

## BAB II

### LANDASAN TEORI

#### 2.1 Konsep Dasar Sistem Manajemen Database

Sebuah sistem manajemen basis data relasional atau dalam bahasa Inggrisnya dikenal sebagai *Relational Database Management System (RDBMS)* adalah sebuah program komputer (atau secara lebih tipikal adalah seperangkat program komputer) yang didisain untuk mengatur/memanajemen sebuah basisdata sebagai sekumpulan data yang disimpan secara terstruktur, dan melakukan operasi-operasi atas data atas permintaan penggunaannya. Contoh penggunaan DBMS ada banyak sekali dan dalam berbagai bidang kerja, misalnya akuntansi, manajemen sumber daya manusia, dan lain sebagainya. Meskipun pada awalnya DBMS hanya dimiliki oleh perusahaan-perusahaan berskala besar yang memiliki perangkat komputer yang sesuai dengan spesifikasi standar yang dibutuhkan (pada saat itu standar yang diminta dapat dikatakan sangat tinggi) untuk mendukung jumlah data yang besar, saat ini implementasinya sudah sangat banyak dan adaptatif dengan kebutuhan spesifikasi data yang rasional sehingga dapat dimiliki dan diimplementasikan oleh segala kalangan sebagai bagian dari investasi perusahaan.

Data merupakan sekumpulan fakta yang disusun, kelompokkan, dikumpulkan, disimpan, dan kemudian diproses oleh suatu sistem manajemen database, sedangkan sistem manajemen database merupakan sekumpulan data

yang telah disusun dan diproses sehingga memiliki makna. Database dapat berupa dokumen, atau laporan. Merupakan kumpulan data dari perangkat keras(*Hardware*) dan perangkat lunak(*Software*) serta perangkat manusia(*Brainware*) yang akan mengolah data.

### 2.1.1. Konsep Dasar Sistem Informasi Berbasis Web

#### 1. Konsep Dasar Sistem Informasi Berbasis Web Dengan Aplikasi PHP Dan MySQL

Sistem Informasi adalah seperangkat komponen yang saling berhubungan yang berfungsi mengumpulkan, memproses, menyimpan dan mendistribusikan informasi untuk mendukung pembuatan keputusan dan pengawasan dalam organisasi. Web atau WWW (*Word Wide Web*) adalah sebuah metode baru yang berjalan didunia internet yang berkembang dengan cepat, dengan media ini dapat menciptakan puluhan bahkan ratusan aplikasi yang berjalan di bawah Web (*Under Web*). PHP adalah salah satu aplikasi program yang biasa digunakan dalam media internet saat ini. Databasenya adalah MySQL yaitu *database server* yang dapat berjalan didalam media online sehingga *database* ini mudah dimanage oleh penggunanya. (Daniel, 2006).

#### 2. Basis Data

##### a. Pengertian Basis Data

Istilah basis data banyak menimbulkan interpretasi yang berbeda.

Basis Data adalah suatu susunan/kumpulan data operasional lengkap

dari suatu organisasi atau perusahaan yang terorganisir atau dikelola

dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakainya. Sedangkan sistem basis data adalah suatu sistem penyusunan dan mengelola record-record menggunakan komputer untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah organisasi atau perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses pengambilan keputusan (Sofie:2007).

b. *Relational Data Base Management Sistem (RDBMS)*

a. Pengertian RDBMS

Kumpulan file yang saling berkaitan untuk program dan pengelolaannya disebut DBMS (Marlinda, 2004). DBMS adalah istilah yang penting dalam pengelolaan data. Untuk membuat dan mengelola data tersebut, dibutuhkan *software* yang diistilahkan DBMS (*Data Base Management Sistem*). Tugas-tugas yang diemban *software* DBMS ini adalah membuat *database*, menampilkan data yang ada pada *database* tersebut, memodifikasi data pada *database* tersebut, menghasilkan laporan sesuai dengan data yang ada dalam *database*, dan mengamankan data dari pihak-pihak yang tidak berkepentingan (Lindaka, 2003). Beberapa contoh *software* DBMS yang banyak beredar adalah *Oracle*, *Microsoft SQL Server*, *SyBase* dan *MySQL*.

Hampir semua program DBMS merupakan RDBMS (*Relational DataBase Management Sistem*), dimana data yang akan diorganisir dalam sekumpulan tabel yang saling berelasi (berhubungan) (Lindaka, 2003).

b. Konsep *Database Relasional*

Arsitektur Sistem Basis Data dibagi menjadi tiga tingkatan (Nugraha, 2001), yaitu:

a) *internal level*

Disebut juga *internal view*, yaitu tingkatan yang basis datanya secara fisik ditulis atau disimpan di media *storage* atau *level* yang berkaitan erat dengan tempat penyimpanan. Contohnya cara penyimpanan data dalam disk.

Tempat penyimpanan dapat digolongkan sebagai berikut.

(1) *Device Model*, bagaimana penyimpanan datanya.

(2) *Storage Model*, bagaimana data disimpan di dalam tempat penyimpanan data dan hubungannya terhadap data pengaksesannya.

b) *external level*

disebut juga dengan *individual user views*, yaitu tingkatan data di basis data dilihat berdasarkan kebutuhan masing-masing aplikasi di *user* atau *level* yang berkaitan erat dengan para pemakai. Contohnya cara seorang pemakai menggunakan data

c) *conceptual level*

disebut juga *community user view*, yaitu tingkat dari aplikasi yang berbeda digabungkan, sehingga menggunakan basis data secara keseluruhan dengan menyembunyikan penyimpanan data secara fisik atau *level* yang merupakan penghubung dari *internal level* dan *external level*. Hal inilah yang mendasari *Relational Data Base Management Sistem*. Contohnya cara penggunaan data oleh sekelompok pemakai.

