



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Mulyono (2008:1) menjelaskan bahwa Komputer adalah seperangkat alat elektronik yang terdiri atas peralatan input, alat yang mengolah input, dan peralatan output yang memberikan informasi, serta bekerja secara otomatis.

Daulay (2007:17) menjelaskan bahwa Komputer adalah sekumpulan alat logik yang dapat menerima data, mengolah data dan menyimpan data dengan menggunakan program yang terdapat pada memori sistem komputer kemudian memberikan hasil pengolahan tersebut dalam bentuk *output*.

Dari pengertian tersebut, dapat disimpulkan bahwa komputer merupakan suatu alat bantu untuk membaca *input* data secara elektronik kemudian memproses datanya berdasarkan urutan intruksi atau program yang tersimpan sehingga menghasilkan *output* berupa informasi.

2.1.2. Pengertian Program

Kadir dan Triwahyuni (2005:8) menjelaskan bahwa Program adalah deretan instruksi yang digunakan untuk mengendalikan komputer sehingga komputer dapat melakukan tindakan sesuai yang dikendahi pembuatnya.

Munir (2011:13) menjelaskan bahwa Program adalah algoritma yang ditulis dalam bahasa komputer.

Dari pengertian tersebut, dapat disimpulkan bahwa program merupakan kumpulan dari perintah-perintah atau intruksi-intruksi yang disusun sehingga dapat dipahami dan dijalankan oleh komputer.

2.1.3. Pengertian Data

Herdiansyah (2013:8) menjelaskan bahwa Data adalah suatu atribut yang melekat pada suatu objek tertentu, berfungsi sebagai informasi yang dapat dipertanggungjawabkan dan diperoleh melalui suatu metode/instrumen pengumpulan data.



Teguh (2011:87) menjelaskan bahwa Data dapat diartikan sebagai kumpulan fakta-fakta, serangkaian bukti-bukti, sesuatu yang secara pasti diketahui atau serangkaian informasi yang terdapat disekitar kita.

Dari pengertian tersebut, dapat disimpulkan bahwa, data merupakan kumpulan angka-angka, huruf-huruf, simbol-simbol khusus atau gabungan darinya yang belum memiliki arti sehingga dapat diolah untuk menghasilkan suatu informasi.

2.1.4. Pengertian Pengolahan Data

Ladjamudin (2005:9) menjelaskan bahwa Pengolahan Data adalah masa atau waktu yang digunakan untuk mendeskripsikan perubahan bentuk data menjadi informasi yang memiliki kegunaan.

Kristanto (2011:8) menjelaskan bahwa Pengolahan Data adalah waktu yang menggambarkan perubahan bentuk data menjadi informasi yang memiliki kegunaan.

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

2.1.5. Pengertian Aplikasi

Budiharto (2013:5) menjelaskan bahwa Aplikasi merupakan program yang dapat berjalan di komputer tersendiri (*stand alone computer*), dari mulai program yang simpel sampai dengan program besar dan rumit.

2.1.6. Pengertian Penentuan Jurusan

Menurut Kamus Besar Bahasa Indonesia, Penentuan adalah proses, cara, atau perbuatan menentukan.

Menurut Kamus Besar Bahasa Indonesia, Jurusan adalah bagian dari fakultas atau sekolah tinggi yang bertanggungjawab untuk mengelola dan mengembangkan suatu bidang studi, misalnya jurusan akuntansi, jurusan manajemen.



2.1.7. Pengertian Web

Sibero (2014:11) menjelaskan bahwa *World Wide Web* (W3) atau yang lebih dikenal dengan istilah *web* adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia dan lainnya pada jaringan internet.

2.1.8. Pengertian Aplikasi Penentuan Jurusan Berbasis Web Pada SMA YPI

Tunas Bangsa Palembang

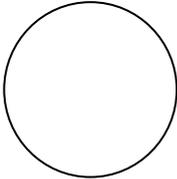
Penulis mendefinisikan Aplikasi Penentuan Jurusan Berbasis Web Pada SMA YPI Tunas Bangsa Palembang merupakan sebuah *software* yang dibangun untuk membantu mempermudah siswa dan juga guru dalam proses pendaftaran dan penyeleksian nilai untuk menentukan jurusan sesuai minat dan bakat yang siswa tersebut miliki.

