

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Web

Menurut (Winarno,2011:1) “*World Wide Web* (yang biasa disingkat WWW atau Web) adalah media informasi global yang bisa dipakai oleh usernya untuk saling bertukar informasi, dan sekarang bahkan fungsinya meluas, mulai dari sosialisasi hingga transaksi”.

Karena *web* sudah merupakan layanan *internet* paling terkenal, banyak orang yang beranggapan bahwa *web* itulah yang disebut *internet*. Padahal *web* hanya salah satu bagian dari *internet*. “*Internet* lebih dulu ada, sebelum *web*, tapi *internet* menjadi sangat terkenal setelah adanya *World Wide Web*” (Winarno,2011:1).

Dalam pembuatan Tugas Akhir ini diperlukan teori-teori yang relevan atau sejalan untuk mendukung kemudahan dalam mempelajari serta merancang *web* yang diharapkan dapat berfungsi secara maksimal dan mudah digunakan. Dengan demikian, program ini akan sangat membantu untuk mempermudah informasi yang dibutuhkan oleh pengguna.

A. Program

Menurut Raharjo (2008:21) “Program komputer tidak lain adalah suatu perangkat lunak (*Software*) yang digunakan untuk keperluan-keperluan aplikatif tertentu di berbagai bidang, baik di lingkungan perusahaan, pendidikan ataupun yang

lainnya”. Perangkat lunak tersebut sebenarnya merupakan suatu runtunan kode-kode program yang ditulis dengan salah satu bahasa pemrograman tertentu dan telah di kompilasi melalui kompilator yang sesuai dengan bahasa pemrograman tersebut. Dengan demikian, untuk membuat sebuah perangkat lunak atau yang lazim dikenal dengan sebutan ‘Program’, kita tentu harus memiliki keahlian dan menguasai salah satu bahasa pemrograman tertentu.

B. *Internet*

Menurut Irawan (2011:42) “*Internet* merupakan kependekan dari kata “*internetwork*”, yang berarti rangkaian komputer yang terhubung menjadi beberapa rangkaian jaringan”. Sistem komputer terhubung secara global dan menggunakan TCP/IP sebagai protokol.

Secara umum internet dapat diartikan sebagai pertukaran informasi dan komunikasi. Semua Informasi bisa didapat dengan mudah dan bebas di *internet* tanpa ada batasan.

Berawal tahun 1957, melalui *Advance Research Project Agency* (ARPA), Amerika Serikat bertekad mengembangkan jaringan komunikasi terintegrasi untuk keperluan militer. Hal ini dilatarbelakangi oleh terjadinya perang dingin antara Amerika Serikat dan Uni Soviet.

Awal kelahiran *internet* pun dimulai, berikut tiga perkembangan penting yang dianggap sebagai cikal-bakal lahirnya *internet* sekaligus sebagai tahapan paling bersejarah yang membuat teknologi jaringan dan *internet* menjadi berkembang seperti sekarang ini.

1. Perkembangan *Internet* pertama

Penemuan *packet switching* pada tahun 1960 oleh ARPA. *Packet switching* adalah pengiriman pesan yang dapat dipecah dalam paket-paket kecil yang masing-masing paketnya dapat melalui berbagai jalur alternatif. *Packet switching* juga memungkinkan jaringan dapat digunakan secara bersamaan untuk melakukan banyak koneksi.

2. Perkembangan *Internet* kedua

Salah satu perkembangan besar jaringan komputer yang dicatat dalam sejarah adalah pengembangan lapisan protokol jaringan yang paling banyak digunakan sekarang yaitu TCP/IP (*Transmission Control Protocol/Internet Protocol*). Protokol ini dikembangkan oleh Robert Khan dan Vinton Cerf pada tahun 1974. Jaringan ARPANET menjadi semakin besar sejak saat itu dan semakin banyak universitas bergabung dan mulailah perusahaan komersial masuk. Protokol TCP/IP menjadi protokol umum yang disepakati sehingga dapat saling berkomunikasi pada jaringan *internet*.



