

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Evans Fuad (2010) dengan judul “Perancangan *E-Commerce* Pada Toko *Online* Mainan Bocah Yogyakarta” melakukan penelitian dengan tujuan mendukung kinerja toko *online* mainan anak lebih cepat dan lebih efektif. Akan tetapi masih terdapat kelemahan karena belum terdapat fitur konfirmasi dari admin kepada pelanggan dan juga laporan penjualan atau laporan bulanan.

Pratama, Bayu Kaesar (2012) melakukan penelitian yang berjudul “Membangun Aplikasi *E-Commerce* Dengan Menggunakan Framework Codeigniter (Studi Kasus Caesar Distro)” Tujuan penelitian ini sendiri adalah memberikan kemudahan untuk pelanggan dan admin pada saat order barang di Caesar Distro. Dari sistem yang telah dibuat tersebut masih memiliki kelemahan, karena belum terdapat fitur laporan penjualan agar memiliki gambaran tentang kondisi penjualan perusahaan secara nyata dan juga masih belum adanya fitur konfirmasi, karena pelanggan hanya melakukan *checkout* tanpa ada konfirmasi balik dari admin.

Elvira Yolanda (2013) dengan judul “Perancangan Aplikasi *E-Commerce* Berbasis *Web* Pada Pempek Rendy” melakukan penelitian dengan tujuan lebih mengenalkan tentang keberadaan Pempek Rendy kepada masyarakat luas melalui *website* dan ingin membangun *system* yang lebih baik dari *system* sebelumnya. Dalam mengembangkan *system* ini masih terdapat kekurangan yaitu masih belum ada fitur konfirmasi setelah melakukan *checkout* dan juga laporan bulanan.

Dari penelitian yang sebelumnya telah dilakukan, bahasa pemrograman PHP dan *database* MySQL telah banyak diterapkan untuk pembangunan sistem berbasis *website*. Dengan begitu, pada penelitian ini akan dibangun *website e-commerce* MAXCloth yang dilengkapi dengan sistem konfirmasi *checkout* dan juga laporan bulanan. Program ini dibangun untuk memberikan kemudahan bagi setiap pelanggan dalam melakukan proses transaksi, dan untuk memudahkan

admin dalam proses pengolahan data. Pengembangan sistem menggunakan metode *waterfall*. Dalam pembangunannya website ini menggunakan *template bootstrap*. *Bootstrap* juga mempunyai keunggulan dengan tampilan yang lebih optimal serta menyesuaikan lebar layar dan dapat diakses melalui berbagai perangkat dengan kecepatan *loading* yang lebih cepat. Pembuatan *website* ini menggunakan bahasa pemrograman PHP, *Javascript*, *CSS* dan *databasenya* menggunakan *MySql*.

2.2. Landasan Teori

2.2.1. Pengertian Website

Fasilitas internet yang dihubungkan dokumen dengan lingkup lokal maupun jarak jauh merupakan pengertian *Website*. *Web page* dan *link* merupakan dokumen pada *website* dapat mempermudah pengguna untuk *browsing* dari satu halaman ke halaman berikutnya (*hypertext*) yang disimpan dalam *server* yang sama maupun *server* diseluruh dunia. Halaman dapat diakses dan dibaca menggunakan *browser* yaitu , *Internet Explorer*, *Mozilla Firefox*, *Google Chrome* dan aplikasi *browser* lainnya. (Yuhefizar, 2008)

2.2.2. Pengertian Sistem Informasi

Sistem informasi adalah komponen yang saling berkaitan dan bekerja sama yang bertujuan untuk mengumpulkan, mengolah, menyimpan, dan menampilkan informasi untuk mendukung dalam pengambilan keputusan, koordinasi, pengaturan, analisa, dan visualisasi pada sebuah organisasi (Laudon, 2012, p. 16). Data dikumpulkan, dikelompokkan dan diolah dengan tujuan untuk menjadikan suatu informasi berharga bagi yang menerimanya adalah pengertian sistem informasi. (Mahyuzir, T.D. 2001)

2.2.3. Pengertian E-Commerce

Website e-commerce menurut William (2008), menjadi salah satu cara transaksi bisnis *online* atau perdagangan secara elektronik telah mengubah bisnis menjadi *online* tidak hanya memperluas pilihan produk dan jasa bagi konsumen, tetapi juga membentuk peluang bisnis baru dan memperkuat bisnis yang telah ada untuk mengembangkan strategi dengan internet.

