



BAB II TINJAUAN PUSTAKA

2.1. Teori Umum

Adapun dalam penyusunan laporan akhir ini penulis akan menuliskan landasan teori, pengertian-pengertian yang terdapat pada penulisan laporan ini adalah sebagai berikut:

2.1.1. Pengertian Komputer

Sutarman (2009:86), menjelaskan bahwa “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.

Asropudin (2013:19), menjelaskan bahwa “komputer adalah alat bantu pemrosesan data secara elektronik dan cara pemrosesan datanya berdasarkan urutan instruksi atau program yang tersimpan dalam memori masing-masing komputer”.

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

2.1.2. Pengertian Sistem

Sutarman (2009:5), menjelaskan bahwa “sistem adalah kumpulan elemen yang saling berhubungan dan berinteraksi dalam satu kesatuan untuk menjalankan suatu proses pencapaian suatu tujuan tertentu”.

Ladjmudin (2012:1), menjelaskan bahwa “sistem adalah suatu urutan kegiatan yang saling berhubungan, berkumpul bersama-sama untuk mencapai suatu tujuan tertentu”.

Hartono (2013:10), menjelaskan bahwa “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)”.



2.1.3. Pengertian Informasi

Hartono (2013:9), menjelaskan bahwa “informasi adalah sekumpulan data yang telah diolah menjadi sesuatu yang memiliki arti dan kegunaan lebih luas”.

Sutarman (2012:14), menjelaskan bahwa “informasi adalah fakta (data) yang diorganisasikan dengan cara tertentu sehingga mereka mempunyai arti bagi si penerima”.

Ladjmudin (2012:9), menjelaskan bahwa “informasi adalah data yang telah diolah menjadi bentuk yang berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun yang akan datang”.

2.1.4. Pengertian Sistem Informasi

Hartono (2013:20), menjelaskan bahwa “sistem informasi adalah rangkaian sistem yang terorganisasi dari sejumlah bagian/komponen yang secara bersama-sama berfungsi atau bergerak menghasilkan informasi”.

Sutarman (2012:13), menjelaskan bahwa “sistem informasi adalah sistem yang mengumpulkan, memproses, menyimpan, menganalisis, menyebarkan *input* dan menghasilkan *output* menjadi informasi untuk tujuan tertentu kepada pengguna atau sistem yang lainnya”.

Ladjmudin (2013:13), menjelaskan bahwa “sistem informasi adalah suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi”.

2.1.5. Metode Pengembangan Sistem

Metodologi penelitian yang digunakan dalam penyusunan Laporan Akhir ini yaitu metodologi *Waterfall*. Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak berurutan, di mana kemajuan terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi, pengujian, dan pemeliharaan (Pressman, Roger S.).

1. Analisis kebutuhan perangkat lunak

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



2. Desain

Desain perangkat lunak adalah proses multi langkah yang focus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan proses pengodean.

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.

4. Pengujian

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

5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

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.

2.2. Teori Judul

Teori ini akan membahas pengertian-pengertian yang terdapat pada judul laporan akhir ini yaitu, sebagai berikut:

2.2.1. Pengertian Aplikasi

Sutabri (2012:147), menjelaskan bahwa “aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya”.

Asropudin (2013:7), menjelaskan bahwa “aplikasi (*application*) adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu”.

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



2.2.2. Pengertian Pengolahan

Phoenix (2009:610), menjelaskan bahwa “pengolahan adalah proses, pembuatan cara mengolah”.

2.2.3. Pengertian Pendataan

Alwi (2005:239), menjelaskan bahwa ”pendataan adalah proses, cara, perbuatan mendata, pengumpulan data; pencarian data”.

2.2.4. Pengertian Usaha

Phoenix (2009:927), menjelaskan bahwa “usaha adalah kegiatan dengan menggerakkan tenaga, pikiran atau badan untuk mencapai sesuatu maksud, pekerjaan, perbuatan, daya upaya, ikhtiar untuk mencapai sesuatu maksud, kerajinan belajar, pekerjaan untuk menghasilkan sesuatu”.

2.2.5. Pengertian Kesehatan

Phoenix (2009:764), menjelaskan bahwa “kesehatan adalah keadaan (hal) sehat; kebaikan keadaan (badan dsb)”.