Kelemahan sistem pemrosesan berkas dapat diatasi dengan sistem basis data ini, dengan keuntungan yang diperoleh dalam pemrosesan data ini adalah:

- a) kemubadziran data akan berkurang,
- b) integritas data,
- c) independensi data,
- d) konsistensi data,
- e) berbagi data,
- f) keamanan data,
- g) lebih mudah dalam penggunaan dan pencarian data.

RDBMS merupakan antar muka bagi pemakai dalam mengorganisasikan *database* yang disusun, pemakai dapat berinteraksi langsung dengan mudah dan praktis dengan menggunakan perintah-perintah yang sederhana yang dibuat dalam suatu bahasa pemrograman

Tujuan RDBMS adalah untuk mempermudah penciptaan penyusunan data dan membebaskan pemrograman dari masalah penyusunan *file* yang kacau dan berantakan.

#### c. Tabel

Tabel merupakan salah satu bagian dari *database*. Tabel akan menampung data yang tersimpan dalam bentuk baris dan kolom. Setiap kolom mempunyai nama yang unik (tidak mungkin kembar), dan dapat mempunyai jenis yang berbeda. Baris pada tabel berisi data yang sesuai dengan jenis *fields*-nya. Dalam suatu tabel kita membutuhkan satu *field* kunci. *Field* kunci adalah sebuah *file* yang unik, dan tidak ada yang sama pada tabel tersebut. Nantinya pada tabel *posting*, tidak perlu mencatat nama, *password* dan *level* dari *record* yang bersangkutan (Daniel, 2006). *Field* kunci digunakan juga untuk menghubungkan antara tabel satu dengan tabel yang lain.

#### d. Relasi Antar Tabel

Dalam sistem *database* ini, data tersimpan dalam beberapa jaringan yang berbeda yang terdistribusi melalui bermacam-macam media yang berbeda. *Database* ini berisi sekumpulan *form*, tiap *form* dapat berinteraksi saat mengeksekusi data pada satu *form* atau secara bersamaan pada beberapa *form*. Setiap *form* dapat memproses eksekusi data lokal dengan data yang telah ditentukan

Sebuah *form* juga dapat mengambil bagian dalam melakukan akses terhadap data pada beberapa *form* yang berbeda.

Macam-macam bentuk topologi distribusi data.

a) *Fully Connected Network*

Keuntungan dari tipe ini adalah jika salah satu rusak maka yang lain masih dapat berjalan, tetapi tidak ada jaminan kerahasiaan data. Karena seluruh bagian dapat melakukan pengaksesan data.

b) *Partially Connected Network*

*Reliability* pada system ini rendah, karena sistem pengaksesan hanya melewati satu jalur dan tidak ada jalur lain. Akan tetapi biaya dapat ditekan. Jaminan kontrol tidak dijamin

c) *Tree Structure Network*

Sistem bersifat sentral, jika pusat rusak maka semua bagian tidak dapat melakukan pengaksesan data.

d) *Ring Network*

Sistem melingkar seperti ini hanya dapat mengakses data pada sistem yang langsung berhubungan, maka sistem kontrol sendiri tidak terjamin.

e) *Star Network*

Pada sistem ini kontrol manajemen data terjamin karena bersifat sentral dengan biaya yang rendah. Tetapi jika pusat

mengalami kerusakan maka data tidak bisa diakses.  
(Sofie,2005)

#### e. Hirarki Data

Data diorganisirkan ke dalam suatu hirarki yang terdiri atas:

##### a) elemen data

Elemen data merupakan satuan data terkecil yang tidak dapat dipecah lagi ke dalam unit lain.

##### b) rekaman

merupakan gabungan dari sejumlah elemen data yang saling terkait. Dalam sistem basis data relational, rekaman biasa disebut dengan tupel atau baris.

##### c) berkas

himpunan dari semua rekaman yang bertipe sama, atau dapat juga dikatakan sebagai kumpulan rekaman data dalam satu objek. Berkas mewakili komponen yang disebut relasi.

#### f. Normalisasi Data

Normalisasi adalah pemrosesan pengelompokan atribut-atribut dan relasi sehingga membentuk struktur relasi antar tabel. Normalisasi merupakan proses pengelompokan elemen-elemen data menjadi suatu tabel-tabel untuk menunjukkan entity dan relasinya. Teknik ini ditemukan pada tahun 1970 oleh E. F. Codd.

*Well-Structure relation* adalah sebuah relasi dengan jumlah kerangkapan data yang sedikit (*Minimum Amount of Redundancy*)

serta memberikan kemungkinan bagi *user* untuk melakukan *insert*, *delete* dan *modify* terhadap baris-baris data pada *relasi* tersebut, yang berakibat tidak terjadi kesalahan atau inkonsistensi data, yang disebabkan oleh operasi-operasi tersebut (Sofie,2005).

