

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Tinjauan Pustaka**

##### **A. Konsep Dasar Sistem Informasi**

Menurut Pressman (2010:155) sistem adalah suatu kumpulan elemen-elemen yang diorganisir untuk mencapai tujuan tertentu dengan memproses informasi yang ada.

Menurut Turban (2010:415), informasi adalah data yang telah diatur sedemikian rupa, sehingga memiliki arti dan nilai bagi orang yang menerimanya.

Menurut Turban (2010:415), sistem informasi adalah sebuah proses untuk mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk sebuah tujuan yang spesifik, dan kebanyakan sistem informasi dikomputerisasi.

Metode waterfall sering dinamakan 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 ke para pelanggan/pengguna (deployment), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan menurut Pressman (2012).

## **B. *E-Commerce***

*E-Commerce* merupakan sistem perdagangan dengan mempergunakan *web* secara *Online*, mempunyai banyak definisi yang dianut oleh para ahli, yaitu: Menurut McLeod Pearson (2008: 59) menyatakan bahwa Perdagangan elektronik atau yang disebut juga *E-Commerce* adalah penggunaan jaringan komunikasi dan komputer untuk melakukan proses bisnis. Pandangan populer dari *E-Commerce* adalah penggunaan internet dan komputer dengan browser untuk pembelian dan penjualan produk.

Berdasarkan karakteristiknya *E-Commerce* dapat dibedakan menjadi beberapa jenis, yaitu:

### **a. *Business to Business* (B2B)**

Jenis ini mempunyai karakteristik berupa trading partners yang sudah saling mengetahui dan antara mereka sudah terjalin hubungan yang berlangsung cukup lama. Informasi yang dimiliki hanya ditukar dengan partner tersebut. Pertukaran data dilakukan berulang-ulang dan berkala dengan format data yang sudah disepakati bersama. Salah satu pelaku tidak harus menunggu rekan mereka untuk mengirim data. Model yang umum digunakan adalah *peer to peer* dimana processing intelligence dapat didistribusikan di kedua pelaku bisnis.

### **b. *Business to Customer* (B2C)**

Jenis *E-Commerce* ini memiliki karakteristik yang terbuka untuk umum, dimana informasi disebarluaskan secara umum pula dan dapat diakses secara bebas. Pelayanan yang digunakan bersifat umum sehingga dapat digunakan oleh orang

banyak, contoh: dikarenakan sistem berbasis *web* sudah umum digunakan. Pelayanan berdasarkan permintaan, produsen harus siap memberikan respon sesuai dengan permintaan konsumen, sering dilakukan sistem pendekatan *Client Server*.

c. *Customer to Customer (C2C)*

Jenis *E-Commerce* ini memiliki karakteristik penjualan dapat dilakukan seorang secara langsung ke pada konsumen lainnya baik berupa produk dan jasa. Contoh: seorang konsumen menjual barang dan diiklankan dalam *web* tersebut, dan dibeli oleh konsumen lain yang turut serta mengaksesnya.

d. *Customer to Business (C2B)*

Karakteristik yang dimiliki oleh jenis *E-Commerce* ini adalah model bisnis dimana konsumen menciptakan nilai dan perusahaan mengkonsumsi nilai tersebut. Contoh: ketika konsumen menulis *review* maka konsumen sebagai individu memberikan dan menciptakan nilai bagi perusahaan.

