

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Website

2.1.1. Website

Menurut Yuhefizar, (2009:2) ”*website* adalah keseluruhan halaman-halaman *web* yang terdapat dalam sebuah domain yang mengandung informasi”. Dapat kita simpulkan bahwa *website* atau sering juga dikenal sebagai *World Wide Web* ini adalah sebuah informasi yang terangkum dalam halaman-halaman *web*. Jenis-jenis *web* terbagi menjadi 2 hal yaitu statis dan dinamis. Menurut Yudhanto (2014:3) ”*Web* dikatakan Statis (*Static Website*) jika tampilan halaman yang di akses harus dilakukan dengan mengubah *script* langsung pada halaman tersebut.” Statis disini bukan diartikan sebagai halaman *web* yang selalu bergerak-gerak seperti layaknya animasi dan sebagainya, misalnya, *web* profile organisasi, dan lain-lain.

Menurut Yudhanto (2014:3) ”Suatu halaman *web* dikatakan dinamis jika perubahan tampilan halaman yang diakses dilakukan tanpa mengubah *script*, tetapi hanya menambah atau mengubah inputan pada halaman tersebut.” Contohnya seperti *Website* berita, www.detik.com. Adapun jenis-jenis *web* berdasarkan tujuannya yang terbagi menjadi 5 yaitu:

1. *Personal Web*, *website* ini berisikan sebuah informasi data pribadi seseorang.
2. *Corporate Web*, merupakan *website* yang dimiliki oleh perusahaan-perusahaan.

3. *Portal Web*, merupakan sebuah *website* yang mempunyai banyak layanan contohnya seperti layanan berita dan lain-lain.
4. *Forum Web*, merupakan sebuah *website* yang bertujuan sebagai media diskusi.
5. Di samping itu juga ada *Website e-Government, e-Banking, e-Payment, e-Procurement* dan lain-lain.

2.1.2 Internet

Menurut Hidayatullah (2014:1) bahwa “*Internet* adalah jaringan global yang menghubungkan komputer-komputer di seluruh dunia”. Sedangkan menurut Yuhefizar (2008:2) bahwa “*Internet* adalah rangkaian hubungan jaringan komputer yang dapat diakses secara umum diseluruh dunia, yang mengirimkan data dalam bentuk paket data berdasarkan standar *Internet Protocol (IP)*. Jadi dari kedua pendapat buku itu dapat disimpulkan bahwa internet adalah sebuah jaringan komputer yang dapat menghubungkan seluruh komputer di dunia.

2.1.3. Web Browser

“*Web browser* adalah aplikasi perangkat lunak yang memungkinkan penggunaanya untuk berinteraksi dengan teks, image, video, games dan informasi lainnya yang berlokasi pada halaman *web* pada *Word Wide Web (WWW)* atau *Local Area Network (LAN)*”. Menurut Limantara (2009:1). *Web browser* dapat menampilkan *email*, mengelola HTML, sebagai input dan menjadikan halaman *web* sebagai hasil output yang informatife. Dengan menggunakan *web browser*, para pengguna *internet* dapat mengakses berbagai informasi yang terdapat di internet

dengan mudah. Fungsi dari *web browser* adalah untuk menampilkan dan melakukan interaksi dengan dokumen-dokumen yang disediakan oleh *web server*. Beberapa *web browser* yang populer saat ini antara lain :

1. *Mozilla Firefox*

Menurut *Smith (2012:30)* “*Firefox* atau juga biasa dikenal dengan nama *Mozilla Firefox* adalah *web browser* yang dikembangkan oleh *Mozilla*.” Sampai saat ini *firefox* adalah *web browser* yang paling populer dan juga handal selain itu *firefox* dapat menjelajah *internet* dengan cepat dan ringan. Banyak sekali *addons/plugins* atau *extention* tambahan yang disediakan oleh *firefox* untuk meningkat kemampuan *browsing* di internet. Dengan fitur yang lengkap tersebutlah *Firefox* menjadi *Web Browser* terpopuler.