## 2.2 Perangkat Analisa Sistem

Perangkat Analisa Sistem adalah suatu fase dimana diperlukan suatu keahlian perancang untuk elemen-elemen komputer yang akan menggunakan sistem baru. Ada beberapa alat bantu yang digunakan dalam desain sistem yaitu :

### 2.2.1 Data Flow Diagram (DFD)

*DFD* merupakan salah satu alat pemodelan proses dari sistem yang paling sering digunakan. *DFD* adalah representasi dari sebuah sistem secara grafis digambarkan dengan sebuah simbol tertentu untuk menunjukkan perpindahan data dalam proses suatu sistem (Bentley,L, 2004). Meskipun diberi nama *DFD*, namun penekanan pada *DFD* lebih pada prosesnya. Dalam hal ini *DFD* menunjukkan perpindahan dan perubahan data dalam suatu sistem. Simbol-simbol yang digunakan dalam *DFD* hanya terdiri dari empat macam yaitu arus data, proses, *entity* (*external entity*), dan simpanan data. Berikut uraian mengenai empat macam simbol yang ada :

#### a. Arus Data atau *Data Flow*

adalah aliran yang menunjukkan perpindahan data dari satu bagian ke bagian yang lain dalam sistem. *Data flow* dalam *DFD* disimbolkan dengan

anak panah dan diberi nama atau keterangan di sampingnya yang

menunjukkan data apa yang mengalir. Contoh arus data dapat dilihat pada gambar 2.1.



Gambar 2.1 Simbol arus data

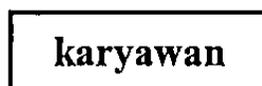
a. **Proses**

adalah simbol yang mengubah suatu data dari suatu bentuk menjadi bentuk yang lain. Atau dengan kata lain, proses menerima *input* data dan mengeluarkan *output* data lain yang telah diproses. Gambar 2.2. merupakan contoh dari simbol proses.



Gambar 2.2 simbol proses

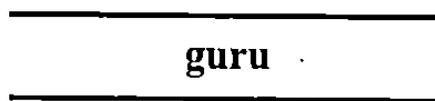
c. **Entity** adalah tempat penyimpanan data dalam suatu sistem, baik secara manual maupun secara elektronik. Simpanan data digunakan jika suatu proses perlu menggunakan data tersebut lagi. Contoh simpan data dapat dilihat pada gambar 2.3.



Gambar 2.3 simbol *entity*

d. **Simpanan data** adalah seseorang, sekelompok orang, sebuah departemen di dalam maupun di luar organisasi, atau sebuah sistem lain yang memberikan

input untuk sistem yang ada atau menerima output dari sistem yang ada. *External entity* juga disebut *terminator*, karena merupakan batas dari sebuah sistem. Dalam *DFD entity* disimbolkan dengan sebuah kotak persegi panjang. Contoh arus data *entity* dapat dilihat pada gambar 2.4.



Gambar 2.4 simbol simpanan data

*DFD* yang menunjukkan batas-batas dari sebuah sistem informasi adalah *Context Diagram*, yaitu *top-level view* dari sebuah sistem. Dalam menggambar *Context Diagram*, hanya dibuat satu proses saja yang menggambarkan keseluruhan proses dari sistem tersebut dan beberapa *entity* di sekelilingnya yang berhubungan (Kasiman, 2006).

## 2.2.2 *Entity Relationship Diagram*

### 2.2.2.1 Pengertian dan fungsi

*Entity Relationship Diagram* merupakan peralatan pembuatan model data yang paling fleksibel, dapat diadaptasi untuk berbagai pendekatan yang mungkin diikuti dalam pengembangan sistem. *Entity Relationship Diagram* merupakan diagram yang sering dipakai dalam proses pendokumentasian data dengan mengidentifikasi jenis entitas dan hubungannya (Dittman, K.C. (2004).

Dengan menggunakan *ERD* dapat dilihat dengan jelas hubungan antar *file* dalam *database*. Melalui *ERD* ini diharapkan programmer dapat menentukan seperti

apakah program yang akan dibuat. Hal ini sangat membantu dalam merevisi program atau dalam pengembangan program tersebut nantinya.

#### 2.2.2.2 Simbol yang digunakan

Beberapa symbol/notasi yang digunakan dalam membuat *ERD*, antara lain *Entity*, *Atribut*, dan *Relation*.

##### 1. *Entity*

Menggambarkan tabel yang akan dipakai dengan nama entity di bagian atas dan *atribut* dibagian bawah, seperti ditunjukkan pada tabel.

Tabel 2.1 siswa

Siswa		
<u>siswa id</u>	<pi>	<u>int</u>
Nm_siswa		TXT20
Almt_siswa		TXT50
Identifier 1	<pi>	

2. *Identifier* adalah *attribute* yang mengidentifikasikan sebuah entity secara unik, contoh ialah Siswa\_id untuk siswa. Dalam *ERD*, *identifier* ditulis dengan garis bawah.

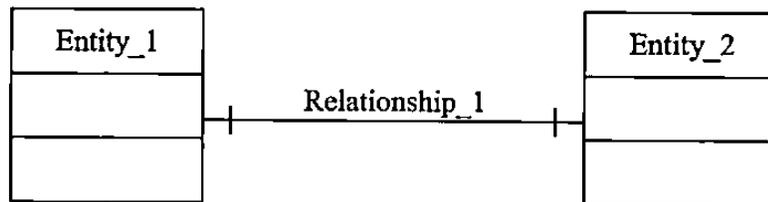
##### 3. *Relation*

Menggambarkan relasi atau hubungan antara dua entitas. Ada empat macam relasi yang ada pada *ERD*, yaitu:

###### a. *One to One Relationship*

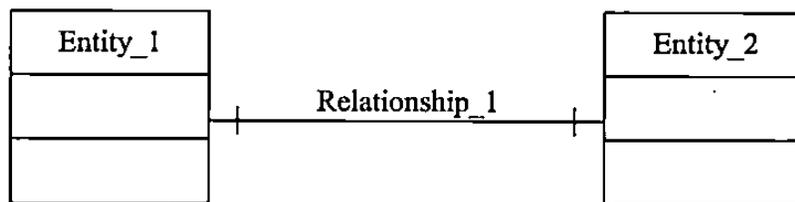
Relasi ini berarti anggota entitas A berhubungan dengan paling banyak dengan satu anggota entitas B, dan begitu juga sebaliknya anggota entitas B berhubungan dengan paling banyak satu anggota entitas A. Relasi ini bernotasi 1 : 1. Relasi *One to One* dapat dibedakan menjadi dua macam yaitu

*obligatory* dan *non obligatory*. *Obligatory* bila semua anggota dari suatu entitas harus berpartisipasi atau mempunyai hubungan dengan entitas yang lain. *Non obligatory* bila satu anggota entitas tidak harus mempunyai hubungan dengan anggota entitas lain.

