



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Aplikasi

Asropudin (2013:6), “*Application* atau aplikasi merupakan *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word, Microsoft Excel”.

Sutarman (2012:285), “Program aplikasi adalah program-program yang dibuat oleh suatu perusahaan komputer untuk para pemakai yang beroperasi dalam bidang-bidang umum, seperti toko, penerbitan, komunikasi, penerbangan, perdagangan, dan sebagainya”.

Jadi aplikasi atau program aplikasi adalah *software* atau program-program yang dibuat untuk membantu para pemakai dalam mengerjakan tugas-tugas tertentu.

2.1.2. Pengertian Data

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

Sutarman (2012:3), “Data adalah fakta dari sesuatu pernyataan yang berasal dari kenyataan, dimana pernyataan tersebut merupakan hasil pengukuran atau pengamatan. Data dapat berupa angka-angka, huruf-huruf, simbol-simbol khusus, atau gabungan darinya”.

Secara keseluruhan, data merupakan kumpulan angka-angka, huruf-huruf, simbol-simbol khusus atau gabungan darinya yang belum memiliki arti sehingga dapat diolah untuk menghasilkan suatu informasi.

2.1.3. Pengertian Pengolahan Data

Ladjamudin (2013:9), “Pengolahan data adalah masa atau waktu yang digunakan untuk mendeskripsikan perubahan bentuk data menjadi informasi yang memiliki kegunaan”.



Sutarman (2012:4) mengatakan, “Pengolahan data (data processing) adalah proses perhitungan/transformasi data *input* menjadi informasi yang mudah dimengerti ataupun sesuai dengan yang diinginkan”.

Jadi, dapat disimpulkan bahwa pengolahan data adalah proses perhitungan/transformasi data menjadi informasi yang memiliki kegunaan.

2.1.4. Pengertian Umroh

Nashrullah (2015:14) mengatakan, “Umroh adalah ibadah yang insya Allah mendatangkan keutamaan besar serta pahala agung bagi pelakunya, jika dilakukan secara ikhlas dan semata-mata mengharap ridha-Nya”.

Jaelani (2015:26) mengatakan, “Umroh adalah menziarahi Ka’bah, dengan tujuan untuk melaksanakan ibadah dengan niat untuk memperoleh ridho Allah SWT melalui ketentuan rukun dan syarat-syarat umroh itu sendiri.

Jadi, dapat disimpulkan bahwa umroh adalah ibadah dengan niat untuk memperoleh ridho dari Allah dan mendatangkan keutamaan besar serta pahala bagi yang melaksanakannya.

2.1.5. Pengertian CV

Marwoto (2014:27) mengatakan, “*CV* adalah persekutuan yang didirikan oleh satu atau beberapa orang yang menyetorkan modalnya dengan tanggung jawab terbatas sebesar modal yang disetorkan itu dan tidak bersedia memimpin (disebut sekutu komanditer), dengan satu atau beberapa orang yang bertindak sebagai pemimpin perusahaan yang bertanggung jawab tidak terbatas atas kekayaan pribadinya (sekutu komplementer).

Suhendi (2014:43) mengatakan, “Persekutuan komanditer (*commanditaire venotnnooschap* atau *CV*) adalah suatu persekutuan yang didirikan oleh dua orang atau lebih.

Jadi, dapat disimpulkan bahwa *CV* adalah persekutuan yang didirikan dua orang atau lebih yang menyetorkan modalnya dengan tanggung jawab terbatas (sekutu komanditer) dan tidak terbatas (sekutu komplementer) atas kekayaan pribadinya.



2.1.6. Pengertian Web

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

Kadir (2008:356), “Aplikasi *web* atau aplikasi berbasis *web* (*web-based-application*) adalah aplikasi yang dijalankan melalui *browser*”.

Jadi, *website* atau aplikasi *WEB* adalah sebuah halaman berisikan informasi, iklan serta program interaksi yang dijalankan melalui *browser*.

2.1.7. 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*”.

Sutarman (2012:283), “Internet adalah kumpulan dari berbagai macam jenis komputer yang saling terhubung dengan menggunakan media telekomunikasi (telepon, wireless, satelit, dan sebagainya) dengan jangkauan seluruh dunia/global”.

