



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

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

Menurut Wahyudi (2012:3), komputer adalah peralatan (*device*) yang menerima data (*input*) dan menyimpan (*storage*) kemudian diproses (*process*) untuk menghasilkan data dalam bentuk lain (*output*).

Menurut Sutarman (2012:86), komputer adalah suatu rangkaian peralatan elektronik yang bekerja secara bersama-sama. Komputer dapat melakukan rangkaian pekerjaan secara otomatis melalui instruksi (program) yang diberikan, dan alat pengolah data menjadi informasi melalui proses tertentu.

2.1.2. Pengertian Sistem

Hartono (2013:9) mengemukakan, sistem yakni suatu benda atau entitas (yaitu himpunan dari berbagai bagian atau komponen), dan sekaligus juga suatu proses atau metode atau cara untuk mencapai tujuan (yaitu saling berhubungan secara terorganisasi berdasar fungsi-fungsinya).

Menurut Ladjamudin (2013:6), sistem adalah suatu bentuk integrasi antara satu komponen dengan komponen lainnya.

Menurut Sutarman (2012:86), sistem adalah kumpulan dari bagian-bagian (subsistem) yang terkait menjadi satu bentuk mekanisme kerja yang memberikan fungsi dan manfaat tertentu.



2.1.3. Pengertian Basis Data (*Database*)

Kristanto (2008:79) menjelaskan, basis data (*database*) adalah kumpulan data yang dapat digambarkan sebagai aktivitas dari satu atau lebih organisasi yang berelasi.

Menurut Ladjamudin (2015:130), *database* adalah kumpulan file yang saling terintegrasi, yang dapat diakses oleh siapapun dengan *software* aplikasi.

Sutarman (2012:15) mengemukakan, *database* adalah sekumpulan file yang saling berhubungan dan terorganisasi atau kumpulan *record-record* yang menyimpan data dan hubungan di antaranya.

2.1.4. Pengertian Aplikasi

Menurut Asropudin (2013:6), aplikasi (*application*) adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu.

Menurut Sutabri (2012:148), aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya.

Hendrayudi (2009:143) menjelaskan, aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (khusus).

2.1.5. Pengertian Monitoring

Monitoring adalah pemantauan yang dapat dijelaskan sebagai kesadaran (*awareness*) tentang apa yang ingin diketahui, pemantauan berkadar tingkat tinggi dilakukan agar dapat membuat pengukuran melalui waktu yang menunjukkan pergerakan ke arah tujuan atau menjauh dari itu.

<http://id.wikipedia.org/wiki/Monitoring>

2.1.6. Pengertian Pembebanan

Pembebanan adalah *Nomina (kata benda)* proses, cara, perbuatan membebani atau membebankan sesuatu.

<http://kamus.cektkp.com/pembebanan/>



2.1.7. Pengertian Trafo Distribusi

Trafo Distribusi adalah merupakan suatu komponen yang sangat penting dalam penyaluran tenaga listrik dari gardu distribusi ke konsumen.

<http://muhamadrizki Fauzika dila.blogspot.com/2012/06/trafo-distribusi.html>

2.1.8. Pengertian Perseroan Terbatas (PT)

Berdasarkan Pasal 1 UUPT No. 40/2007 pengertian Perseroan Terbatas (Perseroan) adalah badan hukum yang merupakan persekutuan modal, didirikan berdasarkan perjanjian, melakukan kegiatan usaha dengan modal dasar yang seluruhnya terbagi dalam saham, dan memenuhi persyaratan yang ditetapkan dalam undang-undang ini serta peraturan pelaksanaannya.

2.1.9. Pengertian PLN

Perusahaan Listrik Negara (disingkat **PLN**) adalah sebuah BUMN yang mengurus semua aspek kelistrikan yang ada di Indonesia.

http://id.wikipedia.org/wiki/PT._Perusahaan_Listrik_Negara

2.1.10. Pengertian Aplikasi *Monitoring* Pembebanan Trafo PT PLN WS2JB

Rayon Ampera

Aplikasi *Monitoring* Pembebanan Trafo PT PLN WS2JB Rayon Ampera adalah suatu program berbasis *web* yang menggunakan bahasa pemrograman *PHP* dan *MySQL* sebagai *database*, yang digunakan untuk memantau proses pemakaian trafo distribusi agar tidak terjadi kerusakan trafo yang menyebabkan gangguan listrik konsumen PT PLN.

2.2. Teori Khusus

2.2.1. *Data Flow Diagram* (DFD)

Kristanto (2008:61) mengemukakan, *data flow diagram*/DFD adalah suatu model logika atau proses yang dibuat untuk menggambarkan dari mana asal data dan ke mana tujuan data yang keluar dari sistem, di mana data disimpan, proses



apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.


