

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Penulis coba mengemukakan beberapa teori yang diambil dari berbagai buku sebagai landasan teori untuk pembuatan *website* yang dibuat penulis. Teori-teori itu tidak dijelaskan secara keseluruhan, melainkan sebagian yang ada kaitannya serta mendukung dengan *website* yang akan dibuat.

##### **2.1.1 Pengertian *E-Commerce***

Perdagangan Elektronik (*E-Commerce = electronic commerce*) adalah bagian dari *e-lifestyle* yang memungkinkan transaksi jual beli dilakukan secara online dari sudut tempat mana pun (Hidayat, 2008:5).

*E-Commerce* juga dapat diartikan sebagai suatu proses berbisnis dengan menggunakan teknologi elektronik yang menghubungkan antara perusahaan, konsumen dan masyarakat dalam bentuk transaksi elektronik dan pertukaran/penjualan barang, servis, dan informasi secara elektronik (Munawar, 2009:1).

Sedangkan pengertian *E-Commerce* (Perdagangan Elektronik) menurut Jony Wong (2010:33) adalah pembelian, penjualan dan pemasaran barang serta jasa melalui sistem elektronik. Seperti televisi, radio dan jaringan komputer atau internet.

### 2.1.2 Komponen *E-Commerce*

*E-Commerce* memiliki beberapa komponen standar yang dimiliki dan tidak dimiliki transaksi bisnis yang dilakukan secara offline, yaitu (Hidayat, 2008:7):

1. **Produk** : Banyak jenis produk yang bisa dijual melalui internet seperti komputer, buku, musik, pakaian, mainan, dan lain-lain.
2. **Tempat menjual produk (*a place to sell*)** : tempat menjual adalah internet yang berarti harus memiliki domain dan hosting.
3. **Cara menerima pesanan** : email, telpon, sms dan lain-lain.
4. **Cara pembayaran** : Cash, cek, bankdraft, kartu kredit, internet payment (misalnya paypal).
5. **Metode pengiriman** : pengiriman bisa dilakukan melalui paket, salesman, atau didownload jika produk yang dijual memungkinkan untuk itu (misalnya software).
6. **Customer service** : email, formulir on-line, FAQ, telpon, chatting, dan lain-lain.

### 2.1.3 Website

#### a. Website

Menurut Sibero (2014:11), “ *Website (world wide web /W3)* adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia, dan lainnya pada jaringan *internet*.”

Menurut Kustiyahningsih dan Rosa Anamisa (2011:4) “*website* merupakan salah satu layanan yang didapat oleh pemakai komputer yang

terhubung ke internet *hypertext* untuk menampilkan data berupa teks, gambar, suara, animasi dan data multimedia lainnya”.

Kesimpulannya, *website* adalah sebuah informasi melalui internet, yang merupakan kumpulan-kumpulan dokumen yang banyak tersebar di beberapa server yang ada di penjuru dunia dan terhubung menjadi satu jaringan melalui jaringan yang disebut internet. Situs web biasanya ditempatkan setidaknya pada sebuah server web yang dapat diakses melalui jaringan seperti *internet*, atau punjaringan wilayah lokal (LAN) melalui alamat *internet*, yang dikenali sebagai URL. Gabungan atas semua situs yang dapat diakses publik di internet disebut pula sebagai *World Wide Web* atau lebih dikenal dengan singkatan WWW.

#### b. *Internet*

Menurut Sibero (2014:10), “Internet (*interconnected Network*) adalah jaringan komputer yang menghubungkan antar jaringan secara global, *internet* juga dapat disebut jaringan dalam suatu jaringan yang luas”.

Menurut Kustiyahningsih (2011:2). Secara sederhana dapat dikatakan internet adalah sebuah “jaringan komputer dunia”, semua berbicara dengan bahasa yang sama. Banyak keuntungan yang didapat dari jaringan komputer, diantaranya produktivitas dan efisien. Jaringan komputer area atau local dapat dibagi menjadi empat yaitu :

1. *Lokal Area Network* (LAN), yaitu jaringan komputer dimana komputer-komputer yang terhubung masih dalam satu area atau lokasi.
2. *Wide Area Network* (WAN), yaitu koneksi antara LAN-LAN yang berbeda lokasi/area. Ciri utamanya adalah memiliki bandwidth yang terbatas

karena disesuaikan dengan fungsi harga, adanya masalah keterlambatan antarstasiun.

3. *Metropolitan Area Network (MAN)*, yaitu sama seperti LAN hanya saja lebih luas areanya semisal dalam satu kota/daerah dengan range mencapai 50km.
4. *Internet*, yaitu kepanjangan dari interconnection networking atau juga yang telah menjadi international networking merupakan suatu jaringan yang menghubungkan komputer diseluruh dunia tanpa dibatasi oleh jumlah unit menjadi satu jaringan yang bisa saling mengakses.

c. *Web Browser*

Menurut Sibero (2014:12), “*Web Browser* adalah aplikasi perangkat lunak yang digunakan untuk mengambil dan menyajikan sumber informasi *web*”. Sumber informasi *web* diidentifikasi dengan *Uniform Resource Identifier (URI)* yang dapat terdiri dari halaman *web*, video, gambar, ataupun konten lainnya. Contoh : *Firefox, Internet Explorer, Safari, Google Chrome, Opera*.

d. *Web Server*

Menurut Sibero (2014:11), “*Web Server* adalah sebuah komputer yang terdiri dari perangkat keras dan perangkat lunak, perangkat keras *web server* tidak berbeda dengan komputer rumah atau PC, yang membedakan adalah kapasitas dan kapabilitasnya.

Perbedaan tersebut dikarenakan *Web Server* bekerja sebagai penyedia layanan yang dapat di akses oleh banyak pengguna, sehingga dibutuhkan kapasitas dan kapabilitas yang besar dibandingkan PC. Contoh: *Apache Software*

*Foundation, Microsoft Corp, Igor Sysoev, Lighttpd, Jigsaw, Koanlogic, Oracle Corp, iMatic Corp, Aprelium Tech, Zeus Teknologi Ltd.*

Menurut Sadeli (2013:2) menyimpulkan bahwa “*Web Server* adalah sebuah perangkat lunak server yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan *web browser* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML”.

#### **2.1.4 Bahasa Pemrograman**

##### a. PHP (*Personal Home Pages*)

Menurut Sibero (2014:49), “PHP (*Personal Home Page*) adalah pemrograman *interpreter* yaitu proses penerjemahan baris sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan”. Pemrograman PHP dapat ditulis dalam dua bentuk yaitu penulisan baris kode PHP pada *file* tunggal (*singlefile*) dan penulisan kode PHP pada halaman html (*embedded*).

