

## BAB II

### LANDASAN TEORI

#### 2.1 Konsep Dasar Web

##### 1. Internet

Menurut Irawan (2011:2) “*Internet* merupakan kependekan dari kata *Internetwork* yang berarti rangkaian komputer yang terhubung menjadi beberapa rangkaian jaringan”. Sistem computer terhubung secara global dan menggunakan TCP/IP (*Transmission Control Protocol/Internet Protocol*) sebagai protocol. Secara umum *internet* dapat di artikan sebagai pertukaran informasi dan komunikasi. Semua informasi bias didapatkan dengan mudah dan bebas di *internet* tanpa ada batasan. Berawal tahun 1957 , melalui (*Advanced Reseach Projects Agency*) ARPA, Amerika Serikat bertekat mengembangkan jaringan komunikasi terintegrasi untuk keperluan militer. Hal ini dilator belakangi oleh terjadinya perang dingin antara Amerika Serikat dengan Uni Soviet. Awal kelahiran *internet* pun dimulai, berikutnya 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 alternative jalur. *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 protocol jaringan yang paling banyak digunakan sekarang yaitu TCP/IP (*Transmission Control Protocol/Internet Protocol*). Protokol adalah suatu kumpulan aturan untuk berhubungan antar jaringan. Protokol ini dikembangkan oleh Robert Kahn 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 protocol umum yang disepakati sehingga dapat saling berkomunikasi pada jaringan *internet*.

## 3) Perkembangan *Internet* Ketiga

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

### 1. *Web Browser*

Menurut Dhanta (2008:55) "*Browser* atau lengkapnya *web browser*, yaitu sebuah program yang dipergunakan untuk menjelajahi *internet* dan menggali informasi yang ada di dalamnya". Istilah saat ini *web browser* yang paling banyak

digunakan adalah *Internet Explorer* yang merupakan salah satu fasilitas yang tersedia pada *Microsoft Windows*. Apabila komputer sudah menggunakan *Microsoft Windows*, maka tidak perlu lagi membeli atau memasang program *Internet Explorer*, karena pada saat *Windows* di instal, maka secara otomatis program *web browser* akan ikut terinstal di komputer. Saat ini juga dikenal program-program *browser* lainnya diluar paket *Windows*, seperti *Netscape*, *Opera* dan *Mozilla Firefox*. Namun secara umum sebenarnya masing-masing program *web browser* ini memiliki fasilitas utama yang sama.

## 2. *Web Server*

Menurut Mulhim (2013:2) "*Web server* adalah perangkat lunak yang berfungsi menerima permintaan (*request*) berupa halaman *web* melalui HTTP atau HTTPS dari klien yang dikenal dengan *browser web* dan mengirimkan kembali dokumen HTML". Jenis-jenis *web server* yang banyak tersedia saat ini yaitu:

### 1) *Apache Web Server*

*Apache Web Server* adalah *server web* yang dapat dijalankan dibanyak system operasi (Unix, BSD, Linux, Microsoft Windows, dan Novell Netware serta *platform* lainnya) yang berguna untuk melayani dan memfungsikan situs *web*. Protokol yang digunakan untuk melayani fasilitas *web* atau WWW ini menggunakan HTTP. *Apache* merupakan perangkat lunak *open source* dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation*.

## 2) IIS (*Internet Information Service*)

IIS atau *Internet Information Service* adalah sebuah HTTP *web server* yang digunakan dalam system operasi *server Windows*, mulai dari *Windows NT 4.0 Server*, *Windows 2000 Server*, atau *Windows Server 2003*. Layanan ini merupakan layanan terintegrasi dalam *Windows 2000 Server*, *Windows 2003 Server* atau sebagai *add-on* dalam *windows NT 4.0*. Layanan ini berfungsi sebagai pendukung protocol TCP/IP yang berjalan dalam lapisan aplikasi (*application layer*).

## 3) Apache Tomcat

*Apache Tomcat* adalah sebuah *open source web server* yang dikembangkan oleh *Apache Software Foundation (ASF)*. *Tomcat* mengimplementasikan Java Servlet dan juga Java *Server Page (JSP)* spesifikasi dari *Oracle Corporation*, dan menyediakan Java HTTP *web server* yang bisa dijalankan dengan kode Java.

## 2. MySQL

Menurut Kadir (2008:34), “MySQL adalah salah satu jenis *database server* yang sangat terkenal”. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahan dasar untuk mengakses *databasenya*. Selain itu ia bersifat *open source* (Anda tidak perlu membayar untuk menggunakannya) pada berbagai *platform*, (kecuali untuk jenis *enterprise*, yang bersifat komersial). Perangkat MySQL sendiri bisa di download dari <http://www.mysql.com>. MySQL termasuk jenis RDBMS (*Relational Database Manajemen System*). Itulah sebabnya, istilah seperti tabel, baris dan kolom digunakan pada MySQL, semua *database* mengandung satu atau beberapa kolom.