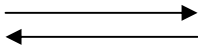
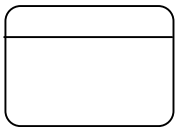
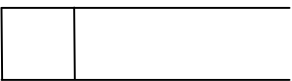
Menurut Sukamto dan Shalahuddin (2013:70), *data flow diagram* (DFD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi yang di (i)aplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).

Menurut Al-Fatta (2007:107), *data flow diagram*/DFD merupakan diagram yang digunakan untuk menggambarkan proses-proses yang terjadi pada sistem yang akan dikembangkan. Dengan model ini, data-data yang terlibat pada masing-masing proses dapat diidentifikasi.

Ada beberapa simbol DFD yang dipakai untuk menggambarkan data beserta proses transformasi data, antara lain:

1. Teknik Gane dan Sarson

Tabel 2.1. Simbol- Simbol *Data Flow Diagram* Gane & Sarson


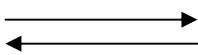
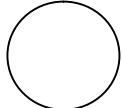

Simbol	Keterangan
	<i>External Entity</i> , yaitu merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan <i>input</i> atau menerima <i>output</i> dari sistem.
	<i>Data Flow</i> , digunakan untuk menggambarkan aliran data dari satu proses ke proses lainnya.
	<i>Process</i> , digunakan untuk mentransformasikan data secara umum.
	<i>Data Store</i> , digunakan untuk menyimpan data seperti: suatu <i>file</i> , suatu arsip, suatu kotak, suatu tabel dan suatu agenda.

(Sumber: Hanif Al Fatta, 2007:107)



2. Teknik Yourdan Dan De Marco

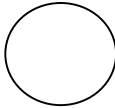
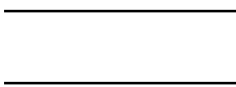
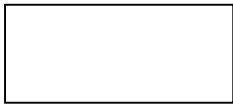
Tabel 2.2. Simbol- Simbol *Data Flow Diagram* Yourdan dan De Marco

Simbol	Keterangan
	<i>External Entity</i> , yaitu merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan <i>input</i> atau menerima <i>output</i> dari sistem.
Simbol	Keterangan
	<i>Data Flow</i> , digunakan untuk menggambarkan aliran data dari satu proses ke proses lainnya.
	<i>Process</i> , digunakan untuk mentransformasikan data secara umum
	<i>Data Store</i> , digunakan untuk menyimpan data seperti: suatu <i>file</i> suatu arsip, suatu kotak, suatu tabel dan suatu agenda.


(Sumber: Hanif Al Fatta, 2007:108)

Ada beberapa simbol DFD yang dipakai untuk menggambarkan data beserta proses transformasi data, antara lain:

Tabel 2.3. Simbol- Simbol *Data Flow Diagram* menurut Sukamto dan Shalahuddin

Notasi	Keterangan
	Proses atau fungsi atau prosedur. Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.
	<i>Field</i> atau basis data atau penyimpanan (<i>storage</i>). Nama yang diberikan pada sebuah penyimpanan biasanya berupa kata benda.
	<i>Entitas</i> atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang. Nama yang diberikan biasanya berupa kata benda.



Simbol	Keterangan
	Aliran data; merupakan data yang dikirim antarproses, dari penyimpanan ke proses atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>). Nama yang diberikan biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data_siswa” atau tanpa kata data misalnya “siswa”.

(Sumber: Sukamto dan Shalahuddin, 2013:71)

2.2.2. Kamus Data (*Data Dictionary*)

Menurut Sukamto dan Shalahuddin (2013:73), kamus data (*data dictionary*) adalah kumpulan daftar elemen data yang mengalir pada sistem sehingga masukan dan keluaran dapat dipahami secara umum (memiliki standar cara penulisan).

Kristanto (2008:72) mengemukakan, kamus data adalah kumpulan elemen-elemen atau simbol-simbol yang digunakan untuk membantu dalam penggambaran atau pengidentifikasian setiap *field* atau file di dalam sistem.

Adapun simbol-simbol dalam kamus data adalah sebagai berikut:

Tabel 2.4. Simbol-simbol Kamus Data (*Data Dictionary*)

Simbol	Keterangan
=	Disusun, terdiri atas, mendefinisikan, diuraikan menjadi, artinya
+	Dan
()	Opsional/Pilihan (boleh Ya atau Tidak)
[]	Salah satu alternative
*	Komentar, catatan, keterangan
@	Identifikasi atribut kunci, petunjuk (<i>key field</i>)
	Pemisah alternatif simbol []

(Sumber: Andri Kristanto, 2008:66)



3.2.3. Pengertian *Flowchart*

eWolf (2011:16) Menjelaskan, *flowchart* adalah simbol-simbol pekerjaan yang menunjukkan bagan aliran proses yang saling terhubung. Jadi setiap simbol *flowchart* melambangkan pekerjaan dan instruksinya. *Flowchart* digunakan untuk mempermudah penyusunan program. Dengan menggunakan *flowchart*, logika pemrograman lebih dapat dipahami dan dianalisis, sehingga dapat menentukan kode-kode pemrograman yang sesuai dengan pekerjaannya.

