

BABII

LANDASAN TEORI

2.1. Tinjauan Pustaka

Untuk mendapatkan kesempurnaan dalam penyusunan sebuah skripsi diperlukan suatu tinjauan pustaka. Dengan tinjauan pustaka diharapkan mampu memberikan kontribusi yang besar terhadap penyusunan sebuah skripsi. Dengan demikian dapat dikatakan bahwa semakin banyak tinjauan pustaka yang dibutuhkan semakin mendekati sempurna pula penyusunan sebuah skripsi. Isi dari tinjauan pustaka dapat berupa dasar-dasar teori yang berhubungan dengan skripsi yang akan dibahas, dimana dasar teori ini digunakan sebagai acuan awal dalam penyusunan skripsi. Berikut adalah beberapa teori yang berhubungan dengan skripsi yang akan dibahas.

2.1.1 Pengertian Sistem

Menurut Kadir (2014:61), “sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan”. Sebagai gambaran jika dalam sebuah sistem terdapat sebuah elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem.

Ada 3 elemen yang membentuk sebuah sistem yaitu :

1. Masukan

Segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk di proses.

2. Proses

Bagian yang melakukan perubahan dari *input* menjadi *output* yang berguna, misalnya berupa informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna, misalnya sisa pembuangan atau limbah.

3. Keluaran

Hasil dari pemrosesan, misalnya berupa suatu informasi, saran, cetakan laporan, dan lain-lain.

Berdasarkan pengertian diatas dapat disimpulkan sistem adalah cara yang kita lakukan untuk mencapai tujuan yang telah kita buat mulai dari memasukan sesuatu memprosesnya kemudian menghasilkan keluaran.

2.1.2 Pengertian Informasi

Informasi merupakan sesuatu yang dihasilkan dari pengolahan data. Data yang sudah ada dikemas dan diolah sedemikian rupa sehingga menjadi sebuah informasi yang berguna. Berikut adalah definisi informasi berdasarkan berbagai sumber.

Informasi merupakan suatu hasil dari pemrosesan data menjadi sesuatu yang bermakna bagi yang menerimanya, sebagaimana dikemukakan oleh Vercellis (2009: 7) *“Information is the outcome of extraction and processing activities carried out on data, and it appears meaningful for those who receive it in a specific domain .”* Selain merupakan hasil dari pengolahan data, informasi juga menggambarkan sebuah kejadian, sebagaimana dikemukakan oleh Wawan dan Munir (2006: 1) bahwa “Informasi merupakan hasil dari pengolahan data dalam suatu bentuk yang menggambarkan suatu kejadian-kejadian (event) yang nyata (fact) dengan lebih berguna dan lebih berarti “.

2.1.3 Pengertian Sistem Informasi

Menurut Darmawan dan Fauzi (2013:13), “sistem informasi adalah kumpulan dari subsistem yang saling berhubungan satu sama lain dan bekerja sama untuk mengelolah data menjadi informasi yang berguna”.

Berdasarkan pengertian diatas, dapat disimpulkan bahwa sistem informasi merupakan cara kita menentukan hal apa saja yang kita perlu untuk mengelolah data yang telah kita peroleh menjadi informasi yang berguna.

2.1.4 E-Commerce

Menurut Sunarto (2009:48) “*E-commerce* merupakan konsep baru yang bisa digambarkan sebagai proses jual beli barang atau jasa pada *world wide web internet* atau proses jual beli atau pertukaran produk, jasa dan informasi termasuk *internet*”.

Ruang lingkup *e-commerce* dapat dibagi menjadi beberapa jenis, antara lain adalah sebagai berikut:

1. *Business to Business (B2B)*

Merupakan sistem komunikasi bisnis antar pelaku bisnis atau transaksi secara elektronik antar perusahaan yang dilakukan secara rutin dan dalam kapasitas produk yang besar.

2. *Bussiness to Consumer (B2C)*

Merupakan sistem bisnis antar pelaku bisnis dengan konsumen untuk memenuhi kebutuhan tertentu pada saat tertentu.