2.2.4. PHP

PHP adalah singkatan rekursif dari *Hypertext Preprocessor*, adalah Bahasa pemrograman yang dapat digunakan untuk tujuan umum, sama seperti bahasa pemrograman lain, seperti: C, C++, Pascal, Python, Perl, Ruby dan sebagainya. Dalam proses pembuatan halaman *web*, PHP tidak memerlukan kode yang panjang seperti pada kode Perl dan Python (misalnya) karena kode PHP dapat disisipkan dalam kode HTML (Budi Raharjo, 2015). Fungsi utama PHP dalam membangun *website* adalah untuk melakukan pengolahan data pada *database*. (Rohi Abdulloh, 2016)

Menurut (Budi Raharjo, 2015), Pemfokusan PHP lebih kepada proses pembuatan aplikasi *web* yang disebut: *server-side scripting*, dengan kemampuan yang lebih optimal dan diaplikasikan menjadi 3 tipe aplikasi, yaitu: aplikasi *web* (*server-side scripting*) bertujuan agar kode dapat dijalankan oleh PHP Interpreter, *server web* dalam keadaan aktif. Akses halaman *web* itu sendiri dapat menggunakan aplikasi *web browser* untuk tipe yang pertama. Tipe aplikasi kedua adalah program CLI (*command-line scripting*) memiliki keunggulan untuk melakukan tugas regular dengan pengerjaan dibalik layar seperti proses *backup* data. Aplikasi *desktop* (GUI) merupakan tipe ketiga dalam aplikasi yang bertujuan untuk mengembangkan aplikasi desktop dengan PHP.

2.2.5. HTML

HyperText Markup Language merupakan penjabaran dari HTML untuk menulis halaman web. Standar pemformatan dokumen teks berkembang menjadi HTML yaitu *Standard Generalized Markup Language* (SGML). HTML pada dasarnya merupakan dokumen ASCII atau teks biasa, yang di rancang untuk tidak tergantung pada suatu sistem operasi tertentu.

Fungsi utama dari HTML adalah memberikan perintah pada browser untuk melakukan manipulasi tampilan melalui tag-tag yang ditulis dalam HTML. Dengan demikian, browser akan menghasilkan tampilan yang sesuai dengan perintah-perintah yang sudah dibuat atau ditetapkan terlebih dahulu. (Koesheryatin, T.S. 2014)

2.2.6. Bootstrap

Bootstrap memiliki pengertian *framework front-end* yang intuitif dan powerful untuk pengembangan aplikasi web yang lebih cepat dan mudah. *Bootstrap* menggunakan HTML, CSS, dan *Javascript*. Salah satu kelebihan *Bootstrap* adalah *framework* ini berisi kumpulan tool yang gratis untuk membuat layout *web* yang fleksibel dan responsive. *Framework* ini juga memiliki komponen interface bagus lainnya. (Jubie Enterprise. 2016)

2.2.7. MySql

My Sequel memiliki definisi yaitu sebuah perangkat lunak dengan manajemen sistem berbasis data SQL atau DBMS. *My Sequel* atau *MySql* adalah perangkat lunak pada *Relational Database Management System* (RDBMS) dengan manajemen *database* sebagai basis data didalamnya. Dengan kata lain, *MySql* ini banyak digunakan pada *web* sebagai *database* sehingga data semakin terintegrasi antara *database dekstop* dengan *database web*. (Alexander F.K. Sibero. 2011)