Cara penulisan skrip PHP ada dua macam, yaitu *singlefile* dan *embedded*.

Contoh :

*Singlefile*

```
echo "<html>";
```

```
echo "<head>";
```

```
echo "<title>contoh php</title> ";
```

```
echo "</head>";
```

```
echo "</body>";
```

```
echo "<p> dibawah ini adalah tulisan hasil dari PHP</p>";
```

```
echo "teks dari PHP";
```

```
echo "</html>";
```

*embedded*

```
<html>
```

```
    <head>
```

```
        <title> contoh PHP</title>
```

```
    </head>
```

```
    </body>
```

```
        <p> dibawah ini adaah tulisan dari PHP</p>
```

```
        <?PHP
```

```
            Echo "teks dari PHP";
```

```
        ?>
```

```
    </body>
```

```
</html>
```

## b. HTML (*Hypertext Markup Language*)

Menurut Sibero (2014:19) "HTML (*Hyper Text Markup Language*) adalah bahasa yang digunakan pada dokumen *web* sebagai bahasa untuk pertukaran

dokumen web”. Struktur dokumen HTML terdiri dari *tag* pembuka dan *tag* penutup.

Dalam penulisan struktur Dokumen HTML terdapat 3 hal yang harus di perhatikan :

1. *HTML*

Setiap dokumen *HTML* diawali dan ditutup dengan *tag HTML*.

2. *HEAD*

Bagian *head* biasanya berisi *tag TITLE*, *Meta tag*, dan semua *script* java atau yang lainnya yang di eksekusi di *browser*. Di bagian inilah kita memberikan *bookmark* untuk keperluan pencarian (*searching*).

3. *BODY*

Pada bagian ini digunakan untuk menampilkan *text*, *image*, *link*, dan semua yang ditampilkan pada *web page*.

Menurut Kustiyaningsih dan Anamisa (2011:13) “HTML kependekan dari *Hyper Text Markup Language*. Dokumen HTML adalah *file text* murni yang dapat dibuat dengan editor teks sembarang”. Dokumen ini dikenal sebagai *web page*. File-file HTML ini berisi instruksi-instruksi yang kemudian diterjemahkan oleh browser yang ada dikomputer *client (user)* sehingga isi informasinya dapat ditampilkan secara visual di computer pengguna (*user*).

- c. *CSS (Cascading Style Sheet)*

Menurut Supardi (2010:128) mengemukakan bahwa “CSS merupakan format atau style untuk tampilan web, dapat dibuat dalam halaman bersama perintah HTML atau file tersendiri dengan nama perluasan *.css*”.

CSS dikembangkan untuk menata gaya pengaturan haman web. Pada walanya CSS dikembangkan pada SGML pada tahun 1970. Pada bulan Desember 1996 W3C mengenalkan spesifikasi CSS level 1 atau dikenal sebagai CSS! Yang mendukung format huruf, warna pada teks, dan lain-lain. Pada bulan Mei 1998 W3C mengeluarkan CSS2 yang didalamnya terdapat fungsi pengaturan letak elemen, dan versi terbaru CSS yang mampu melakukan desain *website*, diantaranya animasi warna hingga animasi 3D. Saat ini CSS telah mendukung banyak bahasa seperti HTML, XHTML, XML, SVG (*Scalable Vector Graphics*), dan Mozilla XUL (*XML User Interface Language*). Berikut contoh sederhana penulisan CSS :

```
<div style = "font-weight:bold">
<p>Contoh paragraph dalam DIV dengan style font:bold</p>
<span> Contoh span dalam DIV dengan style font-weight:bold</span>
</div>
```

#### d. Adobe Dreamweaver CS6

Menurut Andi (2013:2) Adobe Dreamweaver CS6 merupakan versi terbaru dari Adobe Dreamweaver yang sebelumnya adalah Adobe Dreamweaver CS5. Aplikasi Adobe Dreamweaver CS6 memberi tampilan yang lebih baik dan tentu saja semakin mudah dalam penggunaannya. Aplikasi ini mengintegrasikan beragam fitur untuk memenuhi kebutuhan pengembangan website, termasuk pembuatan halama web dan pengelolaannya.

Adobe Dreamweaver CS6 menyertakan banyak *tool* yang berkaitan dengan pengkodean sperti HTML, CSS, XML, dan pemrograman *Client Side*, yaitu *JavaScript* dengan penggunaan yang sangat mudah dan *user friendly*.

Aplikasi ini juga mendukung pemrograman *Script Server Side* seperti PHP, *Active Server Page (ASP)*, *ASP.NET*, *ASP Java Script*, *ASP VB Script*, *Cold Fusion*, dan *Java Server Page (JSP)*.

Fasilitas yang ada pada Adobe Dreamweaver CS6 memberikan kemudahan kepada user untuk melakukan pengeditan karena ditampilkan secara visual. Penambahan desain dan fungsi pada halaman web tidak harus dituliskan dalam baris kode. Anda tinggal memilih dan menempatkan komponen web dengan melakukan drag ke dalam dokumen web secara langsung dan cepat.

Selain itu, Adobe Dreamweaver CS6 juga dapat meng-import dan menyisipkan *image* atau *movie* yang dibuat dari aplikasi lainnya seperti *file flash* (.SWF) ataupun FLV.

#### e. *JQuery*

Menurut Sibero (2014:218) “*JQuery* adalah library *Javascript* yang berfungsi memudahkan pembuatan program pada browser internet”. Dengan *Jquery*, tidak perlu mengetikkan kode *JavaScript* secara panjang lebar. Tinggal memanggil fungsi-fungsi yang sudah disediakan *Jquery*. *Jquery* dilengkapi beragam fitur untuk memanipulasi *event* dan elemen pada dokumen HTML. Dengan *Jquery*, pengaturan serta manipulasi dokumen HTML dapat dilakukan dengan mudah. Pembuatan animasi dan ajax pun lebih mudah dilakukan dengan bantuan *Jquery*. Dengan *Jquery*, pekerjaan koding halaman web yang menjadi lebih ringan karena tidak harus bermain langsung di level *JavaScript*.

### 2.1.5 Basis Data

#### a. Pengertian Basis Data (*database*)

Menurut Anhar (2010:45) “*Database* adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari *field* atau kolom”. Struktur *file* yang menyusun sebuah *database* adalah *Data Record* dan *Field*.

Menurut Kustiyahningsih (2011:145) Basis Data adalah sekumpulan informasi yang diatur agar mudah dicari. Dalam arti umum basis data adalah sekumpulan data yang diproses dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan tepat, yang dapat digambarkan sebagai aktivitas dari satu atau lebih organisasi yang berelasi.

