



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Sutarman (2012:2), menurut Hamacher “komputer adalah mesin penghitung elektronik yang cepat dan dapat menerima informasi *input digital*, kemudian memprosesnya sesuai dengan program yang tersimpan di memorinya, dan menghasilkan *output* berupa informasi.”

Sutarman (2012:2), menurut Blissmer “komputer adalah alat elektronik yang mampu melakukan beberapa tugas yaitu menerima *input*, memproses *input* sesuai dengan instruksi yang diberikan, menyimpan perintah dan hasil pengolahannya, serta menyediakan *output* dalam bentuk informasi.”

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 computer.”

2.1.2. Pengertian Internet

Asropudin (2013:48), “Internet merupakan istilah umum yang dipakai untuk menunjuk *network* tingkat dunia yang terdiri dari komputer dan layanan servis atau sekitar 30 sampai 50 juta pemakai komputer dan puluhan sistem informasi termasuk *e-mail*, *Gopher*, *FTP* dan *world wide web*.”

2.1.3. Pengertian Informasi

Sutabri (2012:22), “Informasi adalah data yang telah diklasifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan”.

Hutahaean (2014:9), “Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan berarti bagi penerimanya. Data kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata”.



2.1.4. Pengertian Web

Asropudin (2013:109), “*Website* adalah sebuah kumpulan halaman (*webpages*) yang diawali dengan halaman muka (*homepage*) yang berisikan informasi, iklan serta program interaksi.”

Hidayat (2010:2), “*Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman.”

2.2. Teori Judul

2.2.1. Pengertian Sistem

Sutabri (2012:10) menjelaskan, “Sistem diartikan sebagai kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu.”

Pratama (2014:7) menjelaskan, “Sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama.”

2.2.2. Pengertian Sistem Informasi

Pratama (2014:9) menjelaskan, “Sistem informasi merupakan gabungan dari empat bagian utama. Keempat bagian utama tersebut mencakup perangkat lunak (*software*), perangkat keras (*hardware*), infrastruktur, dan Sumber daya manusia (SDM) yang terlatih.”

Sutabri (2012:22) menjelaskan, “Sistem informasi adalah suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi organisasi yang bersifat manajerial dalam kegiatan strategi dari suatu organisasi untuk menyediakan laporan-laporan yang diperlukan.”



2.2.3. Pengertian Pendaftaran

Suharso dan Retnoningsih (2014:112) menjelaskan, “Pendaftaran adalah pencatatan nama, alamat, dan sebagainya dalam daftar perihal mendaftar (mendaftarkan).”

Hizair (2013:139) menjelaskan, “Pendaftaran adalah kegiatan mencatat nama, alamat dsb dalam daftar perihal mendaftar.”

2.2.4. Pengertian Pembayaran

Suharso dan Retnoningsih (2014:79) menjelaskan, “Pembayaran adalah perbuatan membayarkan atau membayar.”

2.2.5. Pengertian Umroh

Arifin (2013:15) menjelaskan, “UMRAH ialah berkunjung ke Baitullah untuk melakukan thawaf, sa’i dan cukur demi mengharap ridha Allah.”

Maksum (2013:42) menjelaskan, “Umrah menurut bahasa bermakna ziarah (berkunjung), dan menurut syara’, umrah adalah menziarahi Ka’bah, melakukan thawaf di sekelilingnya, sa’i antara Shafa dan Marwah dan mecukur atau menggunting rambut.”

Idjal (2014:26) menjelaskan, “Umrah secara etimologis berarti ziarah atau mengunjunginya, yaitu mengunjungi Baitullah untuk beribadah kepada Allah Swt dengan syarat-syarat yang telah ditetapkan.”

2.2.6. Pengertian Sistem Informasi Pendaftaran dan Pembayaran Online Jamaah Umroh pada PT Fikri Umroh Tour dan Travel.

Merupakan suatu sistem yang dapat mempermudah dalam pengolahan informasi mengenai pendaftaran dan pembayaran jamaah umroh pada PT Fikri Umroh Tour dan Travel.