Berikut adalah beberapa simbol standar yang sering digunakan dalam pemrograman komputer, yaitu:

Tabel 2.5. Simbol-simbol *Flowchart*

Simbol	Nama Simbol	Keterangan
	Garis Alir (<i>Flow Line</i>)	Simbol untuk menghubungkan antar proses atau antar simbol
	<i>Connector</i>	Simbol konektor untuk menyambung proses pada lembar kerja yang sama
	<i>Offline Connector</i>	Simbol konektor untuk menyambung proses pada lembar kerja yang berbeda
	<i>Process</i>	Simbol pemrosesan yang terjadi pada sebuah alur kerja
	<i>Manual</i>	Simbol yang mendefinisikan proses yang dilakukan secara manual
	<i>Decision</i>	Simbol untuk memutuskan proses lanjutan dari kondisi tertentu
	<i>Terminal</i>	Simbol <i>start</i> atau <i>end</i> yang mendefinisikan awal atau akhir dari sebuah <i>flowchart</i>
	<i>Input/Output</i>	Simbol <i>input/output</i> yang mendefinisikan masukan dan keluaran proses



Simbol	Nama Simbol	Keterangan
	<i>Magnetic Tape</i>	Simbol masukan atau keluaran dari atau ke sebuah pita magnetik
	<i>Document</i>	Simbol masukan atau keluaran dari atau ke sebuah dokumen
	<i>Display</i>	Simbol yang menyatakan piranti keluaran, seperti layar monitor, printer, dll
	<i>Data Storage</i>	Simbol database atau basis data
	<i>Sub Program</i>	Simbol yang menyatakan bagian dari program (sub program)

(Sumber: eWolf Community, 2012:17)


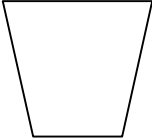

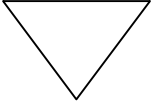
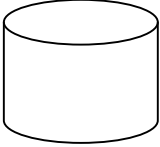

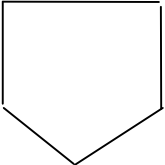
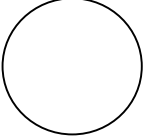

2.2.4. Pengertian *Blockchart*

Menurut Kristanto (2008:75), *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:

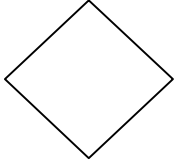


Tabel 2.6. Simbol-simbol dalam *Blockchart*

Simbol	Keterangan
	Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan



Simbol	Keterangan
	Multi dokumen
	Proses manual.
	Proses yang dilakukan oleh komputer.
	Menandakan dokumen yang diarsifkan(arsif manual)
	Data penyimpanan(<i>data storage</i>)
	Proses apa saja yang tidak terdefinisi termasuk aktifitas fisik
	Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang ain
	Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama
	Terminasi yang menandakan awal dan akhir dari suatu aliran



Simbol	Keterangan
	Pengambilan keputusan (<i>decision</i>)
	Layar peraga (<i>monitor</i>)
	Pemasukan data secara manual

(Sumber: Andri Kristanto, 2008:75)

2.2.5. Entity Relational Diagram (ERD)


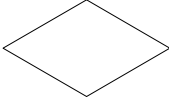


Fatta (2007:123) mengemukakan, *entity relationship diagram*/ERD adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis. Dalam entitas digunakan untuk menghubungkan antar entitas yang sekaligus menunjukkan antar data.

1. Entitas, bisa berupa orang, kejadian, 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.



Simbol-simbol yang digunakan dalam ERD, yaitu:

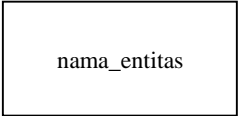
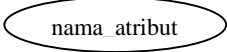
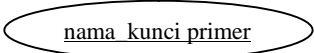
Tabel 2.7. Simbol-simbol *Entity Relationship Diagram (ERD)*

Nama Simbol	Simbol	Keterangan
<i>Entitas</i>		Orang, tempat atau benda memiliki nama tunggal.
<i>Relationship</i>		Menunjukkan hubungan antar 2 entitas. Dideskripsikan dengan kata kerja.
<i>Atribut</i>		Properti dari entitas harus digunakan oleh minimal 1 proses bisnis dipecah dalam detail.
<i>Link</i>		Penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya.

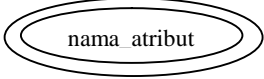
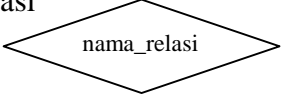
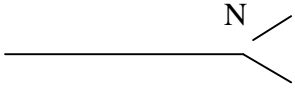
(Sumber: Hanif Al Fatta, 2007:124)

Simbol-simbol yang digunakan dalam ERD, yaitu:

Tabel 2.8. Simbol- Simbol *Entity Relationship Diagram (ERD)*

Simbol	Deskripsi
Entitas / <i>Entity</i> 	Entitas merupakan data inti yang akan tersimpan; 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 tabel.
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).