2. *Safari*

Menurut *Oneto (2009:24)* “ *Safari* adalah *browser* yang dibuat oleh *Apple Inc* yang ditunjukkan untuk pengguna sistem operasi *Mac Os (Macintos)*”. *Browser* ini bisa digunakan pada sistem operasi *windows* dan *browser* ini merupakan *browser* penjelajah tercepat diantara *browser-browser* lainnya.

3. *Opera*

Menurut *Oneto (2009:23)* “*Browser Opera* adalah salah satu *browser* terbaik dari segi *interface*, sekuriti dan kecepatannya”. Kini *browser opera* menjadi *browser* tercepat dalam versi *mobile*. Walau terkadang penggunaan *browser Opera* pada *desktop/pc* sering dipandang sebelah mata, namun pada rilis terbarunya *opera* dapat mengalahkan *firefox* dalam segi kecepatan menjelajahnya.

4. *Google Chrome*

Menurut Oneto dan Yosep (2009:24) “*Google Chrome* merupakan *browser* buatan Google yang diluncurkan pada 2 September 2008.” Walau *Google Chrome* baru saja dibuat namun *Google Chrome* telah bisa menyaingi *web browser* lainnya. Hal ini dikarenakan *Google Chrome* adalah *Web Browser* yang dibuat oleh penguasa internet saat ini yaitu *Google*.

2.1.4. *Web Server*

Menurut Sadeli (2014:2) “*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.” Penggunaan *web server* dalam mendukung penetrasi layanan teknologi informasi sangat besar, terutama dalam menyampaikan informasi kepada pengguna. Banyak sekali *vendor-vendor* penyedia *webserver* mulai dari yang berbayar maupun gratis (*opensource*), dengan masing-masing menawarkan kelebihan yang dimilikinya. *Vendor-vendor* tersebut seperti *Internet Information Services (IIS)* yang merupakan kepemilikan *Microsoft*, *Apache*, *Nginx* dan *Google Web Server*. Namun yang populer digunakan adalah *Apache* dan *IIS*.

a. *Apache Web Server*

Menurut Sadeli (2014:2) “*Apache (Server HTTP Apache atau Server Web/WWW Apache)* adalah *web server* yang dapat dijalankan di banyak sistem

operasi (Unix, BSD, Linux, Micosoft dan Novell Netware serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs *web*”. *Apache* memiliki tingkat kehandalan dan kestabilan yang tinggi serta fitur-fitur yang terbilang lengkap, hal ini yang membuat banyak pemakai mempercayakan *apache* sebagai *web server* mereka. *Apache web server* dapat berjalan pada sistem operasi berbasis *Windows* atau *Unix*.

b. Internet Information Services (IIS)

Menurut Sadeli (2014:3) “IIS (Internet Information Service) adalah sebuah HTTP *web server* yang digunakan dalam sistem operasi server *Windows*, mulai dari *Windows NT 4.0 Server*, *Windows 2000 Server* atau *Windows Server 2003*”. *Web server* buatan *Microsoft* ini Berbeda dengan *Apache* yang *open source*, untuk mendapatkan dukungan teknis penuh maka otomatis kita sebagai pengguna harus membeli lisensi IIS dari *Microsoft*. IIS menawarkan sistem jaringan antar muka yang mudah, aman dan stabil. Sesuai dengan perusahaan dimana IIS dikembangkan, maka IIS hanya bisa berjalan pada sistem operasi berbasis *Windows*.

2.2. Bahasa Pemrograman

2.2.1. PHP (*Personal Home Page*)

PHP adalah bahasa pemrograman yang ditunjukan untuk kepentingan pembuatan aplikasi *web*. Menurut Kadir (2013:17), “PHP merupakan bahasa skrip yang ditempatkan dalam *server* dan diproses diserver.” PHP sendiri mempunyai fungsi untuk menterjemahkan script/kode PHP yang kita buat. PHP juga merupakan

merupakan aplikasi *web* yang dinamis, dinamis disini adalah dapat membuat halaman *web* yang dapat dikendalikan oleh data. Dengan PHP ini kita dapat membuat beragam aplikasi *web*,mulai dari halaman *web* yang sederhana sampai dengan aplikasi kompleks yang membutuhkan koneksi ke database. Sampai saat ini telah banyak database yang didukung oleh PHP dan kemungkinan akan terus bertambah. Database tersebut adalah Base, DBM, mSQL, SQL, ODBC, Oracle,Postgres,Sybase,Velocity, HTML.