2.3. Teori Khusus

2.3.1. Pengertian DFD (*Data Flow Diagram*)

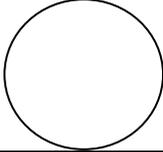
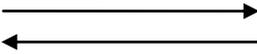
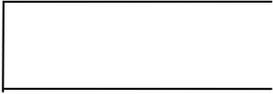
Sutabri (2012:116-120), mengatakan “Secara umum *Data Flow Diagram* adalah suatu *network* yang menggambarkan suatu sistem automat/komputerisasi, manualisasi, atau gabungan dari keduanya, yang penggambarannya disusun dalam bentuk kumpulan komponen sistem yang saling berhubungan sesuai dengan aturan mainnya.

Keuntungan penggunaan DFD adalah memungkinkan untuk menggambarkan sistem dari level yang paling tinggi kemudian menguraikannya menjadi level yang lebih rendah (dekomposisi). Sedangkan kekurangan penggunaan DFD adalah tidak menunjukkan proses pengulangan (*looping*), proses keputusan dan proses perhitungan.

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*)

No	Nama dan Simbol	Keterangan
1.	<i>External Entity</i> 	Digunakan untuk menggambarkan asal atau tujuan data.
2.	Proses 	digunakan untuk proses pengolahan atau transformasi data.
3.	<i>Data Flow</i> 	digunakan untuk menggambarkan aliran data yang berjalan.
4.	<i>Data Store</i> 	digunakan untuk menggambarkan data <i>flow</i> yang sudah disimpan atau diarsipkan.

(Sumber : Sutabri (2012:116-120)).



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 keluar.

2.3.1.3. 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

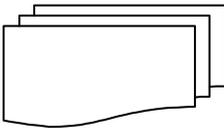
Diagram ini dibuat untuk menggambarkan arus data secara lebih mendetail lagi dari tahapan proses yang ada di dalam diagram nol.

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:

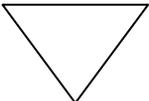
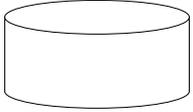
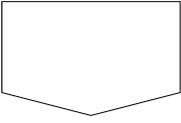
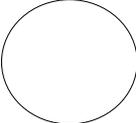
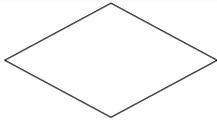
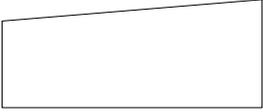
Tabel 2.2. Simbol-simbol *Blockchart*

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*

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.
10.		Terminasi yang menandakan awal dan akhir dari suatu aliran.
11.		Pengambilan keputusan (<i>decision</i>).
12.		Layar peraga (<i>monitor</i>).
13.		Pemasukan data secara manual.

(Sumber:Kristanto(2008:75)).



2.3.3. Pengertian ERD (*Entity Relational Diagram*)

Al Fatta (2007:121-124) menjelaskan, ”ERD (*Entity Relationship Diagram*) adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis.”

1. Entitas

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

2. Relationship

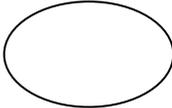
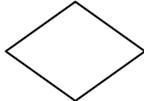
- a. Hubungan antar 2 entitas.
- b. Entitas pertama dalam relationship disebut entitas induk, entitas kedua disebut sebagai entitas anak.
- c. Relationship harus memiliki nama yang berupa kata kerja.
- d. Relationship berjalan 2 arah.

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)

No	Simbol	Keterangan
1.		Entity Name, adalah Orang, tempat, atau benda dimana data akan dikumpulkan.
2.		Atribut, adalah properti dari entitas harus digunakan oleh minimal 1 proses bisnis dipecah dalam detail.
3.		Relationship, menunjukkan hubungan diantara sejumlah entitas.
4.		Garis, sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

(Sumber: Al Fatta (2007:121-124)).