2.2. Teori Khusus

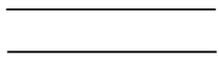
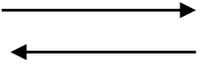
2.2.1. Pengertian *Data Flow Diagram* (DFD)

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

Tabel 2.1. Simbol-Simbol *Data Flow Diagram* (DFD)

No	Simbol	Keterangan
1.		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. Catatan : Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.

Lanjutan **Tabel 2.1.** Simbol-Simbol *Data Flow Diagram* (DFD)

No	Simbol	Keterangan
2.		<p><i>File</i> atau basis data 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</i> (ERD)), <i>Conceptual Data Model</i>(CDM), <i>Physical Data Model</i> (PDM).</p> <p>Catatan: Nama yang diberikan pada sebuah penyimpanan biasanya kata benda.</p>
3.		<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 sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan</p> <p>catatan :</p> <p>nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) berupa kata benda</p>
4.		<p>Aliran data; merupakan data yang dikirimantar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>).</p> <p>catatan :</p> <p>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>

Sumber : Sukamto dan Shalahuddin (2014:71-72)



Sukamto dan Shalahuddin (2014:72) menjelaskan tentang tahapan-tahapan perancangan dengan menggunakan *Data Flow Diagram* yaitu:

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 kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu di-*breakdown* lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 yang di-*breakdown*.

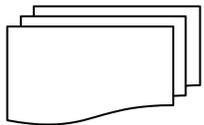
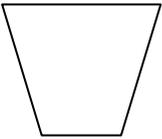
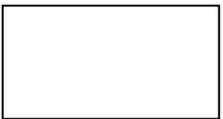
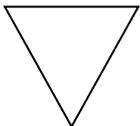
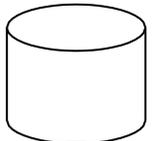
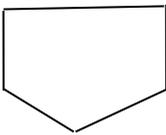
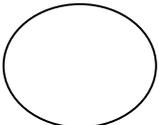
4. Membuat DFD Level 3 dan seterusnya

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 2.

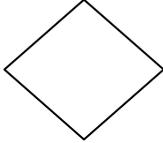
2.2.2. Pengertian *Blockchart*

Kristanto (2011:75) menjelaskan, *Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *Blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.

Tabel 2.2. Simbol-Simbol *Blockchart*

No	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan
2.		Multi-Dokumen
3.		Proses Manual
4.		Proses dilakukan oleh komputer
5.		Menandakan Dokumen yang diarsipkan (dokumen manual)
6.		Data Penyimpanan (<i>storage</i>)
7.		Proses apasaja yang tidak terdefinisikan termasuk aktivitas fisik
8.		Terminasi yang mewakili symbol 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.

Lanjutan **Tabel 2.2.** Simbol-Simbol *Blokchart*

No	Simbol	Keterangan
10.		Terminal yang menandakan awal dan akhir dari suatu aliran.
11.		Pengambilan keputusan (<i>Decision</i>).
12.		Layar peraga (monitor)
13.		Pemasukkan data secara manual.

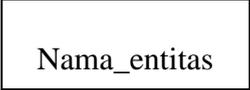
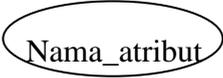
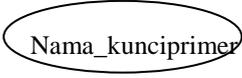
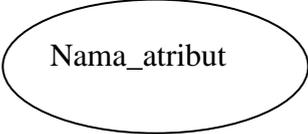
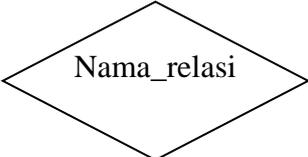
Sumber : Kristanto (2011:69-71)

2.2.3. Pengertian *Entity Relationship Diagram* (ERD)

Sukamto dan Shalahuddin, (2014:50) menjelaskan , *Entity Relationship Diagram* (ERD) adalah pemodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basis data relational.