2.2.6. Pengertian Sekolah

Phoenix (2009:767), menjelaskan bahwa “sekolah adalah bangunan atau lembaga untuk belajar dan member pelajaran; menurut tingkatannya ada Sekolah Taman Kanak-kanak: TK; Sekolah Dasar: SD; Sekolah Lanjutan Pertama: SLTP; Sekolah Lanjutan Atas: SLTA; atau SMA; SPG: Sekolah Pendidikan Guru; Sekolah Teknik Mesin, dll; waktu atau pertemuan ketika murid-murid diberi pelajaran; usaha menuntut ilmu pengetahuan, kepandaian, pelajaran, belajar disekolah, pergi ke sekolah”.

2.2.7. Pengertian Unit

Phoenix (2009:922), menjelaskan bahwa “unit adalah bagian terkecil dari suatu yang dapat berdiri sendiri; kadar yang digunakan untuk mengukur (menilai dsb) sesuatu; dasar ukuran; kata bantu bilangan”.

2.2.8. Pengertian Pelaksana

Alwi (2005: 488), menjelaskan bahwa “pelaksana adalah orang (panitia, organisasi, dsb) yang mengerjakan atau melaksanakan (rancangan dsb)”.



2.2.9. Pengertian Teknis

Phoenix (2009:854), menjelaskan bahwa “teknis adalah Mengenai, menurut teknik, secara teknik”.

2.2.10. Pengertian Dinas

Phoenix (2009:190), menjelaskan bahwa “dinas adalah jawatan, segala sesuatu mengenai atau ada hubungan dengan jawatan, bukan pertikelir; pakaian dinas; surat dinas”.

2.2.11. Pengertian Aplikasi Pengolahan Pendataan Usaha Kesehatan Sekolah (UKS) pada Kantor Unit Pelaksana Teknis Dinas (UPTD) Kecamatan Ilir Timur I Palembang

Merupakan suatu aplikasi yang mengolah data-data dari hasil pendataan usaha kesehatan sekolah (UKS) yang dilakukan oleh unit pelaksana teknis dinas (UPTD) kecamatan ilir timur 1 palembang.

2.3. Teori Khusus

2.3.1. Data Flow Diagram (DFD)

Kristanto (2008:55), *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.

Al-Fatta (2007:105), “*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.

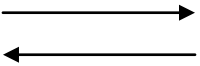
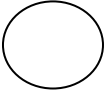
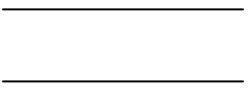
Shalahuddin (2013:70), “*data flow diagram (DFD)* adalah representasi grafik yang menggambarkan aliran informasi dan transformasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*)”.

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



1. Teknik Yourdan Dan De Macro

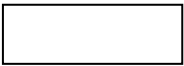
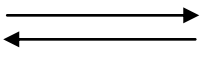
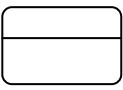
Tabel 2.1 Simbol- Simbol *Data Flow Diagram* Yourdan & De Marco

No	Simbol	Keterangan
1.		<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
2.		<i>Data Flow</i> , digunakan untuk menggambarkan aliran data dari satu proses ke proses lainnya
3.		<i>Process</i> , digunakan untuk mentransformasikan data secara umum
4.		<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 Gane dan Sarson

Tabel 2.2 Simbol- Simbol *Data Flow Diagram* Gane & Sarson

No	Simbol	Keterangan
1.		<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
2.		<i>Data Flow</i> , digunakan untuk menggambarkan aliran data dari satu proses ke proses lainnya
3.		<i>Process</i> , digunakan untuk mentransformasikan data secara umum

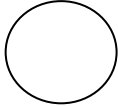
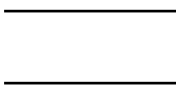
**Lanjutan Tabel 2.2** Simbol- Simbol *Data Flow Diagram* Gane & Sarson

No	Simbol	Keterangan
4.		<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)

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

Tabel 2.3 Simbol- Simbol *Data Flow Diagram* menurut Shalahuddin

No	Notasi	Keterangan
1.		Proses atau fungsi atau prosedur. Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.
2.		<i>Field</i> atau basis data atau penyimpanan (<i>storage</i>). Nama yang diberikan pada sebuah penyimpanan biasanya berupa kata benda.
3.		<i>Entitas</i> atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang. Nama yang diberikan biasanya berupa kata benda.
4.		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: Shalahuddin, 2013:71)



2.3.2. Kamus Data (*Data Dictionary*)

Jogiyanto (2005:725), “kamus data atau *data dictionary* atau disebut juga dengan istilah *systems data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi”.