Basis data adalah sekumpulan relasi data persisten yang secara logika terhubung dimana data tersebut merupakan deskripsi dari satu atau lebih aktivitas dari organisasi yang bersangkutan yang digunakan oleh aplikasi sistem perusahaan yang dibagi dan saling berhubungan yang dirancang demi memenuhi kebutuhan organisasi. Sedangkan sistem basis data dapat disimpulkan sebagai sekumpulan aplikasi program yang saling berinteraksi dengan basis data melalui DBMS dan basis data itu sendiri dan juga merupakan suatu sistem penyimpanan *record* yang sudah terkomputerisasi.

*Data Base Management System* (DBMS) adalah suatu sistem perangkat lunak yang didesain untuk menangani pengelolaan dan penggunaan dari suatu kumpulan data, serta memungkinkan pengguna untuk mendefinisikan, membuat, memelihara, dan mengendalikan akses terhadap sistem basis data.

Fungsi-fungsi dasar yang harus didukung DBMS adalah :

1. Pendefinisian data (*data definition*)

DBMS harus dapat menerima pendefinisian data (skema eksternal, skema konseptual dan semua asosiasi pemetaan) dari sumber dan mengkonversikan ke dalam bentuk objek yang sesuai.

2. Manipulasi data (*data manipulation*)

DBMS harus dapat menangani permintaan untuk mengambil, memperbaharui atau menghapus data yang sudah ada di basis data, maupun menambah data kedalam basis data.

3. Optimalisasi dan eksekusi (*optimization and execution*)

Permintaan Data Manipulation Language (DML) harus diproses dikomponen pengoptimalisi yang bertujuan untuk menentukan cara yang efisien untuk implementasi permintaan. Permintaan yang telah dioptimalisasi kemudian dieksekusi dibawah kendali re-time manager.

4. Keamanan dan integritas data (*data security and integrity*)

DBMS harus mengawasi permintaan pengguna dan menolak gangguan yang dapat membahayakan keamanan dan integrity constraint yang sudah ditentukan oleh Database Administrator (DBA).

5. Perbaikan data dan konkurensi (*data recovery and concurrency*)

DBMS dapat juga disebut *Transaction Processing Monitor* (TPM) harus melakukan kendali perbaikan dan konkurensi.

6. Kamus data (*data dictionary*)

Kamus data berisi “data mengenai data” adalah definisi dari objek lain disistem. Semua skema dan pemetaan, berbagai sistem keamanan, dan

*integrity constraint* akan disimpan, baik dari sumber maupun bentuk objek didalam kamus data.

#### 7. Kinerja (*performance*)

DBMS harus dapat mengerjakan semua tugas seefektif mungkin.

#### b. MySQL (*My Structured Query Language*)

Menurut Sibero (2014:97), "MySQL atau dibaca 'My Sekuel'" dengan adanya suatu RDBMS (*Relational Data Base Manajemen System*) yaitu aplikasi sistem yang menjalankan fungsi pengolahan data".

Selain itu, MySQL juga merupakan program pengakses *database* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi *multi-user* (banyak pengguna). Saat ini *database* MySQL telah digunakan hampir oleh semua pemrogram *database*, terlebih dalam pemrograman *web*. Kelebihan lain dari MySQL adalah penggunaan bahasa *query* yang dimiliki SQL (*Structured Query Language*).

Menurut Sadeli (2013:10), "MySQL adalah database yang menghubungkan *script php* menggunakan perintah *query* dan *escaps character* yang sama dengan *php*. MySQL mempunyai tampilan client yang mempermudah Anda dalam mengakses database dengan kata sandi untuk mengijinkan proses yang bisa Anda lakukan".

MySQL, Anda bisa membaca dengan *My-Ess-Que-Ell*, merupakan *relational database management system (RDBMS)*. Jika Anda bingung dengan istilah RDBMS, bayangkan saja bahwa Anda sedang membangun situs yang

menjual CD atau kaset. Tentu saja di dalam situs tersebut, Anda harus mempunyai data harga masing-masing CD atau kaset, mengurutkan data tersebut berdasarkan nama atau harga (agar pengunjung yang hendak membeli dapat dengan mudah mencari judul atau harga yang diinginkan), menyimpan transaksi penjualan yang terjadi, dapat melaporkan rekap penjualan pada periode tertentu, memperbaharui harga CD atau kaset (jika toko Anda sedang mengadakan sale). Bagaimana Anda mengelola data tersebut? itulah kegunaan RDBMS. Atau lebih tepatnya, disitulah peranan MySQL dalam situs Anda, yaitu menyimpan data toko Anda (termasuk didalamnya data barang yg dijual, transaksi serta melaporkan pelaporan).

MySQL didistribusikan secara *open source* (dan gratis) mulai tahun 1996, namun mempunyai sejarah pengembangan sejak tahun 1979. Walaupun didistribusikan secara gratis, namun MySQL mempunyai versi komersial.

MySQL merupakan database yang sangat populer di kalangan pengembang situs yang sifatnya dinamis. Berikut ini adalah keunggulan dari MySQL :

1. Cepat. Sejak awal, MySQL dikembangkan dengan konsep database yang berkecepatan tinggi dalam penyajian data.
2. Tidak mahal. Walaupun mempunyai versi komersial, namun sesungguhnya MySQL dapat di-download dengan gratis.
3. Gampang digunakan. Anda dapat berinteraksi dengan MySQL menggunakan perintah SQL yang sederhana. Perintah SQL adalah standar bahasa untuk RDBMS.

4. Berjalan pada berbagai sistem operasi. MySQL berjalan dengan baik di banyak sistem operasi, seperti Windows, Linux, Mac OS, Unix (Solaris, AIX dan DEC Unix), FreeBSD, OS/2, Irix dan lainnya.
5. Dukungan penggunaan banyak tersedia. Ada banyak grup diskusi tentang MySQL yang ditawarkan di situs MySQL, atau Anda dapat juga bertanya MySQL AB dengan sedikit biaya.

c. PHP MyAdmin

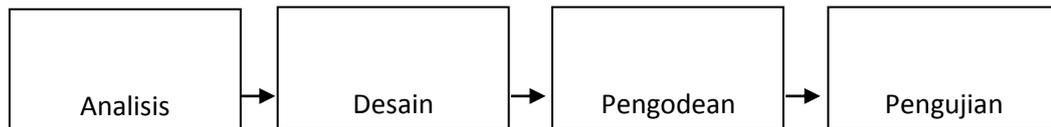
Menurut Sadeli (2013:10) “*Php MyAdmin* adalah Sebuah *software* yang berbentuk seperti halaman situs yang terdapat pada *web server*”. Fungsi dari halaman ini adalah sebagai pengendali database MySQL sehingga pengguna MySQL tidak perlu repot untuk menggunakan perintah-perintah SQL. Karena dengan adanya halaman ini semua hal tersebut dapat dilakukan hanya dengan meng-klik menu fungsi yang ada pada halaman *PHP MyAdmin*. *Php MyAdmin* dijalankan dengan cara mengetik <http://localhost/phpmyadmin> pada *web browser*.