Al Fatta (2007:121) menjelaskan, ERD (*Entity Relationship Diagram*) adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis. Entitas biasanya menggambarkan jenis informasi yang sama. Dalam entitas digunakan untuk menghubungkan antar entitas yang sekaligus menunjukkan hubungan antar data. Pada akhirnya ERD bisa juga digunakan untuk menunjukkan aturan-aturan bisnis yang ada pada sistem informasi yang dibangun.

**Tabel 2.3.** Simbol-Simbol *Entity Relationship Diagram* (ERD)

No	Simbol	Keterangan
1.	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 komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
2.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
3.	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)
4.	Atribut multinilai/ <i>multivalued</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja

Lanjutan **Tabel 2.3.** Simbol-Simbol *Entity Relationship Diagram* (ERD)

No	Simbol	Keterangan
6.		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.

Sumber: Sukamto dan Shalahuddin (2014:50-51)

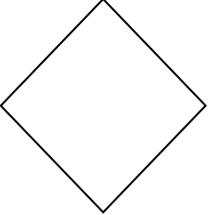
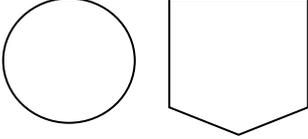
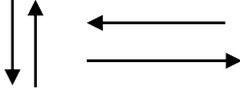
2.2.4. Flowchart Diagram

Supardi (2013:51) menjelaskan, *Flowchart* merupakan diagram alur yang sering digunakan dalam membuat atau menggambarkan logika program. Namun, *flowchart* juga dapat menggambarkan jalannya sistem.

Tabel 2.4. Simbol-simbol dalam *Flow Chart*

No	Simbol	Keterangan
1.		Simbol dokumen Menunjukkan dokumen <i>Input</i> dan <i>output</i> baik untuk proses manual, mekanik, atau komputer
2.		Simbol <i>input</i> / <i>output</i> digunakan untuk mewakili data <i>input</i> / <i>ouput</i>
3.		Menunjukkan <i>input/output</i> yang menggunakan kartu plong (<i>punched card</i>).

Lanjutan **Tabel 2.4.** Simbol-Simbol dalam *Flowchart*

No	Simbol	Keterangan
4.	<p>Simbol Keputusan</p> 	Simbol keputusan (<i>decision symbol</i>) digunakan untuk suatu penyeleksian kondisi didalam program
5.	<p>Simbol titik terminal</p> 	Simbol titik terminal (<i>terminal point symbol</i>) Digunakan untuk menunjukkan awal dan akhir dari suatu proses.
6.	<p>Simbol proses</p> 	Simbol proses digunakan untuk mewakili suatu proses.
7.	<p>Simbol persiapan</p> 	Simbol persiapan (<i>preparation symbol</i>) digunakan untuk memberi nilai awal suatu besaran
8.	<p>Simbol penghubung</p> 	Simbol penghubung (<i>connector symbol</i>) digunakan untuk menunjukkan sambungan dari bagan alir yang terputus dihalaman yang masih sama atau dihalaman lainnya.
9.	<p>Simbol garis alir</p> 	Menunjukkan arus dari proses

Sumber: Supardi (2013:53-59)



2.2.5. Pengertian Kamus Data

Sukanto dan Shalahuddin (2014:73) menjelaskan, Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan).

Table 2.5. Simbol-simbol dalam Kamus Data

No	Simbol	Arti
1	=	disusun atau terdiri atas
2	+	Dan
3	[]	baik ...atau...
4	{ } ⁿ	N kali diulang/bernilai banyak
5	()	Data operasional
6	*...*	Batas komentar

Sumber : Sukanto dan Shalahuddin (2014:74)

2.3. Teori Program

2.3.1. Pengertian Basis Data (*Database*)

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

Al-Fatta (2007:10) menjelaskan, *database* yaitu kumpulan data dan informasi yang diorganisasikan sedemikian rupa sehingga mudah diakses pengguna sistem informasi.