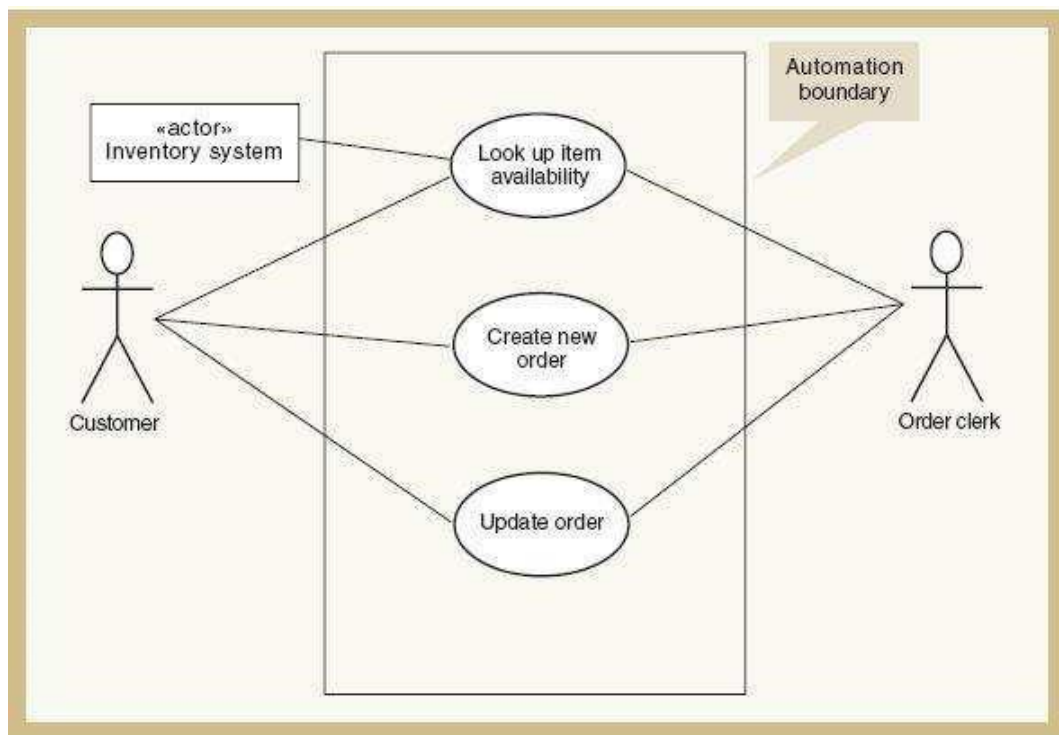
### **C. *Unified Modeling Language (UML)***

Dalam buku *System Analysis and Design In a Changing World sixth edition*, Satzinger, et al (2012:46) mengartikan bahwa “*UML is the standard set of model constructs and notations defined by the Object Management Group (OMG), a standards organization for system development*”. UML adalah suatu set standar konstruksi model dan notasi yang ditetapkan oleh *Object Management Group (OMG)*, suatu standar organisasi untuk pengembangan sistem

## 1. Use Case

*Use case* menurut Satzinger et al., (2010, p.242) merupakan suatu aktivitas yang dilakukan oleh sistem, biasanya merupakan sebuah *respon* untuk permintaan dari pengguna sistem.

Satzinger et al., (2010, p.243) menjelaskan bahwa aktor tidak selalu sama dengan sumber dari peristiwa di *event table* karena aktor di *use case* merupakan orang yang berinteraksi dengan sistem yang mana sistem harus meresponnya.



Sumber : Satzinger et al., (2010, p.243)

Gambar II.1

Notasi Use Case diagram

## 2. Activity Diagram

Menurut Satzinger et al. (2010, p.141) *Activity diagram* merupakan sebuah tipe dari diagram *workflow* yang menggambarkan tentang aktivitas dari pengguna ketika melakukan setiap kegiatan dan aliran sekuensial.