Jadi, internet adalah kumpulan berbagai komputer dan layanan servis yang saling terhubung menggunakan media telekomunikasi dengan jangkauan seluruh dunia/global.

2.1.8. Metodologi Pengembangan Sistem

Sukamto et.al (2013:26) SDLC atau *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik. SDLC memiliki beberapa model dalam penerapan tahapan prosesnya, salah satunya adalah model *waterfall*.

Sukamto dan Shalahuddin (2013:26) model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup



klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut mulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

2.1.8.1. Analisis

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

2.1.8.2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

2.1.8.3. Pembuatan Kode Program

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

2.1.8.4. Pengujian

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

2.1.8.5. Pemeliharaan

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya



kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.1.8.6. Studi Kelayakan

Al Fatta (2007:75), untuk memastikan usulan suatu proposal pendahuluan proyek bisa diteruskan menjadi proyek yang menguntungkan maka proposal proyek harus dievaluasi kelayakannya dari berbagai segi kelayakan, diantaranya:

1. Kelayakan teknis

Kelayakan teknis menyoroti kebutuhan sistem yang telah disusun dari aspek teknologi yang akan digunakan. Jika teknologi yang dikehendaki untuk pengembangan sistem merupakan teknologi yang mudah didapat, murah, dan tingkat pemakaiannya mudah, maka secara teknis usulan kebutuhan sistem bisa dinyatakan layak.

2. Kelayakan operasional

Untuk disebut layak secara operasional, usulan kebutuhan sistem harus benar-benar bisa menyelesaikan masalah yang ada di sisi pemesan sistem informasi. Disamping itu, informasi yang dihasilkan oleh sistem harus merupakan informasi yang benar-benar dibutuhkan oleh pengguna tepat pada saat pengguna menginginkannya.

3. Kelayakan ekonomi

Untuk menganalisis kelayakan ekonomi digunakan kalkulasi yang dinamakan *Cost Benefit Analysis* atau Analisis Biaya dan Manfaat. Adapun tujuan dari analisis biaya dan manfaat ini adalah untuk memberikan gambaran kepada pengguna apakah manfaat yang diperoleh dari sistem baru lebih besar dibandingkan dengan biaya yang dikeluarkan.

2.2. Teori Khusus

2.2.1. Pengertian DFD (Data Flow Diagram)

Berikut ini beberapa definisi tentang DFD yang disajikan oleh beberapa



buku Analisis dan perancangan sistem :

Al Fatta (2007 : 119) menjelaskan DFD merupakan diagram yang digunakan untuk menggambarkan proses-proses yang terjadi pada sistem yang akan dikembangkan.

Sukanto et.al (2014 : 70) menjelaskan DFD adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) atau keluaran (*output*).

Dari definisi diatas, dapat disimpulkan bahwa DFD adalah proses yang menggambarkan semua sistem dari data dengan aturan-aturan tertentu.

Al Fatta (2007 : 106) menyatakan bahwa ada beberapa elemen-elemen yang menyusun suatu DFD. Ada empat elemen yang menyusun suatu DFD, yaitu:

1. Proses

Aktivitas atau fungsi yang dilakukan untuk alasan bisnis yang spesifik, biasa berupa manual maupun terkomputerisasi.

2. Data Flow

Satu data tunggal atau kumpulan logis suatu data, selalu diawali atau berakhir pada suatu proses.

3. Data Store

Kumpulan data yang disimpan dengan cara tertentu. Data yang mengalir disimpan dalam *data store*. Aliran data di-*update* atau ditambahkan ke *data store*.

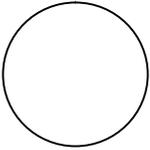
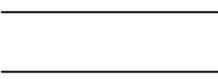
4. External Entity

Orang, organisasi, atau sistem yang berada di luar sistem tetapi berinteraksi dengan sistem.