2.3.4. Pengertian *Flowchart*

EWolf (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:

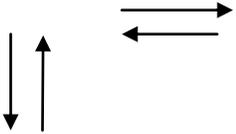
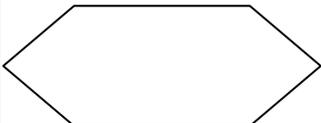
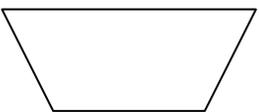
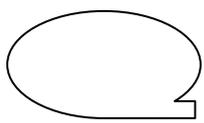
Tabel 2.4. Simbol-simbol *Flowchart*

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



Lanjutan :

Tabel 2.4. Simbol-simbol *Flowchart*

No.	Gambar	Arti
7.		Simbol untuk menghubungkan antar proses atau antar simbol
8.		Simbol yang menyatakan piranti keluaran, seperti layar monitor, printer, dll
9.		Simbol yang mendefenisikan 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

Kristanto (2008:72), menjelaskan "Kamus data adalah kumpulan elemen-elemen atau simbol-simbol yang digunakan untuk membantu dalam penggambaran atau pengidentifikasian setiap *field* atau *file* di dalam system."

Tabel 2.5. Simbol-simbol dalam Kamus Data

No	Notasi	Arti
1.	=	Terdiri atas
2.	+	AND atau dan
3.	()	Opsional
4.	[]	Memilih salah satu alternative
5.	**	Komentar
6.	@	identifikasi atribut kunci
7.		Pemisah alternatif simbol []

(Sumber:Kristanto(2008:72)).

2.4. Teori Program

2.4.1. Basis Data (*Database*)

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.

Madcoms (2011:260), "*Database* berfungsi sebagai penampung data yang diinputkan melalui *form website*. Selain itu dapat juga dibalik dengan menampilkan data yang tersimpan dalam database ke dalam halaman *website*."

2.4.2. Sekilas Tentang *PHP*

2.4.2.1. Pengertian *PHP* (*Hypertext Preprocessor*)

Madcoms (2011:228), *PHP* merupakan singkatan dari *Hypertext Preprocessor*. Pada awalnya *PHP* merupakan kependekatan dari *Personal Home Page* (situs personal) dan *PHP* itu sendiri pertama kali dibuat oleh Rasmus



Lerdorf pada tahun 1995, dan pada saat PHP masih bernama FI (*Form Interpreter*), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data *form* dari *web*. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum. PHP adalah sebuah bahasa *scripting* yang terpasang pada HTML. Sebagian besar sintaksnya mirip dengan bahasa pemrograman C, Java, asp dan Perl, ditambah beberapa fungsi PHP yang spesifik mudah dimengerti.

PHP digunakan untuk membuat tampilan *web* menjadi lebih dinamis. PHP bisa menampilkan atau menjalankan beberapa *file* dalam 1 *file* dengan cara di-*include* atau *require*. PHP itu sendiri suda dapat berinteraksi dengan beberapa database walaupun dengan kelengkapan yang berbeda, yaitu seperti: DBM, FilePro (Personic, Inc), Informix, Ingres, InterBase, Microsoft Access, MSSQL, MySQL, Oracle, PostgrSQL, dan Sybase.

Dari uraian di atas maka dapa diambil 4 point utama tentang PHP:

- a. PHP adalah singkatan dari *Hypertext Preprocessor*.
- b. PHP adalah bahasa *scripting server-side*, artinya di jalankan di *server*, kemudian *outputnya* dikirim ke *clinet (browser)*.
- c. PHP digunakan untuk membuat aplikasi web.
- d. PHP mendukung banyak database (MySQL, Informix, Oeacle, Sybase, Solid, PostgrSQL, Generic ODBC, dll).