### a. Activity Node

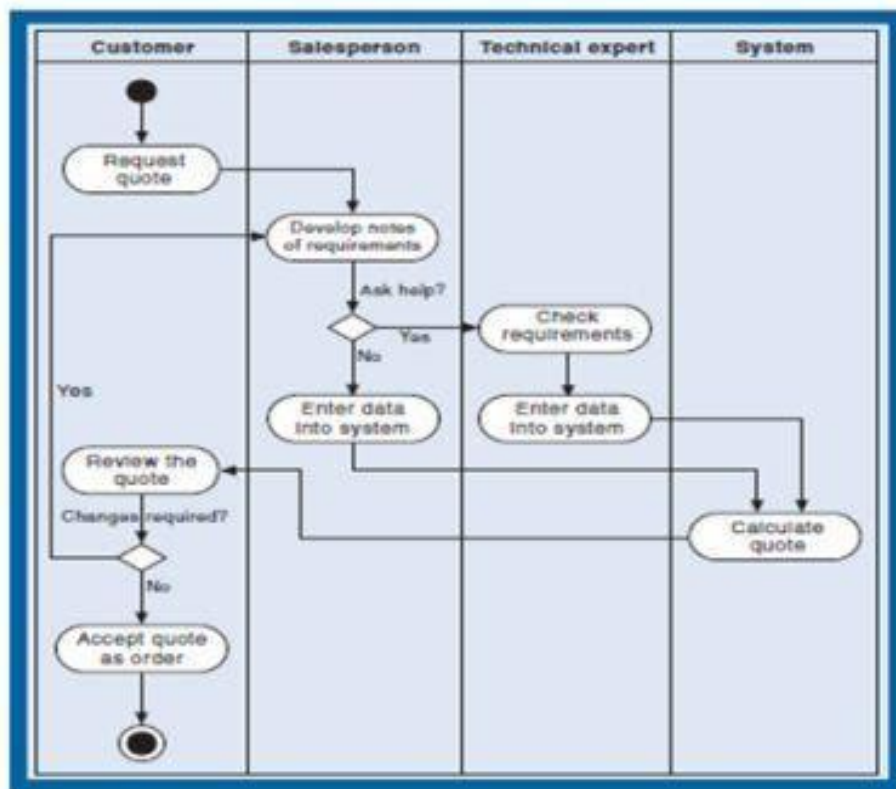
*Activity node* adalah sesuatu yang dilakukan atau yang sedang terjadi dalam *activity diagram*. *Activity node* mempunyai simbol yang hampir mirip dengan *use case* namun mempunyai bentuk yang lebih ramping dan menyerupai bujur sangkar.

### b. Decision Node

*Decision node* disebut *decision diamonds* di dalam *flowchart*. Simbol *diamond* adalah satu elemen yang membuat *activity diagram* mengingatkan *flowchart*, yang berguna untuk memberi kondisi percabangan.

### c. Transition Fork

Keberadaan sebuah *transition fork* untuk menggambarkan tingkah laku yang paralel atau bercabang. Sebuah *transition join* untuk mempertemukan tingkah laku yang paralel. *Activity Diagram* bisa dilihat pada Gambar berikut.



Sumber: *Satzinger et al, 2010, p143*)

Gambar II.2.

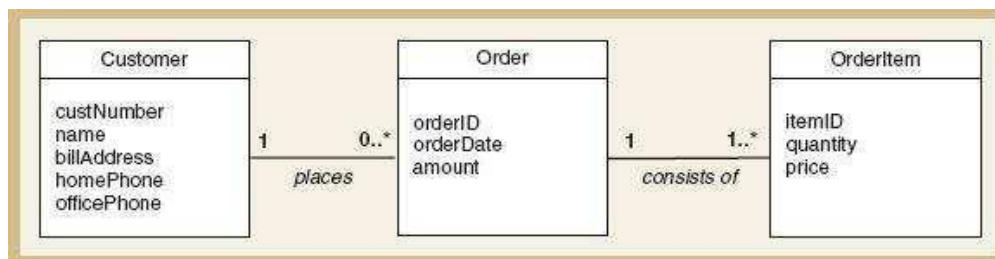
Notasi Activity diagram

### 3. Class Diagram

Menurut Satzinger et al. (2010, p.168) *Class diagram* digunakan untuk menunjukkan objek *class* untuk sistem. Notasinya dari *Unified Modelling Language*(UML), yang telah menjadi standar untuk model yang digunakan dengan pengembangan *system object oriented*.

Salah satu jenis *class diagram* UML menunjukkan hal-hal dalam pekerjaan *domain customer* disebut sebagai *domain model class diagram*. Tipe lain dari notasi *class diagram* UML digunakan untuk membuat desain

*class diagram* ketika merancang *class* perangkat lunak. Di *class diagram*, persegi panjang mewakili kelas, dan garis yang menghubungkan persegi panjang menunjukkan asosiasi antara kelas. Dalam persegi panjang (kotak) terbagi dua, bagian atas berisi nama kelas, dan bagian bawah merupakan atribut kelas. Nama kelas selalu diawali dengan huruf kapital, dan atribut nama selalu diawali dengan huruf kecil. Diagram class digambarkan dengan menampilkan kelas dan asosiasi antara kelas.



