



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Arsopudin (2013:9), mengemukakan “Komputer adalah alat bantu pemrosesan data secara elektronik dan cara pemrosesan datanya berdasarkan urutan instruksi atau program yang tersimpan dalam memori masing-masing komputer.”

Hartono (2013:27), “Komputer adalah sebuah mesin yang dapat dikendalikan melalui perintah (*programmable machine*) yang dirancang untuk secara otomatis melakukan serangkaian urutan perhitungan (*arithmetic*) atau proses-proses yang diurutkan secara logis”.

2.2 Teori Judul

2.2.1 Pengertian Aplikasi

Asropudin (2013:6), berpendapat “*application* adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Ms-Word*, *Ms-Excel*”.

Sujatmiko (2012:22), “Istilah Aplikasi berasal dari bahasa Inggris, yaitu *application* yang dibuat oleh suatu perusahaan komputer untuk membantu manusia dalam mengerjakan tugas-tugas tertentu, misalnya *Ms. Word*, *Ms. Excel*”.

2.2.2 Pengertian Pengolahan Data

George R. Terry Ph. D dalam Hutahean (2014:8), “pengolahan data adalah serangkaian operasi atas informasi yang direncanakan guna mencapai tujuan atau hasil yang diinginkan”.

Sutarman (2012:4) “pengolahan data adalah proses perhitungan atau transformasi dan *input* menjadi informasi yang mudah dimengerti ataupun sesuai dengan yang diinginkan”.



2.2.3 Pengertian Data

Asropudin (2013:22), “Data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti”.

2.2.4 Pengertian Perpindahan

Saripudin (2012:3), “Perpindahan perubahan posisi (kedudukan) suatu benda dalam waktu tertentu”.

2.2.5 Pengertian Penduduk

Somantri dan Huda (2013:116), “Penduduk komponen penting dalam suatu wilayah, syarat berdirinya suatu negara, yaitu penduduk, penduduk sebagai sumber daya manusia memegang peran sentral dalam pembangunan”.

2.2.6 Pengertian Perpindahan Penduduk

Haryati (2007:25), “Perpindahan penduduk adalah ke suatu wilayah ke wilayah lain, baik melewati batas Negara ataupun batas administrasi wilayah suatu Negara dengan tujuan menetap”.

2.2.7 Pengertian Tenaga Kerja

Humalutua (2010:12) “Tenaga Kerja adalah mereka yang melakukan kegiatan ekonomi dengan melakukan pekerjaan bersama-sama dengan orang lain dengan secara mandiri, mereka yang mempunyai kemampuan untuk yang terkait dalam hubungan kerja”.

2.2.7 Pengertian Transmigrasi

Aninandra (2007:39), “Transmigrasi perpindahan penduduk dari satu daerah (pulau) yang berpenduduk padat ke daerah (pulau) lain yang berpenduduk jarang penduduknya di dalam wilayah Republik Indonesia guna kepentingan pembangunan Negara karna alasan-alasan yang dipandang perlu oleh pemerintah berdasarkan ketentuan yang diatur dalam undang-undang”.



2.2.8 Pengertian Aplikasi Pengolahan Data Perpindahan Penduduk Pada Dinas Tenaga Kerja dan Transmigrasi

Aplikasi Pengolahan Data Perpindahan Penduduk Pada Dinas Tenaga Kerja dan Transmigrasi Provinsi Sumatra Selatan adalah aplikasi yang dibuat untuk membantu Dinas Tenaga Kerja dan Transmigrasi Provinsi Sumatra Selatan dalam mengelolah data perpindahan penduduk serta membuat laporan perpindahan penduduk yang nantinya akan diserahkan kepada kepala Dinas Tenaga Kerja dan Transmigrasi Provinsi Sumatra Selatan.

2.3 Teori Khusus

2.3.1 Pengertian DFD (*Data Flow Diagram*)

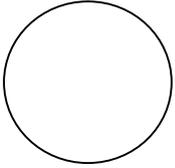
Shalahuddin (2013:70), “*Data Flow Diagram* (DFD) atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggunakan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*ouput*)”.

Keuntungan penggunaan DFD digunakan untuk mempresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD dapat dibagi menjadi beberapa level yang lebih detail. DFD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi.