Simbol	Deskripsi
Atribut Multinilai / <i>Multivalued</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
Relasi 	Relasi yang menghubungkan antarentitas; biasanya diawali dengan kata kerja.
Asosiasi / <i>Association</i> 	Penghubung antara relasi dan entitas di mana di kedua ujungnya punya <i>multiplicity</i> kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas yang lain disebut dengan kardalitas. Misalkan ada kardalitas 1 ke N (sering disebut dengan one to many) menghubungkan entitas A dan B.

(Sumber: Sukanto dan Shalahuddin, 2013:50)

Ada beberapa pedoman yang perlu diperhatikan untuk membuat ERD, diantaranya:

- Entitas harus memiliki banyak kejadian atau realitas.
- Hindari penggunaan atribut yang tidak perlu.
- Berilah label yang jelas untuk semua komponen.
- Pasangkan kardinalitas dan modalitas yang jelas dan benar.
- Pecah atribut menjadi level serendah mungkin yang diperlukan.
- Level harus menrefleksikan istilah-istilah bisnis yang umum.
- Asumsi harus disebutkan dengan jelas.

2.3. Teori Program

2.3.1. Sekilas Tentang PHP

2.3.1.1. Pengertian PHP

Menurut Kadir (2008:2), *PHP* merupakan singkatan dari *PHP Hypertext Preprocessor*) yang merupakan bahasa berbentuk skrip yang ditempatkan dalam



server dan diproses di *server*. Hasilnya yang dikirimkan ke klien, tempat pemakai menggunakan *browser*. *PHP* dirancang untuk membentuk aplikasi *web* dinamis.

Menurut Wahana (2010:1), *PHP* merupakan bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses dan mengolah data secara dinamis.

Madcoms (2010:350) Menjelaskan, bahasa pemrograman *PHP* adalah bahasa pemrograman yang bekerja dalam sebuah *web server*. *Script-script PHP* dibuat harus tersimpan dalam sebuah *server* dan dieksekusi atau diproses dalam *server* tersebut.

Anhar (2010:3), *PHP* merupakan script yang digunakan untuk membuat halaman website yang dinamis.

2.3.1.2. Script PHP

Script PHP berkedudukan sebagai tag dalam bahasa *HTML*. Sebagaimana diketahui, *HTML* (*Hypertext Markup Language*) adalah bahasa standar untuk membuat halaman-halaman *web*. Sebagai contoh, berikut adalah kode *HTML* (disimpan dengan ekstensi *.htm* atau *.html*):

```
<html>
  <head>
    <title>Wellcome To PHP</title>
  </head>
  <body>
    Selamat Datang
  </body>
</html>
```

Bila dijalankan melalui *browser*, kode diatas akan menampilkan hasil seperti berikut:



Gambar 2.1. Tampilan program HTML pada *browser*

2.3.1.3. Teknik Penulisan Script *PHP*

Madcoms (2010:350) mengemukakan, *script PHP* harus selalu diawali dengan `<? Atau <?php` dan diakhiri dengan `?>`. sedangkan untuk menampilkan ke dalam *browser*, digunakan fungsi `echo" data";`. Semua teks yang diketik setelah tanda buka *script* (`<?`) dan tanda tutup *script* (`?>`) akan dieksekusi sebagai suatu *script PHP*. Contoh penulisan *script PHP* adalah sebagai berikut:

Tabel 2.9. *Script Dasar PHP*

<pre><? ?></pre>	Tempat penulisan <i>script PHP</i>
Atau	
<pre><?php ?></pre>	Tempat penulisan <i>script PHP</i>
Dan untuk menampilkan ke <i>browser</i>	
<pre><?php Echo "Welcome to PHP"; ?></pre>	



Di dalam *script PHP* juga dapat membuat komentar dan komentar tersebut tidak akan dieksekusi sebagai sebuah *script*. Beberapa cara untuk memberi keterangan di dalam *script PHP* adalah sebagai berikut:

- a. Gunakan *tag /** dan akhiri *tag */* apabila jumlah keterangan lebih dari 1 baris.
- b. Gunakan *tag //*, *tag* ini digunakan untuk keterangan yang hanya terdiri dari 1 baris saja.
- c. Gunakan *tag #*, *tag* ini juga digunakan untuk 1 baris komentar saja.

Dalam penulisannya, baris perintah *script PHP* selalu diakhiri dengan menuliskan tanda titik koma (;). Satu baris *script PHP* tidak harus berada dalam satu baris, melainkan dapat menuliskan perintah *script PHP* lebih dari satu baris.

2.3.2. Sekilas Tentang MySQL

2.3.2.1. Pengertian MySQL

Kadir (2008:348) mengemukakan, *MySQL* (baca mai-se-kyu-el) merupakan *software* 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.