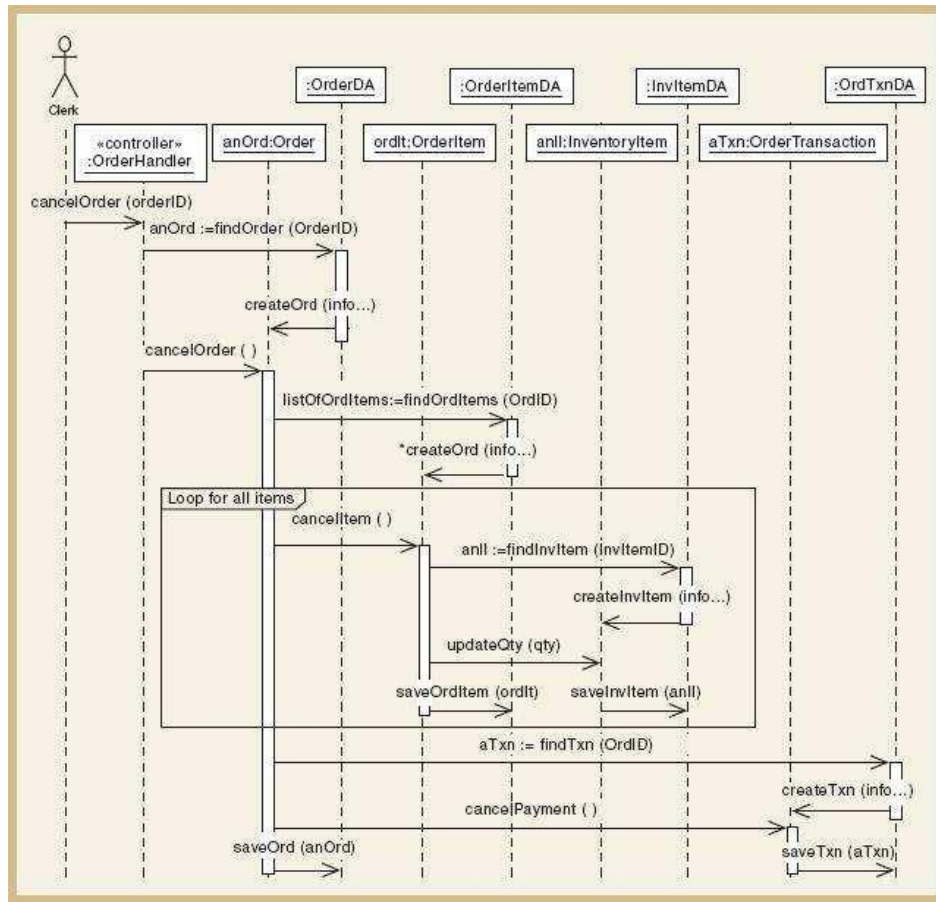
Sumber: *Satzinger et al. (2010, p.168)*

Gambar II.3

### Class Diagram

#### 4. Sequence diagram

Menurut *Satzinger et al. (2010, p.435)* *Sequence Diagram* memiliki tujuan utama untuk mengidentifikasi kolaborasi kelas dan apakah kelas tersebut harus mengirim pesan antara satu sama lain.



Sumber: Whitten (2007:394)

**Gambar II.4**

*Sequence diagram*

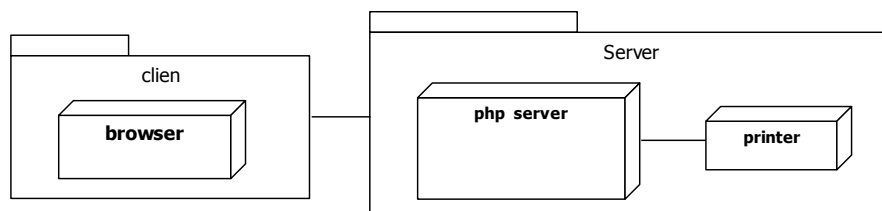
## 5. Component Diagram

Menurut Shalahuddin dan Rosa (2013:148) “Diagram komponen atau component diagram dibuat untuk menunjukkan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem”. Diagram komponen focus pada komponen sistem yang dibutuhkan dan ada di dalam sistem.