2.3.1.1 Simbol *Data Flow Diagram*

Simbol atau lambang yang digunakan dalam membuat *data flow diagram* ada 4 (empat) buah, yaitu sebagai berikut:

Tabel 2.1 Simbol-simbol DFD (*Data Flow Diagram*)

Notasi	Keterangan
	<p>proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program</p> <p>catatan: nama yang diberikan pada sebuah proses biasanya berupa kata kerja</p>

Lanjutan Tabel 2.1 Simbol-simbol DFD (*Data Flow Diagram*)

Notasi	Keterangan
	<p><i>file</i> atau basis data atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (<i>Entity Relationship Diagram</i> (ERD), <i>Conceptual Data Model</i> (CDM), <i>Physical Data Model</i> (PDM))</p> <p>catatan: nama yang diberikan pada sebuah penyimpanan biasanya kata benda.</p>
	<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan</p> <p>catatan: nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda.</p>
	<p>aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>)</p> <p>catatan: nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”</p>

(Sumber: Shalahuddin (2013:72))



2.3.1.2 Aturan Main Data Flow Diagram

Bentuk rambu-rambu atau aturan main yang baku dan berlaku dalam penggunaan *data flow diagram* untuk membuat model sistem adalah sebagai berikut:

- 1) Di dalam *data flow diagram* tidak boleh menghubungkan antara satu *external entity* dengan *external entity* lainnya secara langsung.
- 2) Di dalam *data flow diagram* tidak boleh menghubungkan *data store* yang satu dengan *data store* yang lainnya secara langsung.
- 3) Di dalam *data flow diagram* tidak boleh menghubungkan *data store* dengan *external entity* secara langsung.
- 4) Setiap proses harus ada memiliki *data flow* yang masuk dan ada juga *data flow* yang keluar.

3.3.1.3 Teknik Membuat Data Flow Diagram

Teknik atau cara yang lazim digunakan dalam membuat *data flow diagram* adalah:

1. Mulai dari yang umum atau tingkatan yang lebih tinggi, kemudian diuraikan atau dijelaskan sampai yang lebih detail atau tingkatan yang lebih rendah, yang lebih dikenal dengan istilah *Top-Down Analysis*.
2. Jabarkan proses yang terjadi di dalam *data flow diagram* sedetail mungkin sampai tidak dapat diuraikan lagi.
3. Peliharalah konsistensi proses yang terjadi didalam DFD, mulai dari diagram yang tingkatannya lebih tinggi sampai dengan diagram yang tingkatannya lebih rendah.
4. Berikan label yang bermakna untuk setiap simbol yang digunakan seperti:
 - a. Nama yang jelas untuk *External Entity*
 - b. Nama yang jelas untuk *Proses*
 - c. Nama yang jelas untuk *Data Flow*
 - d. Nama yang jelas untuk *Data Store*



2.3.1.4 Tahapan *Data Flow Diagram*

Langka-langka di dalam membuat *data flow diagram* dibagi menjadi 3 (tiga) tahap atau tingkat konstruksi DFD, yaitu sebagai berikut:

1. Diagram Konteks

Diagram ini dibuat untuk menggambarkan sumber serta tujuan data yang akan diproses atau dengan kata lain diagram tersebut digunakan untuk menggambarkan sistem secara umum/global dari keseluruhan system yang ada.

2. Diagram Nol

Diagram ini dibuat untuk menggambarkan tahapan proses yang ada di dalam diagram konteks, yang penjabarannya lebih terperinci.

3. Diagram Detail

Diagram ini dibuat untuk menggambarkan arus data secara lebih mendetail lagi dari tahapan proses yang ada di dalam diagram nol.

2.3.2 Pengertian *Blockchart*

Kristanto (2008:75), menjelaskan bahwa “*Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu”. Pembuatan *Blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi”.

Simbol-simbol yang sering digunakan dalam *blockchart* dapat dilihat pada tabel berikut ini:

Tabel 2.2 Simbol-simbol *Blockchart*

No.	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan.
2.		Multi dokumen.



Lanjutan Tabel 2.2 Simbol-simbol Blockchart

No.	Simbol	Keterangan
3.		Proses manual.
4.		Proses yang dilakukan oleh komputer.
5.		Menandakan dokumen yang diarsifkan(arsif manual)
6.		Data penyimpanan (<i>data storage</i>)
7.		Proses apa saja yang tidak terdefinisi termasuk aktifitas fisik.
8.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain.
9.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.
11.		Pengambilan keputusan (<i>decision</i>).
12.		Layar peraga (<i>monitor</i>).



Lanjutan Tabel 2.2 Simbol-simbol Blockchart

No.	Simbol	Keterangan
13.		Pemasukan data secara manual.