3. Perkembangan *Internet* ketiga

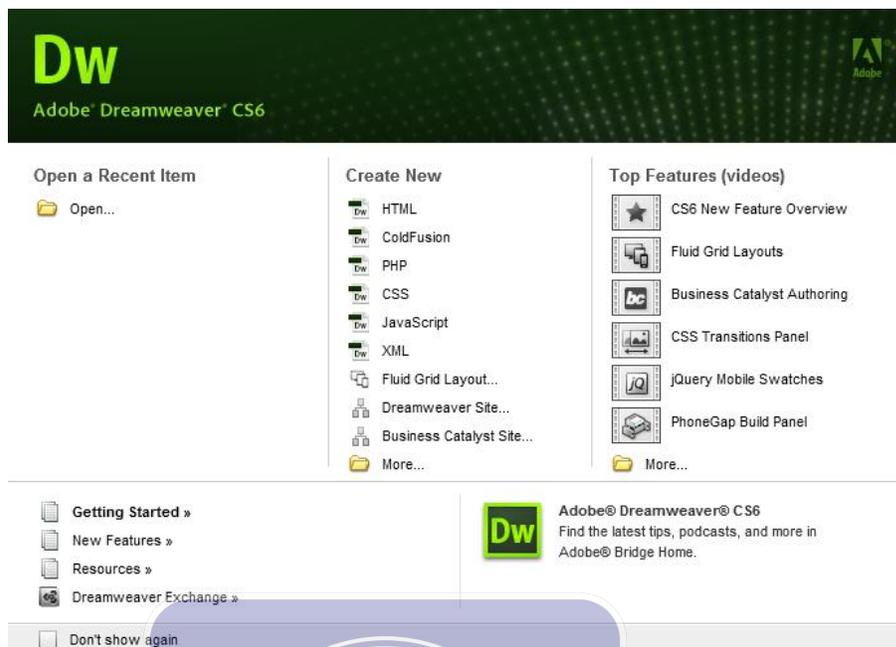
Peristiwa yang paling bersejarah, dan membawa perubahan besar dalam dunia *internet* adalah ketiga terbangunnya aplikasi *World Wide Web* (WWW) pada tahun 1990 oleh Tim Berners-Lee. WWW membuat semua pengguna dapat saling berbagi bermacam-macam aplikasi dan konten, serta saling mengaitkan materi-materi dan informasi yang tersebar di *internet*. Setelah lahir WWW, aplikasi dan konten *internet* pun semakin berkembang hingga saat ini.

C. Adobe Dreamweaver CS6

Menurut Agung (2012:129). “Adobe Dreamweaver adalah sebuah *software web design* yang menawarkan cara mendesain *website* dengan dua langkah sekaligus dalam satu waktu, yaitu mendesain dan melakukan *scripting*”. Salah satu kekuatan Adobe dreamweaver CS6 ini adalah kemampuannya mendukung pemrograman *script server side* seperti *Active Server Pages (ASP)*, *ColdFusion*, *Java Server Pages (JSP)* dan *PHP*. Selain itu, tentunya mendukung pemrograman *clientside* yang sangat terkenal dan dipakai orang, yakni *HTML* dan *Javascript*.

Banyak jenis *Web Editor* yang dapat mengolah *file PHP*, dari *shareware* sampai dengan yang *freeware* dan memiliki kemampuan untuk mendukung pemrograman *server side* dan *client side*. *Server side* digunakan untuk memproses data yang berhubungan dengan *server* seperti pengelola *database*, sedangkan *client side* merupakan bahan pemrograman tambahan.





Sumber: Agung (2012:129)

Gambar II.1.

Kotak Dialog *Starter Page*

Berikut adalah penjelasan menu pada *Starter Page*:

1. *Menu Open a Recent Item*

Menu ini menampilkan daftar nama dokumen yang pernah dibuka oleh Adobe Dreamweaver CS6, anda dapat membuka dokumen tersebut dengan cara memilih nama dokumen yang ada. Selain itu, Anda juga dapat membuka dokumen yang belum pernah dibuka dengan Adobe Dreamweaver CS6 dengan cara memilih *menu open*.

2. *Create New*

Menu ini digunakan untuk membuat dokumen *web* yang baru.

3. *Top Features (Videos)*

Top Features (Videos) merupakan fitur yang disediakan *Dreamweaver* melalui *internet*. Fitur tersebut antara lain *CS6 Features Overview*, *CSS Inspect Mode*, *CSS Enable/Disable*, *Dynamically Related Files*, *Live View Navigation*, *Browser Lab Integartion* dan sebagainya.