Al Fatta (2007 : 107) menjelaskan tentang simbol atau lambang yang digunakan dalam membuat DFD menggunakan simbol De Marco dan Jourdan, yaitu :



Tabel 2.1. Simbol-simbol *Data Flow Diagram*

No.	Notasi	Keterangan
1.		<i>External Entity</i> , adalah kesatuan (entity) di lingkungan luar sistem dapat berupa orang, organisasi atau sistem lainnya yang ada di luar lingkungan, yang memberikan input dan output.
2.		<i>Proses</i> , kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses.
3.		<i>Data Flow</i> , menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem
4.		<i>Data Store</i> , merupakan simpanan dari data yang dapat berupa file, arsip, kotak, tabel, agenda atau buku, dan sebagainya.

Sumber : Al Fatta (2007 : 107)

Berikut ini penjelasan tentang tahapan *Data Flow Diagram* yaitu :

1. Diagram Konteks

Diagram ini dibuat untuk menggambarkan semua proses dalam 1 proses tunggal (proses 0). Diagram konteks juga menunjukkan semua entitas luar yang menerima informasi dari atau memberikan informasi ke sistem.

2. Diagram level Nol

Diagram ini dibuat untuk menunjukkan proses utama yang menyusun keseluruhan sistem.



3. Diagram Detail

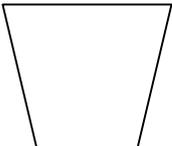
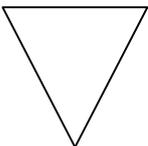
Diagram ini dibuat untuk menggambarkan arus data secara lebih mendetail lagi dari tahapan proses yang ada didalam diagram nol.

2.2.2. Pengertian Blockchart

Kristanto (2008 : 75) menjelaskan, ” *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.

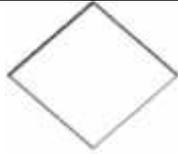
Simbol-simbol yang sering digunakan dalam *Blockchart* dapat dilihat pada tabel berikut ini :

Tabel 2.2. Simbol-simbol dalam *Block Chart* (Kristanto,2008:75-76)

No.	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan.
2.		Multi dokumen.
3.		Proses manual.
4.		Proses yang dilakukan oleh komputer.
5.		Menandakan dokumen yang diarsipkan (arsip manual)



Lanjutan Tabel 2.2. Simbol-simbol dalam *Block Chart* (Kristanto,2008:75-76)

6.		Data penyimpanan (<i>data storage</i>)
7.		Pengambilan keputusan (<i>decision</i>)
8.		Layar Peraga (<i>monitor</i>).
9.		Pemasukan data secara manual.

Sumber: Kristanto (2008:75-76)

2.2.3. Pengertian ERD (*Entity Relational Diagram*)

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

Berikut ini penjelasan tentang Diagram Hubungan Entitas :

1. Entitas

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

2. Atribut

- a. Informasi yang diambil tentang sebuah entitas.
- b. Hanya yang digunakan oleh organisasi yang dimasukan.
- c. Nama atribut harus merupakan kata benda.
- d. Kadang nama entitas diletakan depan nama atribut untuk ketelitian.

3. Relationship

- a. Hubungan antar 2 entitas.
- b. Entitas pertama dalam relationship disebut entitas induk, entitas kedua

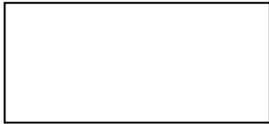
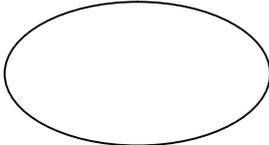
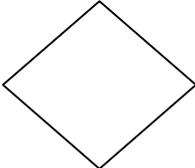


entitas anak.

- c. Relationship harus memiliki nama yang berupa kata kerja.
- d. Relationship berjalan 2 arah.

Berikut ini adalah penjelasan tentang notasi-notasi yang digunakan dalam ERD :

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.		Atribute, 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 : 122)

2.2.4. Pengertian Flowchart

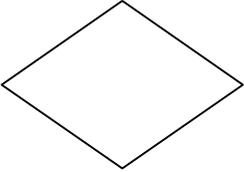
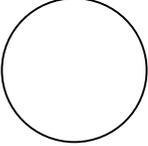
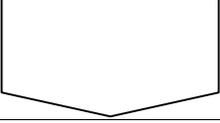
Ladamudin (2013:263) menyatakan, “*Flowchart* adalah bagan–bagan yang mempunyai arus yang menggambarkan langkah–langkah penyelesaian suatu masalah”.