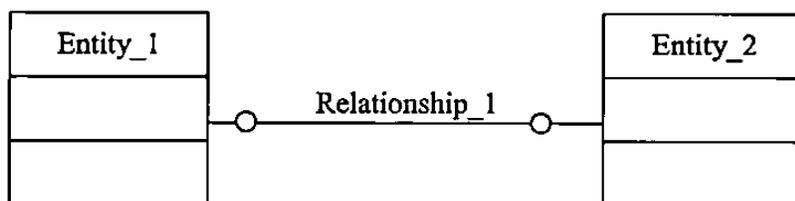


Gambar 2.5. *one to one relationship*.

Gambar 2.5. menjelaskan bahwa entity\_1 mempunyai hubungan *One to One* dengan entity\_2, entity\_1 bersifat *obligatory* terhadap entity\_2 sedangkan entity\_2 *non-obligatory* terhadap entity\_1.



Gambar 2.6. *one to one relationship-obligatory*

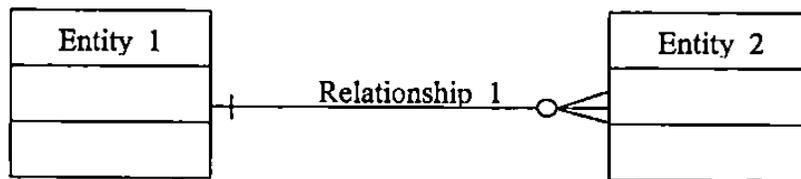


Gambar 2.7. *one to one relationship-non obligatory*

Hubungan *Obligatory dan non obligatory* lain yang dapat terjadi adalah kedua entitas sama-sama *obligatory* (gambar 2.6) atau kedua entitas sama-sama

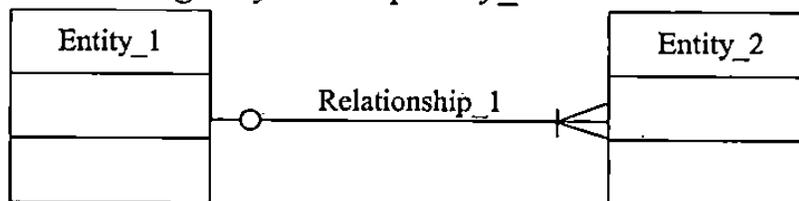
b. *One to Many Relationship*

Relasi ini berarti anggota entitas A dapat berhubungan dengan banyak anggota entitas B, tetapi tidak sebaliknya, dimana setiap anggota entitas B, berhubungan dengan paling banyak satu anggota entitas A. Relasi ini notasinya 1 : N. Relasi *One to Many* dapat dibedakan menjadi dua macam yaitu *obligatory* dan *non obligatory*.



Gambar 2.8. *One to Many Relationship (1)*

Gambar 2.8. menjelaskan bahwa *entity\_1* mempunyai hubungan One to Many dengan *entity\_2*. *Entity\_1* bersifat *non obligatory* terhadap *entity\_2* sedangkan *entity\_2* bersifat *obligatory* terhadap *entity\_1*.

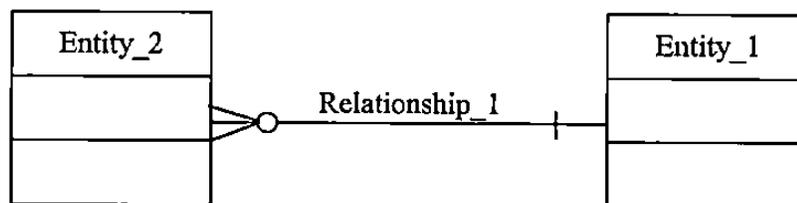


Gambar 2.9. *One to Many Relationship (2)*

Gambar 2.9. menjelaskan bahwa *entity\_1* mempunyai hubungan One to Many dengan *entity\_2*, *entity\_2* bersifat *non obligatory* terhadap *entity\_1* sedangkan *entity\_1* bersifat *obligatory* terhadap *entity\_2*. Hubungan *Obligatory* dan *non obligatory* lain yang dapat terjadi adalah kedua *entity* sama – sama *obligatory*

c. *Many to One Relationship*

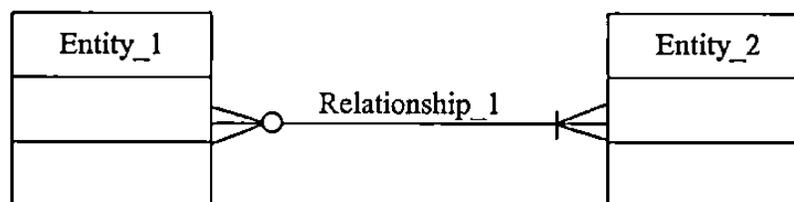
Relasi ini memiliki arti yang sama dengan *One to Many Relationship*, hanya saja pembacaan *entity*-nya dibalik.