(Sumber: Kristanto(2008:75).

2.3.3 Pengertian ERD (*Entity Relational Diagram*)

Shalahuddin (2013:50) menjelaskan:

ERD (*Entity Relationship Diagram*) adalah digunakan untuk pemodelan basis data relasional. ERD memiliki beberapa aliran notasi seperti notasi chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot, dan beberapa notasi lain.

1. Entitas

Entitas bisa berupa orang, kejadian, atau benda di mana data akan dikumpulkan.

2. Relationship

- a. Hubungan antar 2 entitas.
- b. Entitas pertama dalam relationship disebut entitas induk, entitas kedua disebut sebagai entitas anak.
- c. Relationship harus memiliki nama yang berupa kata kerja.
- d. Relationship berjalan 2 arah.

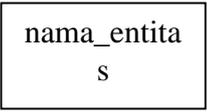
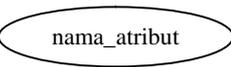
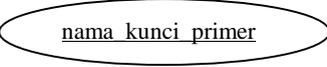
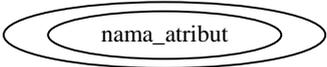
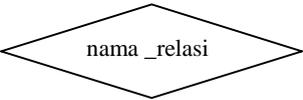
3. Atribut

- a. Informasi yang diambil tentang sebuah entitas.
- b. Hanya yang digunakan oleh organisasi yang dimasukkan dalam model.
- c. Nama atribut harus merupakan kata benda.



Notasi–notasi yang digunakan dalam ERD, yaitu:

Tabel 2.3 Simbol-simbol *Entity Relationship Diagram* (ERD)

Simbol	Keterangan
Entitas / <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama table
Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama)
Atribut multivalai / <i>multivalue</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
Relasi 	<i>Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja</i>

Lanjutan Tabel 2.3 Simbol-simbol *Entity Relationship Diagram* (ERD)

Simbol	Keterangan
Asosiasi / association 	<i>Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian</i> <i>Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan one to many menghubungkan entitas A dan entitas B maka</i>

(Sumber: Shalahuddin (2013:50)).

2.3.4 Pengertian *Flowchart*

EWolf Community (2011:16), “*Flowchart* adalah simbol-simbol pekerjaan yang menunjukkan bagan aliran proses yang saling terhubung, setiap simbol *flowchart* melambangkan pekerjaan dan instruksinya.”

Flowchart digunakan untuk mempermudah penyusunan program. Dengan menggunakan *flowchart*, logika pemrograman lebih mudah dipahami dan dianalisis, sehingga dapat menentukan kode-kode pemrograman yang sesuai pekerjaannya. Seorang *programmer* membuat *flowchart* sebelum menulis kode programnya.



Berikut beberapa simbol standard *flowchart* yang sering digunakan dalam pemrograman komputer:

Tabel 2.4 Simbol-simbol *Flowchart*

No.	Gambar	Arti
1.		Simbol <i>Start</i> atau <i>End</i> yang mendefinisikan awal atau akhir dari sebuah <i>flowchart</i> .
2.		Simbol pemrosesan yang terjadi pada sebuah alur kerja
3.		Simbol Input/Output yang mendefinisikan masukan dan keluaran proses
4.		Simbol untuk memutuskan proses lanjutan dari kondisi tertentu
5.		Simbol Konektor untuk menyambung proses pada lembar kerja yang sama
6.		Simbol Konektor untuk menyambung proses pada lembar kerja yang berbeda
7.		Simbol untuk menghubungkan antar proses atau antar simbol
8.		Simbol yang menyatakan piranti keluaran, seperti layar monitor, printer, dll

Lanjutan Tabel 2.4 Simbol-simbol *Flowchart*

No.	Gambar	Arti
9.		Simbol yang mendefinisikan proses yang dilakukan secara manual
10.		Simbol masukan atau keluaran dari atau ke sebuah dokumen
11.		Simbol yang menyatakan bagian dari program (sub-program)
12.		Simbol masukan atau keluaran dari atau ke sebuah pita magnetic
13.		Simbol database atau basis data

(Sumber:EWolf Community(2011:16)).

2.3.5 Pengertian Kamus Data

Rosa (2013:73), “Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (input) dan keluaran (output) dapat dipahami secara umum (memiliki standard cara penulisan)”. Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur. Kamus data biasanya berisi:

- a. Nama-nama dari data
- b. Digunakan pada merupakan proses-proses yang terkait data
- c. Deskripsi merupakan deskripsi data
- d. Informasi tambahan seperti tipe data, nilai data, dan komponen yang membentuk data.