Kristanto (2008:66), “kamus data adalah kumpulan elemen-elemen atau simbol-simbol yang digunakan untuk membantu dalam penggambaran atau pengidentifikasian setiap *field* atau file di dalam sistem”.

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)”.

Adapun simbol-simbol dalam kamus data adalah sebagai berikut:

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

No	Simbol	Keterangan
1.	=	Disusun, terdiri atas, mendefinisikan, diuraikan menjadi, artinya
2.	+	Dan
3.	()	Opsional/Pilihan (boleh Ya atau Tidak)
4.	[]	Salah satu alternative

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

No	Simbol	Keterangan
5.	*	Komentar, catatan, keterangan
6.	@	Identifikasi atribut kunci, petunjuk (<i>key field</i>)
7.		Pemisah alternatif simbol []

(Sumber: Andri Kristanto, 2008:66)

2.3.3. Pengertian *Flowchart*

eWolf Community (2012:16), “*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*

No	Simbol	Nama Simbol	Keterangan
1.		Garis Alir (<i>Flow Line</i>)	Simbol untuk menghubungkan antar proses atau antar simbol
2.		<i>Connector</i>	Simbol konektor untuk menyambung proses pada lembar kerja yang sama
3.		<i>Offline Connector</i>	Simbol konektor untuk menyambung proses pada lembar kerja yang berbeda
4.		<i>Process</i>	Simbol pemrosesan yang terjadi pada sebuah alur kerja
5.		<i>Manual</i>	Simbol yang mendefinisikan proses yang dilakukan secara manual
6.		<i>Decision</i>	Simbol untuk memutuskan proses lanjutan dari kondisi tertentu
7.		<i>Terminal</i>	Simbol <i>start</i> atau <i>end</i> yang mendefinisikan awal atau akhir dari sebuah <i>flowchart</i>
8.		<i>Input/Output</i>	Simbol <i>input/output</i> yang mendefinisikan masukan dan keluaran proses
9.		<i>Magnetic Tape</i>	Simbol masukan atau keluaran dari atau ke sebuah pita magnetik

Lanjutan Tabel 2.5 Simbol-simbol *Flowchart*

No	Simbol	Nama Simbol	Keterangan
10.		<i>Document</i>	Simbol masukan atau keluaran dari atau ke sebuah dokumen
11.		<i>Display</i>	Simbol yang menyatakan piranti keluaran, seperti layar monitor, printer, dll
12.		<i>Data Storage</i>	Simbol database atau basis data
13.		<i>Sub Program</i>	Simbol yang menyatakan bagian dari program (sub program)

(Sumber: eWolf Community, 2012:17)

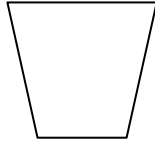
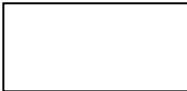
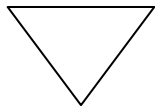
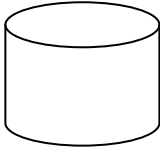
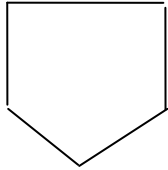
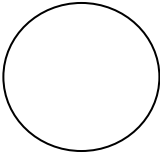
2.3.4. Pengertian *Blockchart*

Kristanto (2008:68), “*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:

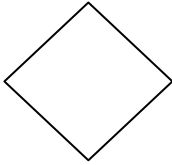
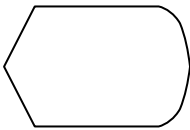
Tabel 2.6 Simbol-simbol dalam *Blockchart*

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

Lanjutan Tabel 2.6 Simbol-simbol *Blokchart*

No	Simbol	Keterangan
3.		Proses manual.
4.		Proses yang dilakukan oleh komputer.
5.		Menandakan dokumen yang diarsifkan(arsif manual)
6.		Data penyimpanan(<i>data storage</i>)
7.		Proses apa saja yang tidak terdefinisi termasuk aktifitas fisik
8.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain
9.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama
10.		Terminasi yang menandakan awal dan akhir dari suatu aliran