2.2.2. HTML (*Hypertext Markup Language*)

Menurut Sutarman (2013:73) “*Hypertext Markup Language* (HTML) merupakan salah satu varian dari SGML (*Standard Generalized Markup Language*), yaitu sebuah standar dari ISO (*International Organization Standardization*) untuk pertukaran dokumen secara elektronik.” HTML sendiri secara formal diumumkan sebagai RFC 1866. HTTP digunakan sebagai protokol dalam pertukaran dokumen. Dalam dokumen *web* , dokumen ini akan dieksekusi oleh *browser*, sehingga *browser* mampu menghasilkan suatu dokumen sesuai dengan keinginan yang mendesain *page*. Dokumen ini mempunyai kemampuan menampilkan gambar, suara, teks, maupun penyediaan link terhadap halaman *web* lainnya, baik dengan alamat yang sama maupun berbeda.

Hypertext Markup Language (HTML) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi di dalam sebuah penjelajah *Web Internet* dan *formatting hypertext* sederhana yang ditulis ke dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yg

terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan ke dalam format ASCII normal sehingga menjadi *homepage* dengan perintah-perintah HTML. HTML saat ini merupakan standar internet yang di definisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C). HTML dibuat oleh kerjasama antara Caillau TIM dengan Berners-lee robert di CERN pada tahun 1989 (CERN adalah Lembaga Penelitian Fisika Energi Tinggi di Jenewa).

2.2.3. JQuery

Menurut Mulhim (2014:2) di dalam bukunya mengatakan bahwa “ *jQuery* adalah *library javascript* yang menyederhanakan *document traversing*, *event handling*, *animating*, dan *AJAX interactions* untuk pengembangan *web* secara cepat”. *jQuery* mampu berjalan disemua *browser-cross browser*. *jQuery* diluncurkan Januari 2006 oleh *John Resig*. *jQuery* merupakan *library* yang paling populer saat ini, karena kecanggihannya *jQuery* banyak digunakan oleh perusahaan besar seperti *Google*, *Dell*, *CBS*, *digg*, *Netflix*, *Bank of America*, *Mozilla*, *Drupal* dan sebagainya.

2.2.4. CSS (Cascading Style Sheet)

Menurut Hidayatullah (2014:53) “dengan CSS kita bisa menyimpan format dan menggunakannya kapan pun dan di mana pun kita inginkan”. Dengan CSS kita bisa membuat efek-efek tertentu untuk konten *web* yang kita buat. Contohnya saja kita ingin bagian header kita pada *web* yang kita buat selalu diberi *font verdana*, ukuran 20 dan berwarna biru. Maka kita definisikan *style* tersebut untuk tag <TH>.

CSS (*Cascading Style Sheet*) sudah didukung oleh hampir semua *web browser* karena CSS telah di standarkan oleh *Word Wide Web Consortium (W3C)*. Jadi CSS ini merupakan pilihan yang tepat untuk kita memformat halaman *web* kita agar terlihat indah dan cantik dimanapun *user* membuka *web* kita.

2.3. Basis Data

2.3.1. Definisi Basis Data

Menurut Hidayatullah (2014:147). “Basis data dapat didefinisikan sebagai himpunan kelompok data yang saling berhubungan yang diorganisasikan.” Jadi dapat disimpulkan bahwa 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.

