

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Artati (2011), dalam penelitian yang berjudul Perancangan dan Pengaplikasian Sistem Penjualan Pada “Distro Smith” Berbasis E-commerce, Smith adalah salah satu distro yang baru berdiri sejak tahun 2010 di Semarang. Mekanisme sistem penjualan pada *distro smith* masih menggunakan sistem konvensional, dimana konsumen harus datang langsung ke distro. Dengan adanya Aplikasi E-commerce pada Distro Smith Semarang dapat memberikan kemudahan kepada masyarakat untuk melakukan pembelian produk tanpa harus datang ke tempatnya, serta memperluas pemasaran dan meningkatkan *customer loyalty*. Sistem ini dikembangkan dengan SDLC (System Development Life Cycle) dan *software* yang digunakan untuk merancang dan mendesain Aplikasi E-commerce yaitu Bahasa pemrograman PHP, text editor Macromedia Dreamweaver 8, Database MySQL dan Adobe Photoshop CS4.

Surya (2011), dalam penelitian yang berjudul Perancangan Sistem Informasi Penjualan Berbasis Web Pada Toko Vivaldi Cell, Vivaldi Cell sebuah toko yang menjual berbagai jenis *handphone* dan aksesoris *handphone*. Pada Vivaldi Cell, proses penjualan belum menggunakan komputerisasi, sehingga waktu yang dibutuhkan sangat lama dan kemungkinan terjadinya kehilangannya beberapa catatan penjualan dan penjualan barang terjadi tidak maksimal. Untuk mengatasi permasalahan tersebut digunakanlah system *E-commerce*. Dengan sistem ini pelanggan tidak perlu repot datang ke toko dan dapat memesannya kapan saja tanpa ada batasan waktu. Perancangan Sistem Informasi Penjualan Handphone Dan Aksesoris Handphone berbasis Web Pada Vivaldi Cell ini dirancang menggunakan macromedia dreamweaver, php, mysql, xampp dan photoshop.

Banu (2012), dalam penelitiannya yang berjudul Sistem Informasi Penjualan Berbasis Web Pada Usantex. Usantex adalah perusahaan yang bergerak di bidang pembuatan, penjualan, dan pemasaran batik. Sangat perlu adanya kemudahan

pelayanan untuk memudahkan pelanggan yang jauh dari jangkauan. Guna memfasilitasi itu semua dapat direalisasikan dengan toko online (*E-commerce*). Aplikasi toko *online* yang dibangun ini bertujuan untuk membantu Usantex dalam hal pemasaran dan penjualan produk. Proses pembuatan program ini dimulai dengan menganalisa sistem yang telah ada di Usantex, kemudian dibuat DFD (Data Flow Diagram), ERD (Entity Relationship Diagram) dan tabel–tabel yang diperlukan. Dilanjutkan dengan mendesain *interface* dan membuat programnya. Aplikasi toko online ini menggunakan bahasa pemrograman PHP, MySQL sebagai *database*, Macromedia Dreamweaver 8 sebagai editor HTML dan CorelDraw sebagai image editor. Aplikasi ini memiliki tiga fungsi utama. Pertama yaitu user umum, setiap pengunjung website bisa melihat katalog produk, menggunakan layanan *website*, tetapi tidak bisa melakukan pemesanan produk. Kedua adalah anggota yang bisa melakukan pemesanan produk. Terakhir yaitu *administrator* yang bisa melakukan pengelolaan data produk, mengelola pesanan dan memperoleh laporan.

Berdasarkan pengujian dan analisis yang telah dilakukan oleh masing-masing peneliti, ketiga penelitian tersebut membuat *website e-commerce* sebagai media pemasaran suatu produk. Ketiga penelitian tersebut berkaitan dengan penelitian yang dibuat oleh penulis yaitu membuat *website e-commerce* sebagai sarana mempermudah pemasaran suatu produk, adapun kelebihan dari penelitian ini yaitu adanya 3 halaman *website* untuk umum, *customer*, dan admin. Serta pada bagian pelaporan transaksi dan produk dapat di unduh berupa file pdf pada halaman admin.

2.2 Landasan Teori

2.2.1 *E-commerce*

Landon (1998), *Website* adalah kumpulan halaman web yang mengandung informasi. Fungsi *website* saat ini sudah hampir dapat melayani berbagai kebutuhan manusia, seperti tersedianya *website e-commerce*. *E-commerce* adalah suatu proses membeli dan menjual produk – produk secara elektronik oleh konsumen dan dari perusahaan ke perusahaan dengan computer sebagai perantara transaksi bisnis.