Supardi (2013:58) menjelaskan Flowchart merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program.

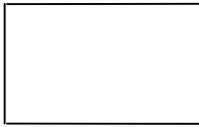


Berikut ini adalah simbol-simbol flowchart :

Tabel 2.4. Simbol-simbol Flowchart

No.	Simbol	Keterangan
1.		<i>Terminal Symbol</i> , digunakan untuk menunjukkan awal dan akhir dari program.
2.		<i>Preparation Symbol</i> , digunakan untuk memberikan nilai awal pada suatu variabel atau counter.
3.		<i>Decision Symbol</i> , digunakan untuk mewakili operasi perbandingan logika.
4.		<i>Processing Symbol</i> , digunakan untuk pengolahan aritmatika dan pemindahan data.
5.		<i>Connector Symbol</i> , digunakan untuk menunjukkan hubungan arus proses yang terputus masih dalam halaman yang sama.
6.		<i>Off page connector symbol</i> , digunakan untuk menunjukkan hubungan arus proses yang terputus masih dalam halaman yang sama.



7.		<i>Annotation Flag Symbol</i> , digunakan untuk memberikan keterangan-keterangan guna memperjelas simbol-simbol yang lain.
----	---	--

Sumber : Supardi (2013:58)

2.2.5. Pengertian Kamus Data

Sukamto et.al (2014 : 73) menjelaskan, ”kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) atau keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan).

Berikut ini penjelasan yang ada didalam kamus data :

Table 2.5 Simbol-simbol dalam Kamus Data

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

Sumber : Sukamto et.al (2014 : 74)

2.3. Teori Aplikasi

2.3.1 PHP

Kadir (2008:2), “PHP merupakan singkatan dari *PHP Hypertext Preprocessor*. Ia merupakan bahasa berbentuk skrip yang ditempatkan dalam server dan diproses di server. Hasilnyalah yang dikirimkan ke klien, tempat pemakai menggunakan *browser*”.

Winarno (2014:49) menjelaskan bahwa “*PHP* adalah bahasa script yang sangat cocok untuk pengembangan Web dan dapat dimasukkan ke dalam *HTML*.”



PHP dapat dikatakan sebagai sebuah server-side embedded script language, artinya sintak-sintak dan perintah program yang Anda tulis akan sepenuhnya dijalankan oleh server tetapi dapat disertakan pada halaman *HTML* biasa”.



Gambar 2.1 Tampilan Logo *PHP*

2.3.1.1 *Script PHP*

PHP adalah program yang fleksibel, artinya *script-script PHP* dapat Anda tuliskan pada sela-sela tag *HTML*. Karena *PHP* memiliki sifat yang dapat berinteraksi dengan program lain, maka kita tidak usah bingung dalam menuliskan skripnya. Jadi, kita dapat membuat sebuah program *PHP* yang di dalamnya berupa tag-tag *HTML* murni yang kemudian disisipi *script PHP*. Akan tetapi, *PHP* juga dapat berdiri sendiri sehingga tidak memerlukan *HTML* murni untuk meletakkan *script-scriptnya*.

Contoh *script*

```
<HTML>
<HEAD>
<TITLE> Contoh Awal </TITLE>
</HEAD>
<?php
    Print ("Avengers Bersatu!");
?>
<BODY>
</BODY>
```



</HTML>

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



Gambar 2.2. Contoh *Script PHP*

2.3.1.2 Teknik Penulisan *PHP*

Ada beberapa cara menuliskan *script PHP*:

- `<? ... Script PHP disini... ?>` atau
- `<?php ... Script PHP disini... ?>` atau
- `<% ... Script PHP disini... %>` atau
- `<SCRIPT language="php"> ... Script PHP disini... </SCRIPT>`

Jadi, semua kata dan *script* yang diletakkan pada daerah *script* akan dianggap sebagai perintah *PHP* sehingga jika terjadi kesalahan atau kata-katanya tidak sesuai dengan program akan dianggap salah dan akan mengakibatkan program yang kita buat menjadi *error*.