2.3.2. DBMS (*Data Base Management System*)

Menurut Nugroho (2008:15) “*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 ke dalam basis data. Optimalisasi dan eksekusi (*optimization and execution*) Permintaan *Data Manipulation Language* (DML) harus diproses di komponen pengoptimalisasi yang bertujuan untuk menentukan cara yang efisien untuk implementasi permintaan. Permintaan yang telah dioptimalisasi kemudian dieksekusi di bawah kendali *reltime* manager.

3. 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).

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

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

5. 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 di dalam kamus data.

6. Kinerja (*performance*)

DBMS harus dapat mengerjakan semua tugas seefektif mungkin.

2.3.3. Xampp

Menurut Wicaksono (2008:7) ”*Xampp* adalah sebuah software yang berfungsi untuk menjalankan *website* berbasis *PHP* dan menggunakan pengolahan data *MySQL* di komputer lokal” dan menurut ANHAR (2010:45) ”*MySQL (My Structure Query Language)* adalah salah satu *Database Management System (DBMS)* dari sekian banyak *DBMS* seperti *Oracle, MS SQL, Postagre SQL*, dan lainnya”. Kepopuleran *MySQL* dimungkinkan karena kemudahannya untuk digunakan, cepat secara kinerja *query*, dan mencukupi untuk kebutuhan *database* perusahaan-perusahaan skala menengah-kecil. Pada *MySQL*, sebuah *database* mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom. Ada beberapa keunggulan *MySQL* diantaranya yaitu :

- a) *MySQL* merupakan program yang *multi-threaded*, sehingga dapat dipasang pada *server* yang memiliki *multi-CPU*.
- b) Didukung program-program umum seperti *C++, Java, Perl, PHP, Python, TCLAPIs*, dll.
- c) Bekerja pada berbagai *platform* (tersedia berbagai versi untuk berbagai sistem operasi).
- d) Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem *database*.
- e) Memiliki sistem sekuriti yang cukup baik dengan *verifikasi host*.

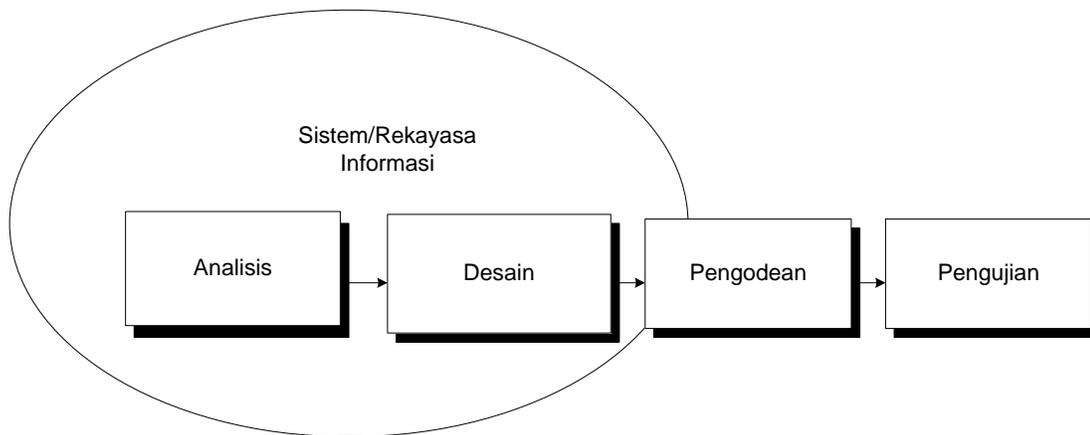
- f) Mendukung ODBC untuk sistem operasi *Microsoft Windows*.
- g) Mendukung *Record* yang memiliki kolom dengan panjang tetap atau panjang bervariasi.