Menurut Kadir (2008:348), *MySQL* adalah nama sebuah *database server* yang menangani akses *database* yang selalu dalam bentuk pernyataan *SQL* (*Structured Query Language*) yaitu suatu bahasa yang digunakan untuk mengakses *database relasional*.

Anhar (2010:21) Menjelaskan, *MySQL* (*My Structure Query Language*) adalah sebuah perangkat lunak sistem manajemen basis data *SQL* (*Database Management System*) atau DBMS dari sekian banyak DBMS, seperti *Oracle*, *MS SQL*, *Postagre SQL*, dan lain-lain.

2.3.2.2. Fungsi-Fungsi MySQL

Untuk dapat menggunakan semua perintah *SQL* pada *PHP*, kita memerlukan fungsi *MySQL*. Fungsi ini berguna untuk mengantarkan perintah *SQL* pada *PHP*



menuju ke *server* sehingga perintah tersebut dapat dieksekusi oleh semua *server MySQL*.

Fungsi-fungsi *MySQL* yang berawalan *mysql_* yang digunakan untuk mengakses *database server MySQL* adalah sebagai berikut:

Tabel 2.10. Fungsi-fungsi *MySQL* dalam *PHP*

No.	Fungsi MySQL	Penggunaan
1	<i>Mysql_connect()</i>	Membuat hubungan ke <i>database MySQL</i> yang terdapat pada suatu host
2	<i>Mysql_close()</i>	Menutup hubungan ke <i>database MySQL</i>
3	<i>Mysql_select_db()</i>	Memilih <i>database</i>
4	<i>Mysql_query()</i>	Mengeksekusi permintaan terhadap sebuah tabel atau sejumlah tabel
5	<i>Mysql_db_query()</i>	Menjalankan suatu permintaan terhadap suatu <i>database</i>
6	<i>Mysql_num_rows()</i>	Memperoleh jumlah baris dari suatu hasil permintaan yang menggunakan <i>SELECT</i>
7	<i>Mysql_affected_rows()</i>	Memperoleh jumlah baris yang dikenai operasi <i>INSERT, DELETE, UPDATE</i>
8	<i>Mysql_num_fields()</i>	Memperoleh jumlah kolom pada suatu hasil permintaan
9	<i>Mysql_fetch_row()</i>	Menghasilkan <i>array</i> /baris yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasil
10	<i>Mysql_fetch_array()</i>	Menghasilkan <i>array</i> /baris yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasil yang akan disimpan dua kali pada <i>array</i> hasil
11	<i>Mysql_fetch_field()</i>	Menghasilkan informasi suatu kolom
12	<i>Mysql_data_seek()</i>	Memindahkan pointer pada suatu himpunan hasil supaya menunjuk ke baris tertentu



No.	Fungsi MySQL	Penggunaan
13	<i>Mysql_field_seek()</i>	Memindahkan pointer pada suatu himpunan hasil supaya menunjuk ke kolom tertentu
14	<i>Mysql_create_db()</i>	Membuat <i>database MySQL</i>
15	<i>Mysql_drop_db()</i>	Menghapus <i>database MySQL</i>
16	<i>Mysql_list_dbs()</i>	Menghasilkan daftar <i>database MySQL</i>
17	<i>Mysql_list_tables()</i>	Memperoleh daftar nama tabel dalam suatu <i>database</i>
18	<i>Mysql_list_fields()</i>	Memperoleh daftar nama kolom dalam suatu <i>database</i>
19	<i>Mysql_fetch_assoc()</i>	Mendapatkan array baris dari suatu <i>recordset</i>
20	<i>Mysql_fetch_lengths()</i>	Mendapatkan panjang baris pada setiap isi <i>field</i>
21	<i>Mysql_fetch_object()</i>	Menghasilkan baris dari <i>recordset</i> sebagai sebuah objek
22	<i>Mysql_field_len()</i>	Mendapatkan informasi panjang maksimum <i>field</i> dalam sebuah <i>recordset</i>
23	<i>Mysql_field_name()</i>	Mendapatkan informasi nama <i>field</i> dalam <i>recordset</i>
24	<i>Mysql_get_client_info()</i>	Mendapatkan informasi tentang <i>MySQL client</i>
25	<i>Mysql_get_host_info()</i>	Mendapatkan informasi tentang <i>MySQL client</i>
26	<i>Mysql_get_server_info()</i>	Mendapatkan informasi tentang <i>MySQL server</i>
27	<i>Mysql_info()</i>	Mendapatkan informasi tentang <i>query</i> terakhir
28	<i>Mysql_ping()</i>	Memeriksa koneksi <i>server</i> dan akan mencoba untuk melakukan koneksi ulang jika koneksi terputus
29	<i>Mysql_result()</i>	Mendapatkan nilai dari <i>field</i> dalam sebuah <i>recordset</i>
30	<i>Mysql_stat()</i>	Menghasilkan informasi status <i>server</i> terkini

