BAB II
TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

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

Hamacher dan Zaky dalam Sutanta (2011:2) mengemukakan, "komputer sebagai mesin penghitung elektronik yang cepat, dapat menerima input digital, memprosesnya sesuai dengan suatu program tersimpan (*stored program*) dan menghasilkan output informasi".

Jogiyanto dalam Sutanta (2011:2) mengemukakan, "komputer sebagai alat elektronik yang dapat menerima input data, dapat mengolah data, dapat memberikan informasi, menggunakan program tersimpan (*stored program*), dapat menyimpan program dan hasil pengolahan, serta bekerja secara otomatis".

Jadi, komputer adalah sebuah alat elektronik yang dapat memproses data dengan cepat sesuai dengan program atau perintah untuk menghasilkan informasi.

2.1.2. Pengertian Perangkat Lunak (*Software*)

Ladjamudin (2005:20), "*software* merupakan kumpulan dari perintah/ fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer melaksanakan tugas tertentu".

Sukamto dan Shalahuddin (2015:2) mengemukakan, "perangkat lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*)".

Jadi, perangkat lunak adalah program komputer yang memiliki aturan tertentu dalam penulisannya dan digunakan untuk memerintahkan komputer mengerjakan tugas tertentu.

2.1.3. Pengertian Teknologi

Sujatmiko (2012:270) mengemukakan, "teknologi adalah penerapan keilmuan yang mempelajari dan mengembangkan kemampuan dari suatu rekayasa dengan langkah dan teknik tertentu dalam suatu bidang".

2.1.4. Pengertian Data

Sujatmiko (2012:76) mengemukakan, "data adalah kumpulan dari angka-angka maupun karakter yang tidak memiliki arti. Data dapat diolah sehingga menghasilkan informasi".

McLeod dalam Yakub (2012:5) mengemukakan, "data adalah kenyataan yang menggambarkan adanya suatu kejadian *(event)*, data terdiri dari fakta *(fact)* dan angka yang secara relatif tidak berarti bagi pemakai".

Jadi, data adalah kumpulan fakta-fakta yang berupa angka, huruf, dan karakter lainnya yang diperoleh dari suatu kejadian dan tidak memiliki arti.

2.1.5. Pengertian Informasi

Sutabri (2012:22) mengemukakan, "informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan".

Sujatmiko (2012:135) mengemukakan, "informasi adalah serangkaian simbol yang disusun menurut aturan tertentu dapat menyampaikan suatu pesan".

Menurut McLeod dalam Yakub (2012:8) "informasi *(information)* adalah data yang diolah menjadi bentuk lebih berguna dan lebih berarti bagi yang menerimanya".

Jadi, informasi adalah kumpulan data yang telah diolah menjadi bentuk yang memiliki arti sehingga dapat digunakan dalam proses pengambilan keputusan oleh penggunanya.

2.2. Teori Khusus

2.2.1. *Data Flow Diagram (DFD)*

Sukamto dan Shalahuddin (2015:70) mengemukakan, "*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*).

Kristanto (2008:61) mengemukakan, "*DFD* adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut". Ada dua tahapan dalam *DFD*, yaitu:

1. Diagram Konteks (*Context Diagram*)

   Diagram konteks adalah sebuah diagram sederhana yang menggambarkan hubungan antara entiti luar, masukan dan keluaran dari sistem. Diagram konteks direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem.

2. *DFD Leveled*

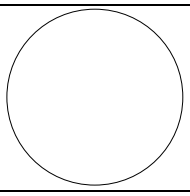
   Model ini menggambarkan sistem sebagai jaringan kerja antar fungsi yang berhubungan satu dengan yang lain dengan aliran dan penyimpanan data. Dalam *DFD leveled* ini akan terjadi penurunan level dimana dalam penurunan level yang lebih rendah harus mampu mempresentasikan proses tersebut ke dalam spesifikasi proses yang jelas. Dalam *DFD leveled* bisa dimulai dari *DFD level 0* kemudian turun ke *DFD level 1* dan seterusnya. Setiap penurunan hanya dilakukan bila perlu.