## 6. Deployment Diagram

Menurut Shalahddin dan Rosa (2013:154) “Diagram deployment atau deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi”. Diagram deployment juga digunakan untuk memodelkan hal-hal berikut:

1. Sistem tambahan (embedded system) yang menggambarkan rancangan device, node dan hardware
2. Sistem client server misalnya seperti gambar berikut:



Sumber :Shalahuddin dan Rosa (2013:154)

**Gambar II.5**

*Deployment diagram*

### **D. Entity Relationship Diagram (ERD)**

Menurut Sutanta (2011:91) “*Entity Relationship Diagram (ERD)* merupakan suatu model data yang dikembangkan berdasarkan objek.” *Entity Relationship Diagram (ERD)* digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logis. *Entity Relationship Diagram (ERD)* didasarkan pada suatu persepsi bahwa *real world* terdiri atas obyek-obyek dasar tersebut. Penggunaan *Entity Relationship Diagram (ERD)* relatif mudah dipahami, bahkan oleh para pengguna yang awam. Bagi perancang atau analis sistem, *Entity*

*Relationship Diagram* (ERD) berguna untuk memodelkan sistem yang nantinya, basis data akan di kembangkan. Model ini juga membantu perancang atau analis sistem pada saat melakukan analis dan perancangan basis data karena model ini dapat menunjukkan macam data yang dibutuhkan dan kerelasian antardata didalamnya. Komponen *Entity Relationship Diagram* menurut Sutanta (2011:91) adalah sebagai berikut :

Entitas Entitas merupakan suatu objek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data. Objek dasar dapat berupa orang, benda, atau hal yang keterangannya perlu disimpan didalam basis data. Untuk menggambarkan sebuah entitas digunakan aturan sebagai berikut :

- 1) Entitas dinyatakan dengan simbol persegi panjang.
- 2) Nama entitas dituliskan didalam simbol persegi panjang.
- 3) Nama entitas berupa kata benda, tunggal.
- 4) Nama entitas sedapat mungkin menggunakan nama yang mudah dipahami

dan dapat menyatakan maknanya dengan jelas. Atribut Atribut merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan dalam basis data. Atribut berfungsi sebagai penjelas pada sebuah entitas. Untuk menggambarkan atribut digunakan aturan sebagai berikut:

- 1) Atribut digambarkan dengan simbol *ellips*.
- 2) Nama atribut dituliskan didalam simbol *ellips*.
- 3) Nama atribut merupakan kata benda, tunggal.
- 4) Nama atribut sedapat mungkin menggunakan nama yang mudah dipahami

dan dapat menyatakan maknanya dengan jelas. Relasi Relasi merupakan hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Aturan

penggambaran relasi adalah sebagai berikut:

- 1) Relasi dinyatakan dengan simbol belah ketupat.
- 2) Nama relasi dituliskan didalam simbol belah ketupat
- 3) Nama relasi berupa kata kerja aktif.
- 4) Nama relasi sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

### **E. Logical Record Structure (LRS)**

Menurut Simarmata (2007:115) “*Logical Record Structure* adalah representasi dari struktur *record-record* pada table-tabel yang terbentuk dan hasil relasi antara himpunan entitas”. Beberapa tipe *record* digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik. Beda LRS dengan diagram *entity relationship* nama tipe *record* berbeda di luar kotak *field* tipe *record* ditempatkan. LRS terdiri dari *link-link* di antara *tipe record*. *Link* ini menunjukkan arah dan satu *tipe record* lainnya. Banyak *link* dari LRS yang diberi tanda *field-field* yang kelihatan pada kedua *link* tipe *record*. Penggambaran LRS mulai dengan menggunakan model yang dimengerti. Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonversikan ke LRS. Metode yang lain dimulai dengan *entity relationship diagram* dan langsung dikonversikan ke LRS.