2.4. Model Pengembangan Perangkat Lunak

Teknik pengembangan perangkat lunak menggunakan metode *incremental* yang sudah dikembangkan dari *waterfall model*, karena metode *incremental* ini terdiri dari tahap-tahap yang memberikan kemudahan, jika pada suatu tahap tidak sesuai atau mengalami kesalahan maka dapat kembali ketahap sebelumnya. Dengan berkembangnya perangkat lunak, pembuat program (*programmer*) secara umum langsung pembuatan pengkodean perangkat lunak, tanpa menggunakan prosedur perangkat lunak. SDLC (*Software Development Life Cycle*) diperkenalkan pada tahun 1960-an, SDLC pertama digunakan oleh para pengusaha besar jaman itu. Dimana sistem-sistem yang dibangun untuk mengelolah informasi kegiatan dan aktivitas yang berpotensi memiliki data yang besar.

Model *Waterfall* adalah suatu proses pengembangan perangkat lunak yang berurutan, di mana kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian. Model ini sering juga disebut sebagai model alur hidup klasik (*classic life cycle*). Menurut Sukamto dan Shalahudin (2013 : 28) “Model Air terjun (*waterfall*) 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: Sukamto dan Shalahuddin (2013:29)

Gambar II.1
ilustrasi model waterfall

a. Analisis kebutuhan perangkat lunak

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

b. 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.

c. Pembuatan 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.

d. Pengujian

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.

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

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan biasa 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 lunak baru.

2.5. Teori Pendukung

Dalam pembuatan suatu program, berhasil tidaknya suatu program tergantung dari persiapan sebelumnya, misalnya membuat *Struktur Navigasi* atau *ERD* dan *Blackbox Testing*. Untuk itu, dalam proses pembuatan program kali ini penulis membutuhkan beberapa peralatan pendukung (*tools system*) yang membantu

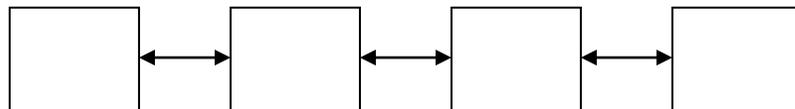
mempermudah penulis dalam membuat suatu program. Adapun peralatan pendukung tersebut terdiri dari :

2.5.1. Struktur Navigasi

Menurut Sutisna (2007:26) “Struktur navigasi merupakan struktur dari sebuah program, dan merupakan bagian yang paling penting di dalam pembuatan multimedia”. Struktur navigasi terbagi menjadi empat, yaitu:

1. Struktur Navigasi *Linear*

Merupakan struktur yang hanya mempunyai satu rangkaian cerita berturut. Pada umumnya dibuat untuk halaman *web* presentasi karena tidak terlalu menuntut keinteraksian tetapi perlu keindahan tampilan dan kemudahan menampilkan data sebagai informasi. Dalam struktur ini tidak boleh ada percabangan.

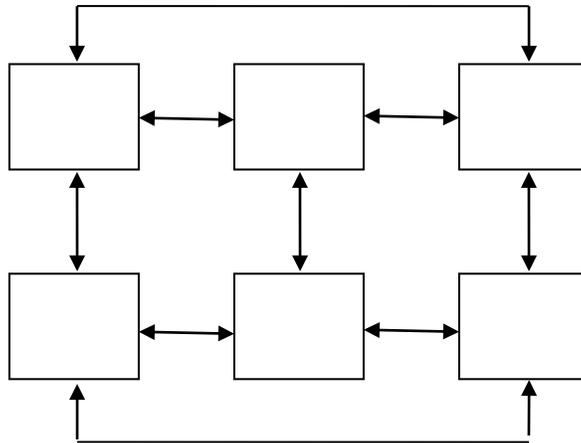


Sumber : Sutisna (2007:29)

Gambar II.2
Struktur Navigasi *Linear*

2. Struktur Navigasi *Non Linear*

Merupakan pengembangan dari struktur navigasi linear. Pada struktur ini diperbolehkan mempunyai cabang, tapi tiap cabang mempunyai kedudukan yang sama. Tidak ada *master page* dan *slave page*.

