



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Teori Umum

##### 2.1.1. Pengertian Komputer

Mulyono (2010:1), “Komputer adalah seperangkat alat elektronik yang terdiri atas peralatan input, alat yang mengolah input, dan peralatan output yang memberikan informasi, serta bekerja secara otomatis”.

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

Sujatmiko (2012:156), “Komputer adalah mesin yang dapat mengolah data digital dengan mengikuti serangkaian perintah atau program. Alat serbaguna ini memegang peran penting dalam teknologi komunikasi”.

##### 2.1.2. Pengertian Data

Sutabri (2013:3), “Data adalah hal, peristiwa, atau kenyataan lain ataupun yang mengandung sesuatu pengetahuan untuk dijadikan dasar guna penyusunan keterangan, pembuatan kesimpulan, atau penetapan keputusan”.

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

Wahyudi (2012:3), “Data adalah suatu file ataupun field yang berupa karakter atau tulisan dan gambar”.

##### 2.1.3. Pengertian Basis Data (*Database*)

Sutanta (2011:35), “Basis data merupakan sekumpulan dari bermacam-macam tipe *record* yang memiliki hubungan antar-*record* dan rincian data terhadap obyek tertentu”.

---



### **1. Lingkungan Luar Sistem (*Environment*)**

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem.

### **2. Penghubung Sistem (*Interface*)**

Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau *interface*.

### **3. Masukan Sistem (*Input*)**

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintanace input*) dan sinyal (*signal input*).

### **4. Keluaran Sistem (*Output*)**

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi sub sistem yang lain.

### **5. Pengolahan Sistem (*Process*)**

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

### **6. Sasaran Sistem (*Objective*)**

Suatu sistem memiliki sasaran dan tujuan yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

#### **a. Analisis kebutuhan perangkat lunak**

Tahap analisis dilakukan secara intensif untuk menspesifikasikan kebutuhan sistem agar dapat dipahami sistem seperti apa yang dibutuhkan oleh *user*.

#### **b. Desain**

Tahap desain adalah proses multi langkah yang fokus pada desain pembuatan program sistem termasuk struktur data, arsitektur sistem, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuha

---



- c. Pembuatan kode program  
Pada tahap pengkodean, desain harus ditranslasikan ke dalam program sistem. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain
- d. Pengujian  
Tahap pengujian fokus pada sistem dari segi logika 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.
- e. Pendukung (*support*) atau Pemeliharaan (*maintenance*)  
Tidak menutup kemungkinan sebuah sistem mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau sistem harus beradaptasi dengan lingkungan baru. Tahap pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan sistem yang sudah ada, tapi tidak untuk sistem baru.

#### **2.1.4. Pengertian Sistem**

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

Yakub (2012:1), “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau tujuan tertentu”.

Fatta (2007:3), “Sistem diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi dan saling bergantung sama lain”.



### 2.1.5. Karakteristik Sistem

Sutabri (2012:20-21) menjelaskan bahwa sistem memiliki karakteristik atau sifat-sifat tertentu yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut :

#### 1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk suatu kesatuan.

#### 2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

#### 3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem.

#### 4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau *interface*.

#### 5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintanace input*) dan sinyal (*signal input*).

#### 6. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi sub sistem yang lain.

#### 7. Pengolahan Sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

---



## 8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki sasaran dan tujuan yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

### 2.2. Teori Khusus

#### 2.2.1. Pengertian DFD (*Data Flow Diagram*)

Sukamto dan Shalahuddin (2014:69) menjelaskan, “*Data Flow Diagram* (DFD) awalnya dikembangkan oleh Chris Gane dan Trish Sarson pada tahun 1979 yang termasuk dalam *Structured Systems Analysis and Design Methodology* (SSADM) yang ditulis oleh Chris Gane dan Trish Sarson. Sistem yang dikembangkan ini berbasis pada dekomposisi fungsional dari sebuah sistem.”

Edward Yourdon dan Tom DeMarco memperkenalkan metode yang lain pada tahun 1980-an di mana mengubah persegi dengan sudut lengkung (pada DFD Chris Gane dan Trish Sarson) dengan lingkaran untuk menotasikan. DFD Edward Yourdon dan Tom DeMarco populer digunakan sebagai model analisis sistem perangkat lunak untuk sistem perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur.