A. Konversi ERD ke LRS. Diagram *entity relationship* diagram harus ke bentuk LRS (struktur record secara logika). Dan bentuk LRS inilah yang nantinya dapat ditransformasikan ke bentuk relasi (tabel).

B. Konversi ERD ke LRS sebuah model sistem yang digambarkan dengan sebuah

ERD akan mengikuti pemodelan tertentu. Dalam kaitannya 16 dengan konversi ke LRS, untuk perubahan yang terjadi adalah mengikuti aturan-aturan berikut:

1. Setiap Entitas diubah ke bentuk kotak dengan nama entitas, berbeda di luar kotak dan atribut berada di dalam kotak.
2. Sebuah relationship kadang disatukan, dalam sebuah kotak bernama entitas, kadang sebuah kotak bersama-sama dengan entitas, kadang disatukan dalam sebuah kotak tersendiri.

C. Konversi LRS ke relasi (tabel) relasi atau tabel adalah bentuk pernyataan data secara grafis 2(dua) dimensi, yang terdiri dari kolom dan baris. Relasi adalah bentuk *visual* dari sebuah *file*, dan tiap *tuple* dalam sebuah *field* atau dalam bentuk lingkaran diagram *entity relationship* dikenal dengan sebutan atribut. Konversi dan *logical record structure*, dikenal dengan cara:

1. Nama logical record structure menjadi relasi.
2. Tiap atribut menjadi sebuah kolom di dalam relasi.

## **F. BAHASA PEMROGRAMAN PHP (*Hypertext Preprocessor*)**

### **1. PHP (*Hypertext Preprocessor*)**

Menurut Anhar (2010:3), “PHP singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat open source”. PHP merupakan script yang terintegrasi dengan HTML dan berada pada server. PHP adalah script yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis yang dimaksud adalah halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru.

Berdasarkan pendapat para ahli yang dikemukakan di atas dapat ditarik kesimpulan bahwa PHP adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML.

## **2. SQL (*Structured Query Language*)**

Menurut Connolly & Begg (2010:184), “SQL (*Structured Query Language*) is an example of transform-oriented language, or language designed to use relations to transform inputs into required outputs”. SQL (*Structured Query Language*) adalah contoh dari *transform-oriented language* atau bahasa yang didesain dengan menggunakan relasi untuk mengubah input menjadi *output* yang diinginkan.

Jadi dapat disimpulkan bahwa SQL (*Structured Query Language*) adalah suatu bahasa yang digunakan untuk mengakses data relasional guna mengubah input menjadi output yang diinginkan.

## **G. Adobe Dreamweaver CS5**

Menurut Puspitosari (2010:1) “*Dreamweaver* merupakan sebuah HTML editor professional untuk mendesain *web* secara visual dan mengelola situasi atau halaman *web*”.

Dalam perkembangannya saat ini *dreamweaver* banyak digunakan para *desainer web* maupun pemrograman *web*. Dengan kemampuan *fasilitas* yang optimal dalam jendela *design*, membuat program ini memberikan kemudahan untuk mendesain *web* untuk para *desainer web* pemula sekalipun. Sedangkan kemampuan *dreamweaver* untuk berinteraksi dengan beberapa bahasa

pemrograman seperti PHP, ASP, *JavaScript*, dan lainnya juga memberikan fasilitas maksimal kepada para *desainer web* yang menyertakan bahasa pemrograman web didalamnya.

*Adobe dreamweaver CS4* merupakan versi terbaru dari *Adobe Dreamweaver* yang sebelumnya adalah *Dreamweaver CS3*. Aplikasi *Adobe dreamweaver CS4* memberikan tampilan yang lebih baik dan semakin mudah dalam penggunaannya. Aplikasi ini mengintegrasikan banyak aspek dalam pengembangan *website*, termasuk pembuatan halaman *web* serta pengelolaan *website*.