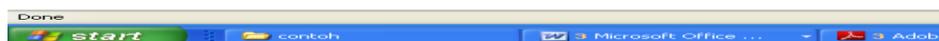
2.4.2.2. Skrip PHP

Berikut adalah kode *HTML* (disimpan dengan ekstensi *.htm* atau *.html*):

```
<HTML>
<HEAD>
<TITLE> Latihan Pertama</TITLE>
</HEAD>
<BODY>
Selamat Belajar PHP<BR>
</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*

Kadir (2008:02), “*MySQL* atau dibaca mai-se-Kyu-el merupakan yang tergolong sebagai DBMS (*Database Management System*) yang bersifat *Open Source*. *Open Source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dibuat untuk membuat *MySQL*), selain bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi.”

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 tabel “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 tabel “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

Kadir (2008:48), menjelaskan “Tipe numeris adalah tipe yang menyatakan bilangan. Secara garis besar, numeris dikelompokkan menjadi tipe bilangan bulat dan tipe bilangan pecahan.”

Kadir (2008:49), 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



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(<i>p</i>)	Tipe bilangan pecahan. Dalam hal ini, <i>p</i> menyatakan jumlah presisi dalam bit.	4 byte jika $\leq p \leq 24$, 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:Kadir (2008:49)).

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

Kadir (2008:51), "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, $L \leq M \leq 65535$
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, $L \leq M \leq 255$
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, $L < 2^{32}$
10.	TINYTEXT	String teks dengan ukuran paling kecil	L+1 byte, $L < 2^8$



Lanjutan :

Tabel 2.8. Tipe String

No	Tipe Data	Keterangan	Kebutuhan Memori
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:Kadir(2008:51)).



2.4.5. Sekilas Tentang Adobe Dreamweaver CS6

2.4.5.1. Pengertian Adobe Dreamweaver CS6

Madcoms (2011:3), “Adobe Dreamweaver CS6 adalah *software* terkemuka untuk membangun dan mengedit web dengan menyediakan kemampuan visual dan tingkat kode, yang dapat digunakan untuk membuat *website* berbasis standar dan desain untuk *desktop*, *mobile*, *smartphone*, *tablet*, dan perangkat lainnya”.

Sadeli (2014:12) menjelaskan, “Dreamweaver merupakan suatu perangkat lunak web editor keluaran Adobe System yang digunakan untuk membangun dan mendesain suatu website dengan fitur-fitur yang menarik dan kemudahan dalam penggunaannya.”

2.4.5. Pengertian XAMPP

Nugroho (2013:1), XAMPP adalah paket program web lengkap yang dapat anda pakai untuk belajar pemrograman web, khususnya PHP dan MySQL, paket ini dapat didownload secara gratis dan legal.

Berikut ini adalah gambar tampilan awal XAMPP.



Gambar 2.2. Tampilan XAMPP

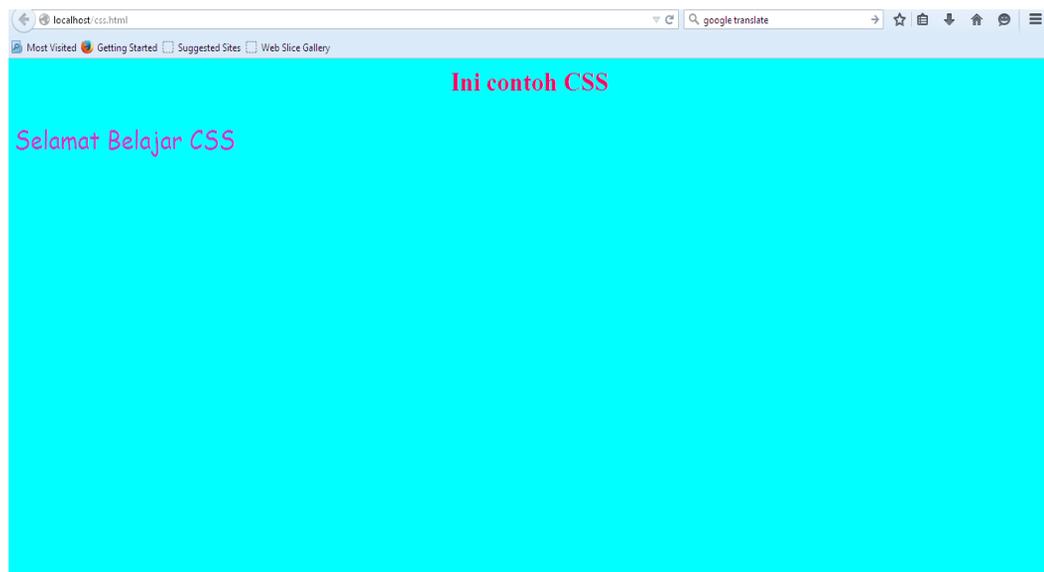


2.4.6. CSS (*Cascading Style Sheets*)

Enterprise (2007:95) menjelaskan, “Secara umum, CSS (*Cascading Style Sheet*) berfungsi untuk menjembatani kelemahan HTML saat kita harus menciptakan desain web yang kaya unsure, seperti desain link yang tanpa garis, tidak selalu berwarna biru, posisi teks yang bisa diatur sedemikian rupa tanpa harus menggunakan table dan sebagainya.”. Contoh cara penggunaan CSS sebagai berikut :