   Adapun simbol-simbol *Data Flow Diagram* adalah sebagai berikut:

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

No.	Nama	Simbol	Keterangan
1.	Entiti luar		Entiti luar merupakan sumber atau tujuan dari aliran data dari atau ke sistem
2.	Aliran data		Aliran data menggambarkan aliran data dari satu proses ke proses lainnya

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

No.	Nama	Simbol	Keterangan
3.	Proses		Proses atau fungsi yang mentransformasikan data secara umum
4.	Berkas atau tempat penyimpanan		Berkas atau tempat penyimpanan merupakan komponen yang berfungsi untuk menyimpan data atau file

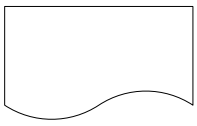
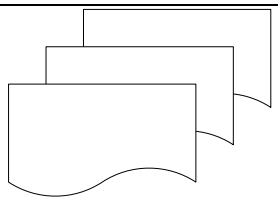
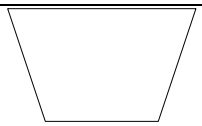
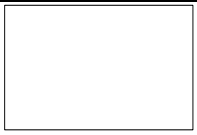
(**Sumber**: Kristanto, 2008:64-65)

2.2.2. *Block Chart*

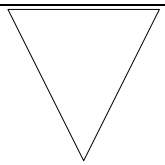
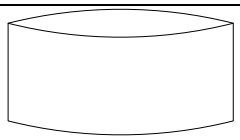
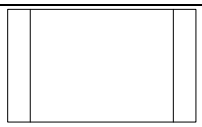
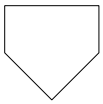
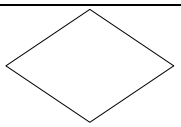
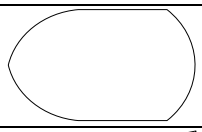
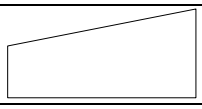
Kristanto (2008:75) mengemukakan, "*block chart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu".

Adapun simbol-simbol *block chart* yang sering digunakan adalah sebagai berikut:

Tabel 2.2. Simbol-Simbol *Block Chart*

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

Lanjutan **Tabel 2.2.** Simbol-Simbol *Block Chart*

No.	Simbol	Arti
5.		Menandakan dokumen yang diarsipkan (arsip manual)
6.		Data penyimpanan (*data storage*)
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.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama
10.		Terminasi yang menandakan awal dan akhir dari suatu aliran
11.		Pengambilan keputusan (*dicision*)
12.		Layar peraga (*monitor*)
13.		Pemasukan data secara manual

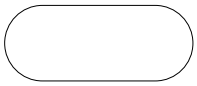
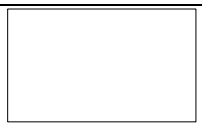
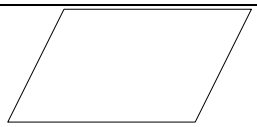
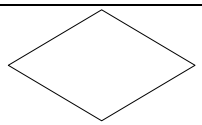
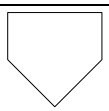
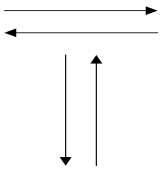
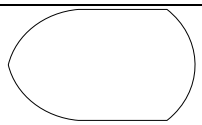
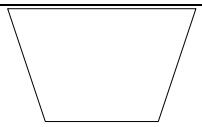
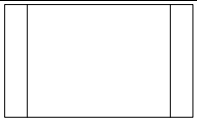
(**Sumber**: Kristanto, 2008:75-77)

2.2.3. *Flowchart*

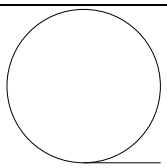
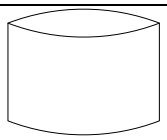
Community (2011:16) mengemukakan, "*flowchart* adalah simbol-simbol pekerjaan yang menunjukkan bagan aliran proses yang saling terhubung. Jadi, setiap simbol *flowchart* melambangkan pekerjaan dan instruksinya".