2.3.2. Pengertian HTML (*HyperText Markup Language*)

Sibero (2014:19) menjelaskan, *HyperText Markup Language* atau HTML adalah bahasa yang digunakan pada dokumen *web* sebagai bahasa untuk pertukaran dokumen *web*.



Suryana dan Koesheryatin (2014:29) menjelaskan, *HyperText Markup Language* (HTML) adalah bahasa yang digunakan untuk menulis halaman *web*. HTML merupakan pengembangan dari *Standard Generalized Markup Language* (SGML).

2.3.3. Pengertian XAMPP

Andi (2007:30) menjelaskan, XAMPP adalah satu paket instalasi *Apache*, *PHP*, dan *MySQL* secara instan yang dapat digunakan untuk membantu proses instalasi ketiga produk yang sama seperti *PHPTriad*.

2.3.4. Pengertian *Cascading Style Sheet* (CSS)

Sibero (2014:112) menjelaskan, *Cascading Style Sheet* dikembangkan untuk menata gaya pengaturan halaman *web*. *Cascading Style Sheet* memiliki arti gaya menata halaman bertingkat, yang berarti setiap satu elemen yang telah diformat dan memiliki anak dan telah diformat, maka anak dari elemen tersebut secara otomatis mengikuti format elemen induknya.

Suryana dan Koesheryatin (2014:101) menjelaskan, CSS (*Cascading Style Sheet*) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu *website*, baik tata letaknya, jenis huruf, warna, dan semua yang berhubungan dengan tampilan.

2.3.5 Pengertian *MySQL*

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



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

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

2.5.5.1. Fungsi-Fungsi *mySQL*

Untuk dapat menggunakan semua perintah *SQL* pada *PHP*, kita memerlukan suatu fungsi *mySQL*. Fungsi ini berguna untuk mengantarkan perintah *SQL* pada *PHP* menuju ke server sehingga perintah tersebut dapat dieksekusi oleh server *mySQL*.

Tabel 2.6. Fungsi-Fungsi *MySQL*

No	Fungsi <i>MySQL</i>	Penggunaan
1	<i>Mysql_affected_rows()</i>	Digunakan untuk mengambil jumlah record hasil eksekusi. Berlaku untuk query insert, update, replace dan delete.
2	<i>Mysql_close()</i>	Digunakan untuk menutup koneksi database.
3.	<i>Mysql_connect()</i>	Digunakan untuk membuka koneksi database.
4.	<i>Mysql_create_db()</i>	Digunakan untuk membuat database baru.
5.	<i>Mysql_db_name()</i>	Digunakan untuk mengambil nama database dari hasil <i>mysql_list_dbs()</i> .
6.	<i>Mysql_drop_db()</i>	Digunakan untuk menghapus suatu database.
7.	<i>Mysql_errno()</i>	Digunakan mengambil nomor index pesan kesalahan.



Lanjutan Tabel 2.6. Fungsi-Fungsi MySQL

No	Fungsi mySQL	Kegunaan
8.	<i>Mysql_error()</i>	Digunakan untuk mengambil pesan kesalahan.
9.	<i>Mysql_fetch_array()</i>	Digunakan mengambil hasil record dan mendefinisikan dalam bentuk array.
10.	<i>Mysql_fetch_assoc()</i>	Digunakan mengambil hasil record seperti <i>mysql_fetch_array</i> dengan tipe <i>MYSQL_ASSOC</i> .
11.	<i>Mysql_free_result()</i>	Digunakan untuk mengosongkan memori hasil proses.
12.	<i>Mysql_insert_id()</i>	Digunakan untuk mengambil nomor ID terakhir setelah eksekusi. Fungsi ini berlaku untuk tabel yang memiliki kolom ID dengan tipe <i>AUTO INCREMENT</i> .
13.	<i>Mysql_list_dbs()</i>	Digunakan untuk mengambil daftar database aktif pada <i>server</i> .
14.	<i>Mysql_num_rows()</i>	Digunakan untuk mengambil jumlah baris data hasil query. Fungsi ini hanya berlaku untuk query dengan perintah <i>SELECT</i> dan <i>SHOW</i> .
15.	<i>Mysql_pconnect()</i>	Digunakan untuk membuka koneksi persisten database.
16.	<i>Mysql_ping()</i>	Digunakan untuk menguji respon server atau membuka kembali koneksi database yang terputus. Fungsi ini berlaku untuk koneksi database menggunakan <i>mysql_connect()</i>
17.	<i>Mysql_query()</i>	Digunakan untuk mengirimkan perintah query pada server.