3. *Consumer to Consumer*

Merupakan sistem komunikasi dan transaksi bisnis antar konsumen untuk memenuhi kebutuhan tertentu pada saat tertentu.

E-commerce merupakan bagian dari *e-business*, dimana cakupan *e-business* lebih luas, tidak hanya sekedar perniagaan tetapi mencakup juga pengkolaborasian mitra bisnis, pelayanan pelanggan, lowongan pekerjaan dll. Selain teknologi jaringan *www*, *e-commerce* juga memerlukan teknologi basis data atau pangkalan data (*database*), e-surat atau surat elektronik (*e-mail*) dan bentuk teknologi non komputer yang lain seperti halnya sistem pengiriman barang, dan alat pembayaran untuk *e-commerce* ini.

2.1.5 Konsep Dasar Pemrograman

1. Pengertian Pemrograman Terstruktur

Pemrograman terstruktur adalah konsep atau paradigma atau sudut pandang pemrograman yang membagi-bagi program berdasarkan fungsi-fungsi atau prosedur-prosedur yang dibutuhkan program komputer. Menurut Sukanto dan Shalahuddin (2016: 67).

Fungsi-fungsi dan prosedur-prosedur ditulis secara sekuensial atau terurut dari atas kebawah sesuai dengan kebergantungan antara fungsi atau prosedur. Pemodelan pada pemrograman terstruktur lebih fokus bagaimana memodelkan data dan fungsi-fungsi atau prosedur-prosedur yang harus dibuat. Jenis paradigma pemrograman yang digunakan dapat dideteksi dari bahasa pemrograman apa yang akan digunakan untuk membuat program, baru setelah itu ditentukan paradigma pemrograman apa yang akan digunakan.

2. PHP (*Personal Home Page*)

Menurut Anhar (2010:3) “PHP (*PHP Hypertext Preprocessor*) adalah bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang diintegrasikan dengan HTML dan berada pada *server (server side HTML embeddeds scripting)*”.

Menurut Kadir (2008:2) “PHP merupakan singkatan dari *PHP Hypertext Preprocessor*. Ia merupakan bahasa berbentuk skrip yang ditempatkan dalam server dan diproses di *server*. Hasilnya yang dikirimkan ke *klien*, tempat pemakai menggunakan *browser*”.

Secara khusus, PHP dirancang untuk membentuk aplikasi *web* dinamis. Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya, Anda bisa menampilkan isi *database* ke halaman web. Pada prinsipnya PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), *Cold Fusion*, ataupun *Perl*. Namun, perlu diketahui bahwa PHP sebenarnya bisa dipakai secara *command line*. Artinya, skrip PHP dapat dijalankan tanpa melibatkan *web server* maupun *browser*. Kelahiran PHP bermula saat Rasmus Lerdorf membuat sejumlah skrip *Perl* yang dapat mengamati siapa saja yang melihat-lihat daftar riwayat hidupnya, yakni pada tahun 1994. Skrip-skrip ini selanjutnya dikemas menjadi *tool* yang disebut "*Personal Home Page*". Paket inilah yang menjadi cikal-bakal PHP.

Pada tahun 1995, Rasmus menciptakan PHP/FI Versi 2. Pada versi inilah pemrograman dapat menempelkan kode terstruktur di dalam tag HTML. Yang menarik, kode PHP juga bisa berkomunikasi dengan *database* dan melakukan perhitungan-perhitungan yang kompleks sambil jalan. Pada saat ini PHP cukup

populer sebagai peranti pemrograman web, terutama di lingkungan Linux. Walaupun demikian, PHP sebenarnya juga dapat berfungsi pada *server-server* yang berbasis UNIX, Windows dan Macintosh.

Pada awalnya, PHP dirancang untuk diintegrasikan dengan *web server* Apache. Namun, belakangan PHP juga dapat bekerja dengan *web server* seperti PWS (*Personal Web Server*), IIS (*Internet Information Server*), dan Xitami Untuk mencoba PHP, Anda tidak perlu menggunakan komputer berkelas server. Hanya dengan sebuah komputer biasa, Anda bisa mempelajari dan mempraktikkan PHP.