Informasi yang ada di dalam perangkat lunak dimodifikasi dengan beberapa transformasi yang dibutuhkan. *Data Flow Diagram* (DFD) atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).

DFD dapat digunakan untuk merepresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD dapat dibagi menjadi beberapa level yang lebih detail untuk merepresentasikan aliran informasi atau fungsi yang lebih detail. DFD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi. Oleh karena itu, DFD lebih sesuai digunakan untuk memodelkan fungsi-fungsi perangkat lunak yang akan diimplementasikan menggunakan pemrograman terstruktur karena pemrograman

---

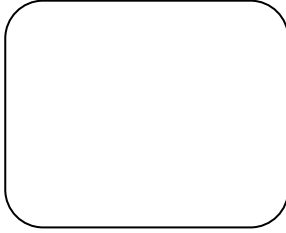
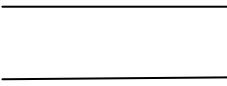


terstruktur membagi-bagi bagiannya dengan fungsi-fungsi dan prosedur-prosedur.

DFD tidak sesuai untuk memodelkan sistem perangkat lunak yang akan dibangun menggunakan pemrograman berorientasi objek. Paradigma pemrograman terstruktur dan pemrograman berorientasi objek merupakan hal yang berbeda. Jangan mencampuradukkan pemrograman terstruktur dan pemrograman berorientasi objek.



Sukanto dan Shalahuddin (2014:71) menjelaskan, “notasi-notasi pada DFD (Chris Gane dan Trish Sarson) adalah sebagai berikut:

**Tabel 2.1** Notasi-notasi pada DFD (Chris Gane dan Trish Sarson)

Notasi	Keterangan
	<p>Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program.</p> <p>Catatan: Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>
	<p><i>File</i> atau basisdata atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (<i>Entity Relationship Diagram (ERD)</i>, <i>Conceptual Data Model (CDM)</i>, <i>Physical Data Model (PDM)</i>)`</p>



**Lanjutan Tabel 2.1** Notasi-notasi pada DFD (Chris Gane dan Trish Sarson)

Notasi	Keterangan
	<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau system lain yang terkait dengan aliran data dari system yang dimodelkan.</p> <p>Catatan: Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda</p>
	<p>Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>)</p> <p>Catatan: Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”</p>

Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD:

1. Membuat DFD Level 0 atau sering disebut juga *Context Diagram*

DFD Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD Level 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

2. Membuat DFD Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. DFD Level 1 merupakan hasil *breakdown* DFD Level 0 yang sebelumnya sudah dibuat.



### 3. Membuat DFD Level 2

Modul-modul pada DFD Level 1 dapat di-*breakdown* menjadi DFD Level 2. Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada tingkat kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di-*breakdown* lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 yang di- *breakdown*.

### 4. Membuat DFD Level 3 dan seterusnya

DFD Level 3, 4, 5 dan seterusnya merupakan *breakdown* dari modul pada DFD Level di-atasnya. *Breakdown* pada level 3, 4, 5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.


Pada satu diagram DFD sebaiknya jumlah modul tidak boleh lebih dari 20 buah. Jika lebih dari 20 buah modul, diagram akan terlihat rumit dan susah untuk dibaca sehingga menyebabkan system yang dikembangkan juga menjadi rumit.

#### 2.2.2. Pengertian *Blockchart*

Kristanto (2008:68) 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.”

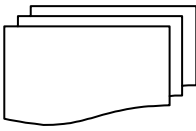
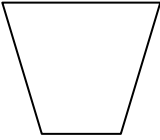

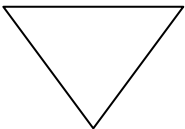
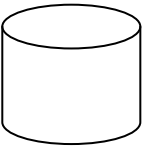
Kristanto (2008:68) menjelaskan, “Simbol-simbol yang sering digunakan dalam *blockchart* dapat dilihat pada tabel berikut ini:

**Tabel 2.2 Simbol-simbol dalam *Block Chart***

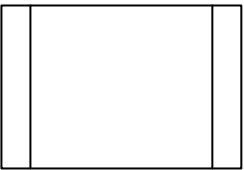
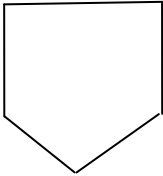
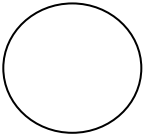

