

BAB II TINJAUAN PUSTAKA

1.1 Tinjauan Pustaka

Tinjauan pustaka dilakukan berdasarkan kajian pada penelitian terdahulu penelitian-penelitian terdahulu menjadi bahan acuan dan kajian terhadap penelitian yang dilakukan saat ini. Kajian yang diperoleh dapat dijadikan acuan perbandingan yang sesuai dengan topik penelitian, yaitu mengenai sistem perancangan dan implementasi *e-commerce* berbasis *website* yang digunakan sebagai media promosi bisnis di suatu organisasi atau perusahaan.

Untuk meningkatkan kinerja bisnis pada suatu perusahaan, dibutuhkan *website e-commerce* yang bertujuan untuk mempermudah dan mempercepat transaksi pembelian dan mempermudah konsumen dalam bertransaksi tanpa perlu datang langsung ke toko yang dituju (Djamal Bachmid, 2011). Penelitian yang dilakukan oleh Djamal Bachmid ini berjudul “Pembuatan *Website E-Commerce* pada toko Jawa Tren Menggunakan PHP dan MySQL”.

“Pengembangan Sistem Informasi Manajemen *Order Satoedjari Merch* Menggunakan Model *eXtreme Programing*” (Iqbal, 2015). Penelitian tersebut bertujuan untuk memudahkan *Satoedjari Merch* dalam mengelola *order* yang sebelumnya masih dilakukan dengan cara manual dengan mengembangkan sistem informasi manajemen *order* yang dapat mengelola *order*, harga jual produk, pengeluaran dan laporan keuangan berupa laporan pendapatan, pengeluaran dan profit. Sistem ini dibangun menggunakan metode *eXtreme programming*, bahasa pemrograman PHP dan database *MySQL*.

Deffi Rosdiana Sari (2014) juga melakukan penelitian dengan judul “Membangun Aplikasi Sistem Pemesanan dan Pembayaran Sewa Mobil *Online* Berbasis *Web* pada Studi Kasus di Rental Daras *Corporation*” mengenai *Web Company profile* beserta sistem pemesanan dan pembayarannya. Persamaan yang ada pada penelitian kali ini dan yang sebelumnya adalah sama-sama mengembangkan *web* menggunakan *apache*, PHP dan MySQL untuk implementasi *web* nya. Perbedaannya penelitian yang sekarang dan yang terdahulu adalah

penelitian yang sekarang menggunakan XAMPP sedangkan pada penelitian sebelumnya menggunakan WAMP sebagai *web server* nya. Sedangkan pada penelitian David Setyawan (2015) dengan judul “Perancangan *Web Mobile* sebagai Media Pemasaran pada *I-CLUB* Madiun menggunakan *JQuery mobile*” penelitian yang dikembangkan adalah pembuatan *website* sebagai media pemasaran berbasis *web mobile*. Persamaan penelitian yang terdahulu dan sekarang adalah sama-sama membuat *website* sebagai media pemasaran atau *e-marketing* untuk memperluas jangkauan pemasaran. Perbedaan penelitian yang terdahulu dengan yang sekarang terletak pada sistem yang dikembangkan. Pada *website* yang ada pada penelitian terdahulu hanya untuk *web mobile* pada *smartphone* dan belum bisa menjadi sistem *website* pada *desktop*.

Setelah membandingkan penelitian-penelitian terdahulu dapat disimpulkan bahwa persamaannya adalah sama-sama membuat aplikasi berbasis *web* sebagai media pemasaran atau promosi *online* yang memudahkan konsumen untuk mendapatkan informasi terbaru. Sedangkan perbedaannya adalah aplikasi berbasis *web* ini dibangun dengan konsep bahasa pemrograman PHP dan *database* menggunakan MySQL serta XAMPP sebagai *web server*nya selain itu untuk mengembangkan desain *web* yang lebih *user friendly* dan menarik maka digunakan HTML5 *Responsive* dan *css* sebagai pelengkap desain *web* yang dapat dibuka melalui berbagai *device* atau perangkat.