Adapun kriteria yang harus diperhatikan dalam penulisan *script PHP* adalah sebagai berikut:

- Setiap halaman yang mengandung *script PHP* harus disimpan dengan EXTENSI *PHP* sesuai dengan program *PHP* yang mendukungnya (nama_file.php, atau nama_file.php3, nama_file.php4).



- b. Setiap *script PHP* harus didahului dengan pembuka *PHP* (`<?php` dan lain-lain) dan kemudian diakhiri dengan penutup (`?>`).
- c. Setiap baris *script* harus didahului pernyataan cetak atau dalam program pascal kita sebut *write* atau *REM*, sedangkan dalam *PHP* pernyataan cetak dibedakan menjadi dua yaitu *Print* dan *Echo*. Adapun kriteria penulisannya sebagai berikut:

Print (“ Isi Perintah “)

Printf (“ Isi Perintah “)

Echo “ Isi Perintah “

Echo Perintah

- d. Setiap akhir baris perintah harus diakhiri dengan titik koma (;) misalnya kita akan menuliskan “Hello, Nama Saya Nayla”, maka *script* yang harus anda tuliskan adalah sebagai berikut:

<?php

Echo “Avengers Bersatu!”;

?>

- e. Semua variabel harus diberi tanda string dolar (\$) pada penulisan awalnya. Sebagai contoh, kita akan menuliskan nama sebagai variabel yang isinya Nayla, maka kita harus menuliskan sebagai berikut:

\$nama = Avengers;

- f. Penulisan *comment* atau keterangan didahului dengan pembuka `/*` dan diakhiri dengan `*/`. Biasanya kita menggunakan ini untuk memberikan *comment* yang berbentuk kalimat. Sebagai contoh, kita akan memberi keterangan penulisan nama diatas.

Contoh *Script*:

<?php

/* Untuk menuliskan komentar, biasanya komentar yang berbentuk kalimat */

Echo “Avengers Bersatu!”;

?>



- g. Selain menggunakan tanda */**, penulisan *comment* juga dapat menggunakan tanda *slash* ganda (*//*). Tanda ini hanya digunakan untuk menuliskan pesan yang hanya berisi satu baris saja.

Contoh *Script*:

```
<?php
// Untuk menuliskan komentar satu baris
Echo "Avengers Bersatu!";
?>
```

- h. Semua *script HTML* yang akan digabungkan dalam *script PHP* harus dihilangkan tanda petiknya (" "). Sebagai contoh, jika kita ingin memberi warna *background* pada halaman kita, maka perintah yang kita gunakan dalam *HTML* adalah `<BODY bgcolor="#000000">`. Akan tetapi, jika kita letakkan dalam *PHP*, maka *script* tersebut harus diubah menjadi `<BODY bgcolor=#000000>`. Untuk lebih jelasnya, lihatlah contoh berikut.

Script pada HTML:

```
.....
<BODY bgcolor="#000000">
<?php
Print ("Avengers Bersatu!");
?>
.....
```

Script pada PHP:

```
<?
Echo (" <BODY bgcolor=#3366FF? ");
Print ("Bersatulah Avengers!");
?>
```



2.3.2 Pengertian Basis Data (*Database*)

Menurut Al Fatta (2007 : 10), ” database adalah Kumpulan data dan informasi yang diorganisasikan sedemikian rupa sehingga mudah diakses pengguna sistem informasi”.

Sutarman (2012:15), “*database* merupakan sekumpulan *file* yang saling berhubungan dan terorganisasi atau kumpulan *record-record* yang menyimpan data dan hubungan diantaranya”.

Kadir (2008:3), “*database* berarti koleksi data yang saling terkait. Secara praktis, basis data dapat dianggap sebagai suatu penyusunan data yang terstruktur yang disimpan dalam media pengingat (*hard disk*) yang tujuannya adalah agar data tersebut dapat diakses dengan mudah dan cepat”.

