor[14]. Tabel *activity* diagram dapat dilihat seperti pada Tabel 2.2 berikut.

Tabel 2.2 Tabel *Activity* Diagram

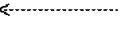
No	Simbol	Nama Simbol	Keterangan
1.		Status Awal	Sebuah simbol yang memiliki sebuah status mulainya aktivitas.
2.		Aktivitas	Simbol sistem yang dimulai dengan kata kerja.
3.		Percabangan /Decision	Percabangan dimana ketika aktivitas memiliki pilhan sebuah keputusan.
4.		Penggabungan/Join	Penggabungan lebih dari satu aktivitas.
5.		Status Akhir	Sebuah simbol aktivitas memiliki sebuah status akhir.
		Swimlane	Menunjukkan kegiatan pemisahan atau pengelompokan berdasarkan peran dan fungsi yang berbeda.

3. Sequence Diagram

Sequence Diagram menunjukkan interaksi antara objek dalam suatu sistem. Urutan Diagram juga menampilkan pesan atau pesanan selama waktu implementasi[15]. Tabel *sequence* diagram dapat dilihat seperti pada Tabel 2.3 berikut.

Tabel 2.3 Tabel *Sequence Diagram*

No	Simbol	Nama Simbol	Keterangan
1.		Actor/Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang diproduksi di luar lebih umum daripada yang lain.
2.		Entity	Bertanggung jawab untuk menyimpan data atau informasi.
3.		Boundary	Elemen sistem yang berinteraksi satu sama lain.
4.		Controll	Mengatur bagaimana informasi diproses untuk sebuah skenario.
5.		Lifeline	Menghubungkan antar elemen <i>sequence</i> .
6.		Activation/ Aktivasi	Tempat di mana barang berpartisipasi dalam urutan yang ditunjukkan saat dikirim atau diterima.

7.		<i>Message Entry</i>	Menggambarkan pesan atau hubungan antara objek yang menunjukkan urutan kejadian.
8.		<i>Message to Self</i>	Simbol ini menunjukkan rangkaian peristiwa yang terjadi dan menjelaskan pesan atau hubungan objek itu sendiri.
9.		<i>Message Return</i>	Menunjukkan dan menjelaskan hasil pengiriman pesan dari kanan ke kiri.