```
<html>
<head>
<style>
body {
background-color: #d0e4fe;
}
h1 {
color: orange;
text-align: center; }
p {
font-family: "TimesNewRoman";
font-size: 20px;
color: blue; }
</style>
</head>
<body>
<h1> Ini contoh CSS</h1>
<p>ini bagian dari paragraph.</p>
</body>p
</html>
```

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



Gambar 2.3. Tampilan CSS (Cascading Style Sheets)

Contoh berikutnya:

```

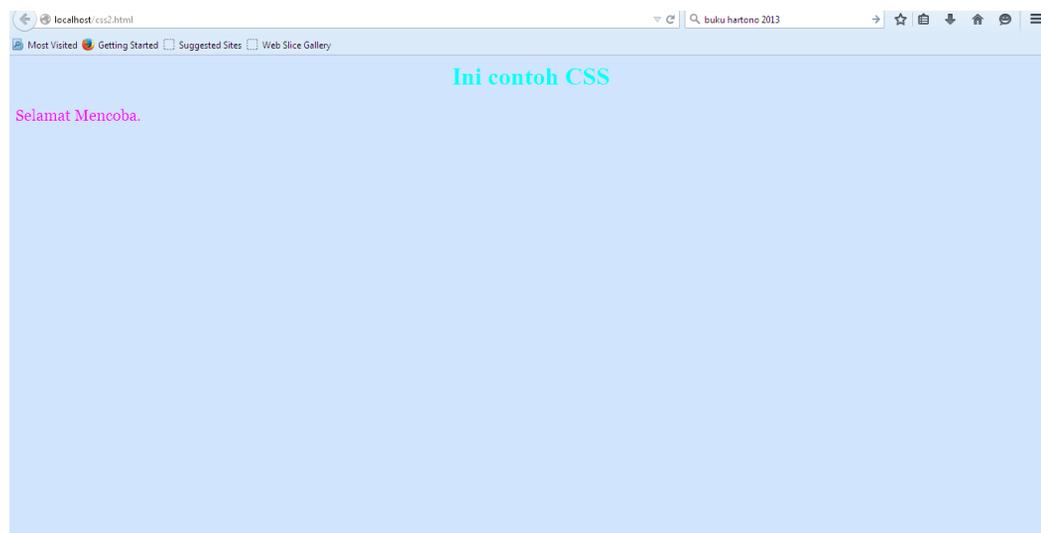
<html>
<head>
<style>
body {
background-color: #d0e4fe;
}
h1{
color: #0FF;
text-align: center; }
p{
font-family: Georgia, "Times New Roman", Times, serif;
font-size: 20px;
color:#F0F; }
</style>
</head>

```



```
<body>  
<h1> Ini contoh CSS</h1>  
<p>Selamat Mencoba.</p>  
</body>  
</html>
```

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



Gambar 3.4 Tampilan CSS (*Cascading Style Sheets*)