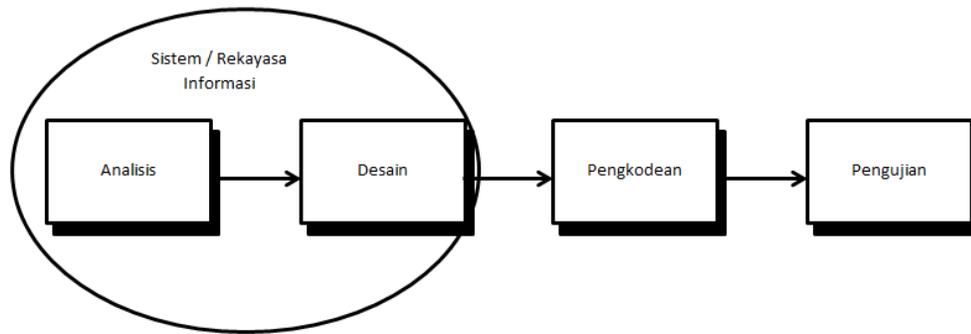
4. *Help Online*

Help Online merupakan menu pilihan untuk menampilkan fasilitas *help* secara *online* melalui *internet*. Anda hanya dapat menggunakan fitur *point 3* dan *4* jika komputer anda terhubung ke *internet*.

D. Model Pengembang Perangkat Lunak

Menurut Rosa dan Shalahuddin (2013:28) “Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sqquential linier*) atau alur hidup klasik (*classic life cycle*)”. Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*). Berikut adalah gambar model air terjun:





Sumber : Rosa dan Shalahuddin (2013:28)

Gambar II.2.

Ilustrasi Model *Waterfall*

1. Analisis kebutuhan perangkat lunak

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

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. 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. Pembuatan kode program

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

4. Pengujian

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

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

Tidak menutup kemungkinan tersebut perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat baru.

Model *waterfall* adalah model SDLC yang paling sederhana. Model ini banyak dan cocok untuk pengembang perangkat lunak dengan spesifikasi yang tidak berubah-ubah.

E. *Hypertext Preprocessor (PHP)*

PHP merupakan perangkat untuk membuat *web* yang dinamis, menurut Anhar (2013:3), “PHP yaitu bahasa pemrograman *web server-side* yang bersifat *open source*”. *Open source* yang berarti kita bisa menggunakannya tanpa perlu membelinya.

PHP memungkinkan pembangunan halaman *web* secara dinamis. Hal seperti ini tidak bisa dilakukan hanya dengan menggunakan HTML. Sebagai gambaran, berkas yang hanya berisi kode HTML tidak mendukung pembuatan aplikasi yang menghasilkan dokumen PDF dan tidak bisa berhubungan dengan *database*.

Seperti telah dijelaskan sebelumnya, PHP termasuk perangkat lunak yang bersifat *free* (gratis). Anda bisa mengunduhnya di *internet* melalui situs www.php.net. Namun perlu diketahui, PHP terkadang dikemas menjadi satu dalam perangkat lunak, misalnya WampServer dan XAMPP. Hal lain yang menarik adalah PHP bersifat *multiplatform*, artinya PHP dapat berjalan pada berbagai sistem seperti Windows, Linux dan UNIX.

Kode PHP dapat diletakan pada kode HTML dengan menggunakan tag.

Contoh :

```
<?php ?>
```

```
<html>
```

```
<head>
```

```
<title> Selamat Datang di HTML</title>
```

```
</head>
```

```
<body>
```

Ini adalah Isi menggunakan html


```
<?php
echo "Ini adalah isi menggunakan PHP";
?>
</body>
</html>
```

Pada contoh di atas, baris

```
<?php
echo "Ini adalah isi menggunakan PHP";
?>
```

adalah kode PHP, dalam prakteknya, kode PHP tidak selalu dikombinasikan dengan HTML.

F. MySQL

MySQL termasuk salah satu jenis *database server* yang sangat terkenal.

Menurut Ramadhan (2007:3) "MySQL merupakan sistem manajemen *database* yang bersifat relasional". MySQL juga mempunyai *query* atau bahasa SQL (*Structure Query Language*) yang *simple* dan menggunakan *escape character* yang sama dengan PHP.