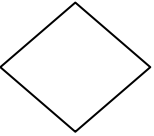
No	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan





Lanjutan Tabel 2.2 Simbol-simbol dalam *Block Chart*

No	Simbol	Keterangan
2.		Multi dokumen
3.		Proses Manual
4.		Proses dilakukan oleh komputer.
5.		Menandakan dokumen yang diarsipkan (arsip manual)
6.		Data penyimpanan ( <i>Storage</i> )

Lanjutan Tabel 2.2 Simbol-simbol dalam *Block Chart*

No	Simbol	Keterangan
7.		Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik.
8.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain.
9.		Terminal yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.
10.		Terminal yang menandakan awal dan akhir dari suatu aliran.
11.		Pengambilan keputusan ( <i>Decision</i> ).

Lanjutan Tabel 2.2 Simbol-simbol dalam *Block Chart*

No	Simbol	Keterangan
12.		Layar peraga ( <i>monitor</i> ).
13.		Pemasukkan data secara manual.

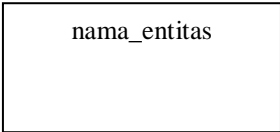
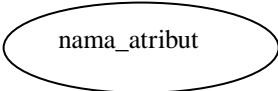
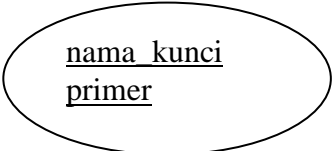
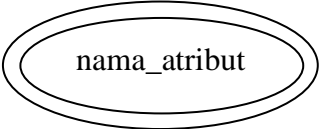
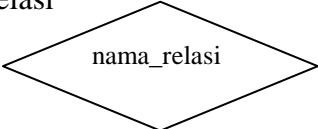
### 2.2.3. Pengertian ERD (*Entity Relationship Diagram*)

Sukamto dan Shalahuddin (2014:50) menjelaskan, tentang pemodelan awal basis data yang paling banyak digunakan adalah: menggunakan *Entity Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen.

Shalahuddin (2014:50) menjelaskan, "Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen:

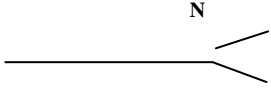


Tabel 2.3 Simbol-simbol ERD dengan notasi Chen

Simbol	Deskripsi
Entitas / <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi computer; 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)
Atribut multivalued / multivalued 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja



Lanjutan Tabel 2.3 Simbol-simbol ERD dengan notasi Chen

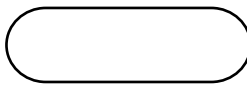
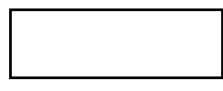
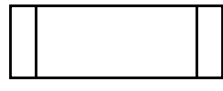
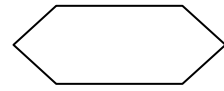
Simbol	Deskripsi
Asosiasi / <i>association</i> 	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian.  Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas

ERD biasanya memiliki hubungan *binary* (satu relasi menghubungkan dua buah entitas). Beberapa metode perancangan ERD menoleransi hubungan relasi *ternary* (satu relasi menghubungkan tiga buah relasi) atau *N-ary* (satu relasi menghubungkan banyak entitas), tapi banyak metode perancangan ERD yang tidak mengizinkan hubungan *ternary* atau *N-ary*.

#### 2.2.4. Pengertian *Flowchart*

Siallagan (2009:6), “menjelaskan simbol-simbol dalam *Flowchart* adalah sebagai berikut:



Tabel 2.4 Simbol-simbol dalam *Flow Chart*

No.	Simbol	Keterangan
1.		Simbol <i>Start</i> atau <i>End</i> yang mendefinisikan awal atau akhir dari sebuah <i>flowchart</i> .
2.		Simbol pemrosesan yang terjadi pada sebuah alur kerja.
3.		Simbol yang menyatakan bagian dari program (sub program).
4.		Persiapan yang digunakan untuk memberi nilai awal suatu besaran.