E-commerce atau yang biasa disebut juga dengan istilah *Ecom* atau *Emmerce* atau EC merupak pertukaran bisnis yang rutin dengan menggunakan transmisi *Electronic Data Interchange* (EDI), email, electronic bulletin boards, mesin faksimili, dan *Electronic Funds Transfer* yang berkenaan dengan transaksi-transaksi belanja di *Internet Shopping*.

2.2.2 Web Server

Nugroho (2004), mengatakan bahwa yang dimaksud dengan web server adalah sebuah bentuk dari server yang khusus digunakan untuk menyimpan halaman *website* atau home page. Sebuah komputer dapat dikatakan sebagai web server apabila komputer tersebut memiliki suatu program server yang disebut PWS atau *Personal Web Service*. PWS ini kemudian nantinya difungsikan agar halaman web yang ada di dalam sebuah komputer server dapat dipanggil oleh komputer klien.

2.2.3 Apache

Menurut Kurniawan (2008), Apache adalah web server yang dapat dijalankan dibanyak sistem operasi (Unix, BSD, Linux, Microsoft windows dan Novell Netware serta Platfrom lainnya) yang berguna untuk memfungsikan situs web”. Protokol yang digunakan untuk melayani fasilitas web ini menggunakan HTTP.

2.2.4 Domain

Menurut Budi Raharjo Imam (2014), *Domain* adalah nama unik yang dimiliki oleh sebuah *website* yang terdiri dari dua bagian utama dan dipisahkan oleh sebuah titik. Misalnya: lintau.info, detik.com, elexmedia.co.id, dan lain-lain. Sebuah nama *domain* bukanlah bersifat hak milik, namun lebih kepada hak sewa, karena kepemilikan sebuah *domain* dibatasi oleh waktu (biasanya per tahun) dan harus membayar, jika waktunya habis anda bisa memperpanjang sewa *domain* untuk tahun-tahun berikutnya.

2.2.5 Hypertext Transfer Protocol (HTTP)

Hypertext Transfer Protocol (HTTP) adalah sebuah protokol jaringan lapisan aplikasi yang digunakan untuk sistem informasi terdistribusi, kolaboratif, dan

menggunakan hipermedia banyak dalam memanfaatkan sumber daya yang dihubungkan dengan link yang disebut dokumen hypertext yang membentuk World Wide Web. <http://www.dosenpendidikan.com>

2.2.6 Hypertext Preprocessor (PHP)

Menurut Budi Raharjo Imam (2015), PHP adalah salah satu program skrip yang dirancang untuk membangun aplikasi *web*. Ketika dipanggil dari *web browser*, program yang ditulis dengan PHP akan di-parsing didalam *web server* oleh interpreter PHP dan terjemahkan ke dalam dokumen HTML, yang selanjutnya akan di tampilkan kembali ke *web browser*. Karena pemrosesan program PHP dilakukan di lingkungan *web server*, PHP dikatakan sebagai bahasa sisi *server (server-side)*. Oleh sebab itu seperti yang telah dikemukakan sebelumnya, kode PHP tidak terlihat pada saat user memilih perintah “*View Source*” pada *web browser* yang mereka gunakan. Selain menggunakan PHP, aplikasi web juga dapat dibangun dengan Java (JSP-*Java Server Pages* dan *Servlet*), Perl, Python, Ruby, maupun ASP (*Active Server Pages*).

2.2.7 XAMPP

Menurut Riyanto (2015), XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data MySQL di komputer lokal. XAMPP berperan sebagai *server web* pada komputer Anda. XAMPP juga dapat disebut sebuah *CPanel server virtual*, yang dapat membantu Anda melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan *internet*. *Software* XAMPP dapat diperoleh secara gratis dari situs www.apachefriends.org atau dapat juga dicari di *google* untuk beberapa versi tertentu. XAMPP adalah perangkat lunak yang bebas, yang mendukung banyak sistem operasi dan merupakan komplikasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri yang terdiri atas program *Apache HTTP server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan pemrograman PHP dan *Perl*.

Menurut Arief (2011) MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengolahan datanya.