MySQL mempunyai tampilan *client* yang mempermudah anda dalam mengakses *database* dengan kata sandi untuk memungkinkan proses yang boleh anda lakukan. Untuk Masuk ke dalam *database* disediakan *user default* yaitu root dengan *password* yang telah dibuat saat proses instalasi.



G. XAMPP

Menurut Nugroho (2013:1). “Xampp adalah paket web lengkap yang dapat anda pakai untuk belajar pemrograman web, khusus php dan mysql, paket ini di *download* dan legal”. XAMPP memudahkan para *web developer* untuk mengembangkan dan membuat sebuah *website* di *localPC/Laptop*, sehingga proses pembuatan sebuah *website* menjadi lebih aman dan cepat dibandingkan melakukan proses pembuatan *website* lewat *online server*.

Sebenarnya ada banyak sekali jenis *software* lain yang dapat menjalankan apache, MySQL, dan PHP. Namun karena proses instalasi dan pengembangan dari XAMPP lebih mudah, maka banyak *developer web* yang menyarankan untuk menggunakan XAMPP untuk menjalankan AMP (Apache, MySQL, PHP). Selain dari pada itu, XAMPP juga berjalan di *platform* (sistem operasi) besar, seperti Macintosh, Windows dan Linux (Ubuntu, Debian, CentOS).



H. Cascading Style Sheet (CSS)

Menurut Sya'ban (2013:37) “CSS (*Cascading Style Sheet*) adalah suatu bahasa yang dikhususkan untuk mengatur gaya atau layout sebuah halaman *web*”. CSS digunakan oleh pembuat halaman *web*, untuk mendefinisikan warna, huruf *layout* dan aspek-aspek presentasi dokumen lainnya. CSS memang didesain untuk memisahkan antara dokumen (yang ditulis menggunakan HTML atau bahasa lain yang sejenis) dengan bentuk presentasi dokumen (ditulis dalam CSS). Pemisahan ini memberikan keuntungan akan adanya peningkatan dalam aksesibilitas isi, menyediakan fleksibilitas lebih dan pengendalian terhadap spesifikasi karakteristik

bentuk presentasi, serta mereduksi kompleksitas dan perulangan-perulangan pada struktur isi.

I. Photoshop CS6

Menurut Dyannita (2009:10) "Photoshop adalah program berbasis *pixel* untuk mengolah hasil fotografi maupun menghasilkan lukisan yang diperoleh dari efek-efek khusus. Di samping kemampuannya sangat bagus untuk memanipulasi foto, program ini juga digunakan untuk mendesain grafis *web*".

Adobe Photoshop, atau bisa disebut Photoshop, adalah perangkat lunak editor citra buatan Adobe System yang dikhususkan untuk pengeditan foto/gambar dan pembuatan efek. Perangkat lunak ini banyak digunakan oleh fotografer digital dan perusahaan iklan sehingga dianggap sebagai pemimpin pasar (*market leader*) untuk perangkat lunak pengolah gambar/foto, dan bersama Adobe Acrobat, dianggap sebagai produk terbaik yang pernah diproduksi Adobe Systems. Versi kedelapan aplikasi ini disebut dengan nama Photoshop CS (*Creative Suite*), versi sembilan disebut Adobe Photoshop CS2, versi sepuluh disebut Adobe Photoshop CS3, versi kesebelas adalah Adobe Photoshop CS4, versi keduabelas adalah Adobe Photoshop CS5, versi ketigabelas adalah Adobe Photoshop CS6, dan versi yang terakhir adalah versi keempatbelas Adobe Photoshop CC.

J. *E-library*

Menurut Fahmi (2012:8) “*E-library* adalah sebuah sistem yang terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*), koleksi elektronik, staff pengola, pengguna, organisasi, mekanisme kerja, serta layanan dengan memanfaatkan berbagai jenis teknologi informasi”.