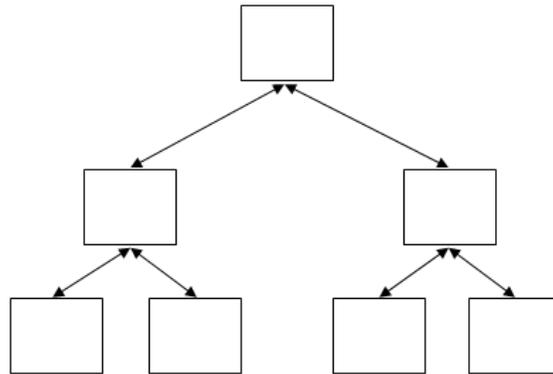


Sumber :Sutisna (2007:31)

Gambar II.3
Struktur Navigasi Non Linear

3. Struktur Navigasi *Hirarki*

Merupakan struktur navigasi yang memerlukan percabangan untuk menampilkan data berdasarkan bagian tertentu. Tampil pada menu pertama disebut *master page*, tampilan pertama dicabangkan lagi dengan nama *slave page*.

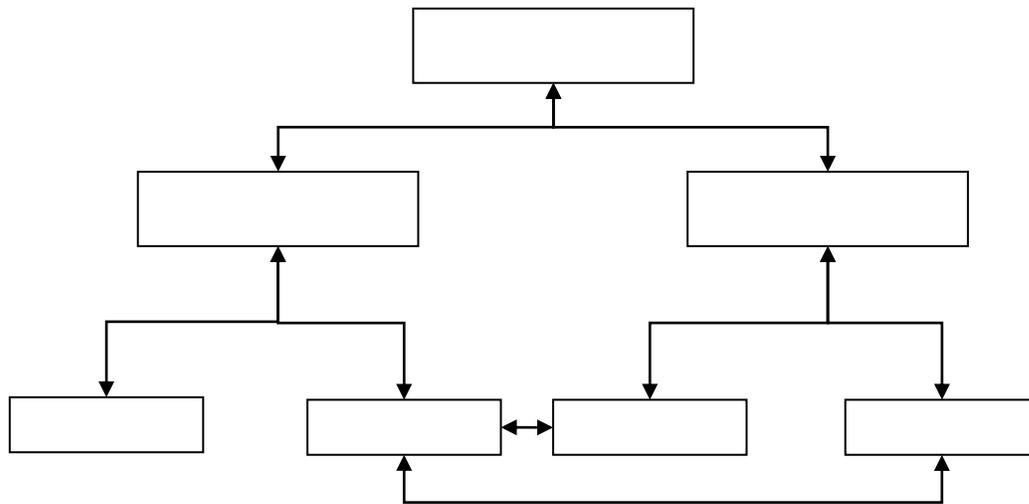


Sumber : Sutisna (2007:33)

Gambar II.4
Struktur Navigasi Hirarki

4. Struktur Navigasi Campuran

Struktur navigasi campuran (*composite*) merupakan gabungan dari struktur sebelumnya dan disebut juga struktur navigasi bebas, maksudnya adalah jika suatu tampilan membutuhkan percabangan maka dibuat percabangan. Struktur ini paling banyak digunakan dalam pembuatan aplikasi multimedia.



Sumber : Sutisna (2007:35)

Gambar I1.5.
Gambar Struktur Navigasi *Composite*

2.5.2. *Enterprise Relationship Diagram*

Menurut Sukamto (2013:50) “ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika”. ERD biasanya memiliki hubungan *binary* (satu relasi menghubungkan dua buah entitas). ERD memiliki beberapa aliran notasi seperti notasi Chen yang dikembangkan oleh Peter Chen, Barker yang dikembangkan oleh Richard Barker, Ian Palmer, dan Harry Ellis, notasi Crow’s Foot, dan beberapa notasi lainnya. Sedangkan menurut Menurut Loonam (2010:17) “sebuah *Entity-Relationship Model* (ERM) merupakan abstrak dan konseptual representasi data. *Entity-Relationship* adalah salah satu metode pemodelan basis data yang digunakan untuk menghasilkan skema konseptual untuk jenis/model data semantik sistem.” Dimana sistem seringkali memiliki

basis data relasional, dan *top-down*. Diagram untuk menggambarkan *Entiti Relationship* ini disebut *Entitiy - Relationship diagram*, *ER diagram* atau *ERD*.