Adapun simbol-simbol *flowchart* adalah sebagai berikut:

Tabel 2.3. Simbol-Simbol *Flowchart*

No.	Simbol	Arti
1.		Simbol ***Start*** atau ***End*** yang mendefinisikan awal atau akhir dari sebuah *flowchart*
2.		Simbol pemrosesan yang terjadi pada sebuah alur kerja
3.		Simbol ***Input/Output*** yang mendefinisikan masukan dan keluaran proses
4.		Simbol untuk memutuskan proses lanjutan dari kondisi tertentu
5.		Simbol konektor untuk menyambung proses pada lembar kerja yang sama
6.		Simbol konektor untuk menyambung proses pada lembar kerja yang berbeda
7.		Simbol untuk menghubungkan antar proses atau antar simbol
8.		Simbol yang menyatakan piranti keluaran, seperti layar *monitor, printer*, dll
9.		Simbol yang mendefinisikan proses yang dilakukan secara manual
10.		Simbol masukan atau keluaran dari atau ke sebuah dokumen
11.		Simbol yang menyatakan bagian dari program (subprogram)

Lanjutan **Tabel 2.3.** Simbol-Simbol *Flowchart*

No.	Simbol	Arti
12.		Simbol masukan atau keluaran dari atau ke sebuah pita *magnetic*
13.		Simbol ***database*** atau basis data

(**Sumber**: Community, 2012:17)

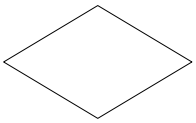
2.2.4. *Entity Relationship Diagram (ERD)*

Yakub (2012:60) mengemukakan, "*ERD* merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem secara abstrak".

Priyadi (2014:20) mengemukakan, "pemodelan basis data dengan menggunakan diagram relasi antar entitas, dapat dilakukan dengan menggunakan suatu pemodelan basis data yang bernama *Diagram Entity-Relationship* (selanjutnya disingkat dengan Diagram E-R)".

Entity Relationship Diagram (*ERD*) memiliki beberapa simbol atau notasi dasar sebagai berikut:

Tabel 2.4. Simbol-Simbol *Entity Relationship Diagram*

No.	Nama	Simbol	Keterangan
1.	Entitas		Notasi untuk mewakili suatu objek dengan karakteristik sama, yang dilengkapi atribut, sehingga pada suatu lingkungan nyata setiap objek akan berbeda dengan objek lainnya.
2.	Relasi		Notasi yang digunakan untuk menghubungkan beberapa entitas berdasarkan fakta pada suatu lingkungan.

Lanjutan **Tabel 2.4.** Simbol-Simbol *Entity Relationship Diagram*

No.	Nama	Simbol	Keterangan
3.	Atribut		Notasi yang menjelaskan karakteristik suatu entitas dan juga relasinya.
4.	Garis penghubung	_____	Notasi untuk merangkaikan keterkaitan antara notasi-notasi yang digunakan dalam *diagram E-R*, yaitu entitas, relasi, dan atribut.

(**Sumber**: Priyadi, 2014:20)

2.2.5. Kamus Data *(Data Dictionary)*

Sukamto dan Shalahuddin (2015:73) mengemukakan, "kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluara (*output*) dapat dipahami secara umum (memiliki standar cara penulisan)".

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

Tabel 2.5. Simbol-Simbol Kamus Data

Simbol	Keterangan
=	Disusun atau terdiri dari
+	Dan
[|]	Baik…atau…
$\{\}^n$	n kali diulang/ bernilai banyak
()	Data opsional
…	Batas komentar

(**Sumber**: Sukamto dan Shalahuddin, 2015:74)

2.3. Pengertian-Pengertian Judul

2.3.1. Pengertian Aplikasi

Sujatmiko (2012:23) mengemukakan, "aplikasi (*application*) adalah program komputer yang dibuat oleh suatu perusahaan komputer untuk membantu manusia dalam mengerjakan tugas tertentu, misalnya *Ms-Word, Ms-Excel*".

2.3.2. Pengertian Pengolahan Data

Ladjamudin (2005:9) mengemukakan, "pengolahan data adalah masa atau waktu yang digunakan untuk mendeskripsikan perubahan bentuk data menjadi informasi yang memiliki kegunaan".

Sutarman (2012:4) mengemukakan, "pengolahan data (*data processing*) adalah proses perhitungan/ transformasi data *input* menjadi informasi yang mudah dimengerti ataupun sesuai dengan yang diinginkan".