(Sumber: Abdul Kadir (2008:36))



2.3.2.3. Keistimewaan MySQL

MySQL memiliki beberapa keistimewaan, antara lain:

1. Portabilitas. *MySQL* dapat berjalan stabil pada berbagai sistem operasi seperti *Windows*, *Linux*, *FreeBSD*, *Mac Os X Server*, *Solaris*, *Amiga*, dan masih banyak lagi.
2. *Multi-user*. *MySQL* dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
3. Perangkat lunak sumber terbuka. *MySQL* didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
4. *Performance tuning*. *MySQL* memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak *SQL* per satuan waktu.
5. Ragam tipe data. *MySQL* memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer*, *float*, *double*, *char*, *text*, *date*, *timestamp*, dan lain-lain.
6. Keamanan. *MySQL* memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
7. Skalabilitas dan Pembatasan. *MySQL* mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
8. Konektivitas. *MySQL* dapat melakukan koneksi dengan klien menggunakan protokol *TCP/IP*, *Unix socket (UNIX)*, atau *Named Pipes (NT)*.
9. Lokalisasi. *MySQL* dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
10. Antar Muka. *MySQL* memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).



11. Klien dan Peralatan. *MySQL* dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.
12. Struktur tabel. *MySQL* memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basis data lainnya seperti *PostgreSQL* ataupun *Oracle*.
13. Perintah dan Fungsi. *MySQL* memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).

2.3.3. Sekilas Tentang Adobe Dreamweaver CS5

2.3.3.1. Pengertian Adobe Dreamweaver CS5

Madcoms (2010:1) menjelaskan, *dreamweaver* merupakan *software* aplikasi yang digunakan sebagai *HTML* editor profesional untuk mendesain *web* secara visual. Aplikasi ini dikenal dengan istilah *WYSIWYG (What You See Is What You Get)*, yang intinya adalah kita tidak harus berurusan dengan *tag-tag HTML* untuk membuat sebuah situs. *Dreamweaver* juga memberikan keleluasaan untuk menggunakannya sebagai media penulisan bahasa pemrograman *web*.

Adobe terus mengeluarkan beberapa varian *Dreamweaver* yang semakin berkembang, salah satunya yaitu *Dreamweaver CS5*. Ada beberapa fitur baru yang dapat ditemukan pada versi ini, diantaranya: integrasi dengan *Adobe Business Catalyst*, integrasi *Adobe Browser Lab*, Integrasi dengan *CMS*, pengecekan *CSS*, dan lain-lain:

2.3.3.2. Starting Page



Gambar 3.2. Tampilan awal Adobe Dreamweaver CS5



Jendela disamping menampilkan 4 menu pilihan, yaitu:

1. *Open a Recent Item*

Menu ini menampilkan daftar nama dokumen yang pernah dibuka oleh *Adobe Dreamweaver CS5*. Dokumen tersebut dapat dibuka dengan cara memilih nama dokumen yang ada.

2. *Create New*

Menu ini digunakan untuk membuat dokumen *web* yang baru. Dokumen *web* dapat dibuat dengan beberapa tipe yang disediakan oleh *Adobe Dreamweaver CS5*.

3. *Top Features* (video)

Merupakan fitur yang disediakan *Dreamweaver* melalui *internet*.

4. *Help online*

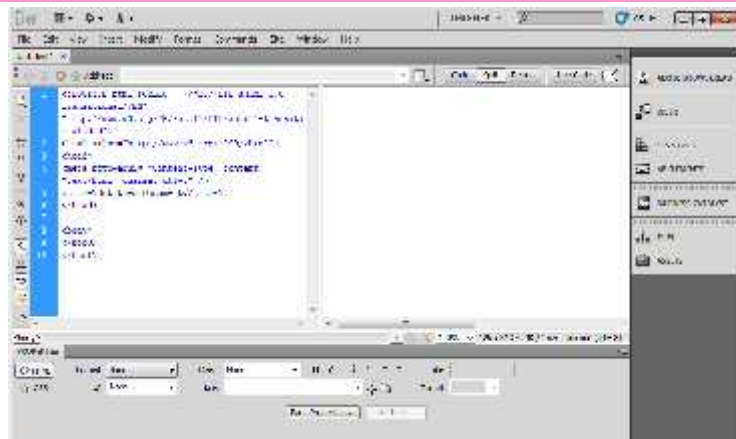
Merupakan menu pilihan untuk menampilkan fasilitas *help* secara *online* melalui *internet*.

2.3.3.3. Ruang Kerja *Adobe Dreamweaver CS5*

Saat pertama kali menjalankan *Dreamweaver* dengan, akan ditampilkan sebuah kotak dialog *Workspace Setup* yang digunakan untuk memilih tampilan ruang kerja.