2.5.3. Komponen ERD

Komponen dalam *Entity Relationship Diagram* adalah sebagai berikut:

a. Entitas

Adalah segala sesuatu yang dapat digambarkan oleh data. Entitas juga dapat diartikan sebagai individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain. Ada dua macam entitas yaitu entitas kuat dan entitas lemah. Entitas kuat merupakan entitas yang tidak memiliki ketergantungan dengan entitas lainnya. Contohnya entitas anggota. Sedangkan entitas lemah merupakan entitas yang kemunculannya tergantung pada keberadaan entitas lain dalam suatu relasi. Entitas digambarkan dengan persegi panjang.

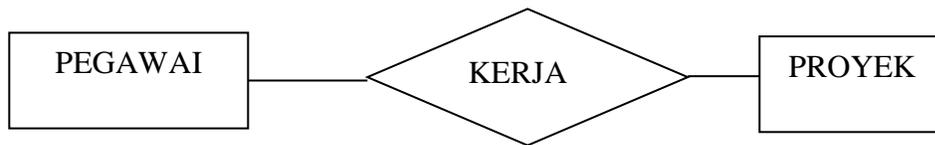
b. Atribut

Atribut merupakan pendeskripsian karakteristik dari entitas. Atribut digambarkan dalam bentuk lingkaran atau *elips*. Atribut yang menjadi kunci entitas atau *key* diberi garis bawah.

c. *Relationship*

Relasi menunjukkan adanya hubungan diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Dalam hal ini digambarkan dengan garis lurus.

Contoh:



Sumber Loonam (2010:20)

Gambar II.6.
Contoh *Relationship*

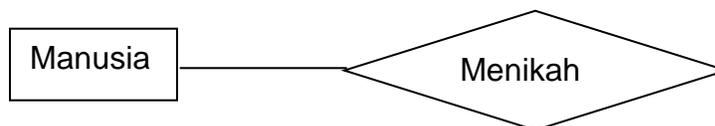
2.5.4. Derajat *Relationship*

Derajat *relationship* yang menjelaskan jumlah *entity* yang berpartisipasi dalam suatu *relationship*. Dalam Derajat *Relationship* terbagi dari tiga tingkatan yakni :

a. *Unary Degree* (Derajat Satu)

Adalah satu buah *relationship* menghubungkan satu buah *entity*.

Contoh :



Sumber: Loonam (2010:21)

Gambar II.7.
Contoh *Unary Degree*

Keterangan :

Manusia menikah dengan manusia, *relationship* menikah hanya menghubungkan *entity* manusia.

b. *Binary Degree* (Derajat Dua)

Adalah satu buah *relationship* yang menghubungkan dua buah *entity*.

Contoh :



Sumber: Loonam (2010:22)

Gambar II.8.
Contoh *Binary Degree*

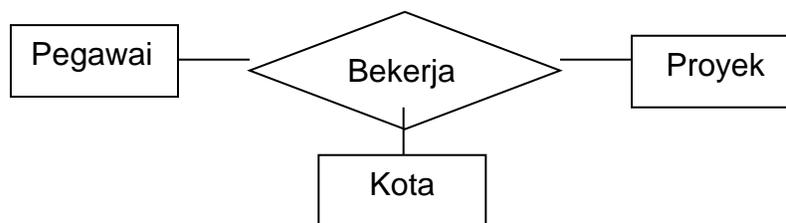
Keterangan :

Pegawai memiliki kendaraan, sebuah *relationship* memiliki menghubungkan *entity* Pegawai dan *entity* Kendaraan.

c. *Ternary Degree* (Derajat Tiga)

Adalah satu buah *relationship* menghubungkan tiga buah *entity*.

Contoh :



Sumber: Loonam (2010:24)

Gambar II.9.
Contoh *Ternary Degree*

Keterangan :

Pegawai pada kota tertentu mempunyai suatu Proyek.