Salah satu *database server* yang cukup terkenal saat ini adalah MySQL.

Adapun keunggulan dari MySQL adalah :

1. Mampu menangani jutaan *user* dalam waktu bersamaan.
2. Mampu menampung lebih dari 50.000.000 *record*.
3. Sangat cepat mengeksekusi perintah.
4. MySQL dapat berjalan di berbagai *platform*, antara lain *Linux*, *windows* dan lain sebagainya.
5. Memiliki *user privilege system* yang mudah dan efisien.

### 3. 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)*, *ASP NET*, *Cold Fusion*, *Java Server Pages (JSP)* dan *PHP*. Selain itu, tentunya mendukung pemrograman *client side* yang sangat terkenal dan dipakai orang, yakni *HTML* dan *JavaScript*

### 4. Xampp

Menurut Aditya (2011:16), *XAMPP* adalah “perangkat lunak bebas, yang mendukung banyak system operasi, merupakan kompilasi dari beberapa program”. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang

ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat system operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis.

### 5. PhpMyAdmin

Menurut Zaki (2008:97) menjelaskan "PhpMyAdmin adalah MySQL *client* yang berupa aplikasi web dan umumnya tersedia di server PHP XAMPP maupun server komersial lainnya". Cara untuk mengakses PhpMyAdmin di web server lokal, anda dapat mengakses PhpMyAdmin menggunakan URL <http://Localhost/PhpMyAdmin>.

### 6. Personal Home Page (PHP)

Menurut Hirin dan Virgi (2011:25) "(*Hypertext Preprocessor*)PHP adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *web* dan dapat ditanamkan pada sebuah skrip HTML". Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, Java, dan Perl serta mudah untuk dipelajari. PHP diciptakan untuk mempermudah pengembangan *web* dalam menulis halaman *web* dinamis dengan cepat, bahkan lebih dari itu dan dapat mengeksplorasi hal-hal yang luar biasa dengan PHP.

Awal mulanya PHP adalah kependekan dari *Personal Home Page* yang dibuat pada tahun 1995 oleh Rasmus Ledorf. Saat itu namanya masih *Form Interpreted*. Pada selanjutnya pembuat PHP merilis kode sumber ke khalayak umum (*open*

*source*) sehingga dengan demikian banyak *programmer* tertarik untuk mengembangkan PHP. Akhirnya pada November 1997 dirilis PHP 2.0 pada versi ini PHP sudah mulai diimplementasikan dalam program C, serta telah disertakan *module-module* tambahan atau dalam PHP sering disebut dengan ekstensi. Pada tahun 1997 juga ada andil sebuah perusahaan bernama Zend, dimana interpreter PHP ditulis ulang menjadi lebih bersih, cepat dan lebih baik. Dan akhirnya pada pertengahan tahun 1998, Zend merilis PHP 3.0 dengan digantinya singkatan dari *Personal Home Page* menjadi *Hypertext Preprocessor*. Pada pengembangan selanjutnya Zend terus memegang peranan penting dalam perkembangan PHP, pada pertengahan tahun 1999 PHP 4.0 dirilis. Pada versi ini lah mulai banyak orang tertarik memakai PHP karena kemampuannya untuk membangun aplikasi *web* kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi. Seiring dengan perkembangan zaman, banyak bahasa pemrograman mulai menerapkan model OOP (*Object Oriented Programming*), tak mau ketinggalan PHP versi 5.0 pun dirilis pada pertengahan tahun 2004 dengan kemampuan barunya yaitu pemrograman berorientasi objek.

Untuk menjalankan *script* PHP pada *website*, ada tiga komponen yang dibutuhkan, yaitu:

- 1) *Web server*, berfungsi untuk mengatur atau memberikan tempat untuk mengeksekusi PHP.
- 2) *Database server*, berfungsi untuk mengelola *database*.

3) *PHP Engine* atau *PHP Paster* adalah program yang digunakan untuk menterjemahkan *script* PHP, kemudian mengeksekusinya.

Selain ketiga komponen diatas, ada beberapa komponen tambahan yang diperlukan sebelum memulai program dalam PHP, yaitu *PHP Editor* yang berfungsi untuk menulis *Script-script* PHP dan *browser* yang berfungsi untuk melihat tampilan-tampilan *script* PHP yang telah di eksekusi.

## 7. HTML

Menurut Anhar (2010:40) “HTML (*Hyper Text Markup Language*) adalah sekumpulan simbol-simbol atau *tag-tag* yang dituliskan dalam sebuah *file* yang digunakan untuk menampilkan halaman pada *web browser*”. Tag-tag HTML selalu diawali dengan `<x>` dan di akhiri dengan `</x>`. Program yang digunakan untuk membuat dokumen HTML menggunakan *HTML Editor*. Ada banyak HTML Editor yang bisa digunakan, di antaranya: *Note Pad*, *Ms. Frontpage*, dan *Dreamwaver*.