d. XAMPP *server*

Menurut (Nugroho, 2014:1) XAMPP merupakan salah satu paket *Web Server* yang didalamnya sudah berisi software Apache, PHP, dan MySQL. Dimana Apache adalah *web server* untuk menjalankan aplikasi *web* berbasis PHP dan PHP adalah kompilator skrip PHP supaya dapat dijalankan paada *browser* seperti Internet Explorer dan Mozilla Firefox sedangkan MySQL adalah database server untuk emnyimpan. Paket *software* ini bersifat gratis dan dapat diperoleh dari situs aslinya, yaitu [www.apachefriends.org](http://www.apachefriends.org).

### 2.1.6 Sistem

Menurut Rosa A.S dan M.Shalahuddin (2013:28) “Model *waterfall* adalah model *SDLC* yang paling sederhana. Model ini hanya cocok untuk pengembangan perangkat lunak dengan spesifikasi yang tidak berubah-ubah”.



Sumber : Rosa A.S dan M.Shalahuddin ( 2013:28 )

#### **Gambar II.1 Ilustrasi Model *Waterfall***

Model *SDLC* air terjun (*waterfall*) sering juga disebut model sekuensial linier atau alur hidup klasik. Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut mulai dari analisis, desain, pengodean, pengujian dan tahap pendukung.

Pembuatan model *waterfall* tentunya tidak terlepas dari tahapan-tahapan yang harus dikerjakan secara terstruktur. Untuk lebih jelasnya tahapan-tahapan pembuatan model *waterfall* adalah sebagai berikut:

#### 1. Analisa kebutuhan

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu didokumentasikan.

## 2. Desain Sistem

Menurut Rosa A.S dan M.Shalahuddin (2011:27) menyatakan bahwa :”Desain perangkat lunak adalah proses multistep yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan”.

## 3. Penulisan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

## 4. Pengujian Program

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

## 5. *Support / Maintenance*

Pada tahapan ini tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah diberikan kepada user. Perubahan ini bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.

Tahap pendukung / *maintenance* dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada tanpa membuat perangkat lunak yang baru.

Dalam penggunaan metode *waterfall* ini ada beberapa keuntungan dan kekurangan seperti berikut :

1. Keuntungan metode *waterfall*

- a. Kualitas dari sistem yang dihasilkan akan baik. Ini dikarenakan oleh pelaksanaannya dilakukan secara bertahap sehingga tidak terfokus pada tahapan tertentu.
- b. Dokumen pengembangan sistem sangat terorganisir, karena setiap tahap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya. Jadi setiap fase atau tahapan akan mempunyai dokumen tertentu.
- c. Metode ini masih lebih baik digunakan walaupun sudah tergolong kuno, daripada menggunakan pendekatan lain. Selain itu, metode ini juga masih masuk akal jika kebutuhan sudah diketahui dengan baik.

2. Kelemahan Metode *Waterfall*

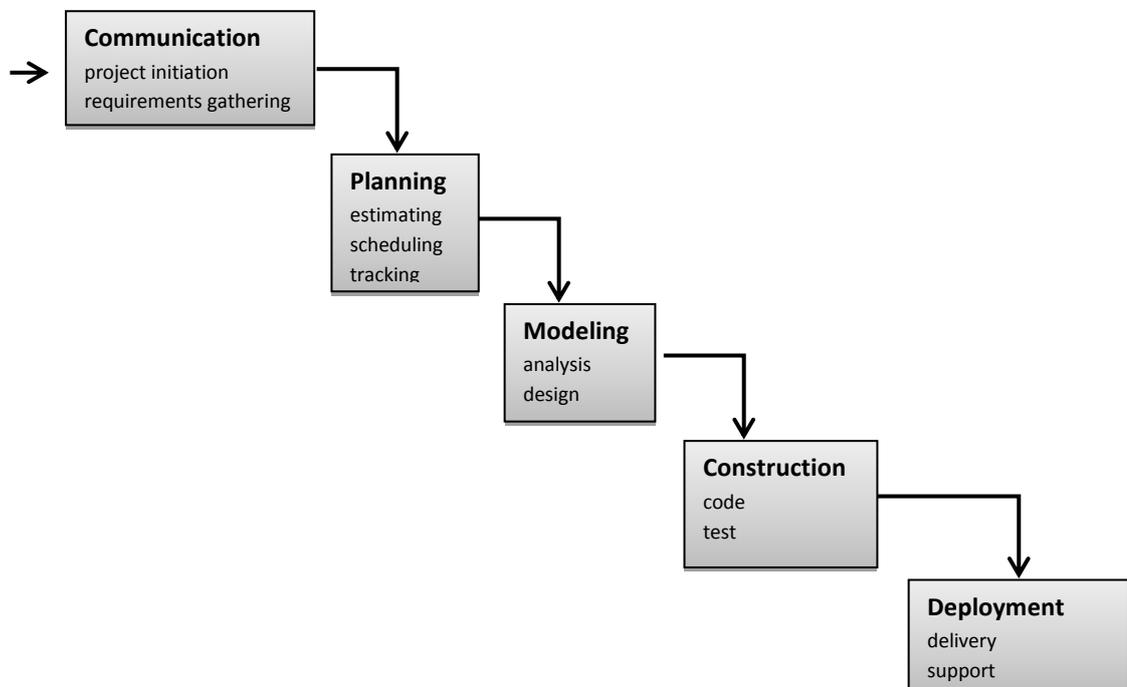
- a. Diperlukan manajemen yang baik, karena proses pengembangan tidak dapat dilakukan secara berulang sebelum terjadinya suatu produk.
- b. Kesalahan kecil akan menjadi masalah besar jika tidak diketahui dari awal pengembangan perangkat lunak yang berakibat pada tahapan selanjutnya tidak akan berjalan sesuai rencana.

- c. *User* sulit menyatakan kebutuhan eksplisit sehingga tidak dapat mengakomodasi ketidakpastian pada saat awal pengembangan perangkat lunak
- d. *User* harus sabar, karena pembuatan perangkat lunak akan dimulai ketika tahap desain sudah selesai. Sedangkan tahap sebelum desain bisa memakan waktu yang lama.