*Adobe Dreamweaver CS4* menyertakan banyak perangkat yang berkaitan dengan pengkodean dan fitur seperti *HTML*, *CSS*, *XML*, dan pemrograman *client side* seperti *javascript* dengan penggunaan yang sangat mudah. *Adobe dreamweaver CS4* juga menggunakan pemrograman *Script Server Side* seperti *PHP*, *Active Server Page (ASP)*, *ASP.Net*, *ASP Javascript*, *ASP Vbscript*, *Coldfusion*, dan *Java Server page (JSP)*.

*Adobe Dreamweaver CS4* juga dilengkapi dengan berbagai fasilitas yang digunakan untuk melakukan semua kegiatan *editing* halaman *web*. Adapun fasilitas adalah sebagai berikut :

a. *Menu Bar*

Berisi menu Pulldown seperti : *File*, *Edit*, *View*, *Insert*, *Modify*, *Text*, *Command*, *Site*, *Window*, dan *Help*.

b. *Document Toolbar*

Berisi tombol-tombol yang digunakan untuk mengatur layar kerja dokumen pada dreaanweaver dengan cepat, diantaranya *show code view*, *show code and design view*, dan *show design view*.

c. *Coding Window*

Berisi kode-kode HTML dan tempat untuk menuliskan kode-kode pemrograman, misalnya PHP atau ASP.

d. *Panel Groups*

Merupakan salah satu fasilitas yang terdapat pada *Adobe Dreamweaver CS4* yang berfungsi untuk membantu dalam mengedit suatu halaman *web* . Panel group terletak disebelah kanan layar kerja, dan terdiri dari beberapa panel, dimana masing – masing panel tersebut memiliki fungsi yang berbeda – beda.

e. *Insert Panel*

Pada *Dreamweaver CS4*, *Insert Panel* atau yang dikenal dengan *insert toolbar* digabungkan dalam panel groups, berbeda pada vesi sebelumnya yang diletakkan secara terpisah. *Insert Panel* berfungsi untuk menyisipkan objek – objek ke dalam halaman *web*.

f. *Tab Common*

Pada tab ini terdapat tool yang dapat digunakan untuk membuat dan menyisipkan objek ke dalam halaman *web*, seperti membuat *hyperlink*, tabel, gambar, media, dan lain sebagainya.

g. *Ruler*

Berfungsi untuk Mempermudah ukuran dalam mendesain atau membuat halaman

#### h. *Web Site Panel*

Digunakan untuk mengatur file-file dan folder-folder yang membentuk situs *Web* anda sehingga mempermudah untuk mencari file dan folder.

#### i. *Document Window*

Menampilkan dokumen dari halaman *web* yang aktif, atau sedang diedit, Pada *Document Window* terdapat *Title* yang menampilkan judul halaman, fasilitas *zoom*, *Document Toolbar*, dan *tag selector*



Sumber: Puspitasari (2010)

**Gambar II.6**

*Dreamweaver*

## H. Pengertian Notepad++

Notepad++ adalah sebuah teks editor yang sangat berguna bagi setiap orang dan khususnya bagi para developer dalam membuat program. Notepad++ menggunakan komponen Scintilla untuk dapat menampilkan dan menyunting teks dan berkas kode sumber berbagai bahasa pemrograman yang berjalan diatas sistem operasi Microsoft Windows. Selain manfaat dan kemampuannya menangani banyak bahasa pemrograman, Notepad ++ juga dilisensikan sebagai perangkat free. Jadi, setiap orang yang menggunakannya tidak perlu mengeluarkan biaya untuk membeli aplikasi ini karena sourceforge.net sebagai layanan yang memfasilitasi Notepad ++ membebaskannya untuk digunakan.

## I. Pengertian Basis Data

Menurut Connolly dan Begg (2010:15), *“A shared collection of logically related data and its description, designed to meet the information needs of an organization”*. Basis data adalah kumpulan data yang terbagi dan terhubung secara logikal dan deskripsi dari data yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi.

## J. Database Management System (DBMS)

Menurut Connolly dan Begg (2010: 17), *Database Management System* (DBMS) adalah suatu sistem perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, memelihara, dan mengatur akses ke basis data. Biasanya DBMS memiliki fasilitas-fasilitas sebagai berikut:

1. Fasilitas mendefinisikan basis data, biasanya menggunakan *Data Definition Language* (DDL). DDL mengizinkan pengguna untuk memspesifikasikan tipe,

struktur, dan batasan aturan mengenai data yang bisa disimpan ke dalam basis data.