2.2.8. PhpMyAdmin

PhpMyAdmin.net membuat sebuah aplikasi *web* dengan nama *PhpMyAdmin*. *PhpMyAdmin* dapat digunakan pada administrasi *database* MySQL. Program tersebut bertujuan untuk mengakses *database* MySQL. Perintah pembuatan tabel menggunakan *form* yang telah tersedia pada aplikasi *web* *PhpMyAdmin* atau langsung dituliskan pada menu *script* SQL. *PhpMyAdmin* dapat dijalankan dengan mengetik `http://localhost/phpmyadmin` pada *web browser*. (Alexander F.K. Sibero, 2011)

2.2.9. Web Server

Menurut Anhar (2010), *Web server* adalah sebuah aplikasi yang memiliki fungsi menerima permintaan untuk memanggil alamat dari pengguna melalui *web browser*. Informasi dikirimkan kembali melalui HTTP (*Hypertext Transfer Protocol*) pada tampilan layar monitor komputer. Disimpulkan bahwa *web server* merupakan sebuah perangkat lunak yang bertugas menerima permintaan client melalui port HTTP ataupun HTTPS dan merubah isi yang ada ke dalam format HTML.

2.2.10. XAMPP

XAMPP adalah sebuah software yang berfungsi untuk menjalankan website berbasis PHP dan menggunakan pengolah data MYSQL di komputer lokal. Peran XAMPP sebagai server web pada komputer lokal. XAMPP juga disebut sebagai sebuah Cpanel server virtual, yang dapat membantu melakukan preview sehingga dapat dimodifikasi website tanpa harus online atau terakses dengan internet. (Wicaksono, 2008:7)

2.2.11. Notepad ++

Notepad ++ adalah sebuah penyunting teks dan penyunting kode sumber yang berjalan di sistem operasi windows. *Notepad ++* menggunakan komponen *Scintilla* untuk dapat menampilkan dan menyunting teks dan berkas kode sumber berbagai bahasa pemrograman.(SourceForge.net, 2011)

2.2.12. Unified Markup Language






Unified Modelling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML dapat dibuat model untuk semua jenis piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam Bahasa pemrograman apapun. (Prastuti Sulistyorini, 2009)

UML yang digunakan dalam pengembangan *website* sistem pengembangan aplikasi perizinan perkuliahan, antara lain:

a. *Activity Diagram*

Activity Diagram merupakan diagram yang digunakan untuk menggambarkan *workflow* (aliran kerja) yang digunakan untuk proses bisnis dalam perangkat lunak. Dalam *activity diagram* mempunyai simbol yang dapat dilihat dalam Tabel 2.1. (Winda Aprianti. 2016)

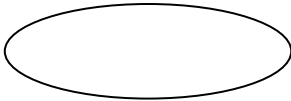
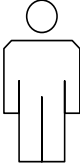

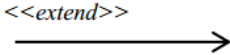

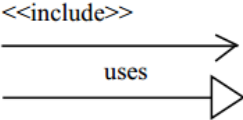
Tabel 2. 1 Simbol-simbol dalam *activity diagram*

| No. | Gambar | Nama | Keterangan |
|-----|---|--------------|---|
| 1 |  | Status Awal | Pada sebuah aktifitas sebuah diagram aktifitas mempunyai sebuah status awal |
| 2 |  | Aktivitas | Aktivitas diawali dengan kata kerja. |
| 3 |  | Percabangan | Pilihan aktivitas lebih dari satu merupakan ssosiasi percabangan. |
| 4 |  | Penggabungan | Dimana aktivitas digabungkan menjadi satu. |
| 5 |  | Status Akhir | Dibutuhkan untuk mengakhiri sebuah diagram aktifitas. |

b. *Use case Diagram*

Use Case mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat (M.Shalahudin & A.S, R. 2013). Simbol yang terdapat pada *use case* dijelaskan dengan Tabel 2.2.