Lanjutan Tabel 2.4 Simbol-simbol dalam *Flow Chart*

No.	Simbol	Keterangan
5.		Simbol <i>Input/Output</i> yang mendefinisikan masukan dan keluaran proses.
6.		Menyatakan penyambung ke simbol lain dalam satu halaman.
7.		Menyatakan penyambung ke halaman lainnya.
8.		Menyatakan pencetakan (dokumen) pada kertas.
9.		Menyatakan <i>desicion</i> (keputusan) yang digunakan untuk penyeleksian kondisi di dalam program.
10.		Menyatakan media penyimpanan drum magnetik.
11.		Menyatakan <i>input/output</i> menggunakan disket.
12.		Menyatakan operasi yang dilakukan secara manual.
13.		Menyatakan <i>input/output</i> dari kartu plong.
14.		Menyatakan arah aliran pekerjaan (proses).



15.		<i>Multidocument</i> (banyak dokumen).
16.		<i>Delay</i> (penundaan atau kelambatan).

### 2.2.5. Pengertian Kamus Data

Sukanto dan Shalahuddin (2014:73) menjelaskan, “Kamus data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada DFD. Kamus data adalah kumpulan daftar elemen data yang mengalir pada system perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan). Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur.

Kamus data biasanya berisi:

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

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut:

**Tabel 2.5 Simbol-simbol dalam Kamus Data**

<b>Simbol</b>	<b>Keterangan</b>
=	disusun atau terdiri dari
+	Dan
[]	baik ...atau...
{ } <sup>n</sup>	n kali diulang/ bernilai banyak
()	data opsional



*...*	batas komentar
-------	----------------

Kamus data pada DFD nanti harus dapat dipetakan dengan hasil perancangan basis data yang dilakukan sebelumnya. Jika ada kamus data yang tidak dapat dipetakan pada tabel hasil perancangan basis data berarti hasil perancangan basis data dengan perancangan dengan DFD masih belum sesuai, sehingga harus ada yang diperbaiki baik perancangan basis datanya, perancangan DFD-nya, atau keduanya.

## 2.3. Teori Judul

### 2.3.1. Pengertian Sistem

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

Yakub (2012:1), “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau tujuan tertentu”.

Fatta (2007:3), “Sistem diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi dan saling bergantung sama lain”.

### 2.3.2. Pengertian Informasi

Ladjamudin (2013:8) menjelaskan, informasi sebagai data yang telah diolah menjadi bentuk yang lebih berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun masa yang akan datang.

Sutarman (2012:4) menjelaskan, informasi adalah hasil kegiatan mengolah data yang memberikan bentuk yang lebih berarti dari suatu kejadian.





### **2.3.3. Pengertian Sistem Informasi**

Sutabri (2012:46), “Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasional organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan”.

Ladjamuddin (2013:13), “Sistem informasi adalah sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan atau untuk mengendalikan organisasi”.

### **2.3.4. Pengertian Akademik**

Nugroho (2013:75) menjelaskan, akademik merupakan sebagai segala sesuatu yang berhubungan dengan proses penunjang kegiatan sekolah atau lembaga pendidikan beserta pelaku didalamnya.

### **2.3.5. Pengertian Web**

Sarwono dan Prihartono (2012:40) menjelaskan *web* merupakan system yang menghubungkan dokumen-dokumen dengan *hypertext* dan yang dapat diakses melalui internet dengan bantuan browser.

### **2.3.6. Pengertian Sistem Informasi Akademik Berbasis Web pada SMA Tri Dharma Palembang**

Pengertian sistem informasi akademik berbasis *web* pada SMA Tri Dharma Palembang adalah sistem informasi online yang digunakan untuk memudahkan siswa dalam mendapatkan informasi-informasi baik informasi tentang profil sekolah, nilai-nilai pelajaran, absensi kelas, guru, jadwal pelajaran, berita dan informasi mengenai kegiatan-kegiatan yang sedang atau berlangsung di SMA Tri Dharma dengan menggunakan Bahasa pemrograman PHP dan database MySQL.

---



## 2.4. Teori Program

### 2.4.1. Sekilas Tentang PHP

#### 2.4.1.1. Pengertian PHP

Badiyanto (2013:32), PHP *Hypertext Preprocessor* adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML/PHP banyak dipakai untuk membuat situs *web* dinamis.