2.2. Teori Pendukung

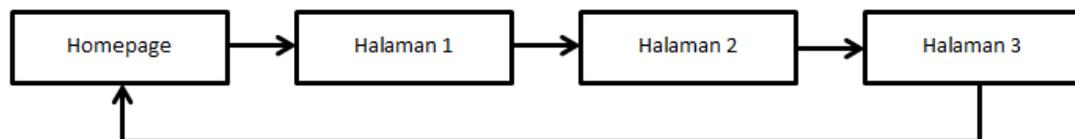
A. Struktur Navigasi

Menurut Sutopo (2007:6) “Dalam Pengembangan *web*, terdapat beberapa model navigasi dasar, yang harus dikenal dengan baik oleh *designer*, karena setiap model navigasi dapat memberikan solusi untuk kebutuhan yang berbeda”.

Macam-macam struktur navigasi antara lain :

1. *Linear Model*

Sebagian besar *website* mempergunakan *linear model*. Informasi di berikan secara sekuensial dimulai dari satu halaman. Beberapa *designer web* menggunakan satu halaman untuk masuk atau keluar dari *website*.



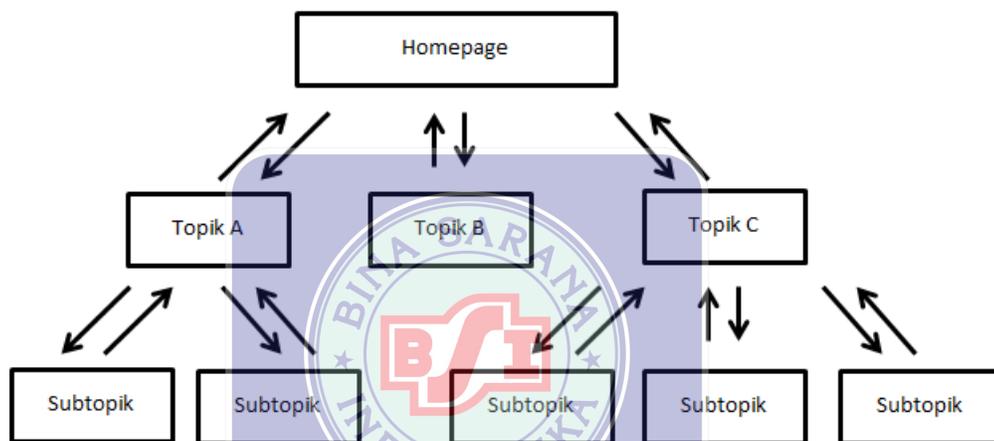
Sumber: Sutopo (2007:6)

Gambar II.3.

Struktur Navigasi *Linear*

2. *Hierarchical Model*

Hierarchical navigation model diadaptasi oleh *top-down design*. Konsep navigasi ini dimulai dari satu *node* yang menjadi *homepage*. Dari *homepage* dapat dibuat beberapa cabang. Apabila diperlukan, dari tiap halaman utama dapat dikembangkan menjadi beberapa cabang lagi. Hal ini seperti struktur organisasi didalam perusahaan.

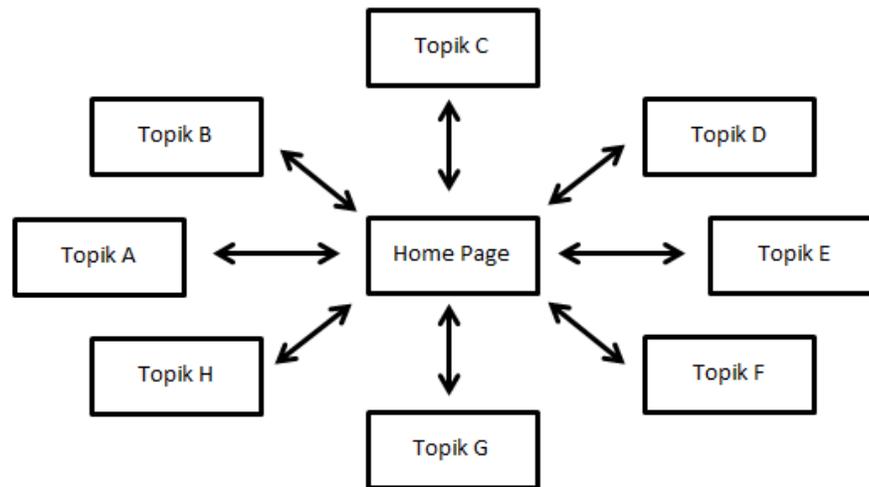


Sumber: Sutopo (2007:6)

Gambar II.4.
Struktur Navigasi *Hierarchical*

3. *Spoke-and-Hub Model*

Dalam *spoke-and-hub model* hanya ada dua macam *link* yaitu dari *homepage* ke halaman tertentu dan dari halaman tertentu ke halaman *homepage*. *Spoke-and-hub model* hanya menggunakan satu *node* untuk menghubungkan satu *node* ke *node* yang lain.