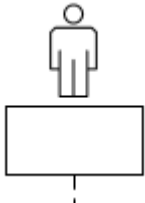
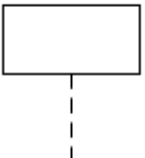


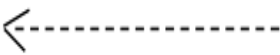
Tabel 2. 2 Simbol yang terdapat pada *use case diagram*

| No. | Gambar | Nama | Keterangan |
|-----|---|---|--|
| 1. |  | <i>Use Case</i> | Fungsi yang tersedia dalam unit saling bertukar pesan antara unit dan aktor. |
| 2. |  | <i>Actor</i> | Simbol manusia bertujuan untuk dapat berinteraksi dengan sistem informasi diluar sistem. |
| 3. |  | Asosiasi atau <i>association</i> | Komunikasi antar aktor dan <i>Use Case</i> yang berpartisipasi. |
| 4. |  | Ekstensi atau <i>extend</i> | Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambah dapat berdiri sendiri walau tanpa <i>Use Case</i> tambahan. |
| 5. |  | Generalisasi atau <i>generalization</i> | Hubungan generalisasi dan spesialisasi antara dua buah <i>Use Case</i> yang mana fungsi yang satu lebih umum dari yang lainnya. |
| 6. |  | <i>Include/Use Case</i> | Bertujuan untuk menjalankan fungsi yang terdapat pada <i>use case</i> . |

c. *Sequence Diagram*

Dalam buku *System Analisis and Dedign in a Charging World*, "System sequence diagram (SSD) adalah diagram yang digunakan untuk mendefinisikan input dan output serta urutan interaksi antara penggunaan dan sistem untuk sebuah use case". Simbol – simbol yang ada dalam *sequence diagram* dapat dilihat pada Tabel 2.3. (John Satzinger, 2010)

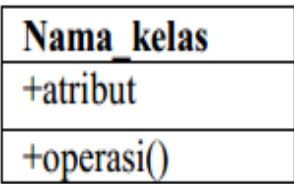
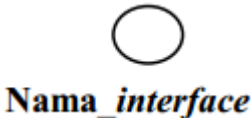



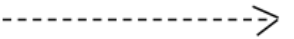

Tabel 2. 3 Simbol-simbol dalam *sequence diagram*.

| No. | Gambar | Nama | Keterangan |
|-----|---|---------------------------|---|
| 1. |  | <i>Actor</i> | Mewakili seorang aktor (orang atau peran yang berinteraksi dengan sistem). |
| 2. |  | <i>Object Lifelines</i> | Objek yang mewakili keseluruhan sistem yang terotomatisasi. |
| 3. |  | <i>Lifelines</i> | Garis putus-putus vertikal yang merupakan perpanjangan objek tersebut, baik aktor maupun objek, sepanjang durasi dari <i>sequence diagram</i> . |
| 4. |  | Anak panah tidak bergaris | Anak panah antara <i>lifeline</i> mewakili <i>message</i> yang dikirim. |
| 6. |  | Anak panah bergaris | Anak panah antara <i>lifeline</i> mewakili <i>message</i> yang diterima oleh aktor dari sistem |

d. *Class Diagram*

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan di buat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode operasi. Simbol-simbol yang ada pada *Class Diagram* dapat di lihat pada Tabel 2.4. (Winda Apriant, 2016)





Tabel 2. 4 Simbol-simbol dalam *class diagram*

| No. | Gambar | Nama | Keterangan |
|-----|---|---|---|
| 1. |  | Kelas | Kelas pada struktur sistem. |
| 2. |  | Antarmuka atau <i>interface</i> | Sama dengan konsep <i>interface</i> dalam prnograman berorientasi objek. |
| 3. |  | Asosiasi atau <i>association</i> | Relasi antarkelas dengan makna umum, sosiasi biasanya disertai dengan <i>multiplicity</i> . |
| 4. |  | Asosiasi berarah atau <i>directed association</i> | Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> . |
| 5. |  | Generalisasi | Relasi antarkelas dengan makna generalisasi-spealisasi (umum khusus) |
| 6. |  | Kebergantungan/ <i>dependency</i> | Relasi antarkelas dengan makna kebergantungan antarkelas. |
| 7. |  | Agregasi/ <i>aggregation</i> | Relasi antarkelas dengan makna semua-bagian. |

e. ERD

Entity Relationship Diagram (ERD) merupakan suatu model data yang dikembangkan berdasarkan objek. ERD digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logis dengan kemudahan pemahaman dalam penggunaannya. Perancangan model membantu dalam analisis sistem dan perancangan dikarenakan adanya kebutuhan dan relasi antar data di dalamnya, keterangan simbol yang ada pada ERD dapat dilihat pada Tabel 2.5. (Sutanta, Edhy. 2011)