Lanjutan Tabel 2.6 Simbol-simbol *Blokchart*

No	Simbol	Keterangan
11.		Pengambilan keputusan (<i>decision</i>)
12.		Layar peraga (<i>monitor</i>)
13.		Pemasukan data secara manual

(Sumber: Andri Kristanto, 2008:68)

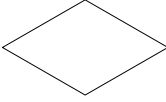
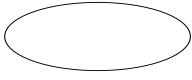
2.3.5. Entity Relational Diagram (ERD)

Al-Fatta (2007:121), *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
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)*

No	Nama Simbol	Simbol	Keterangan
1.	<i>Entitas</i>		Orang, tempat atau benda memiliki nama tunggal.
2.	<i>Relationship</i>		Menunjukkan hubungan antar 2 entitas. Dideskripsikan dengan kata kerja.
3.	<i>Atribut</i>		Properti dari entitas harus digunakan oleh minimal 1 proses bisnis dipecah dalam detail.
4.	<i>Link</i>		Penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya.

(Sumber: Hanif Al Fatta, 2007:124)

2.4. Teori Program

2.4.1. Sekilas Tentang PHP

2.4.1.1. Pengertian PHP

Kadir (2008:2), “*PHP* merupakan singakatan dari *PHP Hypertext Preprocessor*) yang merupakan bahasa berbentuk skrip yang ditempatkan dalam *server* dan diproses di *server*”. Hasilnyalah yang dikirimkan ke klien, tempat pemakai menggunakan *browser*. *PHP* dirancang untuk membentuk aplikasi *web* dinamis.

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

Kadir (2013:17), “*PHP* adalah bahasa pemrograman yang ditujukan untuk kepentingan pembuatan aplikasi *web*”.