### 2.1.7 Model Pengembangan Perangkat Lunak

Menurut Pressman (2010:39) “model *waterfall* adalah model klasik yang bersifat sistematis berurutan dalam membangun *software*”.

Fase-fase dalam model waterfall menurut referensi Pressman:



Sumber : Pressman (2010:39)

**Gambar II.2 Waterfall Pressman**

a. *Communication* (komunikasi)

Langkah ini merupakan analisis atas kebutuhan *software*, dan tahap untuk mengadakan pengumpulan data dengan melakukan pertemuan dengan *costumer*, maupun mengumpulkan data-data tambahan baik yang ada di jurnal, artikel maupun dari *internet*.

b. *Planning* (perencanaan)

Proses *planning* merupakan lanjutan dari proses *communication (analysis requirement)*. Tahapan ini menghasilkan dokumen *user requirement* atau biasa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan *software*, termasuk rencana yang akan dilakukan.

c. *Modeling* (pemodelan)

Proses *modeling* ini akan menerjemahkan syarat kebutuhan ke sebuah perancangan *software* yang dapat diperkirakan sebelum membuat *coding*. Proses ini berfokus pada rancangan struktur data, arsitektur *software*, representasi *interface* dan detail (algoritma) prosedural. Tahap ini menghasilkan dokumen yang disebut *software requirement*.

d. *Construction* (kontruksi)

*Construction* merupakan proses membuat kode. *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu *software*, artinya penggunaan komputer akan

dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka dilakukan *testing* terhadap sistem yang telah dibuat tadi. Tujuan *testing* ini adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki.

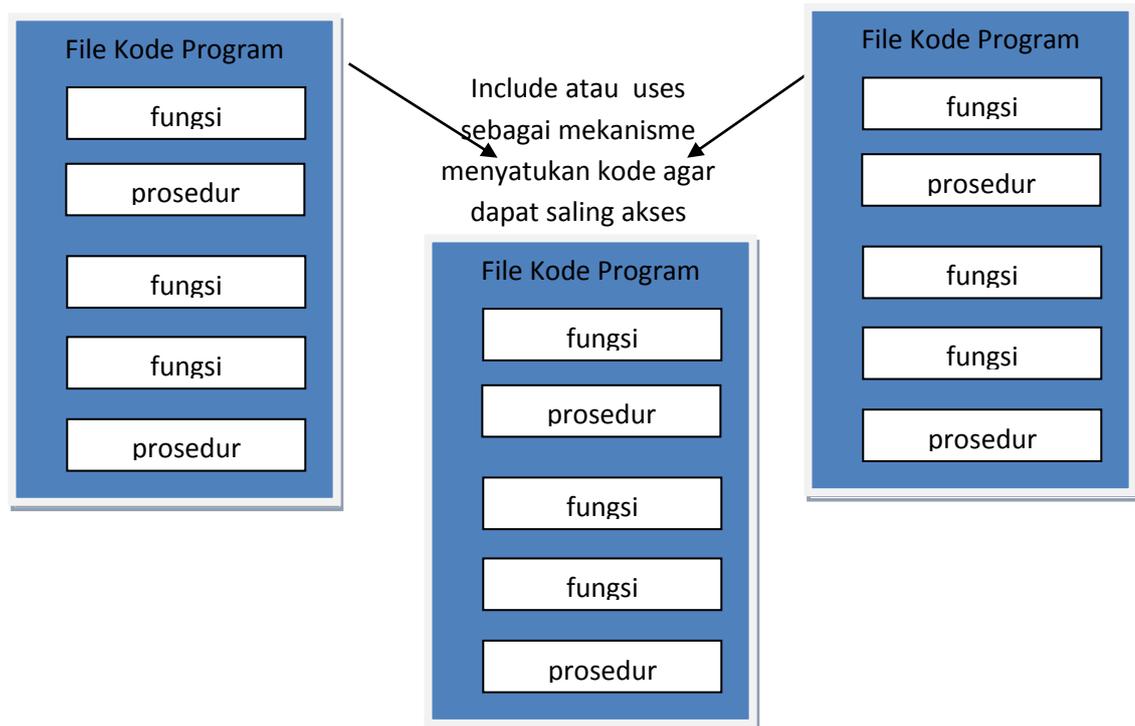
e. *Deployment* (pengembangan)

Tahapan ini bisa dikatakan final dalam merupakan pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*. Kemudian *software* yang telah dibuat harus dilakukan pemeliharaan secara berkala.

## 2.2 Konsep Dasar Pemrograman

“Pemrograman terstruktur adalah konsep atau paradigma atau sudut pandang pemrograman yang membagi-bagi program berdasarkan fungsi-fungsi atau prosedur-prosedur yang dibutuhkan program computer” Rosa A.S dan M.Shalahuddin (2011:62).

Fungsi-fungsi dan prosedur-prosedur ditulis secara sekuensial atau terurut dari atas ke bawah sesuai dengan kebergantungan antar fungsi atau prosedur (fungsi atau prosedur yang dipakai oleh fungsi atau prosedur di bawahnya harus yang sudah ditulis atau dideklarasikan di atasnya).



Sumber : Rosa A.S dan M.Shalahuddin ( 2011:62 )

### **Gambar II.3 Ilustrasi Pemrograman Terstruktur**

Untuk mengetahui lebih jauh lagi, penulis akan menjelaskan beberapa konsep dasar dalam pemrograman terstruktur yang harus dipahami :

#### 1. Modular

Dalam pemrograman secara modular, program dipecah-pecah menjadi modul-modul. Setiap modul menunjukkan fungsi dan tugas tunggal. Modul-modul tersebut ditulis dan dicari kesalahannya secara terpisah. Karena tujuan dan ukuran setiap modul dibatasi, maka terjadinya kesalahan dalam program tersebut dapat dikurangi.

Setiap program mempunyai sebuah modul program utama yang mengontrol semua proses yang terjadi, termasuk mengirimkan control program ke submodul untuk melakukan suatu fungsi tertentu. Submodul ini harus

mengembalikan control program tersebut ke modul utama setelah selesai melaksanakan tugas dan fungsinya.

Pemrograman modular ini diterapkan menggunakan subroutines, yaitu sebuah kumpulan perintah yang melakukan tugas pemrosesan yang terbatas, seperti mencetak sebagian laporan, membaca masukan data, atau menghitung akar pangkat.