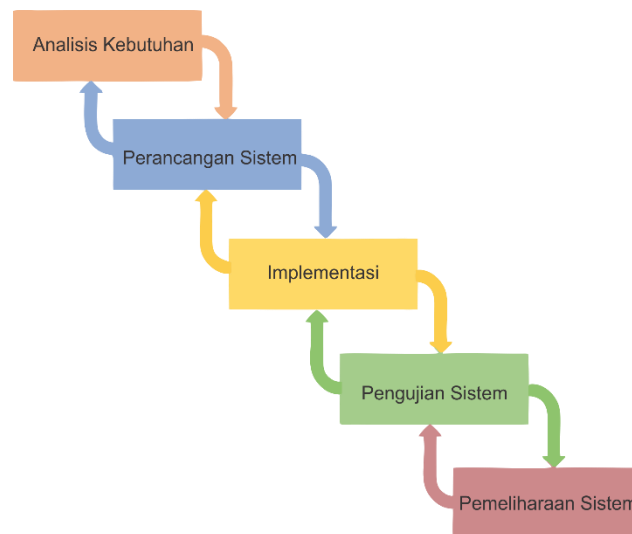
Tabel 2. 5 Keterangan Simbol ERD

| No. | Gambar | Nama | Keterangan |
|-----|---|---------|---|
| 1. |  | Entitas | Kumpulan dari objek yang dapat diidentifikasi secara unik. |
| 2. |  | Relasi | Hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan antara lain; satu ke satu, banyak ke banyak, dan banyak ke banyak. |
| 3. |  | Atribut | Karakteristik dari entity atau relasi yang merupakan penjelasan detail tentang entitas. |
| 4. |  | Garis | Hubungan antara entity dengan atributnya dan himpunan entitas dengan himpunan relasi. |

2.2.13. Metode Waterfall

Metode air terjun atau yang sering disebut dengan metode *waterfall* sering disebut siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem kepada pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan dan juga mempunyai tahapan yang dapat dilihat pada Gambar 2.1.

(Pressman S. Roger , 2012).



Gambar 2. 1 Langkah Penelitian dengan Metode *Waterfall*

Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang berurut yaitu *requirement* (analisis kebutuhan), *design system* (desain sistem), *coding* (pengkodean) dan *testing* (pengujian), penerapan program serta terakhir pemeliharaan. Penjelasan tahap per tahap dari metode *waterfall* sebagai berikut :

1. Analisis Kebutuhan.

Seluruh kebutuhan *software* harus bisa didapatkan dalam fase ini, termasuk didalamnya kegunaan *software* yang diharapkan pengguna dan batasan *software*.

2. Perancangan Sistem.

Tujuan dalam tahap ini untuk memberikan gambaran apa yang seharusnya dikerjakan dan bagaimana tampilannya. Tahap ini membantu dalam menspesifikasikan kebutuhan *hardware* dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan.

3. Implementasi.

Dalam tahap ini dilakukan pemrograman. Pembuatan *software* dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya.

4. Pengujian Sistem.

Di tahap ini dilakukan penggabungan modul-modul yang sudah dibuat dan dilakukan pengujian ini dilakukan untuk mengetahui apakah *software* yang dibuat telah sesuai dengan desainnya dan masih terdapat kesalahan atau tidak.

5. Pemeliharaan Sistem

Pada perangkat lunak yang sudah jadi, semuanya dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada tahap sebelumnya. Perbaikan implementasi *unit* sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

2.2.14. 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 pemrograman. *Black Box Testing* berfokus pada 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 *interface*(antarmuka).
3. Kesalahan dalam struktur data atau akses *database* eksternal.
4. Kesalahan kinerja.
5. Kesalahan inisialisasi dan pemutusan kesalahan