2.3.3 MySQL

Kadir (2008:2), “*MySQL* (baca: mai-se-kyu-el) merupakan *software* yang tergolong *DBMS (Database Management System)* yang bersifat *Open Source*. *Open Source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat *MySQL*), selain tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa diperoleh dengan cara *men-download* (mengunduh) di internet secara gratis”.

Kelebihan dari *MySQL* adalah dapat digunakan untuk aplikasi *multi-user* (banyak pengguna) dan menggunakan bahasa *query* (permintaan) standar *SQL (Structured Query Language)*.

Kekurangan dari *MySQL* adalah untuk koneksi ke bahasa pemrograman *visual* seperti *VB*, *Delphi*, dan *Foxpro*, *MySQL* kurang *support*, karena koneksi ini menyebabkan *field* yang dibaca harus sesuai dengan koneksi dari program visual tersebut, dan ini yang menyebabkan *MySQL* jarang dipakai dalam program visual serta data yang ditangani belum begitu besar.

2.3.3.1 Fungsi-Fungsi MySQL

Untuk dapat menggunakan semua perintah *SQL* pada *PHP*, kita memerlukan fungsi bawaan *PHP (reserved word)* dengan beragam kemampuan, mulai dari menampilkan informasi, memanipulasi informasi, keperluan



kondisional, keperluan perulangan, sampai pengaksesan berbagai *database*, termasuk *MySQL*.

Berikut adalah daftar beberapa fungsi *PHP* yang secara khusus digunakan untuk mengakses dan mengelola *database MySQL*.

Tabel 2.6. Fungsi-fungsi *MySQL*

No	Nama Fungsi	Kegunaan	Bentuk Pemanggilan
1	<i>MySQL_connect()</i>	Membuat hubungan ke <i>database MySQL</i> yang terdapat pada suatu <i>host</i> .	<i>mysql_connect (host, nama_pemakai, password)</i>
2	<i>MySQL_close()</i>	Menutup hubungan ke <i>database MySQL</i> .	<i>mysql_close (pengenal_hubungan)</i>
3	<i>MySQL_select_db()</i>	Memilih <i>database</i>	<i>mysql_select_db (database, pengenal_hubungan)</i>
4	<i>MySQL_query()</i>	Mengeksekusi permintaan terhadap sebuah tabel atau sejumlah tabel.	<i>mysql_query (permintaan, pengenal_hubungan)</i>
5	<i>MySQL_db_query()</i>	Menjalankan suatu permintaan terhadap suatu <i>database</i> .	<i>mysql_db_query (database, permintaan, pengenal_hubungan)</i>
6	<i>MySQL_num_rows()</i>	Memperoleh jumlah baris dari suatu hasil permintaan (<i>query</i>) yang menggunakan <i>SELECT</i> .	<i>mysql_num_rows (pengenal_hasil)</i>
7	<i>MySQL_affected_rows()</i>	Memperoleh jumlah baris yang dikenai operasi <i>INSERT</i> , <i>DELETE</i> , dan <i>UPDATE</i> .	<i>mysql_affected_rows ([pengenal_hubungan])</i>
8	<i>MySQL_num_fields()</i>	Memperoleh jumlah kolom pada suatu hasil permintaan.	<i>mysql_num_fields (pengenal_hasil)</i>