## 2. *Top – Down Programming*

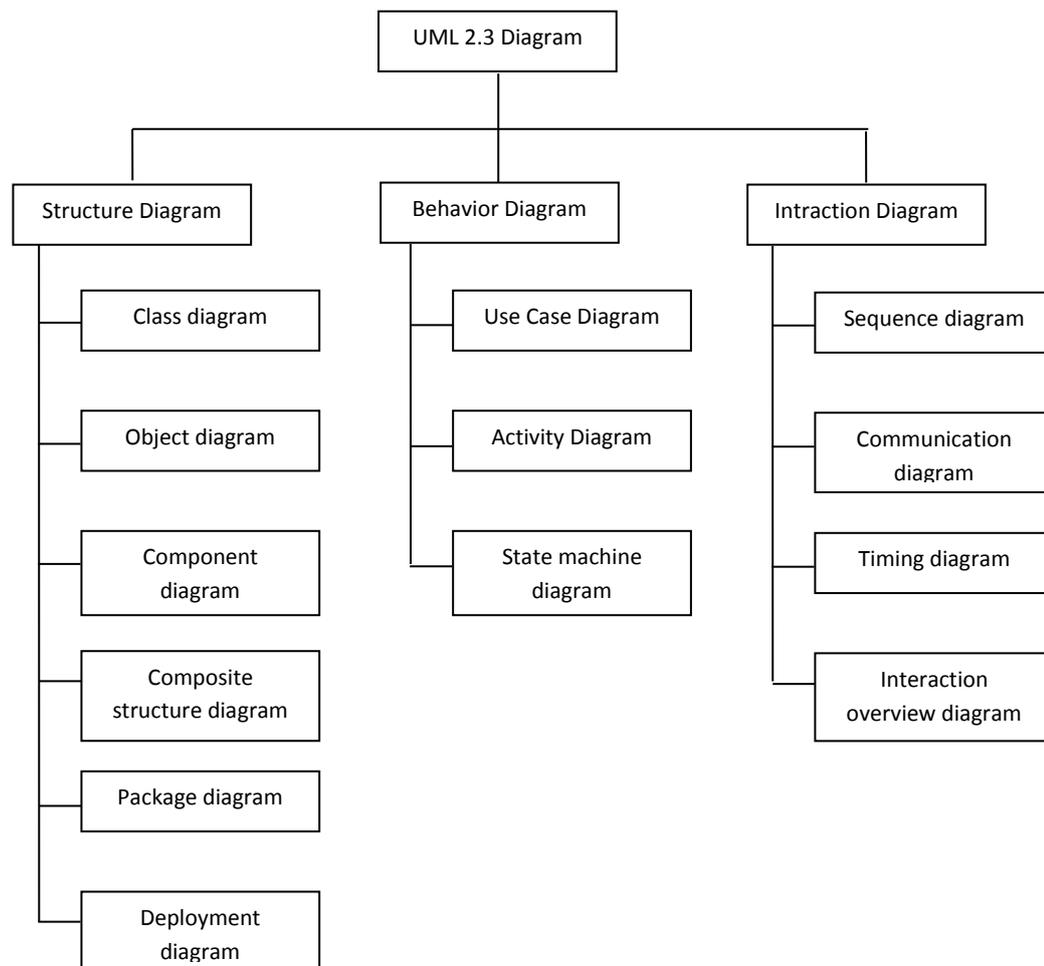
Pendekatan Top-down ini sangat berguna dalam perencanaan pemrograman modular. Dalam pemrograman top-down (atas-bawah), yang pertama harus kita definisikan adalah modul utama. Modul utama yang dimaksud adalah modul yang pertamakali kita jalankan atau modul yang memanggil modul lainnya atau juga modul yang mengakhiri proses tersebut.

### 2.2.1 **Peralatan Pendukung Sistem (*Tools Sistem*)**

Merupakan alat yang digunakan untuk menggambarkan bentuk logika model dari suatu sistem dengan menggunakan simbol-simbol, lambang-lambang, diagram-diagram yang menunjukkan secara tepat arti dan fungsinya. Dalam hal ini penulis menggunakan *tools sistem* menggunakan UML (*Unified Modeling Language*), ERD (*Entity Relationship Diagram*) dan LRS (*Logical Record Structure*).

### 2.2.1.1 UML ( *Unified Modeling Language* )

“Bahasa visual untuk pemodelan dan komunikasi mengenai UML adalah sebuah sistem dengan menggunakan diagram dan teks-teks pendukung” menurut Rosa A.S dan M.Shalahuddin (2011:118).



Sumber : Rosa A.S dan M.Shalahuddin ( 2011:121)

#### Gambar II.4 Diagram UML

Berikut ini penjelasansingkat dari pembagian kategori tersebut.

- a. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan

- b. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
- c. *Interaction diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

Penjelasan tentang masing – masing diagram akan dilakukan pada submodul-submodul berikutnya.

#### 1. *Use Case Diagram*

*Use Case* mendetugaskan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Menggambarkan fungsionalitas yang diterapkan dari sebuah sistem. *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesimple mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut actor dan *use case*.

- a. *Actor* merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, tapi *actor* belum tentu merupakan orang.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau *actor*.

## 2. *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa apa yang dilakukan *actor*, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

- a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
- b. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
- c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

## 3. *Component Diagram*

Diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem. Diagram komponen juga dapat digunakan untuk memodelkan hal-hal berikut :

- a. *Source code* program perangkat lunak.
- b. Komponen *executable* yang dilepas ke user.
- c. Basis data secara fisik.

#### 4. Deployment Diagram

Diagram *deployment* atau *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram deployment juga dapat digunakan untuk memodelkan hal-hal berikut :

- a. Sistem tambahan (*embedded sistem*) yang menggambarkan rancangan *device, node, hardware*
- b. Sistem *client/server*
- c. Sistem terdistribusi murni
- d. Rekayasa ulang aplikasi

#### 2.2.1.2 ERD (*Entity Relationship Diagram*)

Menurut Rosa dan Shalahuddin (2013:50) “ERD adalah bentuk paling awal dalam perancangan basis data relasional.. Sehingga jika menggunakan OODBMS maka perancangan ERD tidak perlu dilakukan. Dalam pembentukan ERD terdapat beberapa komponen yang akan dibentuk diantaranya :

1. Entitas yang merupakan data inti yang akan disimpan, penamaan entitas biasanya lebih ke kata benda dan belum merupakan tabel.
2. Atribut adalah kolom data yang butuh disimpan dalam suatu entitas.
3. Atribut kunci primer adalah kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan.
4. Atribut multivali adalah kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5. Relasi adalah hubungan antar entitas, biasanya diawali dengan kata kerja.
6. Asosiasi adalah penghubung antar relasi dan entitas dimana dikedua

ujungnya memiliki multiplicity kemungkinan jumlah pemakaian, kemungkinan jumlah maksimum keterhubungan antar entitas yang lain disebut dengan kardinalitas.