1.2 Landasan Teori

1.2.1 Sistem

Sistem menurut Abdul Kadir (2002: 54) “Sistem adalah sekumpulan elemen yang saling berkaitan atau terpadu yang dimaksudkan untuk mencapai suatu tujuan”. Sebagai gambaran, jika dalam sebuah sistem terdapat elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama, maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem.

1.2.2 Website

Menurut Abdul Kadir (2009:10). *Website* atau situs juga dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar

diam atau gerak, data *animasi*, suara, *video* dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimanamasing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*). Bersifat statis apabila isi informasi *website* tetap, jarang berubah, dan isi informasinya searah hanya dari pemilik *website*. Bersifat dinamis apabila isi informasi *website* selalu berubah-ubah, dan isi informasinya interaktif dua arah berasal dari pemilik serta pengguna *website*. Contoh *website* statis adalah berisi profil perusahaan, sedangkan *website* dinamis adalah seperti *Friendster*, *Multiply*, dll. Dalam sisi pengembangannya, *website* statis hanya bisa di *update* oleh pemiliknya saja, sedangkan *website* dinamis bisa di *update* oleh pengguna maupun pemilik.

1.2.3 E-commerce

E-Commerce merupakan suatu sistem atau paradigma baru dalam dunia bisnis, yang menggeser paradigma perdagangan tradisional menjadi *electronic commerce* yaitudengan memanfaatkan teknologi ICT (*Information and Communication Technology*), atau dengan kata lain teknologi *internet*. Definisi *e-commerce* secara umum: “Proses membeli, menjual, baik dalam bentuk barang, jasa ataupun informasi, yang dilakukan melalui media *internet*”. Menurut Stefan Probst (Opticom), definisi *e-commerce* adalah “Bisnis yang dilakukan secara *electronic* yang melibatkan aktivitas-aktivitas bisnis berupa *business to business* ataupun *business to konsumen* melalui teknologi *internet*.” *E-business* adalah transaksi yang menggunakan media elektronik yang dipergunakan untuk berjualan atau proses pembelian atau proses pembelian suatu atau beberapa produk menggunakan teknologi ICT. Secara umum, interaksi dan transaksi antara pelaku bisnis yang akan menggunakan teknologi *ecommerce* dapat dikategorikan dalam jenis B2B (*business to business*), B2C (*business to konsumen*), C2B (*konsumen to business*), dan C2C (*konsumen to konsumen*). Andi (2008)

1.2.4 Internet

Menurut Wibowo Turnady (Turnady Weblog:2009 06 Mei 2010) *Internet* adalah sebuah jaringan komputer yang terdiri dari berbagai macam

ukuran jaringan komputer di seluruh dunia mulai dari sebuah PC, jaringan-jaringan lokal berskala kecil, jaringan-jaringan kelas menengah, hingga jaringan-jaringan utama yang menjadi tulang punggung *internet* seperti *NSFnet*, *NEARnet*, *SURAnet*, dan lain-lain.

1.2.5 PHP

Menurut Arif (2011c:43) *PHP* adalah bahasa *server-side-scripting* yang menyatu dengan *HTML* untuk membuat halaman *web* yang dinamis. Karena *PHP* menurut *server-side-scripting* maka sintaks dan perintah-perintah *PHP* akan dieksekusi di *server* kemudian hasilnya akan dikirimkan ke *browser* dengan format *HTML*. Dengan demikian kode program yang ditulis dalam *PHP* tidak terlihat oleh *user* sehingga keamanan halaman *web* lebih terjamin. *PHP* dirancang untuk membuat halaman *web* yang dinamis, yaitu halaman *web* yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman *web*.

1.2.6 PhpMyAdmin

Menurut Sibero (2011e:376) “*phpMyAdmin* adalah aplikasi *web* yang dibuat oleh *phpMyAdmin*. *Net phpMyAdmin* digunakan untuk administrasi *database MySQL*”. Program ini digunakan untuk mengakses *database MySQL*. Perintah untuk membuat *table* dapat menggunakan *form* yang sudah tersedia pada *phpMyAdmin* atau dapat langsung menulis *script* pada menu *sql*. *PhpMyAdmin* dijalankan dengan cara mengetik *http://localhost/phpmyadmin* pada *web browser*.