2.2.8 Cascading Style Sheet (CSS)

Menurut Bunafit Nugroho (2014), *Cascading Style Sheet* adalah bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa *markup*. CSS bekerja sebagai pelengkap pada elemen *HTML* yang kesemuanya itu dapat dikendalikan dengan menggunakan dengan menggunakan sebuah bahasa *script* CSS. Penggunaan CSS dilakukan untuk memperluas kemampuan *HTML* dalam memformat dokumen *web* atau untuk memperindah tampilan *web*. Penulisan kode CSS disisipkan pada *tag* *HTML*. Kode CSS ditulis dengan *tag* `<style>` dan `</style>` dengan mendefinisikan suatu *syle* baru yang kemudian dapat digunakan berulang kali.

2.2.9 JavaScript

Javascript menurut (Sunyoto,2007:17) adalah bahasa *scripting* yang populer di internet dan dapat bekerja di sebagian besar *browser* populer seperti Internet Explorer (IE), Mozilla Firefox, Netscape dan Opera. Kode Javascript dapat disisipkan dalam halaman web menggunakan tag `SCRIPT`.

Beberapa hal tentang Javascript:

1. Javascript didesain untuk menambah interaktif suatu *web*
2. Javascript merupakan sebuah bahasa *scripting*.
3. Bahasa *scripting* merupakan bahasa pemrograman yang ringan.
4. Javascript berisi baris kode yang dijalankan di komputer (*web browser*).
5. Javascript biasanya disisipkan (*embedded*) dalam halaman *HTML*.
6. Javascript adalah bahasa interpreter (yang berarti skrip dieksekusi tanpa proses kompilasi).
7. Setiap orang dapat menggunakan Javascript tanpa membayar lisensi.

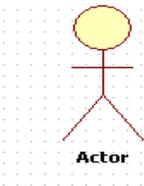
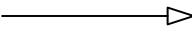
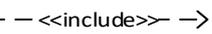
2.2.10 Unified Modeling Language (UML)

Menurut Booch (2005:7) UML adalah Bahasa standar untuk membuat rancangan *software*. UML biasanya digunakan untuk menggambarkan dan membangun, dokumen artifak dari *software-intensive system*. Model UML yang dipakai dalam pengembangan *web e-commerce* UKM CV. Warisan Multi Tenun antara lain *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, *ER Diagram*.

2.2.11 Use Case Diagram

Menurut Whitten, Bentley & Dittman (2004, p. 271) *Use case modeling* adalah sebuah pendekatan yang memfasilitasi pengembangan yang berpusat pada penggunaan. *Use case diagram* digunakan untuk memodelkan proses berdasarkan perspektif pengguna sistem. *Use case diagram* terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. Simbol-simbol yang digunakan dalam *use case diagram* dapat dilihat pada Tabel 2.1.

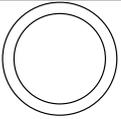
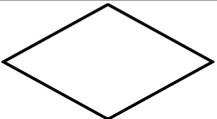
Tabel 2. 1 *Use Case Diagram*

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menggambarkan seseorang yang berinteraksi dengan sistem, di mana hanya bisa menginputkan informasi dan menerima informasi dari sistem dan tidak memegang kendali pada use case
2		<i>Dependency</i>	Sebuah elemen yang bergantung beberapa cara kepada elemen lainnya.
3		<i>Generalization</i>	Sebuah elemen yang menjadi spesialisasi dari elemen yang lain
4		<i>Include</i>	Metode yang harus terpenuhi agar sebuah event dapat terjadi.
5		<i>Extend</i>	Metode yang hanya berjalan di bawah kondisi tertentu
6		<i>Association</i>	Menghubungkan link antar element

2.2.12 Activity Diagram

Menurut Whitten, Bentley & Dittman (2004, p442), *activity* diagram digunakan untuk menggambarkan alur dari proses bisnis atau langkah – langkah *usecase* secara berurutan. Diagram ini juga digunakan untuk menggambar *action* (tindakan) yang akan dieksekusikan ketika suatu proses sedang berjalan dan beserta hasil dari proses eksekusi tersebut. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada Tabel 2.2.