Kadir (2008:358), PHP merupakan bahasa pemrograman skrip yang diletakkan dalam server yang biasa digunakan untuk membuat aplikasi Web yang bersifat dinamis.

Nugroho (2013:153), PHP kepanjangannya Hypertext Preprocessor itu bahasa pemrograman berbasis Web. Jadi, PHP itu adalah bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis web (website, blog, atau aplikasi web).

#### 2.4.1.2. Script PHP

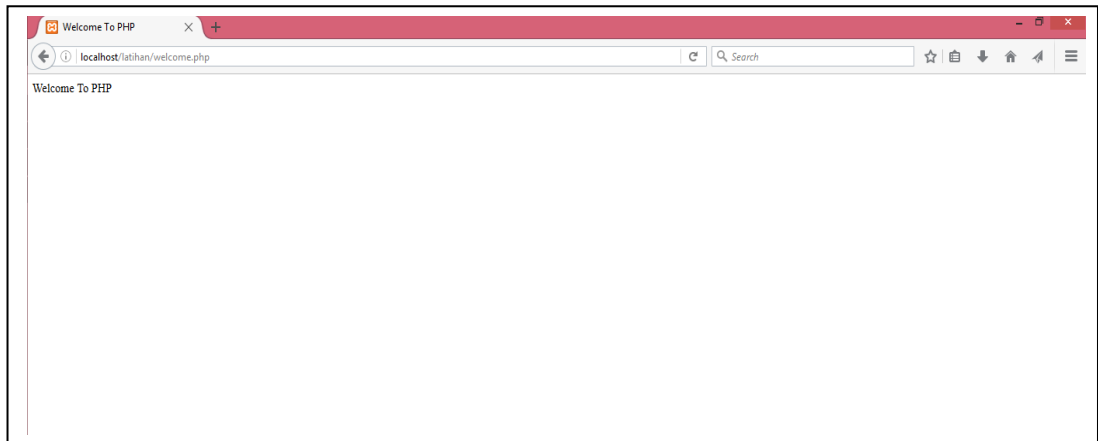
PHP yang merupakan sebuah bahasa scripting yang terpasang pada HTML, dan skrip HTML (*Hyper Text Markup Language*) adalah bahasa standar untuk membuat halaman web. Biasa *file* dituliskan dengan ekstensi *.htm* atau *.html*.

```
<html>
  <head>
    <title>Welcome To PHP</title>

  </head>
  <body>
    Selamat Datang
  </body>
</html>
```



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



**Gambar 2.2.** Tampilan program HTML pada *browser*

#### 2.4.1.3. Teknik Penulisan Script *PHP*

Madcoms (2010:350), *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 2.6.** *Script Dasar PHP*

<pre>&lt;?   [.....] Tempat penulisan script PHP ?&gt;</pre>
Atau
<pre>&lt;?php   [.....] Tempat penulisan script PHP ?&gt;</pre>
Dan untuk menampilkan ke <i>browser</i>



```
<?php
Echo "Welcome to PHP";
?>
```

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:

- 1) Gunakan *tag /\** dan akhiri *tag \*/* apabila jumlah keterangan lebih dari 1 baris.
- 2) Gunakan *tag //*, *tag* ini digunakan untuk keterangan yang hanya terdiri dari 1 baris saja.
- 3) 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 (2008:348), *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.

Winarno (2014:102), *MySQL* adalah sebuah *software database*. *MySQL* merupakan tipe data relasional yang artinya *MySQL* menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan.

Shalahuddin (2013:46), *SQL (Structured Query Language)* adalah bahasa yang digunakan untuk mengelola data pada RDBMS. *SQL* awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.



#### 2.4.2.2. 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

**Tabel 2.7.** 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

Lanjutan Tabel 2.7 Fungsi-fungsi *MySQL* dalam *PHP*

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

Lanjutan Tabel 2.7 Fungsi-fungsi *MySQL* dalam *PHP*

No.	Fungsi <i>MySQL</i>	Penggunaan
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. *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*).