PHP bersifat bebas dipakai. Anda tidak perlu membayar apa pun untuk menggunakan perangkat lunak ini alias *free*. PHP tersedia dalam bentuk kode biner maupun kode sumber yang lengkap. Bagaimana halnya kalau yang diminta adalah sebuah halaman PHP? Prinsipnya serupa dengan kode HTML. Hanya saja, ketika berkas PHP yang diminta didapatkan oleh *web server*, isinya segera dikirimkan ke mesin PHP dan mesin inilah yang memproses dan memberikan hasilnya (bempa kode HTML) ke *web server*. Selanjutnya, *web server* menyampaikan ke klien.

Satu kelebihan dari PHP adalah mampu berkomunikasi dengan berbagai *database* yang terkenal. Dengan demikian, menampilkan data yang bersifat dinamis, yang diambil dari *database*, merupakan hal yang mudah untuk diimplementasikan. Itulah sebabnya sering dikatakan bahwa PHP sangat cocok untuk membangun halaman-halaman web dinamis. Pada saat ini PHP sudah dapat berkomunikasi dengan berbagai *database* meskipun dengan kelengkapan yang berbeda-beda. Beberapa di antaranya BASE, DBM, FilePro (Personix, Inc),

Informix, Ingres, InterBase, Microsoft Access, MS SQL, MySQL, Oracle, PostgreSQL, Sybase.

3. MySQL

Menurut Anhar (2010:21) “MySQL (*My Structur Query Language*) adalah sebuah perangkat lunak system manajemen basis data SQL (*Database Manajement Sistem*) atau DBMS, dari sekian banyak DBMS seperti Oraecle, MS SQL, Postagre SQL, dan lain-lain”. Sedangkan menurut Kadir (2008:348) “MySQL adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *database-nya*”.

MySQL dimiliki dan disponsori oleh sebuah perusahaan swedia, yaitu MySQL AB, MySQL AB memegang hak cipta kode sumbernya. Kedua orang swedia dan satu orang finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larson dan Michael Monty Widenius. Seperti yang telah disebutkan sebelumnya MySQL bersifat gratis atau *open source* sehingga kita dapat menggunakannya secara gratis.

4. Adobe Dreamweaver CS

Dreamweaver adalah sebuah HTML editor profesional untuk mendesain *web* secara *visual* dan mengelola situs atau halaman *web*. *Dreamweaver* merupakan *software* utama yang digunakan oleh *web desainer* maupun *web programmer* dalam mengembangkan suatu situs *web*, karena *dreamweaver* mempunyai ruang kerja, fasilitas dan kemampuan yang mampu meningkatkan produktivitas dan efektivitas dalam desain maupun membangun suatu situs *web* (Madcoms, 2011:13). Ada beberapa kelebihan yang dimiliki *Macromedia*

Dreamweaver diantaranya mampu membuat halaman *web* terlihat konsisten, mudah dan efisiensi dalam penggunaan, mudah untuk mengupload melalui FTP dan bisa dikustom. Selain itu *dreamweaver* memiliki kemampuan memperlihatkan tiga proses yang berbeda seperti *code view* berfungsi untuk hanya menampilkan *script html* saja, *desain view* berfungsi menampilkan kode-kode *html* yang ditulis menjadi sebuah *design/template* yang nantinya akan ditampilkan di *browser*, *split view* berfungsi menampilkan gabungan antara *code view* dan *desain view* pada saat bersamaan sehingga bisa langsung melihat perubahan pada saat mengubah *html*nya. Sedangkan kekurangan yang paling signifikan dari aplikasi *Macromedia Dreamweaver* adalah harga yang terlalu mahal untuk software aslinya.

2.1.6. UML (*Unified Modeling Language*)

Menurut Nugroho (2010:6) “UML (*Unified Modelling Language*) adalah bahasa pemodelan sistem atau perangkat lunak yang berparadigma berorientasi objek”. Pengembangan UML dimulai dari kerjasama Grady Booch dan James Rumbaugh pada tahun 1994 untuk mengkombinasikan 2 metologi terkenal Booch dan OMT (*Object Management Group*). Notasi UML terutama diurutkan dari 3 notasi yang telah ada sebelumnya, yakni Grady Booch OOD (*Object Oriented Design*), Jim Rumbough OMT (*Object Modelling Technique*) dan Ivar jacobson OOSE (*Object Oriented Software Engineering*).

UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standardisasi agar orang di berbagai negara dapat

mengerti pemodelan perangkat lunak. Seperti yang kita ketahui bahwa menyatukan banyak kepala untuk menceritakan sebuah ide dengan tujuan untuk memahami hal yang sama tidaklah mudah, oleh karena itu diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang.

Beberapa literatur menyebutkan bahwa UML menyediakan lima jenis diagram, karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu *statis* atau *dinamis*.

Jenis diagram itu antara lain:

a. *Use Case Diagram*

Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

b. *Activity Diagram*

Diagram aktivitas (*Activity Diagram*) adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.

c. *Deployment Diagram*

Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. Diagram *deployment* berhubungan erat dengan diagram komponen dimana

diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*).

d. *Component Diagram*

Diagram komponen ini memperlihatkan organisasi serta ketergantungan sistem atau perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan ke dalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.

2.1.7. ERD (*Entity Relationship Diagram*)

Menurut Nugroho (2011:57) “Model E-R (ERD) adalah perincian yang merupakan organisasi atau area tertentu”.ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD.

Komponen yang terdapat dalam *Entity Relationship Diagram* menurut Nugroho (2011:57) adalah sebagai berikut:

1. Entity (Entitas)

Pada ER diagram, entity digambarkan dengan sebuah bentuk persegi panjang. Entity adalah sesuatu apa saja yang ada didalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama dengan kata benda dan dapat dikelompokkan dalam empat jenis nama yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu didalamnya).

2. *Relationship*

Pada ER diagram, Relationship dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung (*Relationship*) diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa dengan kalimat aktif atau kalimat pasif). Penggambaran hubungan yang terjadi adalah sebuah bentuk belah ketupat dihubungkan dengan dua bentuk empat persegi panjang.

3. *Relationship Degree*

Relationship degree atau derajat *relationship* adalah jumlah entitas yang berpartisipasi dalam satu relationship. Derajat relationship yang sering dipakai didalam ERD:

a. *Unary Relationship*

Unary relationship adalah model *relationship* yang terjadi diantara *entity* yang berasal dari *entity set* yang sama. Sering juga disebut sebagai *Recursive Relationship* atau *Reflective Relationship*.

b. *Binary Relationship*

Binary Relationship adalah model *relationship* antara *instance* dari suatu tipe entitas (dua *entity* yang berasal dari *entity* yang sama). *Relationship* ini paling umum digunakan dalam pembuatan model data.

c. *Ternary Relationship*

Ternary Relationship adalah merupakan antara *instance* dari tiga tipe entitas secara serentak.

d. *Attribute Value*

Attribute value atau nilai *attribute* adalah suatu occurrence tertentu dari sebuah attribute didalam suatu *entity* atau *relationship*. Ada dua jenis atribut:

- 1) Identifier (key) digunakan untuk menentukan suatu entity secara unik (primary key).
- 2) Descriptor (*nonkey attribute*) digunakan untuk menspesifikasikan karakteristik dari suatu entity yang tidak unik.

e. Kardinalitas (*Cardinality*)

Kardinalitas relasi menunjukkan jumlah maksimum tupelo yang dapat berelasi dengan entitas pada entitas yang lain. Terdapat tiga macam kardinalitas relasi yaitu:

1) *One to One*

Tingkat hubungan satu kesatu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya yang berarti setiap tupel pada entitas A berhubungan dengan paling banyak satu tupel pada entitas B, dan begitu juga sebaliknya setiap tupel pada entitas B berhubungan dengan paling banyak satu tupel pada entitas A.

2) *One to Many atau Many to One*

Tingkat hubungan satu banyak adalah sama dengan banyak kesatu. Tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya satu

kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

3) *One to Many* (satu banyak)

Yang berarti satu tupel pada entitas A dapat berhubungan dengan banyak tupel pada entitas B, tetapi tidak sebaliknya, dimana setiap tupel pada entitas B berhubungan dengan paling banyak satu tupel pada entitas A.