Gambar 2.10. *Many to One Relationship*

d. *Many to Many Relationship*

Relasi ini berarti setiap anggota *entitas A* dapat berhubungan dengan banyak anggota *entitas B*, dan demikian sebaliknya, dimana setiap anggota *entitas B* dapat berhubungan dengan banyak anggota *entitas A*. Relasi *many to many* dapat dibedakan menjadi dua macam yaitu *obligatory* (gambar 2.11) dan *non Obligatory* (gambar 2.12.). Relasi ini notasinya N : M.



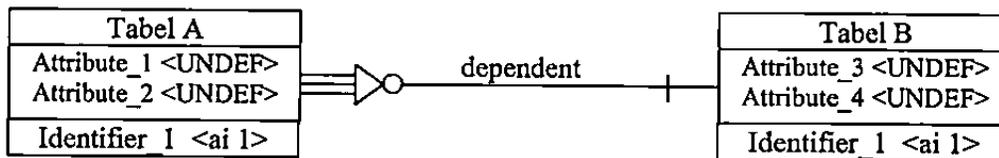
Gambar 2.11. *Many to Many Relationship*

Gambar 2.11. menjelaskan bahwa *entity\_1* mempunyai hubungan many to many dengan *entity\_2*, *entity\_1* bersifat *Obligatory* terhadap *entity\_2* sedangkan *entity\_2* bersifat *non obligatory* terhadap *entity\_1*.

Hubungan obligatory dan non obligatory lain yang dapat terjadi adalah kedua

*entiti* sama sama *obligatori* atau kedua *entiti* sama sama bersifat *non*

Pada keempat macam relasi yang telah disebutkan diatas, bisa terjadi sebuah relasi yang *dependent*. Yang dimaksud *dependent* adalah tabel yang satu tergantung dengan tabel yang lain. Contoh tabel A tergantung dengan tabel B, dapat dilihat pada gambar 2.11.



Gambar 2.12. *Many to One Relationship yang dependent*

### 2.2.2.3 Tata Cara dalam Merancang *Entity Relationship Diagram*

Terdapat tujuh tahap yang harus diikuti dalam membuat *ERD* :

- a. Menyiapkan *ERD* kasar
- b. Membuat analisis data
- c. Memetakan elemen-elemen data pada entitas
- d. Mengidentifikasi Entitas
- e. Mengidentifikasi hubungan
- f. Menyiapkan *ERD* yang telah dimodifikasi
- g. Menelaah *ERD* bersama pemakai dan memperbaikinya

### 2.2.2.4 Kelebihan *Entity Relationship Diagram*

Terdapat beberapa konsep kelebihan dari *Entity Relationship Diagram*, adalah :

- a. Desainnya mendekati pengamatan dan penerimaan pemakai terhadap data.
- b. Konsep data model dengan level tinggi
- c. Tidak untuk mendeskripsikan cara data disimpan dalam komputer

## 2.3 Perangkat Lunak Pendukung

Sesudah mendisain sistem dengan alat bantu dibutuhkan program komputer untuk mengimplementasikannya. Ada beberapa perangkat lunak pendukung yaitu :

### 2.3.1 *MySQL*

*MySQL* adalah database pilihan populer untuk digunakan dalam aplikasi web, dan merupakan komponen utama dari perangkat lunak web banyak digunakan aplikasi LAMP - LAMP merupakan singkatan dari "Linux, Apache, MySQL, PHP". Popularitasnya terkait erat dengan popularitas PHP. MySQL digunakan dalam beberapa situs web yang paling sering dikunjungi di Internet, Platform dan interface salah satu jenis *database server* yang terkenal, disebabkan *MySQL* menggunakan *SQL* sebagai bahasa dasar untuk mengakses *Database*. MySQL termasuk *RDBMS (Relation Database Management Sytem)*. *MySQL* bersifat gratis dalam arti tidak perlu membayar untuk menggunakannya. MySQL dapat digunakan pada berbagai *platform* sistem operasi. Khusus pada sistem operasi *windows*, MySQL bersifat *shareware* (Parenginangin,2006).

*MySQL* sebagai *database*, mempunyai banyak kelebihan, pemakai *database* ini adalah di lingkungan internet aspek yang paling menonjol. *MySQL* terkenal dengan pengolahan data yang cepat walaupun data *record* yang dimasukkan dalam jumlah yang banyak. Sebagai *database server*, *MySQL* dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data, hal ini terbukti untuk *query* yang dilakukan oleh *single user* dan lima kali lebih

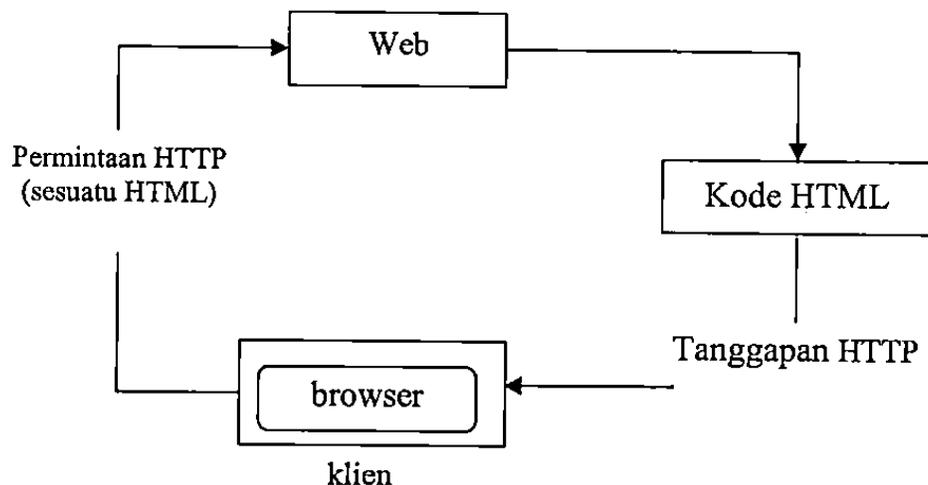
cepat dibandingkan *interbase*. Kemampuan yang cukup menakjubkan untuk sebuah *software* yang diperoleh secara gratis (Parenginangin,2006).