Lanjutan **Tabel 2.6.** Fungsi-Fungsi MySQL

No	Fungsi mySQL	Kegunaan
18.	<i>Mysql_real_escape_string()</i>	Digunakan untuk mengubah karakter tanda baca menjadi bentuk yang aman pada database.
19.	<i>Mysql_result()</i>	Digunakan untuk mengambil data hasil.
20.	<i>Mysql_Select_db()</i>	Digunakan untuk memilih dan mengaktifkan database.
21.	<i>Mysql_unbuffered_query()</i>	Digunakan untuk mengirimkan perintah pada server dan hasilnya tidak disimpan dalam buffer memory.

Sumber : Sibero (2014:71-79)

2.3.6. Adobe Dreamweaver CS5

Wahana (2011:2) menjelaskan, *dreamweaver* adalah salah satu aplikasi yang dapat dimanfaatkan untuk merancang desainweb secara visual atau aplikasi editor web.

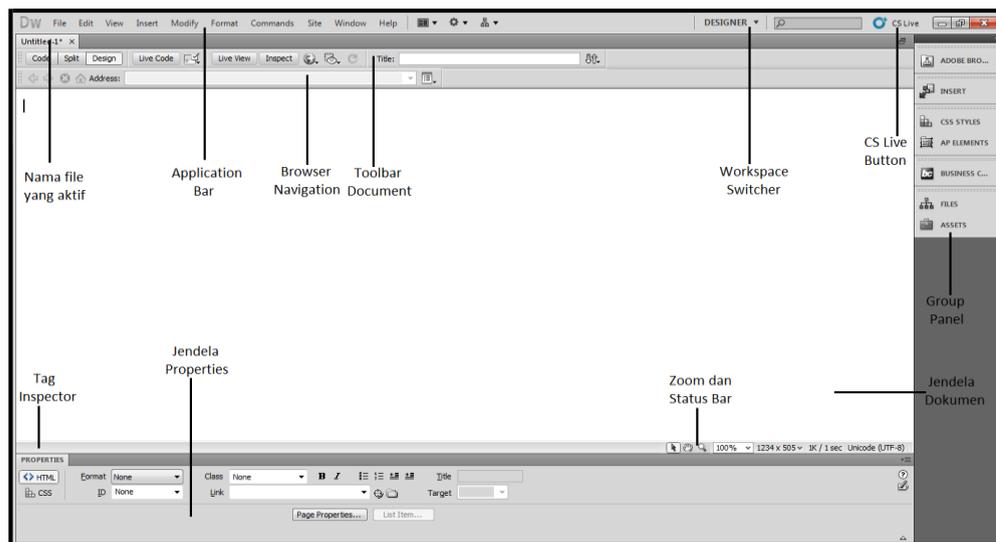
Pada *dreamweaver CS5*, terdapat beberapa kemampuan bukan hanya sebagai *software* untuk desain *Web* saja tetapi juga untuk menyunting kode serta pembuatan aplikasi *Web* dengan menggunakan berbagai bahasa pemrograman *Web*.