4) *Many to One* (banyak kesatu)

Yang berarti setiap tupel pada entitas A dapat berhubungan dengan paling banyak satu tupel pada entitas B, tetapi tidak sebaliknya, dimana setiap tupel pada entitas A berhubungan dengan paling banyak satu tupel pada entitas B.

5) *Many to Many*

Tingkat hubungan kebanyakan terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya. Baik dilihat dari sisi entitas yang pertama, maupun dilihat dari sisi yang kedua yang berarti setiap tupel pada entitas A dapat berhubungan dengan banyak tupel pada entitas B. Dimana setiap tupel pada entitas B dapat berhubungan dengan banyak tupel pada entitas A.

2.1.8. LRS (*Logical Record Structure*)

Menurut Frieyadie (2007:13), *Logical Record Structure* sebelum tabel dibentuk dari field atau atribut entitas secara fisik atau level internal, maka harus dibuatkan suatu bentuk relational model yang dibuat secara *logic* atau level

external dan konsep, dari pernyataan tersebut dibutuhkan yang disebut dengan *Logical RecordStructure* (LRS).

Dalam pembuatan LRS terdapat tiga hal yang dapat mempengaruhi, yaitu:

1. Jika tingkat hubungan (*cardinality*) satu pada satu (*one-to-one*), maka digabungkan dengan entitas yang lebih kuat (*strong entity*), atau digabungkan dengan entitas yang memiliki atribut yang lebih sedikit.
2. Jika tingkat hubungan (*cardinality*) satu pada banyak (*one-to-many*), maka hubungan relasi atau digabungkan dengan entitas yang tingkat hubungannya banyak.
3. Jika tingkat hubungan (*cardinality*) banyak pada banyak (*many-to-many*), maka hubungan relasi tidak akan digabungkan dengan entitas manapun, melainkan menjadi sebuah LRS.

2.1.9. Model Pengembangan Sistem (*Waterfall Model*)

Model air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung *support* (Sukamto dan Shalahuddin, 2016:26).

1. Analisis kebutuhan perangkat lunak

Agar dapat memahami spesifikasi kebutuhan perangkat lunak seperti apa yang dibutuhkan oleh *user*, maka proses pengumpulan data kebutuhan perangkat lunak perlu dilakukan secara intensif. Tahap analisis kebutuhan perangkat lunak ini perlu untuk didokumentasikan.

2. Desain

Tahap ini membuat desain aliran kerja manajemen dan desain pemrograman yang diperlukan untuk pengembangan sistem informasi serta memberikan gambaran yang jelas tentang bagaimana suatu sistem dibentuk. Proses desain mengubah kebutuhan-kebutuhan menjadi bentuk karakteristik yang dimengerti perangkat lunak sebelum dimulai penulisan program. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan kode program

Pada tahap ini, desain yang sudah dibuat harus ditranslasikan ke dalam program perangkat lunak sehingga menghasilkan program komputer yang sesuai dengan desain.

4. Pengujian

Pengujian difokuskan pada logika internal dari perangkat lunak, fungsi eksternal, dan mencari segala kemungkinan kesalahan, dan memeriksa apakah sesuai dengan hasil yang diinginkan.

5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Setelah perangkat lunak diberikan kepada pelanggan, kemungkinan masih dapat mengalami perubahan. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi pada saat pengujian, atau mungkin pelanggan meminta penambahan fungsi untuk beradaptasi dengan lingkungan baru. Tahap ini tidak untuk membuat perangkat lunak baru, tetapi dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada.

Kenyataannya, model air terjun (*waterfall*) sangat jarang dapat dilakukan sesuai alurnya karena beberapa sebab. Berikut menurut Sukamto dan Shalahuddin (2016:28):

1. Perubahan spesifikasi perangkat lunak terjadi di tengah alur pengembangan.
2. Sangat sulit bagi pelanggan untuk mendefinisikan semua spesifikasi di awal alur pengembangan. Pelanggan sering kali butuh contoh (*prototype*) untuk menjabarkan spesifikasi kebutuhan sistem lebih lanjut.
3. Pelanggan tidak mungkin bersabar mengakomodasi perubahan yang diperlukan di akhir alur pengembangan.