### 2.3.2 *Hypertext Markup Language (HTML)*

Pada mulanya *HTML* didesain untuk menjadi sebuah bahasa yang menggambarkan suatu struktur dokumen yang tidak terikat pada perangkat keras atau perangkat lunak tertentu, tetapi pada kenyataannya *HTML* menjadi semacam bahasa untuk mengatur *format* tampilan dokumen saja.

*HTML* adalah kependekan dari *Hypertext Markup Language* dan merupakan bahasa *markup* yang mengatur bagaimana sebuah dokumen ditampilkan pada browser. Standar *HTML* dibuat berdasarkan *SGML* (*standard Generalized markup language*) dan memiliki *DTD* (*Document Type Definition*), yang merupakan suatu dokumen yang mengatur *syntax HTML*

Sebenarnya suatu halaman *web* yang kita lihat di *internet* merupakan hasil dari kode-kode *HTML* yang dipanggil oleh *browser* kita. Struktur *HTML* dasar terdiri dari atas *head* dan *body* yang diapit oleh *tag HTML*. Dalam struktur *head* kita mengenal *tag title* untuk menggambarkan judul suatu halaman *web*, serta elemen penting seperti *tag meta* yang sangat diperlukan untuk *search engine*. Sedangkan dalam struktur *body* kita dapat menyisipkan *image*, *table*, serta media lainnya seperti *flash*, *quicktime movie*, dan sebagainya



Gambar 2.13. Skema Konsep Kerja HTML (Parenginangin,2006).

Konsep kerjanya adalah model kerja *HTML* diawali dengan permintaan suatu halaman *web* oleh *browser*. Berdasarkan *URL (uniform resource locator)* atau dikenal dengan sebutan alamat *internet*, *browser* mendapatkan alamat dari *web server*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Informasi yang disampaikan *web server* antara lain adalah nama *browser*, versinya, dan sistem operasinya. Selanjutnya *web server* akan mencari berkas yang diminta dan diberikan isinya ke *browser*, yang mendapatkan isinya segera melakukan proses penerjemahan kode *HTML* dan menampilkannya ke layar pemakai(Parengin,2006).

### 2.3.3 PHP (*Hypertext preprocessor*)

PHP diperkenalkan pertama kali oleh J Wrynja adalah seorang pria yang memiliki dasar yang matang tentang pemrograman, khususnya pemrograman pada sisi *server*. PHP adalah bahasa pemrograman berbasis web. Bahasa ini mempunyai kelebihan yaitu komabilitasnya dengan berbagai macam jenis *database* dukungan dengan berbagai macam jenis sistem operasi. PHP lebih

cocok dan umum digunakan jika digabungkan dengan *database* MySQL. MySQL dengan PHP seakan-akan dua hal yang tidak dapat dipisahkan. Tentunya untuk dapat menggunakan keduanya dibutuhkan tingkat kemampuan pemrograman tertentu.

Banyak digunakan oleh para pemrogram berlatar belakang C/C++ karena kemiripan *syntaxnya*. *Open source*, karenanya gratis dan bebas. *Database* pasangannya biasanya MySQL, dijalankan bersama *webserver Apache* di atas sistem operasi Linux.

#### a. Dasar Pemrograman PHP

PHPTriad adalah *software installer* PHP secara *instant* yang berjalan pada lingkungan Windows, setelah menginstal PHPTriad anda tidak saja telah menginstal PHP, akan tetapi juga sekaligus telah menginstall *Apache Web Server* dan *Database MySQL*.

PHPTriad jalan di bawah *platform* Windows, Jadi untuk Windows x ataupun Windows 2000, NT, Me, ataupun XP tetap berjalan sama baiknya. Untuk spesifikasi komputer tidak begitu memerlukan spesifikasi tinggi.

Fungsi yang selalu akan ditampilkan dalam bahasa pemrograman ini adalah *echo*, fungsi ini digunakan untuk menampilkan suatu data ke dalam *browser* (Nugroho, 2001).

#### b. Alias untuk parameter PHP

Alias adalah suatu metode untuk pengalamatan terhadap *interpreter* yang akan digunakan. Secara standar pada file *httpd.conf* memiliki

parameter alias yang berguna untuk memberitahukan kepada *Apache* tentang *interpreter* PHP dan *Perl*. Dalam pemrograman ini digunakan PHPTriad, sehingga secara otomatis terpasang pada direktori *C:/Apache/PHP*, sehingga penyetingan *script* aliasnya adalah *ScriptAlias /php/ "c:/apache/php/"*. Sedangkan apabila tidak menggunakan PHPTriad, maka harus diketahui terlebih dahulu tempat penginstalan program PHP (Nugroho, 2001).