Gambar 2.1. Logo *Dreamweaver CS5*



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



Gambar 2.2. 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 2.3. Tampilan *Application Bar*

2. *Toolbar Document*

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



Gambar 2.4. Tampilan *Toolbar Document*

3. Jendela Dokumen

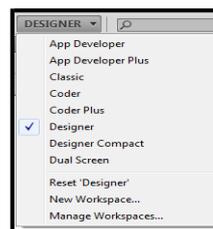
Jendela Dokumen adalah lembar kerja tempat membuat dan mengedit desain halaman *web*.



Gambar 2.5. Jendela Dokumen

4. *Workspace Switcher*

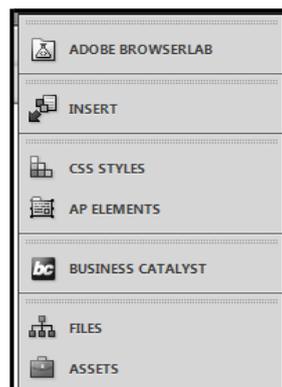
Digunakan untuk mengubah tampilan ruang kerja (*workspace*) Adobe Dreamweaver CS5.



Gambar 2.6. Tampilan *Workspace Switcher*

5. *Panel Groups*

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



Gambar 2.7. Tampilan *Panel Group*



6. Tag Selector

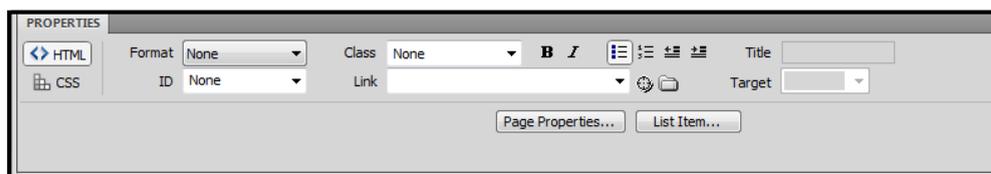
Bagian ini menampilkan hirarki pekerjaan yang sedang terpilih pada jendela dokumen, dapat juga digunakan untuk memilih objek pada jendela desain berdasarkan jenis atau kategori objek tersebut.



Gambar 2.8. Tag Selector

7. Panel Property

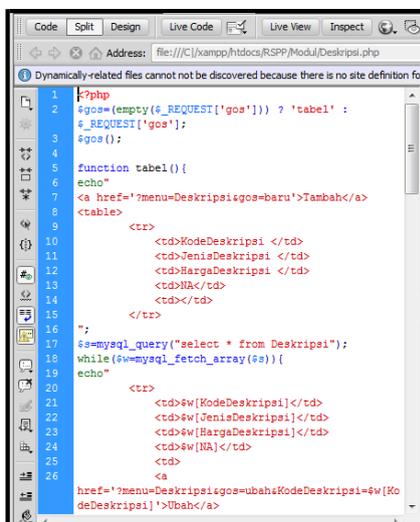
Digunakan untuk melihat dan mengubah berbagai properti objek atau teks pada jendela *design*.



Gambar 2.9. Tampilan Jendela *Properties*

8. Toolbar Coding

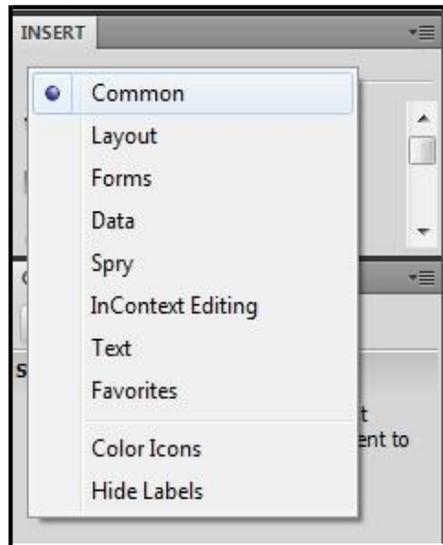
Berisi tombol-tombol yang digunakan untuk melakukan operasi kode-kode standar. Jendela ini hanya akan tampil pada jendela *Code* dan *Split*.