Adapun beberapa derajat *Relationship* diantaranya :

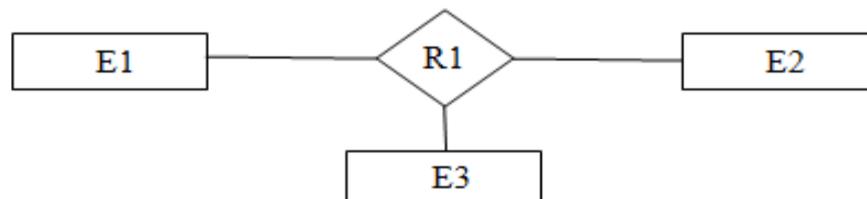
- a. *Binary* adalah satu relasi menghubungkan dua buah entitas.



Sumber : Rosa dan Shalahuddin (2013:52)

**Gambar II.5 Binary**

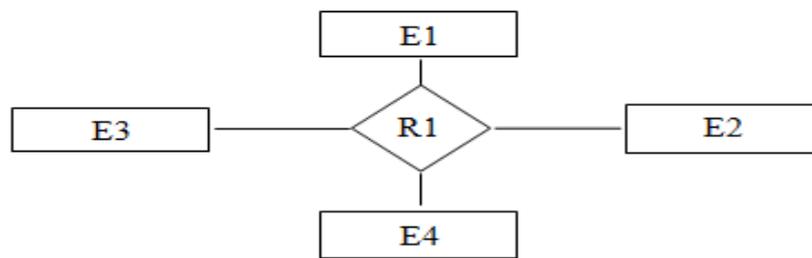
- b. *Ternary* adalah satu relasi menghubungkan tiga buah relasi entitas.



Sumber : Rosa dan Shalahuddin (2013:52)

**Gambar II.6 Ternary**

c. *N-ary* adalah satu relasi menghubungkan banyak entitas.



Sumber : Rosa dan Shalahuddin (2013:52)

**Gambar II.7 N-ary**

### 2.2.1.3 LRS (*Logical Record Structure*)

*Logical Record Structure (LRS)* merupakan representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil relasi antar himpunan entitas pada diagram E-R. Transformasi diagram E-R ke LRS merupakan suatu kegiatan untuk membentuk data-data dari diagram hubungan entitas untuk menentukan entitas pada diagram LRS. 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 ER-diagram dan langsung dikonversikan ke LRS.

Setelah ERD ditransformasikan ke bentuk LRS, maka hasil akhir dari proses transformasi tersebut adalah sebuah diagram yang sudah dapat menggambarkan basis data yang akan digunakan. *Logical Record Structure* terdiri dari *link-link* diantara tipe *record*. *Link* ini menunjukkan arah dari suatu tipe *record* lainnya.

Aturan-aturan dalam melakukan transformasi E-R diagram ke LRS :

1. Setiap entitas akan diubah kebentuk sebuah kotak dengan nama entitas berada diluar kotak, dan atribut berada didalam kotak.
2. Sebuah relasi kadang disatukan dalam sebuah kotak bernama entitas, kadang dipisah dalam kotak tersendiri.

Konversi *Logical Record Structure* 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 yang dalam bentuk lingkaran Diagram E-R dikenal dengan sebutan atribut.

Konversi dari *logical record structure* dilakukan dengan cara :

1. Nama *logical record structure* menjadi nama relasi.
2. Tiap atribut menjadi sebuah kolom didalam relasi.

## **2.3 Teori Pendukung**

### **2.3.1 Struktur Navigasi**

Menurut Prihatna (2008:51) Struktur Navigasi adalah “Susunan menu atau hirarki dari suatu situs yang menggambarkan isi dari setiap halaman dan *link* atau navigasi tiap halaman pada suatu situs *web*”. Struktur navigasi suatu situs *web* sangat dipengaruhi oleh tujuan dari situs *web* yang akan dibuat.

Struktur navigasi dapat digolongkan menurut kebutuhan akan objek, kemudahan pemakaian, keinteraktifitasannya dan kemudahan membuatnya yang berpengaruh terhadap waktu pembuatan suatu situs *web*. Bentuk dasar dari struktur navigasi antara lain:

a. *Linear* (Satu alur)

*Linear* (satu alur) merupakan struktur yang hanya mempunyai satu rangkaian cerita yang berurut. Dengan kata lain struktur ini hanya dapat menampilkan satu demi satu tampilan layar secara berurut menurut urutannya. Tampilan yang dapat ditampilkan pada struktur jenis ini adalah satu halaman sebelumnya atau satu halaman sesudahnya dan tidak dapat menampilkan dua halaman sebelumnya atau dua halaman sesudahnya. Salah satu yang terpenting dari struktur ini adalah tidak diperkenankan terjadinya percabangan

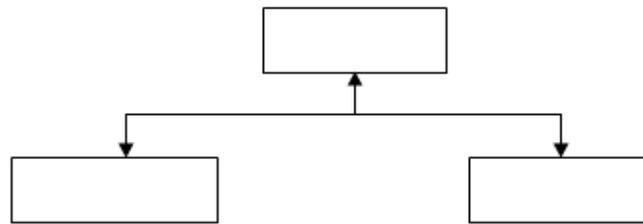


Sumber : Prihatna (2008:51)

**Gambar II.8 Struktur Navigasi *Linear***

b. Hirarki (Bercabang)

Struktur Hirarki (bercabang) ini percabangan untuk menampilkan data berdasarkan kriteria tertentu. Tampilan pada menu pertama akan disebut sebagai *Master Page* (halaman utama kesatu), halaman utama ini akan mempunyai halaman percabangan yang dikatakan *Slave Page* (halaman pendukung). Jika salah satu halaman pendukung dipilih atau diaktifkan, maka tampilan tersebut akan bernama *Master Page* (halaman utama kedua), dan seterusnya. Yang terpenting dari struktur penjejakan ini tidak diperkenankan adanya tampilan secara linear.

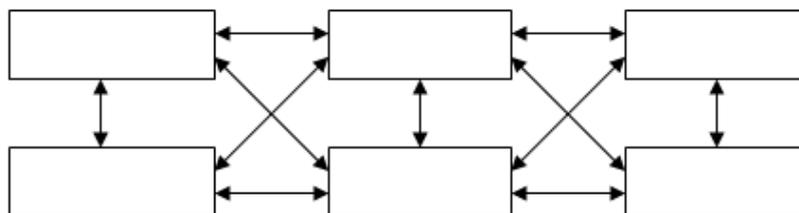


Sumber : Prihatna (2008:51)

**Gambar II.9 Struktur Navigasi Hirarki**

*c. Non Linear* (Tidak Berurut)

Struktur penjejakan *Non Linear* (tidak berurut) merupakan pengembangan dari struktur penjejakan *Linear*. Pada struktur ini diperkenankan membuat penjejakan bercabang. Pemakai bebas menelusuri *website* tanpa dibatasi oleh suatu rute di mana kontrol navigasi dapat mengakses ke semua halaman manapun. Percabangan yang dibuat pada struktur *Non Linear* ini berbeda dengan percabangan pada struktur *Hierarchi*, karena pada percabangan *Non Linear* ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama tidak ada *Master Page* dan *Slave Page*.