1.2.7 Web Server

Menurut Kadir, Abdul (2009), *Web server* merupakan sebuah aplikasi perangkat lunak. Aplikasi *web server* ini dijalankan pada sebuah komputer yang disebut dengan *server*. *Web server* adalah salah satu jenis perangkat lunak yang menyediakan layanan halaman (dokumen) *web* yang dapat diakses di seluruh dunia melalui *internet*.

Web server akan melayani permintaan akses halaman *web* dengan bantuan protokol komunikasi terutama *HTTP*.

Salah satu contoh *platform* tersebut adalah *apache*. *Apache* adalah aplikasi *web server* yang tersedia secara gratis dan disebarakan dengan lisensi *open source*.

1.2.8 UML

Berikut ini definisi *Unified Modeling Language (UML)* menurut para ahli:

1. Menurut (Adi Nugroho: 2005). “*Unified Modeling Language (UML)* adalah alat bantu analisis serta perancangan perangkat lunak berbasis objek”.
2. Menurut (Joomla dari <http://soetrasoft.com:2007>), “*Unified Modeling Language (UML)* merupakan standar *modeling language* yang terdiri dari kumpulan-kumpulan diagram, dikembangkan untuk membantu para pengembang sistem dan *software* agar bisa menyelesaikan tugas-tugas seperti: Spesifikasi, Visualisasi, Desain Arsitektur, Konstruksi, Simulasi dan testing serta Dokumentasi”.





Berdasarkan beberapa pendapat yang dikemukakan diatas dapat ditarik kesimpulan bahwa “*Unified Modeling Language (UML)* adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis OO (*Object Oriented*)”.

a. Use Case Diagram



Menurut Abdul Kadir (2003:54). *Use case* menggambarkan suatu urutan interaksi antara satu atau lebih aktor dan sistem. Dalam fase *requirements*, model *use case* menggambarkan sistem sebagai sebuah kotak hitam dan interaksi antara aktor dan sistem dalam suatu bentuk naratif, yang terdiri dari input *user* dan respon-respon sistem. Setiap *use case* menggambarkan perilaku sejumlah aspek sistem, tanpa mengurangi struktur *internal* nya. Selama pembuatan model *use case* secara paralel juga harus ditetapkan objek-objek yang terlibat dalam setiap *use case*. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *usecase* dapat meng-include

fungsionalitas *usecase* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *usecase* yang di *include* akan dipanggil setiap kali *usecase* yang meng-*include* di *eksekusi* secara normal.

Keterangan simbolis *usecase*.

Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>.</p>
<p>Aktor atau <i>actor</i></p> 	<p>Orang proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya menggunakan benda di awal frase nama aktor.</p>
<p>Asosiasi atau <i>association</i></p> 	<p>Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p>Ekstensi atau <i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang Di tambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.</p>



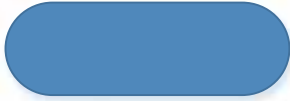
Keterangan simbolis *usecase* (Lanjutan)

Simbol	Deskripsi
Generalisasi atau <i>generalization</i> 	Hubungan generalisasi dan spesialis (umum - khusus) antara dua buah <i>use case</i> dimana fungsi yang umum dari lainnya.
Menggunakan atau <i>include</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.



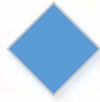

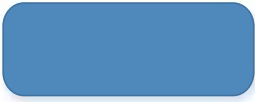
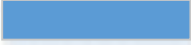

b. Activity Diagram

Activity diagram menurut Martin fowler (2005: h, 163) adalah “teknik untuk menggambarkan logika procedural, proses bisnis, dan jalur kerja”. Dalam beberapa hal, *activity* diagram memainkan peran mirip diagram alir, tetapi perbedaan prinsip antara notasi diagram alir adalah *activity* diagram mendukung behavior parallel. Node pada sebuah *activity* diagram disebut sebagai *action*, sehingga diagram tersebut menampilkan sebuah *activity* yang tersusun *action*.