## 8. CSS (*Cascading Style Sheet*)

Menurut Andi (2009:1) “CSS adalah salah satu fasilitas yang diberikan untuk pemograman HTML sehingga pengaturan atau *desain* atau tampilan *web-page* menjadi lebih baik”. Dengan atribut CSS, *user* memiliki kemampuan untuk mengatur posisi secara *absolute*, mengubah warna, mengubah besar *font*, mengubah *margin*, dan lain sebagainya.

## 9. *Waterfall*

Menurut Soekamto dan M Shalahuddin (2013:28) “Model (*System Development Life Cycle*) SDLC air terjun (*waterfall*) sering disebut juga model skuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*)”. Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara skuensial atau terurut mulai dari analisis, desain, pengodean, pengujian, tahap pendukung (*support*). Berikut adalah penjelasan dari model air terjun (*waterfall*):

### 1) Analisis kebutuhan perangkat lunak

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

### 2) Desain

Desain perangkat lunak adalah proses multi langkah yang focus 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 di implementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

### 3) Pembuatan kode program

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

### 4) Pengujian

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

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

Tidak menutup kemungkinan sebuah 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 lunak baru.

Dari kenyataan yang terjadi sangat jarang model air terjun dapat dilakukan sesuai alurnya karena sebab berikut:

- 1) Perubahan spesifikasi perangkat lunak terjadi di tengah alur pengembangan.

- 2) Sangat sulit bagi pelanggan untuk mendefinisikan semua spesifikasi dari awal alur pengembangan. Pelanggan sering kali butuh contoh (*prototype*) untuk menjabarkan spesifikasi kebutuhan sistem lebih lanjut.

## 2.2 Teori Pendukung

### 1. Struktur Navigasi

Menurut Prihatna (2005:51) “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 di pengaruhi oleh tujuan dari situs *web* yang akan dibuat.

Struktur navigasi adalah struktur atau alur dari sebuah program yang termasuk unsur penting didalam pembuatan aplikasi halaman *website*. Struktur navigasi ini merupakan perancangan hubungan dan rantai kerja dari beberapa lokasi yang berbeda dari halaman-halaman *website*. Selain itu struktur navigasi memberikan kemudahan dalam menganalisa keinteraktifan seluruh objek didalam halaman *website*. Struktur pembuatan halaman *website* terdiri dari *linier*, *non linier*, hirarki dan campuran.

#### 1. *Linier* (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 di tampilkan 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.

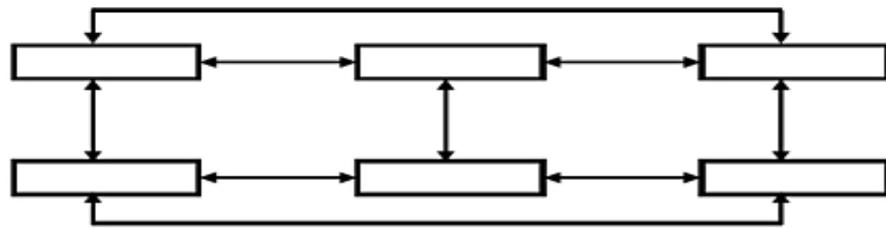


Sumber: Prihatna (2005:51)

**Gambar II.1. Struktur Navigasi *Linier***

## 2. *Non Linier* (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 dimana 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*.

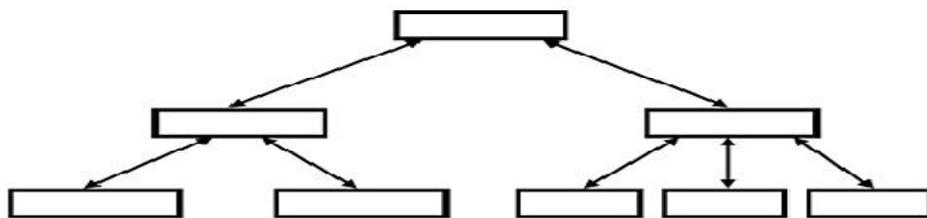


Sumber: Prihatna (2005:51)

**Gambar II.2. Struktur Navigasi *Non – Linier***

### 3. *Hierarchical* (Hirarki)

Struktur *Hierarchi* (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*.