Tabel 2.5 Simbol-simbol dalam Kamus Data

No	Simbol	Keterangan
1.	=	Disusun atau terdiri dari
2.	+	Dan
3.	[]	Baik ...atau...
4.	{ } ⁿ	N kali diulang/bernilai banyak
5.	()	Data opsional
6.	*...*	Batas computer

(Sumber:Rosa(2013:74).

2.4 Teori Program

2.4.1 Basis Data (*Database*)

Rossa (2013:43), “Database adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat di butuhkan”.

Kadir (2014:218), menjelaskan:

Basis Data (*database*) adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi. Basis data dimaksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas.

Dalam proses normalisasi, ada beberapa istilah yang akan dipakai yaitu:

1. *Entity*, adalah konsep informasi yang direkam, meliputi orang, kejadian dan tempat.
2. *Field* (atribut), adalah sesuatu yang mewakili *entity*, misalnya untuk mahasiswa atributnya adalah NIM, nama, alamat dan sebagainya.
3. *Data Value*, adalah informasi yang tersimpan dalam setiap atribut.
4. *Record*, adalah kumpulan atribut yang saling berkaitan satu sama lain dan menginformasikan suatu *entity* secara lengkap.
5. *File*, adalah kumpulan *record* yang mempunyai panjang atribut yang sama tetapi berbeda dengan *data value*.
6. *Database*, adalah kumpulan *file* satu dengan *file* lainnya yang membentuk suatu informasi sistem secara keseluruhan.



2.4.2 Sekilas Tentang PHP

2.4.2.1 Pengertian PHP (*Hypertext Preprocessor*)

Betha (2014:4) menjelaskan:

PHP merupakan secara umum dikenal sebagai bahasa pemrograman script-script yang membuat dokumen HTML secara on the fly yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML dikenal juga sebagai bahasa pemrograman server side.

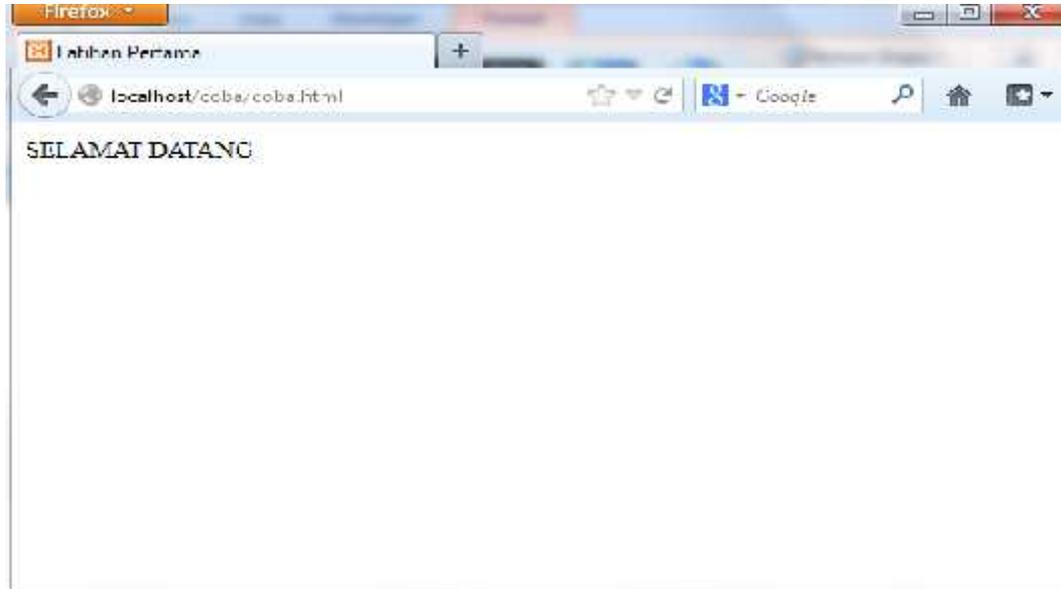
2.4.2.2 Skrip PHP

Betha (2014:5-6), “Skrip PHP berkedudukan sebagai tag dalam bahasa HTML. Sebagaimana diketahui, HTML adalah bahasa standar untuk membuat halaman-halaman *web*”. Sebagai contoh, berikut adalah kode HTML (disimpan dengan ekstensi .htm atau .html).

```
<HTML>
<HEAD>
<TITLE>LatihanPertama</TITLE>
</HEAD>
<BODY>
<?php
SELAMAT DATANG
?>
</BODY>
</HTML>
```