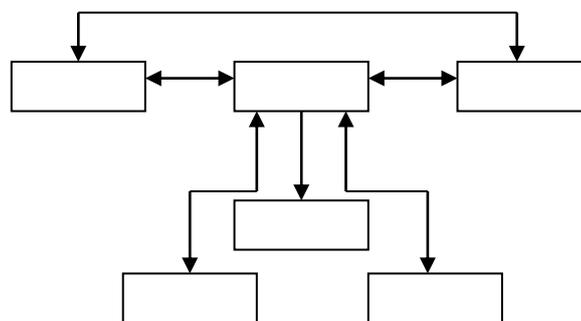
Sumber :Prihatna (2008:51)

**Gambar II.10 Struktur Navigasi *Non Linear***

d. *Composite* (Campuran)

*Composite* (campuran) atau disebut juga struktur penjejakan bebas merupakan gabungan dari ketiga struktur sebelumnya yaitu *Linear*, *Non Linear* dan *Hierarchi*. Jika suatu tampilan membutuhkan percabangan, maka dapat dibuat percabangan, dan bila dalam percabangan tersebut terdapat suatu tampilan yang sama kedudukannya maka dapat dibuat struktur *Linear* dalam percabangan tersebut.

Setiap struktur peta penjejakan seperti yang baru dibahas mempunyai fungsi dan tujuan tersendiri, tidak ada yang lebih baik atau lebih buruk. Penggunaan peta penjejakan bergantung kepada kebutuhan dan tujuan dari *web* yang hendak dibuat. Semakin kompleks peta penjejakan yang digunakan, maka semakin sulit pula pembuatan *page* dari peta penjejakan tersebut.



Sumber :Prihatna (2008:51)

**Gambar II.11 Struktur Navigasi Campuran**

## 2.4 Pengujian Web (*Black-Box Testing*)

Menurut Pressman (2010:495) “*Black-Box testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineers* untuk memperoleh set kondisi *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program”.

*Black-Box testing* berusaha untuk menemukan kesalahan dalam kategori yaitu, fungsi yang tidak benar atau fungsi yang hilang, kesalahan antarmuka, kesalahan dalam struktur data atau akses *database* eksternal, kesalahan perilaku (*behaviour*) atau kesalahan kinerja, inisialisasi dan pemutusan kesalahan.

Tes ini dirancang untuk menjawab beberapa pertanyaan-pertanyaan yang berhubungan dengan sistem yang ada di *website*, diantaranya sebagai berikut :

- a. Bagaimana validitas fungsional diuji?
- b. Bagaimana perilaku dan kinerja sistem diuji?
- c. Apa kelas *input* akan membuat kasus uji yang baik?
- d. Apakah sistem *sensitive* terhadap nilai input tertentu?
- e. Bagaimana batas-batas kelas data yang terisolasi?
- f. Kecepatan dan volume data seperti apa yang dapat ditolerir sistem?
- g. Efek apakah yang akan menspesifikasikan kombinasi data dalam sistem operasi?

## 2.5 Penelitian Terkait

Penulis menggunakan tiga referensi jurnal yang berhubungan dengan penjualan berbasis web untuk mendukung isi dari penulisan tugas ini.

Menurut Acep Angga dkk (2015:1) “Toko Bagja Jaya merupakan salah satu usaha yang menjual berbagai bahan bangunan yang berlokasi di Jl. Bayongbong. Ds. Sukarame Kec. Bayongbong Kab. Garut. Toko bangunan Bagja Jaya masih memakai sistem informasi yang manual, penulis mencoba membuat dan menerapkan aplikasi sistem informasi pada toko bangunan tersebut. Dengan tujuan untuk mempercepat dan memudahkan proses transaksi penjualan bahan bangunan, tanpa harus repot mencatat secara manual ke sebuah buku besar. Pada penelitian ini akan dirancang sebuah Aplikasi Penjualan dan Pembelian Bahan Bangunan dengan menggunakan metode Waterfall dari Ian Sommerville (2003). Tahapannya adalah analisis defines persyaratan, perancangan system dan perangkat lunak, implementasi dan pengujian unit dan integrasi dan pengujian system. Bahasa pemrograman yang digunakan untuk merancang aplikasi ini adalah PHP dan Adobe Dreamweaver untuk pembuatannya. DBMS yang digunakan dalam sistem paka ini adalah MySQL yang terintegrasi dalam aplikasi XAMMP. Hasil yang diperoleh pada penelitian ini adalah suatu aplikasi yang diharapkan dapat membantu dalam manajemen proses penjualan dan pembelian bahan bangunan”.

Menurut Ika Nur Indah (2013:124) “Sistem Informasi merupakan peranan sangat penting dalam kegiatan bisnis di suatu perusahaan atau wirausaha, jika masih menggunakan cara yang konvensional dalam mengolah data tentang transaksi penjualan maka dibutuhkan sistem informasi yang terkomputerisasi yang dapat menunjang kelancaran dalam melakukan transaksi penjualan.”.

Dari hasil pengamatan, dapat diuraikan mengenai permasalahan yang terjadi, yaitu toko yang memanfaatkan sistem penjualan konvensional (diharuskan datang langsung melakukan transaksi) membuat toko mengalami kesulitan dalam pencapaian target penjualan dan proses transaksi berjalan lambat.

Tujuan dari Sistem Informasi penjualan ini adalah membangun suatu Website Penjualan pada CV. Mega Karya Grup yang memiliki fungsi antara lain. Membangun aplikasi *E-Commerce* untuk membantu manajemen CV. Mega Karya Grup dalam melakukan transaksi jarak jauh. Membangun aplikasi transaksi *e-commerce* untuk membantu konsumen dalam melakukan pembelian di CV.

Mega Karya Grup tanpa datang langsung ke toko. Aplikasi ini telah memiliki berbagai fasilitas yang dapat mempermudah konsumen dalam melakukan transaksi (pembelian) seperti : Pembayaran dapat melalui transfer rekening. Mempermudah produsen dalam membuat laporan penjualan dan rekapitulasi laporan toko perbulannya sesuai dengan tahun yang di inginkan secara komputerisasi.