Tabel keterangan simbolis *activity* diagram

Simbol	Deskripsi
<i>Start state</i> 	Menggambarkan awal dari suatu aktivitas yang berjalan pada sistem.
<i>End State</i> 	Menggambarkan akhir dari suatu aktivitas yang berjalan pada sistem.
<i>Activity</i> 	Menggambarkan aktivitas yang dilakukan pada sistem.

Tabel 2.2. Tabel keterangan simbolis *activity* diagram (Lanjutan)

Simbol	Deskripsi
<p data-bbox="363 383 592 416"><i>Transition to self</i></p> 	<p data-bbox="874 383 1356 528">Menggambarkan hubungan antara <i>state</i> atau <i>activity</i> yang kembali kepada <i>state</i> atau <i>activity</i> itu sendiri.</p>
<p data-bbox="363 573 571 607"><i>Transition state</i></p> 	<p data-bbox="874 573 1356 719">Menggambarkan hubungan antara dua <i>state</i>, dua <i>activity</i> atau antara <i>state</i> dan <i>activity</i>.</p>
<p data-bbox="363 748 480 781"><i>Decition</i></p> 	<p data-bbox="874 748 1356 893">Menggambarkan kondisi dari sebuah aktivitas yang bernilai benar atau salah.</p>
<p data-bbox="363 938 496 972"><i>Swimlane</i></p> 	<p data-bbox="874 938 1356 1084">Menggambarkan pembagian atau pengelompokan berdasarkan tugas dan fungsi tersendiri.</p>
<p data-bbox="363 1240 432 1274"><i>State</i></p> 	<p data-bbox="874 1240 1356 1330">Menggambarkan kondisi, situasi atau tempat untuk beberapa aktivitas.</p>
<p data-bbox="363 1431 432 1464"><i>Fork</i></p> 	<p data-bbox="874 1431 1356 1632">Menggambarkan aktivitas yang dimulai dengan sebuah aktivitas dan diikuti oleh dua atau lebih aktivitas yang harus dikerjakan.</p>
<p data-bbox="363 1655 424 1688"><i>Join</i></p> 	<p data-bbox="874 1655 1356 1856">Menggambarkan aktivitas yang dimulai dengan dua atau lebih aktivitas yang sudah dilakukan dan menghasilkan sebuah aktivitas.</p>

c. Class Diagram

Class diagram adalah diagram yang menunjukkan *class* yang ada dari sebuah sistem dan hubungannya secara logika (Henderi, 2008 dalam buku Martin Fowle, UML Distilled, 2005). *Class* diagram menggambarkan struktur statis dari sebuah sistem. Sementara menurut (Whitten L. Jeffery et al 2004:432), *class* diagram adalah gambar grafis mengenai struktur objek statis dari suatu sistem, menunjukkan *class* objek yang menyusun sebuah sistem dan juga hubungan antar *class* objek tersebut.

Tabel 2.3. Keterangan simbolis *class* diagram

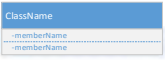




Simbol	Deskripsi
<p><i>Class</i></p> 	<p><i>Class</i> blok-blok pembangunan pada pemrograman berorientasi Obyek, Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas tiga bagian. Bagian atas adalah bagian nama dari <i>class</i>, bagian tengah mendefinisikan <i>property</i> atau atribut <i>class</i>, bagian akhir mendefinisikan <i>method</i> dari sebuah <i>class</i></p>
<p><i>Association</i></p> 	<p>Sebuah asosiasi merupakan sebuah <i>relationship</i> paling umum antara dua <i>class</i> dan dilambangkan oleh sebuah garis yang menghubungkan antara dua <i>class</i>.</p>
<p><i>Compotion</i></p> 	<p>Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>composition</i> terhadap tempat dia bergantung tersebut.</p>

Table 2.3. Tabel keterangan simbolis *class* diagram (Lanjutan)

Simbol	Deskripsi
<p><i>Dependency</i></p> 	<p>Kadangkala sebuah <i>class</i> menggunakan <i>class</i> lain, hal ini disebut <i>dependency</i>. Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada Sebuah <i>class</i> yang menggunakan <i>class</i> yang lain</p>
<p><i>Aggregation</i></p> 	<p><i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi.</p>

1.3 Entity Relationship Diagram (ERD)

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. *ERD* untuk memodelkan struktur data dan hubungan antar data, untuk menggambarannya digunakan beberapa notasi dan simbol.