Bila dijalankan melalui *browser*, kode tersebut membentuk tampilan seperti berikut:



Gambar 2.1 Tampilan PHP Sederhana

2.4.3 Sekilas Tentang *MySql*

2.4.3.1 Pengertian *MySql*

Nugroho Bunafit (2014:15), “*MySql* adalah *database server*, *database* yang dibuat di *MySql* dapat kita akses dalam komputer jaringan, baik untuk aplikasi *single user* maupun maupun aplikasi *multi user*”.

Rosa dan Salahudin (2014:46), “SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS”.

Berikut ini adalah contoh pengaksesan pada DBMS dengan SQL yang secara umum terdiri dari 4 hal sebagai berikut:

- a. Memasukkan data (*insert*)

INSERT INTO Tabel_mahasiswa

(nim, nama, tanggal_lahir)

VALUES

(‘13501058’, ‘Rosa’, ‘1986-01-07’); *Query* tersebut digunakan untuk memasukkan data mahasiswa dengan NIM 13501058, nama Rosa, dan tanggal lahir 1 Januari 1986 ke table “Tabel_mahasiswa”.



- b. Mengubah data (*update*)

UPDATE Tabel_mahasiswa

SET

Tanggal_lahir = '1990-03-04'

WHERE

nim = '13501058';

Query diatas digunakan untuk mengubah data tanggal lahir mahasiswa dengan NIM = 13501058 menjadi 4 Maret 1990 dalam table "Tabel_mahasiswa".

- c. Menghapus data (*delete*)

DELETE FROM Tabel_mahasiswa

WHERE

nim = '13501058';

Query diatas digunakan untuk menghapus data mahasiswa dengan NIM = 13501058 dari table "Tabel_mahasiswa".

- d. Menampilkan data (*select*)

SELECT nim, nama

FROM Tabel_mahasiswa

WHERE

nim = '13501058';

Query diatas digunakan untuk menampilkan data mahasiswa yang tersimpan dalam table "Tabel_mahasiswa" dengan NIM = 13501058.

2.4.3.2 Tipe Data Dalam MySQL

Secara garis besar, *Mysql* menyediakan beberapa kategori tipe data, yaitu :

a. Tipe Numeris

Nugroho Bunafit (2014:17-19), menjelaskan "Tipe numeris adalah tipe yang menyatakan bilangan. Secara garis besar, numeris dikelompokkan menjadi tipe bilangan bulat dan tipe bilangan pecahan".



Nugroho Bunafit (2014:15), juga menjelaskan beberapa tipe data numeris, yaitu:

Tabel 2.6 Tipe Numeris

No	Tipe Data	Keterangan	Kebutuhan Memori
1.	BIT	Menyatakan tipe bit. Satu bit dapat berupa bilangan 0 atau 1.	8 byte
2.	TINYINT[(M)]	Menyatakan bilangan bulat yang berkisar antara -128 sampai 127.	1 byte
3.	BOOLEAN	Untuk menyatakan nilai benar dan salah.	1 byte
4.	SMALLINT[(M)]	Menyatakan data bilangan bulat yang berkisar antara -32768 sampai 32767.	2 byte
5.	MEDIUMINT[(M)]	Menyatakan data bilangan bulat yang berkisar antara -8388608 sampai 8388607.	3 byte
6.	INT[(M)] atau INTEGER[(M)]	Menyatakan data bilangan bulat yang berkisar antara -2147683648 sampai 2147683647.	4 byte
7.	BIGINT[(M)]	Menyatakan data bilangan bulat yang berkisar antara -9223372036854775808 sampai 9223372036854775807.	6 byte

(Sumber: Nugroho Bunafit (2014:15).