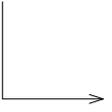
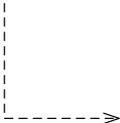
Tabel 2. 2 Activity Diagram

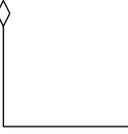
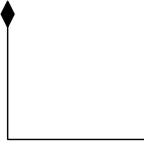
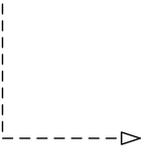
No	Gambar	Nama	Keterangan
1		<i>Start Point</i>	Merupakan awal dalam aktifitas
2		<i>End Point</i>	Akhir dalam aktivitas
3		<i>Activities</i>	Menggambarkan proses kegiatan bisnis
4		<i>Decision Point</i>	Menggambarkan pilihan pengambilan keputusan aktifitas.
5		<i>Swimlane</i>	Berfungsi sebagai pembagi activity diagram yang menunjuki siapa yang melakukan aktifitas

2.2.13 Class Diagram

Menurut Whitten, Bentley & Dittman (2004, p441), class diagram menggambarkan struktur objek yang terdapat pada sebuah sistem. Diagram ini menunjukkan objek – objek yang terdapat pada suatu sistem dan relasi antar objek – objek tersebut. Class memiliki tiga area pokok yaitu: Nama, Atribut dan Metode / Operation. Atribut dan Metoda dapat mempunyai sifat *Private* (tidak dapat dipanggil dari luar class), *Protected* (Hanya dapat dipanggil oleh class yang bersangkutan dan anak yang mewarisinya) dan *Public* (dapat dipanggil siapa saja). Berikut adalah contoh dari sebuah kelas. Class Diagram secara khas meliputi: Nama Kelas (*Class Name*), Atribut (*Attributes*), Operasi (*Operations*), dan Relasi (*Relationships*).

Tabel 2. 3 Class Diagram

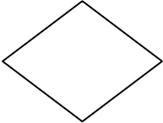
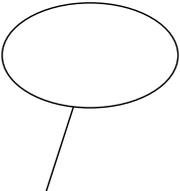
No	Gambar	Nama	Keterangan
1		<i>Assosiation</i>	Hubungan statis antar kelas. Asosiasi menggambarkan kelas yang memiliki atribut berupa kelas lain
2		<i>Directed Assosiation</i>	Asosiasi dengan makna kelas yang satu digunakan oleh kelas yang lain.
3		<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus) atau untuk menyatakan hubungan <i>intheritance</i>
4		<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.

No	Gambar	Nama	Keterangan
5		<i>Agregation</i>	Hubungan yang menyatakan bahwa suatu kelas menjadi atribut bagi kelas lain.
6		<i>Composition</i>	Bentuk khusus dari agregasi dimana kelas yang menjadi bagian diciptakan setelah kelas menjadi <i>whole</i> .
7		<i>Realization</i>	Hubungan antar kelas dimana sebuah kelas memiliki keharusan untuk mengikuti aturan yang ditetapkan oleh kelas lainnya

2.2.14 Entity Relationship Diagram (ERD)

Menurut Whitten, Bentley & Dittman (2004, p280) pemodelan data adalah teknik untuk mendefinisikan persyaratan bisnis untuk sebuah database atau teknik yang digunakan untuk mengatur dan mendokumentasikan data sistem. Biasanya disebut dengan *database modeling* atau pemodelan *database*. Terdapat beberapa alat atau tools yang digunakan untuk membuat pemodelan data. Salah satunya adalah *Entity Reliationship Diagram* (ERD) yang merupakan pemodelan data yang menggunakan beberapa notasi untuk menggambarkan data dalam konteks entitas dan hubungan yang digambarkan oleh data tersebut. Untuk menggambarkan digunakan beberapa notasi dan simbol seperti pada tabel 2.3.