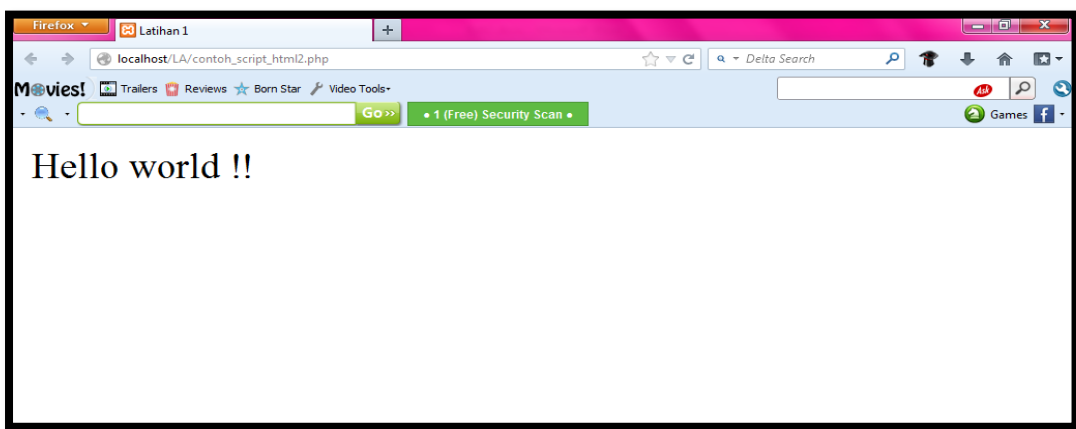
2.4.1.2. Script PHP

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>Latihan 1</title>
  </head>
  <body>
    Hello world !!
  </body>
</html>
```

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



Gambar 2.1 Tampilan program HTML pada *browser*

2.4.1.3. Teknik Penulisan Script *PHP*

Madcoms (2010:350) menyatakan, “*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 2.8 Script Dasar *PHP*

<?	
.....	Tempat penulisan <i>script PHP</i>
?>	
Atau	
<?php	
.....	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:

- Gunakan *tag /** dan akhiri *tag */* apabila jumlah keterangan lebih dari 1 baris.
- Gunakan *tag //*, *tag* ini digunakan untuk keterangan yang hanya terdiri dari 1 baris saja.
- 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.

2.4.2. Sekilas Tentang MySQL

2.4.2.1. Pengertian MySQL

Kadir (2013:412), “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), “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”.

3.4.2.2. 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 2.9 Fungsi-fungsi *MySQL* dalam *PHP*

No.	Fungsi <i>MySQL</i>	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
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>



Lanjutan Tabel 2.9 Fungsi-fungsi MySQL dalam PHP

No.	Fungsi MySQL	Penggunaan
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 (2008:360) dan eWolf Community (2012:157-168))

2.4.2.3. 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. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. *Multi-user*. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.



-
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. Perintah dan Fungsi. *MySQL* memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
 7. Keamanan. *MySQL* memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
 8. 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.
 9. Konektivitas. *MySQL* dapat melakukan koneksi dengan klien menggunakan protokol *TCP/IP*, *Unix socket (UNIX)*, atau *Named Pipes (NT)*.
 10. Lokalisasi. *MySQL* dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