Lanjutan Tabel 2.6. Fungsi-fungsi *MySQL*

9	<i>MySQL_fetch_row()</i>	Menghasilkan suatu array yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasil.	<i>mysql_fetch_row</i> (<i>pengenal_hasil</i>)
10	<i>MySQL_fetch_array()</i>	Mempunyai kegunaan serupa dengan <i>mysql_fetch_row()</i> . Hanya saja, setiap kolom akan disimpan dua kali pada array hasil.	<i>mysql_fetch_array</i> (<i>pengenal_hasil</i>)
11	<i>MySQL_fetch_field()</i>	Memperoleh informasi suatu kolom.	<i>mysql_fetch_field</i> (<i>pengenal_hasil</i> [, <i>nomor_kolom</i>])
12	<i>MySQL_data_seek()</i>	Memindah <i>pointer</i> pada suatu himpunan hasil supaya menunjuk ke baris tertentu.	<i>mysql_data_seek</i> (<i>pengenal_hasil</i> , <i>nomor_baris</i>)
13	<i>MySQL_field_seek()</i>	Memindah <i>pointer</i> pada suatu himpunan hasil supaya menunjuk ke kolom tertentu.	<i>mysql_data_seek</i> (<i>pengenal_hasil</i> , <i>nomor_kolom</i>)
14	<i>MySQL_create_db()</i>	Menciptakan <i>database MySQL</i> .	<i>mysql_create_db</i> (<i>database</i> [, <i>pengenal_hubungan</i>])
15	<i>MySQL_drop_db()</i>	Menghapus <i>database MySQL</i> .	<i>mysql_drop_db</i> (<i>database</i> [, <i>pengenal_hubungan</i>])
16	<i>MySQL_list_dbs()</i>	Menghasilkan daftar <i>database MySQL</i> .	<i>mysql_list_dbs</i> ([<i>pengenal_hubungan</i>])
17	<i>MySQL_list_tables()</i>	Memperoleh daftar nama tabel dalam suatu <i>database</i> .	<i>mysql_list_tables</i> (<i>database</i> [, <i>pengenal_hubungan</i>])
18	<i>MySQL_list_fields()</i>	Memperoleh daftar nama kolom dalam suatu <i>database</i> .	<i>mysql_list_fields</i> (<i>database</i> [, <i>pengenal_hubungan</i>])

Sumber: Kadir (2008:360-379)



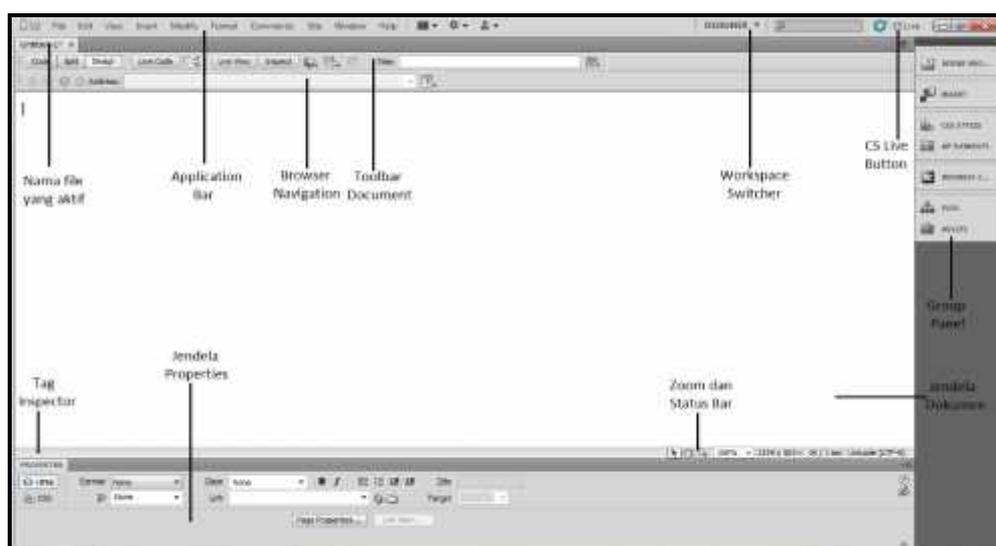
2.3.4 Adobe Dreamweaver CS5

Madcoms (2011:2), “*dreamweaver* adalah sebuah *HTML editor* profesional untuk mendesain *Web* secara *visual* dan mengelola situs atau halaman *Web*”. Pada *dreamweaver CS5*, terdapat beberapa kemampuan bukan hanya sebagai *software* untuk desain *Web* saja tetapi juga untuk menyunting kode serta pembuatan aplikasi *Web* dengan menggunakan berbagai bahasa pemrograman *Web*.