Lanjutan Tabel 2.6 Tipe Numeris

No	Tipe Data	Keterangan	Kebutuhan Memori
8.	FLOAT[(M,D)]	Tipe bilangan pecahan yang dapat menampung bilangan antara $-3.4028234466E+38$ sampai dengan $1.175494351E-38$, 0 , dan $1.175494351E-38$ sampai $3.4028234466E+38$.	4 byte
9.	DOUBLE[(M,D)]	Tipe bilangan pecahan yang dapat menampung bilangan antara $-1.79769313348623157E+308$ sampai dengan $2.2250738585072014E-308$, 0 , dan $2.2250738585072014E-308$ sampai $1.79769313348623157E+308$.	8 byte
10.	FLOAT(p)	Tipe bilangan pecahan. Dalam hal ini, p menyatakan jumlah presisi dalam bit.	4 byte jika $p \leq 24$, 8 byte jika $25 \leq p \leq 35$
11.	DEC[(M[,D])] atau DECIMAL[(M[,D])]	Menyatakan bilangan dengan sejumlah digit di belakang titik desimal.	

(Sumber: Nugroho Bunafit (2014:15).)



b. Tipe Tanggal dan Waktu

Kadir (2008:51), “menjelaskan bahwa tipe tanggal dan waktu untuk menangani waktu dan tanggal”.

Tabel 2.7 Tipe Tanggal dan Waktu

No	Tipe Data	Keterangan	Kebutuhan Memori
1.	DATE	Menyatakan tipe untuk data tanggal. Tanggal yang didukung berkisar antara ‘1000-01-01’ sampai dengan ‘9999-12-31’. Format : ‘tahun-bulan-tanggal’.	3 byte
2.	TIME	Menyatakan tipe untuk data waktu yang berformat : ‘jam:menit:detik’.	3 byte
3.	DATETIME	Menyatakan tipe data yang menggabungkan tanggal dan waktu. Format : ‘tahun-bulan-tanggal jam:menit:detik’.	8 byte
4.	TIMESTAMP	Tipe yang nilainya akan diisi secara otomatis dengan tanggal dan jam saat ada operasi INSERT atau UPDATE.	4 byte
5.	YEAR	Tipe untuk menyatakan tahun.	1 byte

(Sumber:Kadir(2008:51)).

c. Tipe String

Rosa dan Salahudin (2014:46), “String berarti deretan karakter. Sebuah string dapat tersusun atas nol, satu, atau banyak karakter. String yang tidak mengandung satu karakter pun dinamakan string kosong”.



Tabel 2.8 Tipe String

No	Tipe Data	Keterangan	Kebutuhan Memori
1.	CHAR(M)	String dengan panjang tetap yaitu M karakter. Nilai terbesar untuk M adalah 255.	M byte, $0 \leq M \leq 255$
2.	CHAR	Sinonim dengan CHAR(1)	1 byte
3.	VARCHAR(M)	String dengan panjang variabel. M menyatakan jumlah maksimum karakter yang bisa disimpan. Nilai terbesar untuk M yaitu 65535.	L+1 byte, $0 \leq L \leq M$ dan $0 \leq M \leq 65535$
4.	BINARY(M)	Serupa dengan VARCHAR(M), tetapi disimpan dalam bentuk biner.	M byte, $0 \leq M \leq 255$
5.	VARBINARY(M)	Serupa dengan VARCHAR(M), tetapi disimpan dalam bentuk biner.	L+1 byte, $0 \leq L \leq M$ dan $0 \leq M \leq 255$
6.	TINYBLOB	BLOB dengan ukuran paling kecil	L+1 byte, $L < 2^8$
7.	BLOB[(M)]	BLOB dengan ukuran lebih besar dari pada TINYBLOB	L+2 byte, $L < 2^{16}$
8.	MEDIUMBLOB	BLOB berukuran medium	L+3 byte, $L < 2^{24}$
9.	LONGBLOB	BLOB berukuran paling besar	L+4 byte, $L < 2^{32}$



Lanjutan Tabel 2.8 Tipe String

No	Tipe Data	Keterangan	Kebutuhan Memori
10.	TINYTEXT	String teks dengan ukuran paling kecil	L+1 byte, $L < 2^8$
11.	TEXT[(M)]	String teks dengan ukuran lebih besar dari pada TINYTEXT	L+2 byte, $L < 2^{16}$
12	MEDIUMTEXT	String teks berukuran medium	L+3 byte, $L < 2^{24}$
13	LONGTEXT	String teks berukuran paling besar	L+4 byte, $L < 2^{32}$
14	ENUM('nilai1', 'nilai2',...)	Tipe data yang nilai kolom berupa salah satu string yang disebutkan pada ENUM. Jumlah elemen ENUM dapat mencapai 65535.	1 atau 2 byte, bergantung pada jumlah nilai enumerisasi
15	SET('nilai1', 'nilai2',...)	Merupakan suatu objek string yang berisi nol, satu, atau beberapa nilai. Sebuah kolom bertipe SET maksimum berisi 64 elemen yang berbeda. Nilai antar-elemen ditulis dengan pemisah koma.	1,2,3,4, atau 8 byte, bergantung pada jumlah anggota himpunan

(Sumber: Rosa dan Salahudin (2014:47-49).