 11. Antar Muka. *MySQL* memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
 12. 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*.
 13. Struktur tabel. *MySQL* memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basis data lainnya seperti *PostgreSQL* ataupun *Oracle*.
-



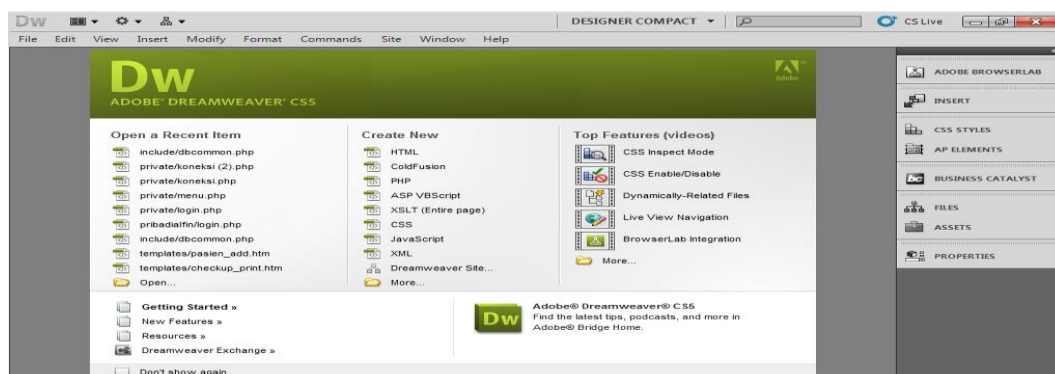
2.4.3. Sekilas Tentang Adobe Dreamweaver CS5

2.4.3.1. Pengertian Adobe Dreamweaver CS5

Madcoms (2010:1), “*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 Tou 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 Bussiness Catalyst*, integrasi *Adobe Browser Lab*, Integrasi dengan *CMS*, pengecekan *CSS*, dan lain-lain.

2.4.3.2. Starting Page



Gambar 2.2 Tampilan awal Adobe Dreamweaver CS5

Jendela diatas 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

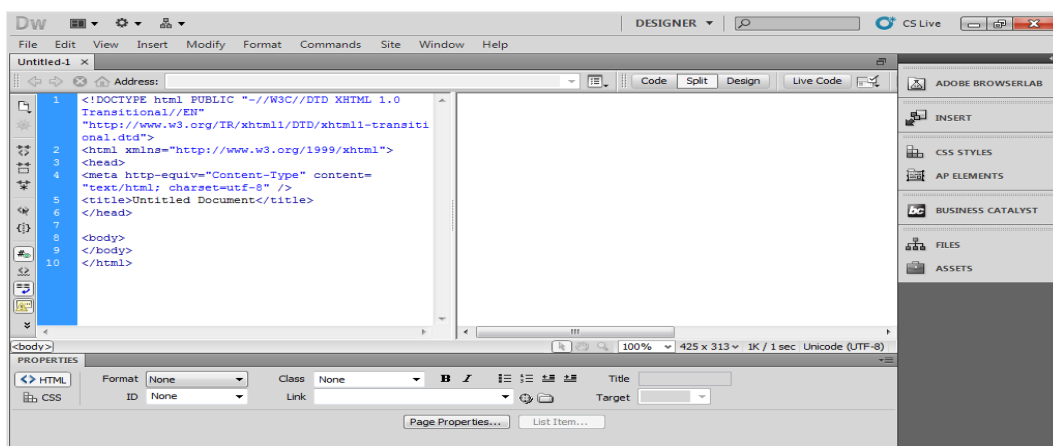
Merupakan menu pilihan untuk menampilkan fasilitas *help* secara *online* melalui *internet*.

2.4.3.3. 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.

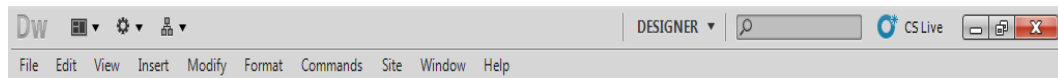
Ruang kerja *Dreamweaver CS5* memiliki komponen-komponen yang memberikan fasilitas dan ruang untuk menuangkan kreasi saat bekerja, seperti *Application Bar*, *Toolbar Document*, *Document Window*, *Workspace Switcher*, *Panel Groups*, *Tag Selector*, *Property Inspector*, *Toolbar Browser Navigation*.



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.



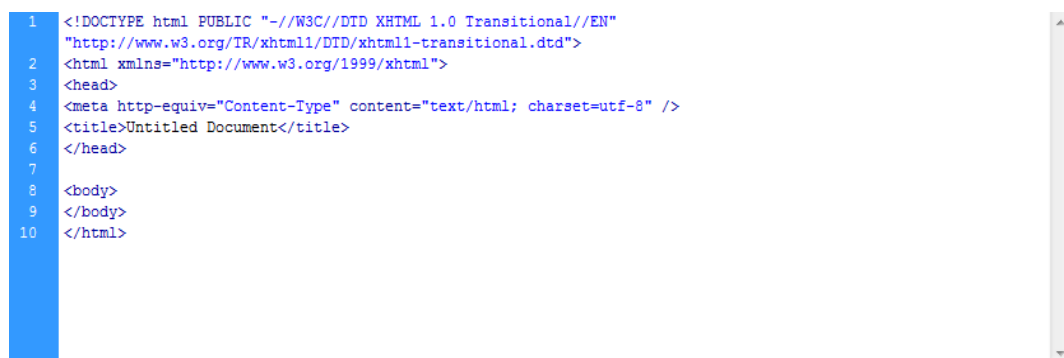
Gambar 2.4 *Application Bar* pada *Dreamweaver CS5*

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*.



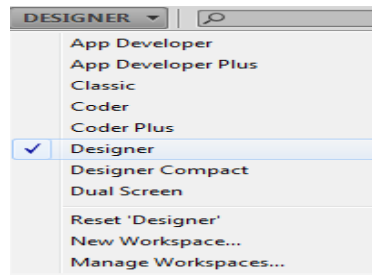
Gambar 2.5 *Toolbar Document* pada *Dreamweaver CS5*

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



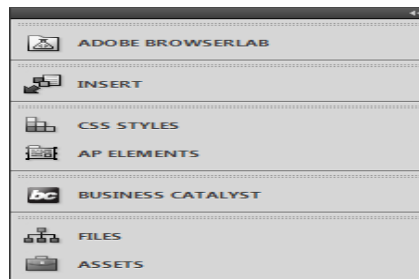
Gambar 2.6 *Document window* pada *Dreamweaver CS5*

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.



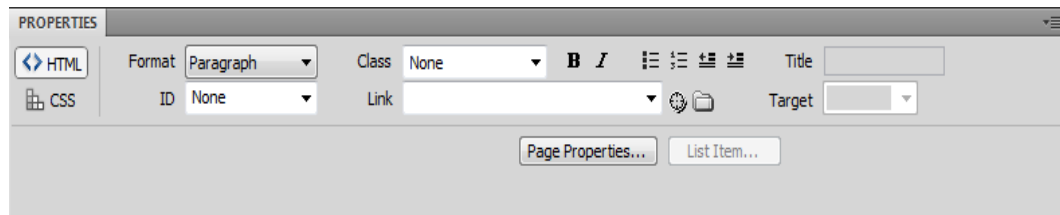
Gambar 2.8 Panel Groups pada Dreamweaver CS5

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.



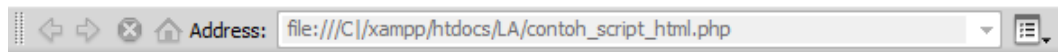
Gambar 2.9 Tag Selector pada Dreamweaver CS5

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*