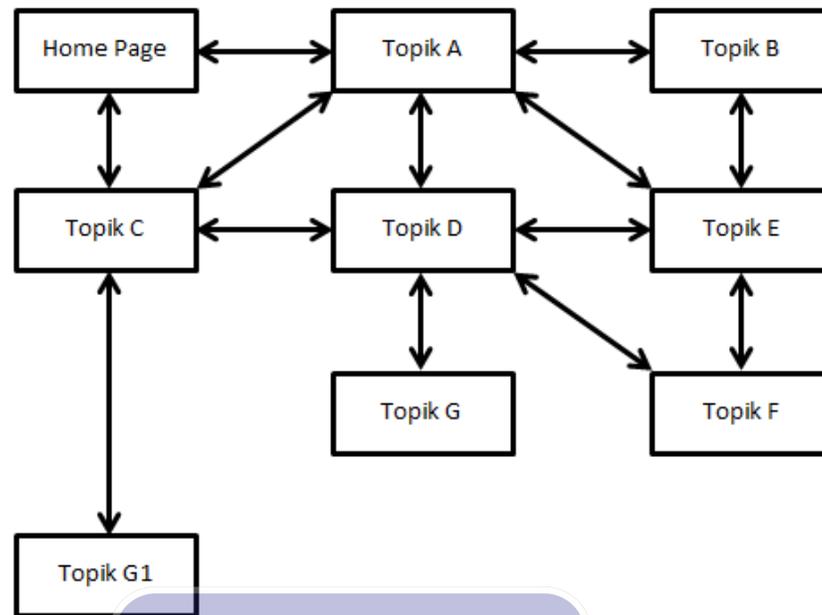


Sumber: Sutopo (2007:7)

Gambar II.5.
Struktur Navigasi Spoke and Hub

4. *Full Web Model*

Full web model memberikan kemampuan *hyperlink* yang banyak. *Full web model* banyak digunakan karena *user* dapat mengakses semua topik dan subtopik dengan cepat. Namun kelemahan dari model ini, yaitu dapat berakibat *user* kehilangan cara untuk kembali ke topik sebelumnya.



Sumber : Sutopo (2007:8)

Gambar II.6.
Struktur Navigasi Full Web

B. ERD (*Entity Relationship Diagram*)

1. Pengertian *Entity Relationship Diagram*

Menurut Rosa dan Shalahuddin (2013:50) “ERD atau *Entity Relationship Diagram* merupakan sebuah diagram yang digunakan untuk merancang hubungan antar tabel-tabel dalam basis data”. Berikut adalah yang sering digunakan pada ERD untuk mencanang tabel-tabel beserta relasinya dalam basis data:

ERD terbagi atas tiga komponen, yaitu entitas (*entity*), atribut (*attribute*), dan relasi atau hubungan (*relation*), secara garis besar entitas merupakan dasar yang

terlibat dalam sistem atribut atau *field* berperan sebagai penjelas dari entitas, dan relasi atau hubungan menunjukkan hubungan yang terjadi antara dua entitas.

a. Entitas (*entity*)

Entitas (*entity*) menunjukkan obyek-obyek dasar yang terkait didalam sistem. Obyek dasar dapat berupa orang, benda atau hal lain yang keterangannya perlu disimpan dalam basis data. Untuk menggambarkan entitas dilakukan dengan mengikuti aturan-aturan sebagai berikut:

- 1) Entitas dinyatakan dengan simbol persegi panjang.
- 2) Nama entitas berupa kata benda tunggal.
- 3) Nama entitas sedapat mungkin menggunakan nama yang dipahami dan menyatakan maknanya dengan jelas.

b. Atribut (*Attribute*)

Atribut sering juga disebut sebagai *property*, merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan sebagai basis data. Atribut berfungsi sebagai penjelas sebuah entitas untuk menggambarkan atribut yang dilakukan dengan mengikuti aturan sebagai berikut:

- 1) Atribut dinyatakan dengan simbol *ellipse*.
- 2) Nama atribut dituliskan dalam simbol *ellipse*.
- 3) Nama atribut berupa kata benda tunggal.
- 4) Nama atribut sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.
- 5) Atribut dihubungkan dengan entitas yang bersesuaian dengan menggunakan garis.



c. Relasi (*relation*)

Relasi atau hubungan yang kejadian atau transaksi yang terjadi diantara dua entitas yang keterangannya perlu disimpan dalam basis data. Aturan penggambaran relasi antar *entity* adalah:

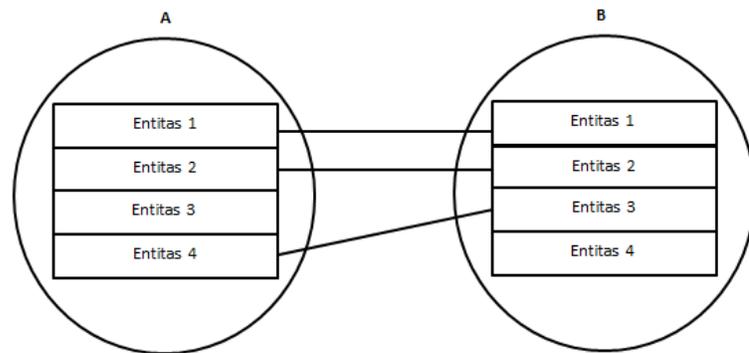
- 1) Relasi dinyatakan dengan simbol belah ketupat.
- 2) Nama relasi didalam simbol belah ketupat.
- 3) Relasi menghubungkan dua entitas.
- 4) Nama relasi menggunakan kata kerja aktif (diawali awalan me) tunggal.
- 5) Nama relasi sedapat mungkin menggunakan nama yang mudah dipahami dan dapat dinyatakan maknanya dengan jelas.

2. Derajat Relasi (Kardinalitas)

Menurut Fathansyah (2007:18) "Kardinalitas relasi merujuk kepada hubungan maksimum yang terdiri dari himpunan entitas yang satu ke himpunan entitas yang lain dan begitu juga sebaliknya". Kardinalitas relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dapat berupa:

a. Satu ke satu (*one to one*)

Berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, dan begitu juga sebaliknya setiap entitas pada himpunan entitas B berhubungan dengan paling banyak satu entitas himpunan B.



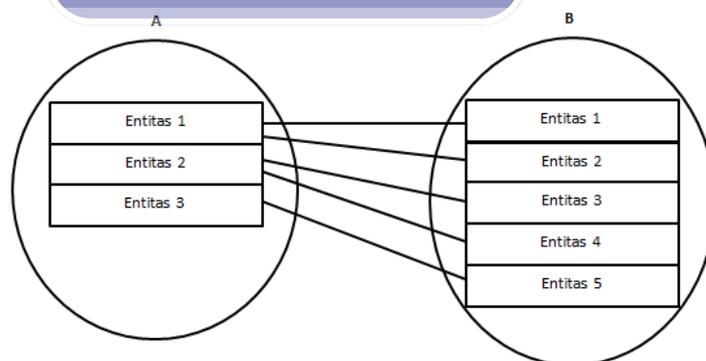
Sumber: Fathansyah (2007:77)

Gambar II.7.

Kardinalitas Satu ke satu

- b. Banyak ke satu (*many to one*)

Berarti setiap entitas pada himpunan entitas A dan dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.



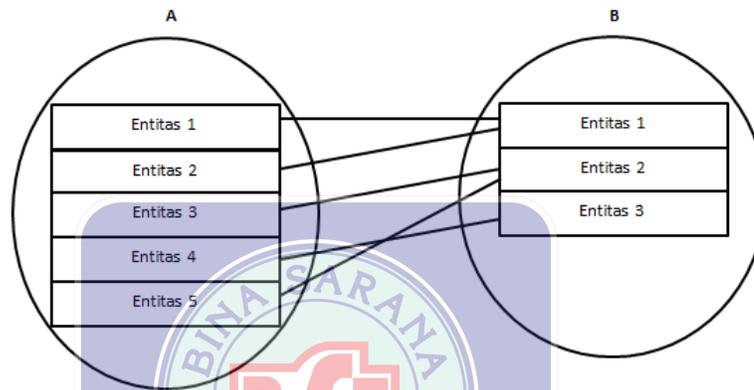
Sumber: Fathansyah (2007:78)

Gambar II.8.