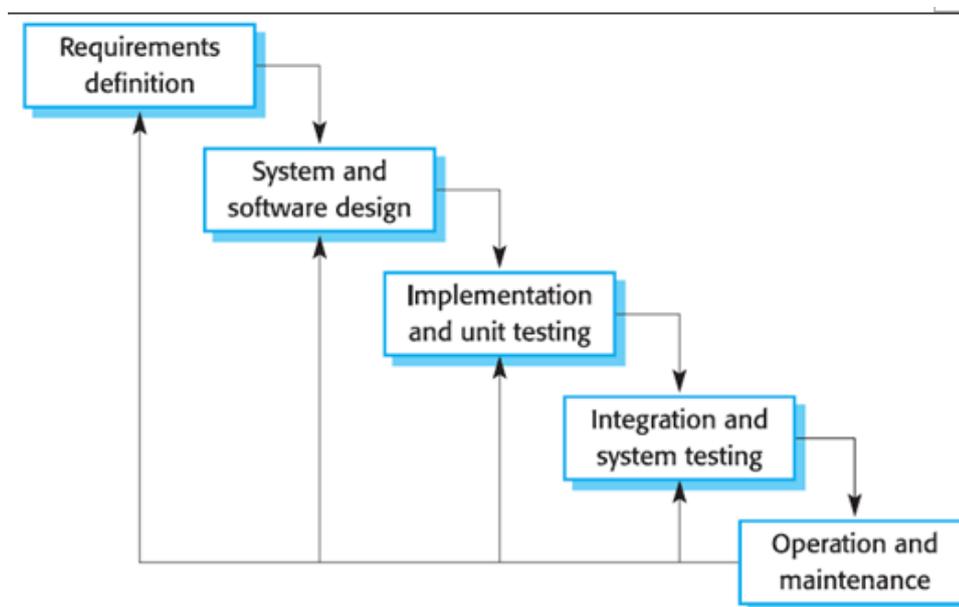
Tabel 2. 4 Tabel ER Diagram

No	Gambar	Nama Gambar	Keterangan
1		<i>Entitas</i>	Suatu objek yang dapat diidentifikasi dalam lingkungan pemakai
2		<i>Relasi</i>	Menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda
3		<i>Atribut</i>	Berfungsi mendeskripsikan karakter entitas(atribut yang berfungsi sebagai key diberi garis bawah).
4		<i>Association</i>	Sebagai penghubung antara relasi dengan entitas, relasi, dan entitas dengan atribut.

2.3 Software Development Life Cycle

Dalam alur penelitian, metode yang digunakan adalah model SDLC (*Software Development Life Cycle*). SDLC adalah suatu kerangka yang menggambarkan beberapa kegiatan yang dilakukan melalui beberapa tahap dalam pembuatan sebuah *software* (Fatta, 2007).

Selain itu, SDLC juga penting untuk proses *maintenance software* itu sendiri. Model SDLC yang dipakai dalam pengembangan aplikasi adalah model *Waterfall*. *Waterfall model* adalah sebuah contoh dari proses perencanaan dimana semua proses kegiatan harus terlebih dahulu direncanakan dan dijadwalkan sebelum dikerjakan. *Waterfall Model* atau *Classic Life Cycle* merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Disebut *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. (Sommerville, 2011),



Gambar 2. 1 Model Waterfall

Terdapat 5 tahapan metode *Waterfall* yang dapat dilihat pada gambar 2.1 di mulai dari analisis dan definisi kebutuhan hingga operasi dan pemeliharaan. Berikut penjelasan dari tahapan metode *Waterfall*:

1. Analisis dan definisi kebutuhan. Layanan, batasan, dan tujuan sistem ditentukan melalui konsultasi dengan *user*.
2. Perancangan sistem dan perangkat lunak. Proses perancangan sistem membagi persyaratan dalam sistem perangkat keras atau perangkat lunak. Kegiatan ini menentukan arsitektur sistem secara keseluruhan,

Perancangan melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar.

3. Implementasi dan pengujian unit. Pada tahap ini, perancangan perangkat lunak direalisasikan dengan program atau unit program. Pengujian ini melibatkan verifikasi bahwa setiap unit telah memenuhi spesifikasinya.
4. Integrasi dan pengujian sistem. Unit program atau program individual diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa kebutuhan sistem telah dipenuhi.
5. Operasi dan pemeliharaan, yaitu mengoperasikan program di lingkungannya dan melakukan pemeliharaan. Biasanya ini merupakan fase siklus hidup yang paling lama.

2.4 Metode Pengujian Sistem

Pengujian sistem menyajikan anomali yang menarik bagi rekayasa perangkat lunak pada proses perangkat lunak, perekrayasa berusaha membangun perangkat lunak dari konsep abstrak ke implementasi yang dapat dilihat, baru kemudian dilakukan pengujian.

2.4.1 Pengujian *Black Box*

Menurut Roger S. Pressman (2010) *Black Box Testing* atau Pengujian Kotak Hitam, berfokus pada persyaratan fungsional dari perangkat lunak. Artinya, teknik *Black Box Testing* memungkinkan untuk mendapatkan set kondisi masukan yang sepenuhnya akan melaksanakan semua persyaratan fungsional untuk suatu program. *Black box testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineer* untuk memperoleh *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program. *Black box testing* berusaha untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi yang tidak benar atau fungsi yang hilang.
2. Kesalahan antarmuka.
3. Kesalahan dalam struktur data atau akses *database* eksternal.
4. Kesalahan kinerja.
5. Kesalahan inisialisasi dan pemutusan kesalahan.