Entity Bekerja menghubungkan *Entity* Pegawai, Proyek dan Kota

2.5.5. Cardinality Ratio Constraint

Menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain.

Jenis-jenis *Cardinality* :

a. One To One (1 : 1)

Yaitu perbandingan antara *entity* pertama dengan *entity* kedua berbanding satu berbanding satu.

Contoh :



Sumber: Loonam (2010:25)

Gambar II.10.
Contoh *One To One*

b. One To Many (1 : M)

Yaitu perbandingan antara *entity* pertama dengan *entity* kedua berbanding satu berbanding banyak.

Contoh :



Sumber: Loonam (2010:26)

Gambar II.11
Contoh *One To Many*

c. Many To One (M : 1)

Yaitu perbandingan antara *entity* pertama dengan *entity* kedua berbanding banyak berbanding satu.

Contoh :



Sumber: Loonam (2010:26)

Gambar II.12
Contoh *Many to One*

d. Many To Many (M : M)

Yaitu perbandingan antara *entity* pertama dengan *entity* kedua berbanding banyak berbanding banyak.

Contoh :



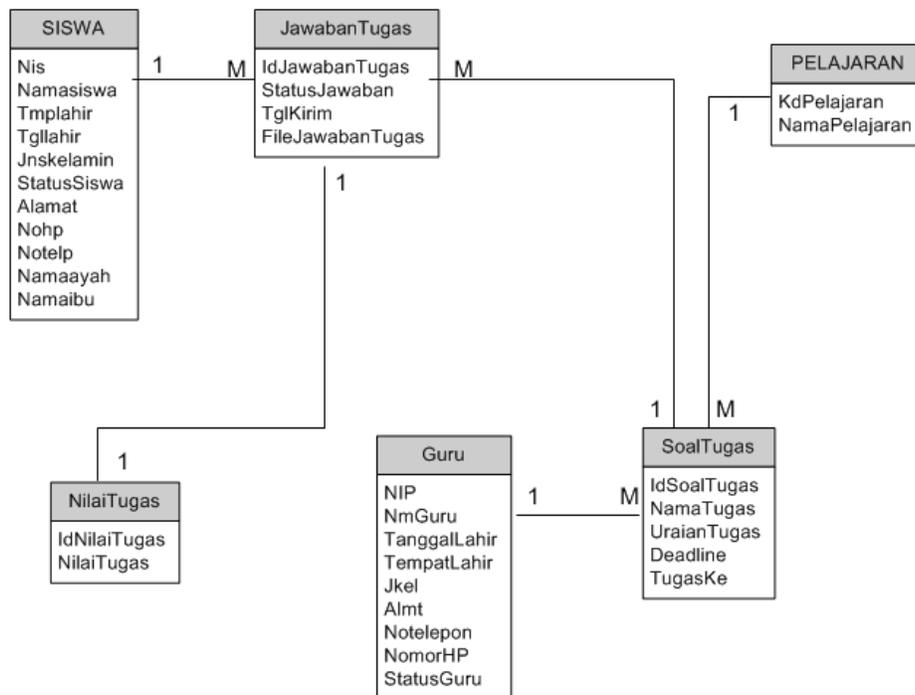
Sumber: Loonam (2010:27)

Gambar II.13
Contoh *Many To Many*

2.5.6. LRS (*Logical Record Structure*)

Menurut Purnomo (2008:15) “pengertian LRS adalah representasi dari struktur record-record pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas yang digunakan untuk menentukan kardinalitas, jumlah table dan *Foreign Key (FK)*.”

Berikut adalah contoh gambar LRS :



Sumber: Purnomo (2008:15)

Gambar II.14
Contoh Gambar LRS

2.5.7. Pengujian Web

Menurut Pressman (2010:95), “pengujian *Black Box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak.” Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dieksekusi pada perangkat lunak dan kemudian keluar dari perangkat lunak dicek apakah telah sesuai yang diharapkan.

Pengujian *Black Box* berusaha menemukan kesalahan dalam kategori :

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