### c. Konfigurasi *Web Server*

Cara pengetesan konfigurasi yang telah dilakukan adalah dengan menjalankan *start apache* yang ada pada program. Kemudian membuka sebuah halaman *website* dengan IE atau program lain yang juga dapat digunakan untuk *browsing* di internet. Untuk memanggil halaman PHP ketikkan <http://localhost>, apabila berhasil maka tidak terdapat pesan *error* pada halaman tersebut tetapi akan tampil ucapan *welcome*. Tampilan akan berbeda-beda sesuai dengan jenis MySQL yang digunakan.

Data yang terdapat didalam *database* akan ditampilkan ke *website* dalam bentuk dokumen *HTML*. Oleh karena itu, data tersebut perlu dimanipulasi agar dapat dipahami dengan mudah oleh *user*. Proses memanipulasi data menggunakan bahasa pemrograman *PHP*. *PHP* adalah sebuah *script* yang bersifat *server side* (diolah di *server*) yang ditambahkan pada *HTML* untuk membuat *web* menjadi lebih menarik, dinamis, dan interaktif. Salah satu kelebihan *PHP* adalah dapat

digunakan di sistem operasi *Windows* maupun *Linux*. *PHP* mudah digunakan dan memiliki fitur yang lengkap untuk memelihara sebuah *website* (Zandstra, 2000).

*PHP* merupakan kepanjangan dari *Hypertext preprocessor*. *PHP* merupakan *server side scripting language* atau sering disebut merupakan sebuah bahasa pemrograman *website* yang langsung berhubungan dengan *server* dan *client* hanya menerima hasil dari tampilan yang telah dikemas oleh bahasa pemrograman *Web PHP* ini. Secara keseluruhan bahasa *PHP* ini seringkali ditulis di atas konteks bahasa *HTML*. Tidak seperti *HTML*, bahasa *PHP* tidak langsung dikirim ke *client* secara langsung melalui *server*, tetapi oleh *PHP* fungsi-fungsi yang telah dibuat dikirim ke *client* dalam bentuk modul-modul. Elemen *HTML* berdiri sendiri sendiri di dalam *script page* yang telah dibuat namun kode *PHP* dibuat untuk dieksekusi sehingga *HTML* tanpa menggunakan fungsi-fungsi tertentu sudah dapat tampil, sedangkan *PHP* tidak dapat dilihat secara kasat mata oleh *client* karena hanya berupa modul yang bekerja di *server*. Bahasa *PHP* tidak hanya berupa kode teks yang dikemas untuk mengeksekusi sebuah fungsi namun juga dapat *query database*, membuat *image*, membaca dan menulis sebuah *file*, dan *remote server*. Jadi secara umum *HTML* dan *PHP* dikombinasikan sehingga memunculkan sebuah tampilan yang dikirim ke *client*.

*PHP* dapat berjalan dengan dukungan dua buah aplikasi yang berbeda, yaitu *Apache* sebagai *server* maintain-nya dan *MySQL* sebagai *database*-nya. *Apache* saat ini merupakan salah satu *maintain web server* yang paling banyak digunakan oleh kalangan penggunaan web karena kedinamisannya di dalam tugas sebagai *web server*. *Apache* mampu menerima semua *server side script language*

yang ada termasuk di dalamnya *java server page* dan *ASP*, sedangkan *MySQL* merupakan salah satu dari sekian banyak *database* yang powerful di dalam menangani masalah *database*. *PHP* bekerja di dalam *Apache* sebagai modul sedangkan *MySQL* bekerja sebagai *connector* web dengan data yang diperlukan. Dengan lengkapnya penggunaan *PHP*, bukan berarti *PHP* tidak dapat berjalan di *maintain web server* lainnya, *PHP* mampu juga berjalan diatas *maintain web server* milik *Windows* atau yang sering disebut *IIS (integrated informasi sistem)*.

#### a. Cara Kerja *PHP* Secara Umum

Kode program *PHP* menyatu dengan *tag-tag HTML* dalam satu *file*. Perintah dari *PHP* selalu dimulai dengan tanda '<?' dan diakhiri dengan tanda '?>'. Untuk mengakhiri setiap akhir perintah yang dilaksanakan atau diproses diberi tanda ';' . File yang berisi *tag HTML* dan kode *PHP* ini diberikan eksistensi ini, pada saat *file* diakses, *server* tahu bahwa file ini mengandung kode *PHP*. *Server* menterjemahkan kode ini akan menghasilkan *ouput* dalam bentuk *tag HTML* yang dikirim ke *browser client* yang mngakses *file* tersebut.

Pada *PHP* tidak ada deklarasi variable yang dibutuhkan. Sebuah variable dapat diakses kapan saja dengan nama yang harus berlainan antara variable yang satu dengan yang lainnya. Penanda variable dari *PHP* adalah tanda '\$ ' pada bagian awal suatu kata seperti '\$query' ataupun '\$hasil'. Sedangkan untuk struktur bahasa pemrograman *PHP* mirip dengan struktur bahasa pemrograman C, contohnya : printf echo for while dan masih banyak lagi lainnya

b. Kelebihan-Kelebihan dari *PHP*

*PHP* pada dasarnya merupakan sebuah bahasa *web* yang sangat *powerful*, dimana terdapat beberapa alasan untuk bekerja dengan *PHP*. Untuk beberapa *project* yang akan dibuat dengan *PHP*, akana ditemukan beberapa proses yang lebih cepat daripada yang kita dengar sebelumnya. Selain itu *PHP* memiliki keunggulan antara lain :

1. Kecepatan Pengembangan

*PHP* memperbolehkan *programmer* untuk memisahkan kode-kode *HTML* dengan kode elemen *script PHP*, maka akan mengurangi penggunaan waktu proses jika akan melakukan eksekusi pada sebuah fungsi. Sebagaimana kita ketahui *HTML* menggunakan beberapa *tag* yang mengharuskan sistem untuk menyelesaikan satu *tag* demi *tag* untuk mendapatkan hasil yang di perlukan. Dalam beberapa hal, *PHP* juga dapat memisahkan proyek antara *design* dengan bagian eksekusi sehingga ketika sebuah modul dieksekusi tidak perlu semuanya dieksekusi *design*-nya.