Model air terjun (*waterfall*) ini telah menjadi dasar dari model-model yang lain dalam melakukan perbaikan-perbaikan model pengembangan perangkat lunak, walaupun ada berbagai kelemahan yang dimiliki model air terjun (*waterfall*) ini. Model air terjun (*waterfall*) ini sangat cocok digunakan untuk kebutuhan pelanggan yang sudah sangat memahami dan kemungkinan terjadi perubahan kebutuhan selama proses pengembangan perangkat lunak kecil. Hal positif dari model air terjun (*waterfall*) adalah struktur tahap pengembangan sistem sangat jelas, dokumentasi dihasilkan di setiap tahap pengembangan, dan sebuah tahap dijalankan setelah tahap sebelumnya selesai dijalankan, sehingga tidak ada tumpang tindih pelaksanaan tahap.

2.2 Penelitian Terkait

Menurut Handayani dan Purnama (2013:18) dalam Jurnalnya yang berjudul Pembuatan Website E-Commerce Pada Distro Java Trend dijelaskan bahwa Distro Java Trend mempunyai beberapa kekurangan dalam rangka menunjang proses penjualan barang. Penjual yang saat ini dilakukan hanya sebatas konsumen datang memilih barang dan membayar transaksi pemesanan juga dilakukan melalui BBM dan media situs pertemanan Facebook. Dengan mengandalkan sistem penjualan dari media tersebut, dirasa masih kurang dalam memperbanyak tingkat penjualan barang sedangkan dengan menggunakan penjualan barang menggunakan sistem online, *customer* dapat melakukan transaksi penjualan tanpa adanya batasan waktu dan batasan tempat. Selain itu dengan menggunakan penjualan online maka data-data yang ada dibuat secara terstruktur dan terkomputerisasi untuk memudahkan dalam mengelola data member maupun laporan transaksi. Tujuan dari proyek ini adalah membangun suatu Website Penjualan pada Distro Java Trend yang memiliki fungsi antara lain. Membangun aplikasi *e-commerce* untuk membantu manajemen Distro java Trend dalam melakukan transaksi jarak jauh, serta membantu konsumen dalam melakukan pembelian. Pembayaran dapat melalui paypal dan transfer rekening. Bagi produsen mempermudah dalam membuat laporan pemesanan, pembayaran untuk rekapitulasi Distro Java Trend perbulannya sesuai dengan tahun yang diinginkan secara komputerisasi.

Menurut Haryanti dan Irianto (2011:8) dalam Jurnalnya yang berjudul Rancang Bangun Sistem Informasi E-Commerce Untuk Usaha Fashion Studi Kasus Omah Mode Kudus dijelaskan bahwa Omah Mode kudus salah satu toko di kota Kudus yang bergerak di bidang fashion, seperti : baju, celana, topi, jaket, kaos, sweater dan lainnya. Namun, dengan sistem yang berjalan sekarang, masih banyak masalah yang terjadi baik di pihak vendor maupun di pihak konsumen terutama yang berada diluar kota Kudus. Pihak vendor selama ini belum memiliki media khusus untuk mempromosikan produknya. Dan konsumen masih harus langsung datang ke toko apabila ingin mendapatkan informasi produk dan hendak memesan produk sesuai dengan yang diinginkan, serta harus menghubungi vendor via sms / telepon jika ingin mengetahui perkembangan stok, koleksi terbaru. Berdasarkan latar belakang yang telah dipaparkan, penelitian ini akan merancang suatu sistem *e-commerce* yaitu sistem informasi penjualan produk pakaian berbasis web. Sistem *e-commerce* ini akan menampilkan semua persediaan dari berbagai jenis produk yang ditawarkan lengkap dengan spesifikasi dan harga produk. Sehingga penjual dan pembeli dapat mengakses ketersediaan produk serta dapat bertransaksi secara online kapan saja dan dimana saja.