2. Fasilitas untuk mengizinkan pengguna untuk menambah, mengedit, menghapus dan mendapatkan kembali data dari basis data, biasanya menggunakan *Data Manipulation Language* (DML). Ada pula suatu fasilitas yang melayani pengaksesan data yang disebut *query language*. Bahasa yang diakui adalah *Structured Query Language* (SQL), yang merupakan standard dari DBMS.
3. Fasilitas untuk mengontrol ke basis data (DCL). Contoh :
  - a. Sistem keamanan yang mencegah *customer* yang tidak punya otoritas untuk akses data.
  - b. Suatu sistem terintegrasi yang memelihara konsistensi penyimpanan data.
  - c. Suatu Sistem kontrol pengembalian data yang mana dapat mengembalikan data ke keadaan sebenarnya apabila terjadi kegagalan perangkat keras atau perangkat lunak.
  - d. Terdapat suatu katalog yang dapat di akses oleh pengguna, yang menjelaskan data didalam basis data tersebut.

Menurut Connolly dan Begg (2010:18), Komponen DBMS terbagi menjadi lima yaitu :

1. *Hardware* (perangkat keras)

*Hardware* dapat berkisar dari komputer tunggal, mainframe tunggal, hingga jaringan komputer. *Hardware* yang dipakai tergantung pada kebutuhan organisasi dan *Database Management System* (DBMS) yang digunakan. Sebuah

DBMS memerlukan jumlah minimum memori dan hardisk untuk bekerja, tetapi konfigurasi yang minimum tidak memberikan performa yang handal.

## 2. *Software* (perangkat lunak)

Komponen perangkat lunak terdiri dari perangkat lunak DBMS dan program aplikasi beserta sistem operasi (OS), termasuk jaringan perangkat lunak jika DBMS digunakan melalui jaringan.

## 3. Data

Data merupakan data terpenting dalam DBMS khususnya sudut pandang dari *end customer* mengenai data, dimana data berfungsi sebagai jembatan antara komponen mesin dengan komponen manusia.

## 4. *Procedures*

Prosedur merupakan panduan dan aturan dalam membuat dan menggunakan basis data. Prosedur didalam basis data berupa : login ke dalam basis data, penggunaan fasilitas DBMS atau aplikasi program, cara menjalankan dan menghentikan DBMS, membuat *backup database*, menangani kerusakan hardware atau software, mengubah struktur table, mengumpulkan basis data dari beberapa *disk*, meningkatkan kinerja atau membuat arsip data pada *secondary storage*.

## 5. *People* (manusia)

Komponen terakhir yaitu manusia yang terlibat dengan sistem tersebut.

### **K. Black Box**

Menurut Rizky (2011:264) “*Black box testing* adalah tipe testing yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya”. Sehingga para tester memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat isinya, tapi cukup dikenal proses testing di bagian luar.

### **2.2. Penelitian Terkait**

Pada penelitian yang dilakukan Rara Sri Artati Rejeki dkk (2011) dengan judul perancangan dan pengaplikasian sistem penjualan pada “Distro Smith” berbasis Ecommerce. Software penelitian yang digunakan dalam penelitian ini adalah bahasa pemrograman *PHP*, text editor *Macromedia Dreamweaver 8*, Database *MySQL* dan *Adobe Photoshop CS4*. Tujuan penelitian ini memudahkan konsumen untuk mendapatkan barang yang diinginkan tanpa harus datang langsung ke toko serta mempermudah konsumen dalam melakukan proses transaksi secara *online* serta memudahkan customer dalam mengetahui produk apa saja yang ditawarkan oleh distro smith tanpa harus datang langsung ke toko tersebut.

Menurut Eka Wulansari Fridayanthie (2015) Dalam memberikan peranan sistem yang lebih baik, maka diperlukan sebuah sistem informasi penjualan barang didalam pengelolannya sehingga dapat mengoptimalkan kinerja perusahaan dalam kegiatan penjualan barang.