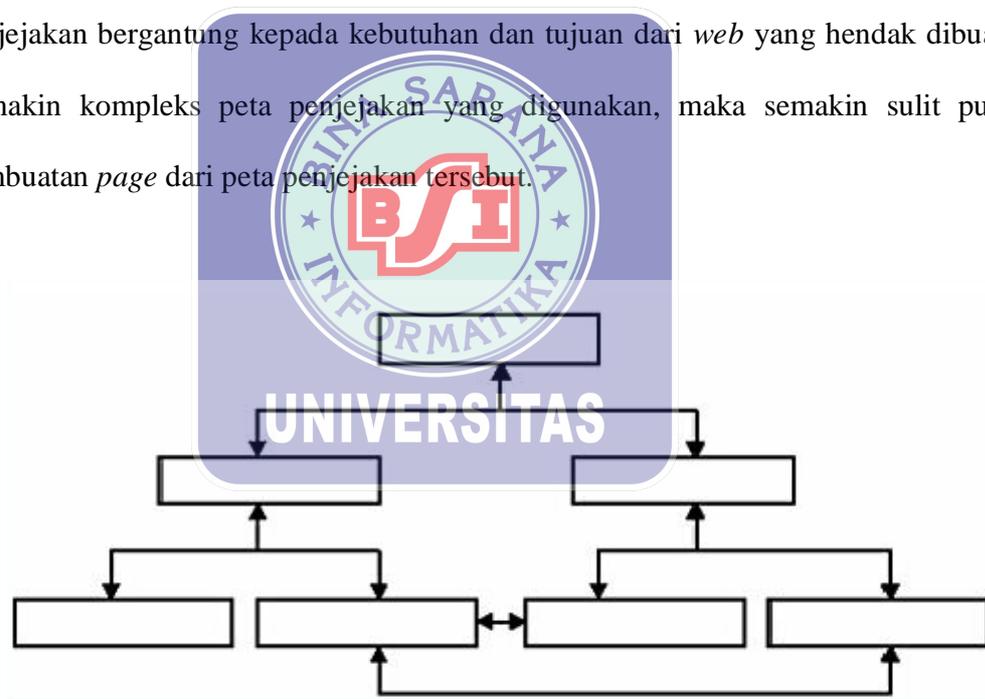
Sumber: Prihatna (2005:51)

**Gambar II.3. Struktur Navigasi *Hierarchical***

#### 4. *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 di bahas 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 (2005:51)

**Gambar II.4. Struktur Navigasi *Composite***

## 2. ERD (*Enterprise Relationship Diagram*)

Menurut Indrajani (2011:18) *Entity Relation (ER) Modeling* adalah sebuah perancangan basis data yang dimulai dengan mengidentifikasi data-data terpenting yang disebut dengan *entitas* dan hubungan antara *entitas-entitas* tersebut yang digambarkan dalam suatu model. Terdapat lima macam komponen utama ERD Menurut Indrajani (2011:18), yakni :

### 1. *Entity* (entitas)

Entitas adalah konsep dasar dalam pemodelan basis data berupa individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Kumpulan individu entitas disebut sebagai himpunan entitas.

Dalam ERD entitas dilambungkan dengan simbol kotak, entitas berupa :

- a. Objek yang dapat diidentifikasi
- b. Bisa dibedakan dari objek lain
- c. Mempunyai karakteristik tertentu, dimana informasi yang berkaitan dengan objek yang bersangkutan dikumpulkan.

### 2. *Attribute* (atribut)

Atribut adalah karakteristik atau properti dari entitas yang menyajikan penjelasan detail tentang entitas tersebut. Atribut bisa bersifat wajib (harus ada), atau opsional (bisa ada, bisa pula tidak). Atribut dilambungkan dengan simbol *elips*.

### 3. *Unique Identifier*

Sesuai dengan namanya, *unique identifier* adalah atribut yang bernilai berbeda untuk masing-masing individu dalam himpunan entitas. Untuk membedakannya dengan atribut-atribut biasa, dalam gambar ERD, *unique identifier* ditandai dengan garis bawah.

### 4. *Relationship* (hubungan antar entitas)

Kelebihan utama basis data dibandingkan penyimpanan dalam *file* terletak pada kemampuan untuk saling berinteraksi antar data.

### 5. Kardinalitas

*Relationship* secara lengkap tidak hanya dilihat dari nama relasinya, tetapi juga jumlah kejadian di entitas yang terlibat dalam relasi tersebut. Hubungan ERD yang paling umum pada *binary relationship*, diantaranya :

1. *One To One* : Hubungan antara *file* satu dengan *file* yang kedua adalah satu berbanding satu.
2. *One To Many* : Hubungan antara *file* yang satu dengan *file* yang kedua adalah satu berbanding banyak atau banyak berbanding satu.
3. *Many To Many* : Hubungan antara *file* yang satu dengan *file* yang kedua adalah banyak berbanding banyak.

Menurut Soekamto dan M. Shalahudin (2013:275) "*Black Box Testing* yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan

kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan”. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses *login* maka kasus uji yang dibuat adalah:

1. Jika *user* memasukan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
2. Jika *user* memasukan nama pemakai (*username*) dan kata sandi (*password*) yang salah, mislanya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.