2. *PHP* adalah program *open source*

*Open source* dapat diartikan *free* atau gratisan, tidak peduli gratis pada waktu *download PHP* atau aplikasi *PHP* ini sendiri. *PHP* merupakan sebuah bahasa *web programming* yang gratis, bener-bener gratis untuk komersial, gratis untuk digunakan di dalam *home industry, dan lain sebagainya*. Secara khusus *PHP* menawarkan pada penggunaanya untuk mengembangkan *PHP* sebagai penerapan yang

dimiliki oleh penggunanya. Selain itu dengan adanya visi seperti itu maka kemungkinan untuk memperbaiki *error* yang ada dapat dilakukan oleh semua pengguna *PHP* di seluruh dunia. Selain itu jika pengguna yang lain menemukan adanya *bug* atau *error* maka kita juga akan dapat langsung mengetahuinya.

### 3. Kemampuan

Secara khusus *performance* yang dimiliki oleh *PHP* sudah sangat bahkan terlalu baik untuk sebuah bahasa pemrograman *web* yang dipakai oleh semua pengguna bahasa *web programming* di dunia ini. Karena kemampuannya yang *open source* itu pula *PHP* mampu berkembang sangat cepat bahkan versi *PHP4*-nya sudah memiliki kemampuan yang sama dengan *ASP* bahkan *ASP*.

*Net* padahal *PHP* tidak secara khusus mengembangkan *PHP* ke bentuk yang lebih baik. Versinya yang terakhir ini sangatlah *reliable* sekali terutama beberapa fungsi yang dimilikinya merupakan pengembangan dari fungsi-fungsi sebelumnya.

### 4. Mudah disesuaikan

*PHP* pada dasarnya dan secara khusus ditulis untuk digunakan di berbagai macam *platform* dan *opreting sistem* serta juga *PHP* didesain untuk bisa bekerja dengan berbagai macam *server* dan *database*. *Programmer* bisa mengerjakan *PHP* dari *UNIX* kemudian mengintegrasikanya ke *Windows* tanpa da kesulitan yang berarti.

Demikian pula sebaliknya jika kita telah membuat suatu aplikasi

*Windows* maka kita juga tidak akan kesulitan mengintegrasikannya kembali ke dalam *UNIX*.

#### 5. Interaksi dengan berbagai macam *Database*

Salah satu kelebihan *PHP* adalah mampu berkomunikasi dengan berbagai macam *database* yang terkenal. Dengan demikian, menampilkan data yang bersifat dinamis, yang diambil dari *database*, merupakan hal yang mudah untuk di implementasikan pada saat ini.

#### 2.3.4 *Apache*

*Apache* adalah perangkat lunak *server web* yang dapat dijalankan pada banyak sistem operasi (*Unix, BSD, Linux, Microsoft Windows* dan *Novell Netware* serta *platform* lainnya) yang berguna untuk melayani dan memfungsikan situs *web*. *Apache* juga didukung oleh sejumlah antarmuka pengguna berbasis grafik yang memungkinkan penanganan *server* menjadi mudah. *Apache* merupakan perangkat lunak sumber terbuka dikembangkan oleh komunitas terbuka yang terjadi yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation* ([www.apache.org](http://www.apache.org)).

Asal mula nama *Apache* berasal ketika sebuah *server web* populer yang dikembangkan pada awal 1995 yang bernama NCSA HTTPd 1.3 memiliki sejumlah perubahan besar terhadap kode sumbernya (patch). Saking banyaknya patch pada perangkat lunak tersebut sehingga disebut sebuah *server* yang memiliki banyak patch ("a patchy" server). Tetapi pada halaman FAQ situs web

resminya, disebutkan bahwa "*Apache*" dipilih untuk menghormati suku asli Indian Amerika *Apache*(Indian), yang dikenal karena keahlian dan strategi perangnya.

*Apache* merupakan perangkat lunak *open source* yang menjadi alternatif dari server web Netscape (sekarang dikenal sebagai *Sun Java System Web Server*). Sejak April 1996 *Apache* menjadi terpopuler di internet. Pada Mei 1999, *Apache* digunakan di 57% dari semua *server web* di dunia. Pada November 2005 persentase ini naik menjadi 71%. (*Netcraft Web Server Survey*, November 2005).

### 2.3.5 XAMPP

*XAMPP* adalah perangkat lunak yang *open source* sering disebut gratisan yang mendukung banyak sistem operasi dari beberapa program untuk menjalankan fungsinya sebagai *server* yang berdiri sendiri yang terdiri atas program *Apache Http Server*, *MySQL Database*, dan penterjemahan bahasa yang di tulis dengan bahasa pemrograman *PHP* dan *Perl*, *XAMPP* adalah nama yang merupakan singkatan dari X( empat sistem oprasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU(General Public Licence)* dan bebas yang merupakan *web server* yang mudah digunakan juga mampu melayani halaman yang dinamis. Saat ini, *XAMPP* tersedia dalam buntut sistem operasi *Microsoft Windows*, *Linux*, *Sun Solaris* dan *Mac OS X*. kelebihan dari *XAMPP* sebagai *Server* juga memiliki mod *ssl* yang berfungsi sebagai *SSL Server*