Gambar 2.3. Logo *Dreamweaver CS5*

Tampilan utama ruang kerja *Adobe Dreamweaver CS5* adalah sebagai berikut:



Gambar 2.4. Tampilan Ruang Kerja *Adobe Dreamweaver CS5*



1. *Application Bar*

Berada di bagian paling atas jendela aplikasi *Dreamweaver CS5*. Baris ini berisi tombol *workspace (workspace switcher)*, *CS Live*, menu dan aplikasi lainnya.



Gambar 2.5. Tampilan *Application Bar*

2. *Toolbar Document*

Berisi tombol-tombol yang digunakan untuk mengubah tampilan jendela dokumen, sebagai contoh tampilan *Design* atau tampilan *Code*. Juga dapat digunakan untuk operasi-operasi umum, misalnya untuk melihat hasil sementara halaman *web* pada jendela *browser*.



Gambar 2.6. Tampilan *Toolbar Document*

3. Jendela Dokumen

Adalah lembar kerja tempat membuat dan mengedit desain halaman *web*.



Gambar 2.7. Jendela Dokumen

4. *Workspace Switcher*

Digunakan untuk mengubah tampilan ruang kerja (*workspace*) *Adobe Dreamweaver CS5*.



Gambar 2.8. Tampilan *Workspace Switcher*



5. Panel Groups

Merupakan kumpulan-kumpulan panel yang saling berkaitan, panel-panel ini dikelompokkan pada judul-judul tertentu berdasarkan fungsinya. Panel ini digunakan untuk memonitor dan memodifikasi pekerjaan.



Gambar 2.9. Tampilan *Panel Groups*

6. Tag Selector

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.



Gambar 2.10. *Tag Selector*

7. Panel Property

Digunakan untuk melihat dan mengubah berbagai properti objek atau teks pada jendela *design*.

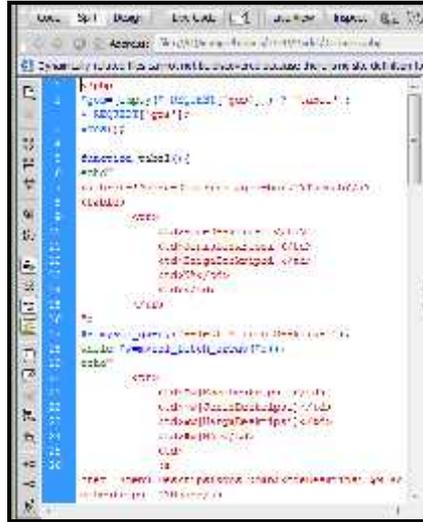


Gambar 2.11. Tampilan Jendela *Properties*



8. *Toolbar Coding*

Berisi tombol-tombol yang digunakan untuk melakukan operasi kode-kode standar. Jendela ini hanya akan tampil pada jendela *Code* dan *Split*.



Gambar 2.12. Tampilan *Toolbar Coding*

9. *Panel Insert*

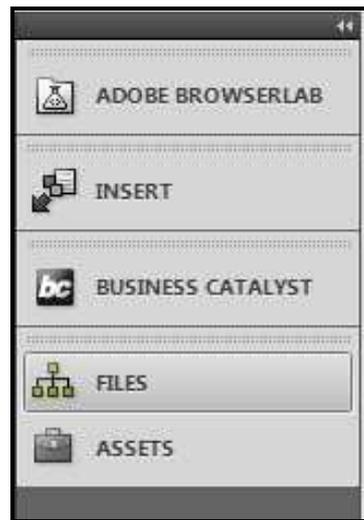
Berisi tombol-tombol untuk menyisipkan berbagai jenis objek, seperti *image*, tabel, atau objek media ke dalam jendela dokumen.



Gambar 2.13. Tampilan *Panel Insert*

10. Panel Groups

Kumpulan panel yang saling berkaitan satu sama lainnya yang dikelompokkan dibawah satu judul, seperti: *Insert*, *Files*, *CSS Styles*, dan sebagainya.



Gambar 2.14. Tampilan *Panel Groups*

11. Panel Files

Digunakan untuk mengatur *file-file* dan *folder-folder* yang membentuk situs *Web* Anda.



Gambar 2.15. Tampilan *Panel Files*