### **2.4.3. 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 juga yang biasa dikenal dengan istilah WYSIWYG (What You See Is What You Get), yang intinya adalah bahwa Anda tidak harus berurusan dengan tag-tag HTML untuk membuat sebuah situs. Selain itu, *Dreamweaver* juga memberikan keleluasaan kepada Anda untuk menggunakannya sebagai media penulisan bahasa pemrograman web”.

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

---



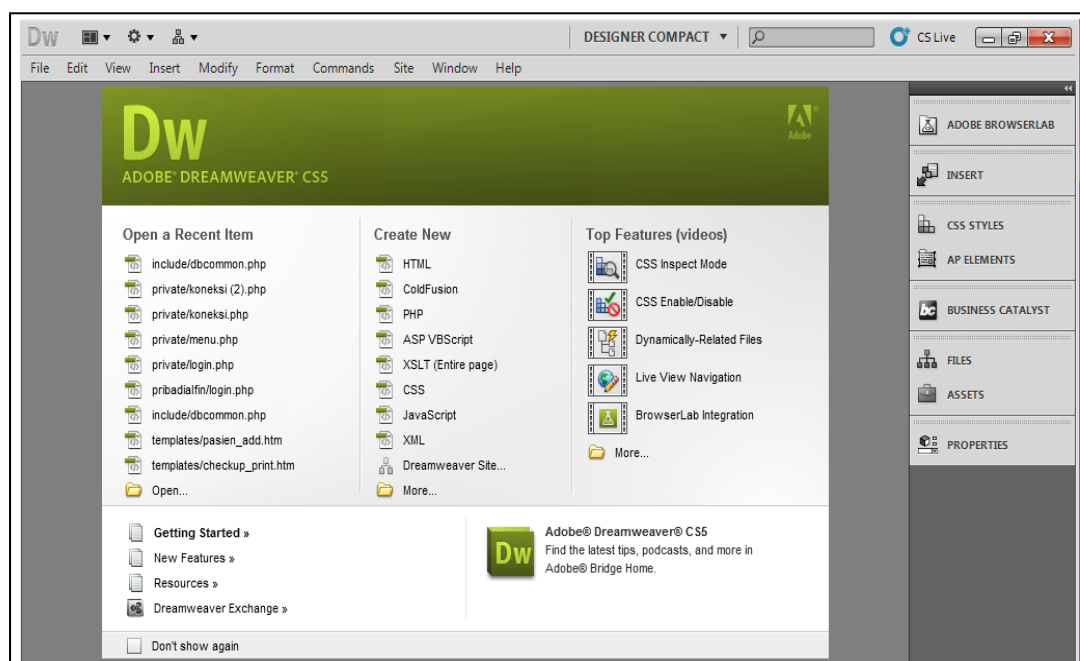


Madcoms (2012:2), “*Dreamweaver* adalah sebuah HTML editor profesional untuk mendesain web secara visual dan mengelola situs atau halaman web”.

Sulistya (2013:1), “*Dreamweaver* adalah sebuah editor profesional yang menggunakan HTML untuk mendesain web secara visual dan mengelola situs atau halaman 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.

#### 2.4.3.2. Starting Page



**Gambar 2.3.** 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*

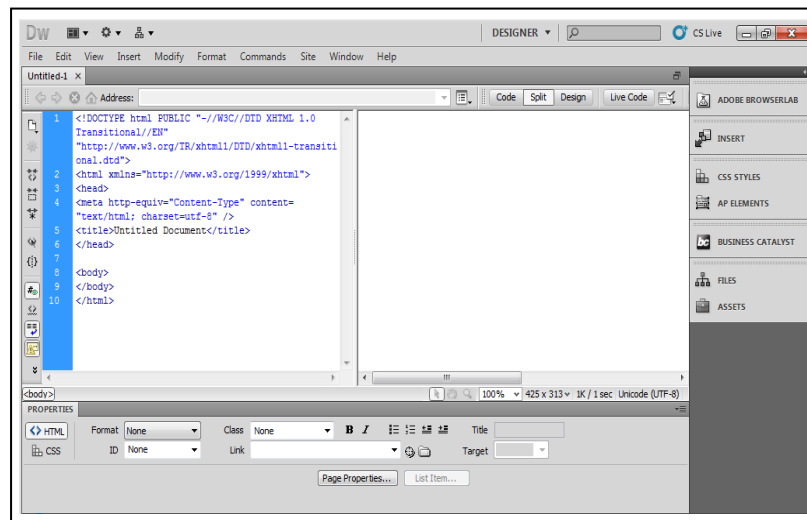
Saat pertama kali menjalankan *Dreamweaver* dengan, akan ditampilkan sebuah kotak dialog *Workspace Setup* yang digunakan untuk memilih tampilan ruang kerja.