Gambar 2.10. Tampilan *Toolbar Coding*

9. *Panel Insert*

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



Gambar 2.11. Tampilan *Panel Insert*

10. *Panel Groups*

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



Gambar 2.12. Tampilan *Panel Groups*



11. *Panel Files*

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



Gambar 2.13. Tampilan *Panel File*

2.3.7. *PHP*

2.3.7.1. Pengertian *PHP*

Andi (2012:76) menjelaskan bahwa “*PHP* merupakan bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses dan mengolah data secara dinamis. *PHP* dapat dikatakan sebagai sebuah server-side embedded script language, artinya sintak-sintak dan perintah program yang Anda tulis akan sepenuhnya dijalankan oleh server tetapi dapat disertakan pada halaman *HTML* biasa”.



Gambar 2.14 Tampilan Logo *PHP*



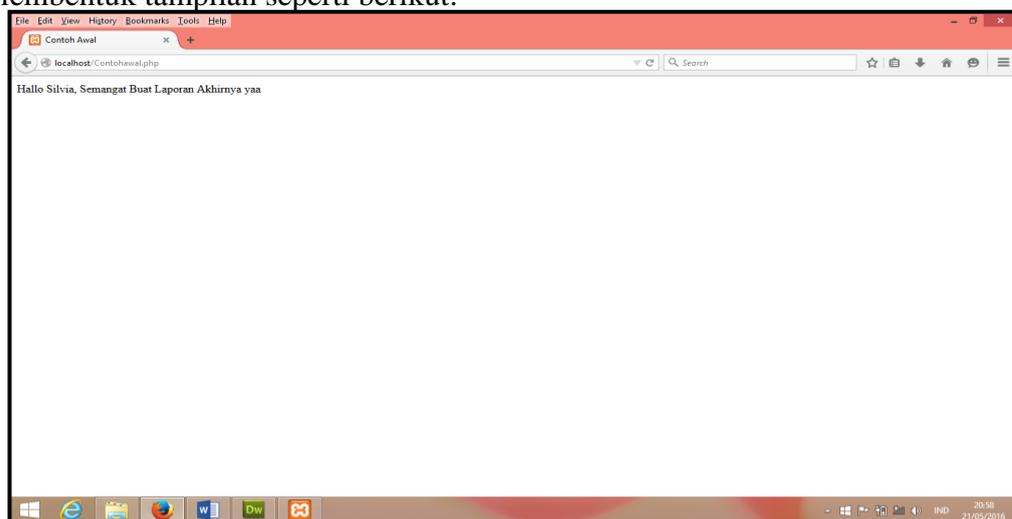
2.3.7.2. Script PHP

Script PHP disimpan sebagai *plain text* dalam format ASCII, yang memungkinkan *script* dapat ditulis hampir disemua editor teks seperti notepad, wordpad, dan lain-lain. Script PHP adalah kode yang disertakan disebuah halaman HTML dan kode tersebut dijalankan oleh server yang akan dieksekusi sebelum dikirim ke browser.

Contoh *script*:

```
<HTML>
<HEAD>
<TITLE> Contoh Awal </TITLE>
</HEAD>
<?php
    Print ("Hallo Silvia, Semangat Buat Laporan Akhirnya yaa
    ");
?>
<BODY>
</BODY>
</HTML>
```

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



Gambar 2.15. Contoh Script PHP



2.3.7.3. Teknik Penulisan *PHP*

Penulisan *script* PHP memiliki aturan sebagai berikut :

1. Diawali `<?PHP` dan diakhiri `?>` atau diawali `<? Dan diakhiri ?>`
2. `<script language="php">` dan `</script>`
3. `<% dan %>`
4. *Script* PHP bersifat *case-sensitive*, sehingga penulisan huruf besar dan huruf kecil pada *script* PHP harus diperhatikan. Untuk nama *identifier* yang dibuat oleh *programmer*, sedangkan yang *noncase-sensitive* untuk *built-in* dari PHP.