1. ***Designer Workspace***, adalah sebuah penggabungan ruang kerja menggunakan MDI (*Multiple Document Interface*), dimana semua jendela *document* dan panel-panel tergabung di dalam satu jendela aplikasi yang benar, dengan sisi panel group di sebelah kanan.
2. ***Coder Workspace***, adalah ruang kerja yang tergabung sama, tapi letak sisi panel group berada di sebelah kiri.

Ruang kerja *Dreamweaver CS5* memiliki komponen-komponen yang memberikan fasilitas dan ruang untuk menuangkan kreasi saat bekerja, seperti *Application Bar*, *Toolbar Document*, *Document Window*, *Workspace Switcher*, *Panel Groups*, *Tag Selector*, *Property Inspector*, *Toolbar Browser Navigation*.



Gambar 2.3. Tampilan Ruang Kerja *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.



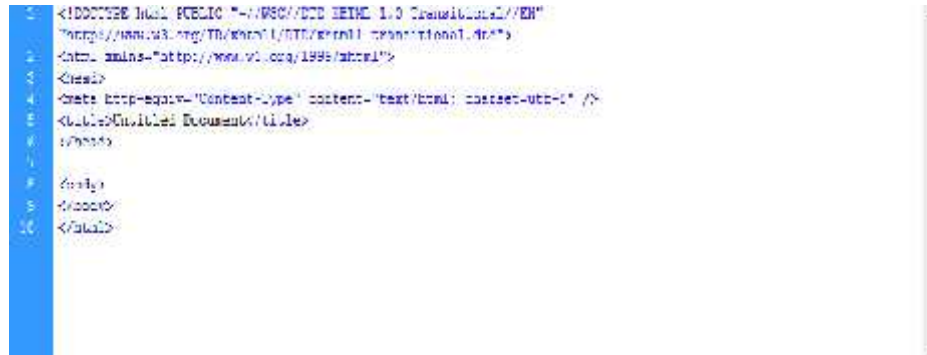
Gambar 2.4. *Application Bar* pada *Dreamweaver CS5*

2. **Toolbar Document**, berisi tombol-tombol 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*.



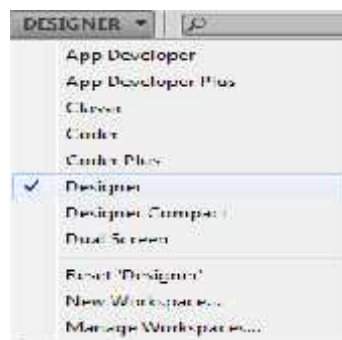
Gambar 2.5. *Toolbar Document* pada *Dreamweaver CS5*

3. **Document Window** (Jendela Dokumen), lembar kerja tempat membuat dan mengedit desain halaman *web*.



Gambar 2.6. Document window pada Dreamweaver CS5

4. **Workspace Switcher**, digunakan untuk mengubah tampilan ruang kerja (*workspace*) Dreamweaver CS5



Gambar 2.7. Workspace Switcher pada Dreamweaver CS5

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



Gambar 2.8. Panel Groups pada Dreamweaver CS5

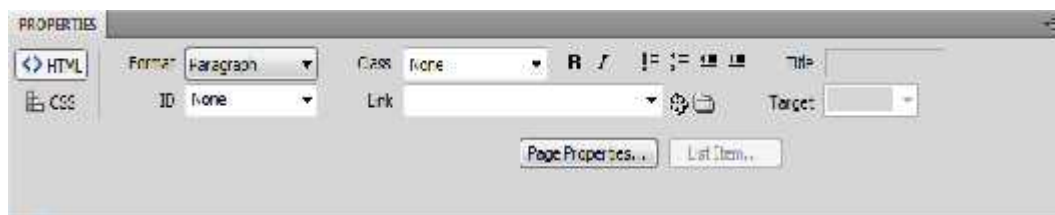


6. **Tag Selector**, terletak di bagian bawah jendela dokumen, satu baris dengan *status bar*. Bagian ini menampilkan hirarki pekerjaan yang sedang terpilih pada jendela dokumen, dapat juga digunakan untuk memilih objek pada jendela desain berdasarkan jenis atau kategori objek tersebut. *Tag Selector* juga menampilkan informasi format dari bagian yang sedang aktif pada lembar kerja desain.



Gambar 2.9. *Tag Selector* pada *Dreamweaver CS5*

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.



Gambar 2.10. *Property Inspector* pada *Dreamweaver CS5*

8. **Toolbar Browser Navigation**, toolbar ini merupakan *toolbar* baru yang ada di dalam *Dreamweaver CS5* dan letaknya tepat berada di atas jendela dokumen. Toolbar ini berisi tombol-tombol yang digunakan sebagai navigasi di dalam *browser*.