Menurut Brady dan Loonam (2010), *ERD* merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh sistem analis dalam tahap analisis persyaratan proyek pengembangan sistem. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain *database* relasional yang mendasari sistem informasi yang dikembangkan. *ERD* bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk *database*.

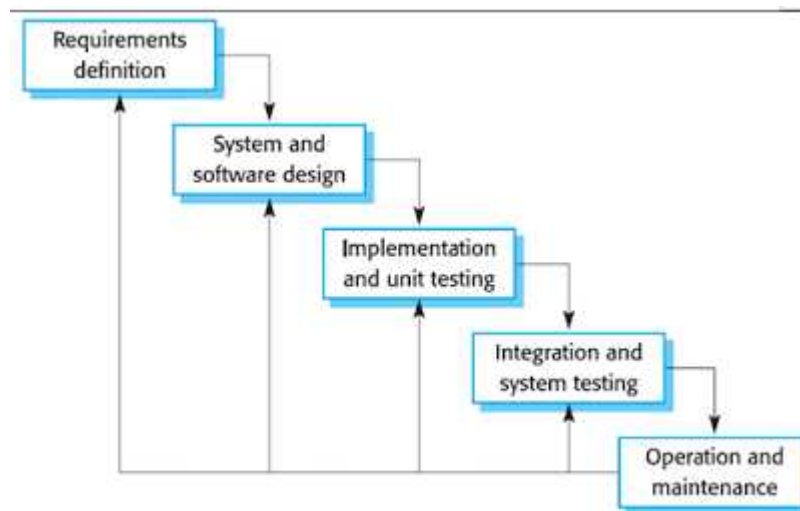
Notasi	Keterangan
	Entitas, adalah suatu objek yang dapat diidentifikasi dalam lingkungan tertentu.
	Relasi, menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
	Atribut, merupakan mendefinisikan besaran/nilai (atribut yang berfungsi sebagai key di beri garis bawah).
	Garis, sebagai pendukung antara relasi dengan entitas, dan antar entitas lainnya.

Sumber: [1] Dwiyo Nugroho, Database ER.

Gambar 2.1. Simbol-simbol *ERD*

1.4 Metodologi *Waterfall*

Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak berurutan, dimana kemajuan dipandang sebagai terus mengalir ke bawah melewati fase-fase perencanaan, pemodelan, implementasi dan pengujian. Berikut adalah gambaran pengembangan perangkat lunak berurutan.



Gambar 2.2. Metode *Waterfall*

1.5 Metode Pengujian Sistem

Pengujian sistem menyajikan anomali yang menarik bagi perancang perangkat lunak. Pada proses perangkat lunak, perancang berusaha membangun perangkat lunak dari konsep abstrak ke implementasi yang dapat di lihat, baru kemudian di lakukan pengujian.

1.6 Pengujian *Black Box*

Menurut Hanif (2007). Dalam pengujian perangkat lunak ada dua yaitu *white box testing* dan *black box testing*. Dari kedua metode itu, pada skripsi dipilih menggunakan *black box testing* karena dianggap lebih tepat dibanding *white box testing*. Perangkat lunak memerlukan seperangkat tes untuk pencarian kesalahan fungsi-fungsi dalam aplikasi sehingga dalam hal ini *black box testing* lebih sesuai. Pengujian ini digunakan untuk mengetahui apakah fungsi-fungsi dalam perangkat lunak sudah sesuai dengan yang diharapkan.

Menurut Roger S. Pressman (2010), *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.

Kesalahan inisialisasi dan pemutusan kesalahan