2.4.4 Sekilas Tentang Adobe Dreamweaver CS5

2.4.4.1 Pengertian Adobe Dreamweaver CS5

Madcoms (2010:1), “*Adobe Dreamweaver* merupakan software aplikasi yang digunakan sebagai HTML editor profesional untuk mendesain web secara visual. Aplikasi ini juga yang biasa dikenal dengan istilah WYSIWYG “*what you see is what you get*” yang intinya adalah bahwa Anda tidak harus berurusan dengan tag-tag HTML untuk membuat sebuah situs”.

Dreamweaver juga memberikan keleluasaan kepada Anda untuk menggunakan sebagai media penulisan bahasa pemrograman web. Dengan kemampuan fasilitas yang optimal dalam jendela Design membuat program ini memberikan kemudahan untuk mendesain web meskipun untuk para web desainer pemula sekalipun.

Adobe kembali mengeluarkan varian terbaru *Dreamweaver* yaitu *Dreamweaver CS5*. Ada beberapa *fitur* baru yang dapat ditemukan pada versi terbaru *dreamweaver CS5* ini, diantaranya: integrasi dengan *Adobe Business Catalyst*, integrasi *Adobe Browser Lab*, integrasi dengan *CMS*, pengecakan *CSS*, dll. Selain itu juga ada beberapa *fitur* dalam versi *Dreamweaver CS4* yang tidak disertakan atau ditemukan kembali dalam *Dreamweaver CS5*, diantaranya: fasilitas pembuatan *web* album foto, penambahan *Flash Paper*, beberapa *beaviour javascript*, dll.

2.4.4.2 Ruang Kerja Adobe Dreamweaver CS5

Adobe Dreamweaver CS5 mempunyai ruang kerja yang dapat digunakan untuk mendesain sebuah halaman web. Selain itu kita juga dapat mengubah tampilan umum dari ruang kerja *Adobe Dreamweaver CS5*. Adapun elemen-elemen ruang kerja dari *Adobe Dreamweaver CS5* adalah sebagai berikut:

1. **Application Bar**, Berada dibagian paling atas jendela aplikasi *Adobe Dreamweaver CS5*. Baris ini berisi tombol *workspace (workspace switcher)*, *CS Live*, Menu dan aplikasi lainnya.
2. **Toolbar Document**, Berisi tombol-tombol yang yang digunakan untuk mengubah tampilan jendela dokumen, sebagai contoh tampilan design atau



- tampilan kode. Selain itu juga dapat digunakan untuk operasi-operasi umum, misalnya untuk melihat hasil sementara halaman web pada jendela *browser*.
3. **Jendela Dokument**, Lembar kerja tempat membuat dan mengedit desain halaman web.
 4. **Workspace Switcher**, Digunakan untuk mengubah tampilan ruang kerja (*workspace*) *Adobe Dreamweaver CS5*. Sebagai contoh mengubah tampilan menjadi tampilan *classic*, yaitu tampilan ruang kerja *Dreamwaver* versi sebelumnya.
 5. **Panel Groups**, Kumpulan panel yang saling berkaitan, panel-panel ini dikelompokkan pada judul-judul tertentu berdasarkan fungsinya. Panel ini digunakan untuk memonitor dan memodifikasi pekerjaan. Secara default, panel groups berisi panel *Adobe Browser Lab*, *Adobe Business Catalyst*, *Insert*, *CS5 Styles*, *AP Element* dan *Files*.
 6. **Tag Selector**, Terletak dibagian bawah jendela dokumen, satu baris dengan status bar. Bagian ini menampilkan hirarki pekerjaan yang sedang terpilih pada jendela dokumen, dapat juga digunakan untuk memilih objek pada jendela desain berdasarkan jenis atau kategori objek tersebut. Tag selector juga menampilkan informasi format dari bagian yang sedang aktif pada lembar kerja desain.
 7. **Property Inspector**, Digunakan untuk melihat dan mengubah berbagai *property* objek atau teks pada jendela desain. *Property* untuk satu objek dengan objek lainnya selalu berbeda-beda. Jendela ini tidak dapat diuraikan pada tampilan jendela kode.
 8. **Toolbar Standard**, Baris toolbar ini berisi tombol-tombol yang mewakili perintah pada menu *File* dan *Edit*, diantaranya adalah perintah *New*, *Open*, *Save*, *Save All*, *Cut*, *Copy*, *Paste*, *Undo* dan *Redo*.
 9. **Toolbar Style Rendering**, Secara default disembunyikan. *Toolbar* ini berisi tombol-tombol untuk menampilkan desain dalam media yang berbeda. Selain itu juga digunakan untuk mengaktifkan dan menonaktifkan *style CSS*.
 10. **Toolbar Coding**, Berisi tombol-tombol yang digunakan untuk melakukan operasi kode-kode standar. Jendela ini hanya tampil pada jendela *Code*.
-