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.

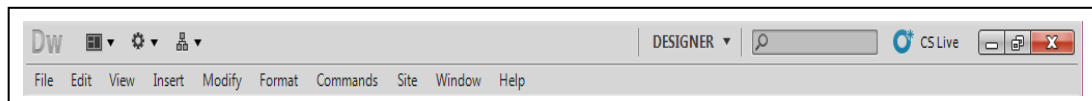
Ruang kerja *Adobe Dreamweaver CS5* memiliki komponen-komponen yang memberikan fasilitas dan ruang untuk menuangkan kreasi saat bekerja, seperti *Application Bar, Toolbar Document, Document Window, Workspace Switcher, Panel Groups, Tag Selector, Property Inspector, Toolbar Browser Navigation*.

---



**Gambar 2.4.** 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.** Application Bar pada Adobe 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 2.6.** Toolbar Document pada Adobe Dreamweaver CS5

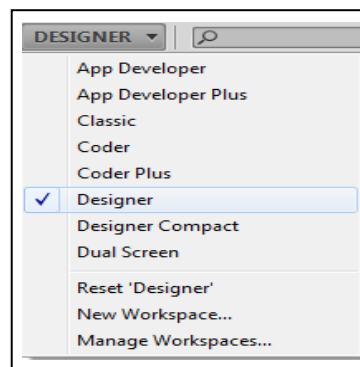


3. **Document Window** (Jendela Dokumen), lembar kerja tempat membuat dan mengedit desain halaman *web*.

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
3 <html xmlns="http://www.w3.org/1999/xhtml">  
4 <head>  
5 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
6 <title>Untitled Document</title>  
7 </head>  
8 <body>  
9 </body>  
10 </html>
```

**Gambar 2.7.** *Document window* pada *Adobe Dreamweaver CS5*

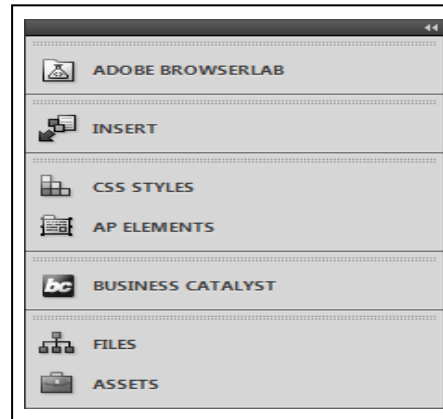
4. **Workspace Switcher**, digunakan untuk mengubah tampilan ruang kerja (*workspace*) *Adobe Dreamweaver CS5*



**Gambar 2.8.** *Workspace Switcher* pada *Adobe 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.



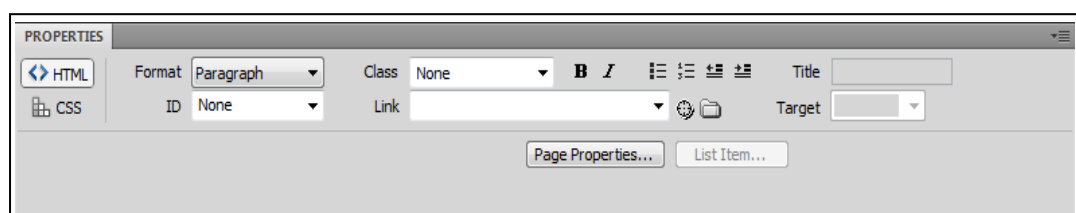
**Gambar 2.9.** Panel Groups pada Adobe Dreamweaver CS5

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 2.10.** Tag Selector pada Adobe 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 2.11.** Property Inspector pada Adobe 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 2.12.** *Toolbar Browser Navigation* pada *Adobe Dreamweaver CS5*