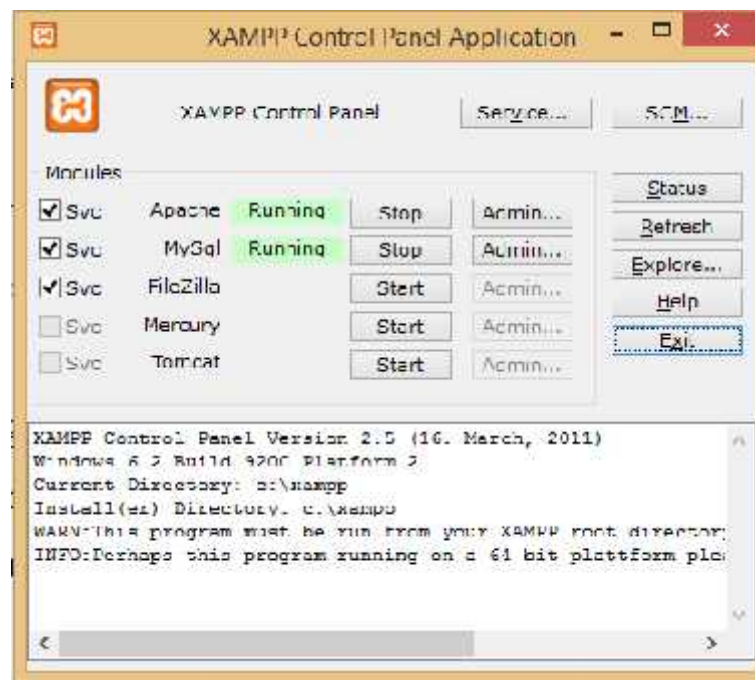


Gambar 2.11. *Toolbar Browser Navigation* pada *Dreamweaver CS5*



2.3.4 Pengenalan XAMPP

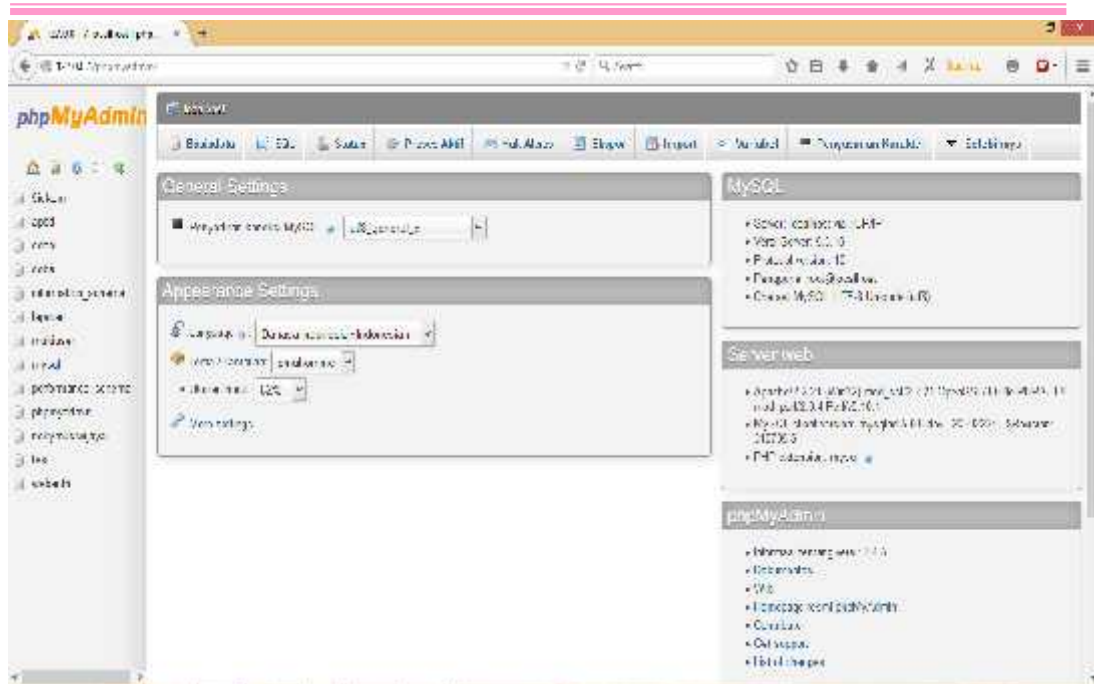
Nugroho (2013 : 1) menjelaskan, XAMPP adalah paket program web lengkap yang dapat Anda pakai untuk belajar pemrograman web, khususnya PHP dan MySQL, paket ini dapat didownload secara gratis dan legal.”



Gambar 2.12 Tampilan XAMPP

2.3.5. Pengertian phpMyAdmin

Nugroho (2013:71) menjelaskan, “phpMyAdmin adalah tools *MySQL Client* berlisensi *Freeware*, Anda dapat mengunduhnya secara gratis pada situs resminya di www.phpmyadmin.net.



Gambar 2.13 Tampilan phpMyAdmin