Kardinalitas Satu ke banyak

c. Banyak ke satu (*many to one*)

Berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas himpunan entitas B.

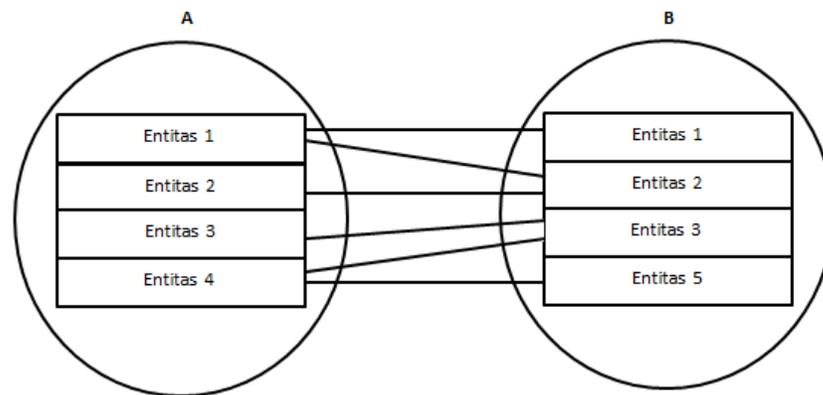


Sumber: Fathansyah (2007:78)

Gambar II.9.
Kardinalitas Banyak ke satu

d. Banyak ke banyak (*many to many*)

Berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, demikian juga sebaliknya, dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.



Sumber: Fathansyah (2007:79)

Gambar II.10.

Kardinalitas Banyak ke banyak

C. LRS (*Logical Record Structure*)

Menurut Ladjamuddin (2006:210) Aturan-aturan dalam melakukan transformasi ERD ke LRS adalah sebagai berikut:

1. Setiap *entity* akan diubah ke bentuk sebuah kotak dengan nama *entity* berbeda di luar kotak dan atribut berada didalam kotak.
2. Sebuah relasi kadang disatukan dalam kotak bersama *entity* kadang dipisah dalam sebuah kotak tersendiri.

D. Pengujian *Black-Box*

Menurut Ladjamuddin (2006:379) “Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan perancang perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program”.

Pengujian *black-box* bukan merupakan alternatif dari teknik *white-box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan dari pada metode *white-box*.

Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi-fungsi yang tidak benar atau hilang,
2. Kesalahan interface,
3. Kesalahan dalam struktur data atau akses *database* eksternal,
4. Kesalahan kinerja,
5. Inisialisasi dan kesalahan terminasi.

Tidak seperti pengujian *white-box* yang dilakukan pada saat awal proses pengujian, pengujian *black-box* cenderung diaplikasikan selama tahap akhir pengujian. Karena pengujian *black-box* memperhatikan struktur kontrol, maka perhatian berfokus pada *domain* informasi. Pengujian didesain untuk menjawab pertanyaan berikut :

1. Banyak validasi fungsional diuji?
2. Kelas *input* apa yang akan membuat *test case* menjadi baik ?
3. Apakah sistem sangat sensitif terhadap harga *input* tertentu ?
4. Bagaimana batasan dari suatu data diisolasi ?
5. Kecepatan data apa dan *volume* data apa yang dapat ditolerir oleh sistem ?
6. Apa pengaruh kombinasi tertentu dari data terhadap operasi sistem ?

Dengan mengaplikasikan teknik *black-box*, maka menarik serangkaian *test case* yang memenuhi kriteria berikut:

- (1) *test case* yang mengurangi dengan harga lebih dari satu, jumlah *test case* tambahan yang harus didesain untuk mencapai pengujian yang dapat dipertanggung jawabkan.
- (2) *test case* yang memberi tahu sesuatu mengenai kehadiran atau ketidakhadiran kelas kesalahan, daripada memberi tahu kesalahan yang berhubungan banyak dengan pengujian spesifik yang ada.