11. **Toolbar Browser Navigation**, Toolbar ini merupakan toolbar baru yang ada didalam *Adobe Dreamweaver CS5* dan letaknya tepat berada diatas jendela dokumen. *Toolbar* ini berisi tombol-tombol yang digunakan sebagai navigasi didalam *browser*.

2.4.5 Pengertian XAMPP

Nugroho (2014:16), “*XAMPP* adalah paket *web server* yang didalamnya terdapat *software MySQL Server, Apache web server, PHP, dan tools PhpMyAdmin*. Dalam satu kali install, maka Anda sudah memiliki semuanya”.

Berikut ini adalah gambar tampilan awal *XAMPP*.



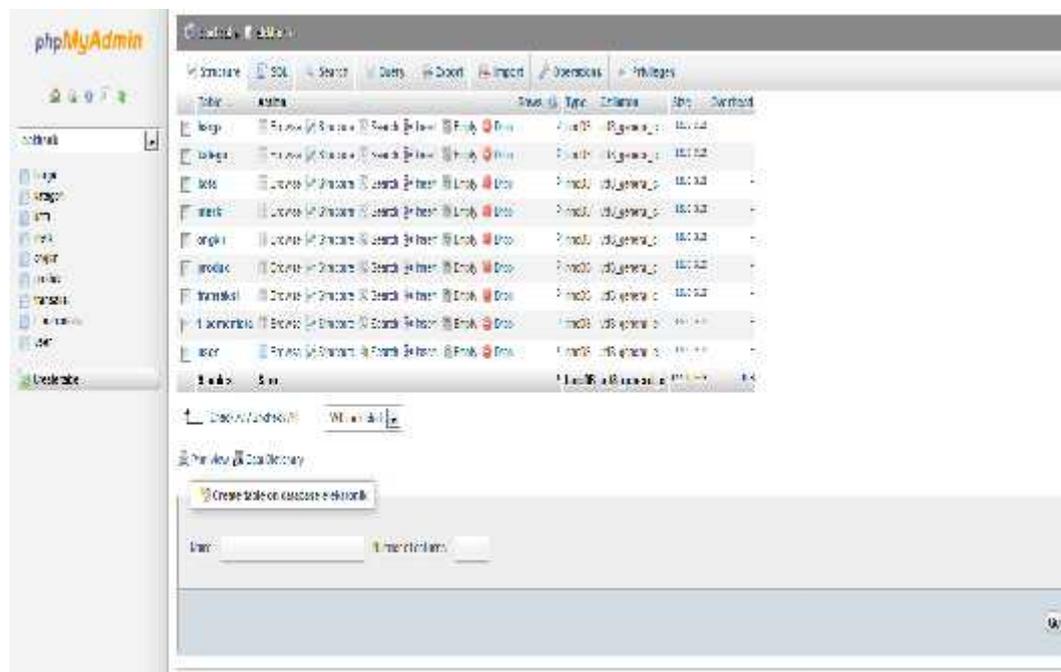
Gambar 2.2 Tampilan *XAMPP*



2.4.6 Pengertian PHP MyAdmin

Nugroho (2014:15), “*PHP MyAdmin* adalah aplikasi manajemen *database server MySQL* berbasis *web* menggunakan *PhpMyAdmin*. Anda dapat menjalankan aplikasi ini lewat browser layaknya mengakses *Internet*”. Berikut ini caranya.

1. Pastikan modul *MySQL* dan *Apache* sudah dijalankan.
2. Buka *Browser* Anda, misalnya; *Internet Explorer* atau *firefox*
3. Masukkan alamat <http://localhost/phpmyadmin>, lalu *Enter*.



Gambar 2.3 Tampilan *Database*