Kristanto (2008:8) mengemukakan, "pengolahan data adalah waktu yang digunakan untuk menggambarkan perubahan bentuk data menjadi informasi yang memiliki kegunaan".

Jadi, pengolahan data adalah waktu yang digunakan dalam mengubah data menjadi informasi, sehingga informasi tersebut dapat dimanfaatkan oleh penggunanya.

2.3.3. Pengertian Keberatan

Peraturan Menteri Keuangan Republik Indonesia Nomor 9/PMK.03/2013 BAB I Pasal 1 mengemukakan, "keberatan adalah surat yang diajukan oleh Wajib Pajak kepada Direktur Jenderal Pajak mengenai keberatan terhadap suatu surat ketetapan pajak atau pemotongan atau pemungutan pajak oleh pihak ketiga".

2.3.4. Pengertian Kantor Wilayah Direktorat Jenderal Pajak Sumatera Selatan dan Kepulauan Bangka Belitung

Kantor Wilayah Direktorat Jenderal Pajak Sumatera Selatan dan Kepulauan Bangka Belitung adalah instansi vertikal Direktorat Jenderal Pajak yang berada di bawah dan bertanggung jawab langsung kepada Direktur Jenderal Pajak dengan

wilayah kerja meliputi provinsi Sumatera Selatan dan Kepulauan Bangka Belitung.

2.3.5. Aplikasi Pengolahan Data Keberatan pada Kantor Wilayah Direktorat Jenderal Pajak Sumatera Selatan dan Kepulauan Bangka Belitung

Aplikasi Pengolahan Data Keberatan pada Kantor Wilayah Direktorat Jenderal Pajak Sumatera Selatan dan Kepulauan Bangka Belitung adalah suatu aplikasi yang digunakan oleh penanggungjawab penyelesaian data keberatan untuk mengelola data keberatan yang masuk ke bidang Keberatan, Banding, dan Pengurangan sehingga mempermudah pertukaran, pencarian, dan proses pengolahan data keberatan.

2.4. Teori Program

2.4.1. *PHP (PHP Hypertext Preprocessor)*

2.4.1.1. Pengertian *PHP*

Sidik (2014:4) mengemukakan, "*PHP* merupakan secara umum dikenal sebagai bahasa pemrograman *script script* yang membuat dokumen *HTML* secara *on the fly* yang dieksekusi di *server web*, dokumen *HTML* yang dihasilkan dari suatu aplikasi bukan dokumen *HTML* yang dibuat menggunakan editor teks atau editor *HTML*. Dikenal juga sebagai bahasa pemrograman *server side*".

Sibero (2013:49) mengemukakan, "*PHP* adalah pemrograman *interpreter* yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan".

Nugroho (2013:153) mengemukakan, "*PHP* itu bahasa pemrograman berbasis *web*. Jadi, *PHP* itu adalah bahasa program yang digunakan untuk membuat aplikasi berbasis *web* (*website, blog*, atau aplikasi *web*)".

Jadi, *PHP* adalah bahasa pemrograman yang menerjemahkan baris kode sumber menjadi kode mesin sehingga dapat dibaca oleh komputer dan digunakan untuk membuat aplikasi berbasis *web*.

2.4.1.2. Tata Bahasa Dasar *PHP*

Sibero (2013:51) berpendapat sebagai berikut.

Tata bahasa penulisan baris kode *PHP* terdiri dari dua bentuk yaitu penulisan baris kode menggunakan format *PHP* maupun penulisan baris kode dengan format campuran antara *HTML* dan *PHP*. Berikut dibawah ini beberapa hal penting perlu diketahui sebelum memulai pembuatan program *PHP*:

1. Penulisan suatu program *PHP* harus diawali dengan *tag* **<?php** dan diakhiri dengan *tag* **?>**
2. Penulisan baris program *PHP* harus diakhiri tanda titik koma **(;)**
3. Penulisan pada *PHP* adalah *case-sensitive*, yaitu antara huruf **a** dan huruf **A** adalah berbeda.
4. Penulisan komentar atau baris yang tidak akan diproses oleh *PHP*
   a. Untuk komentar pada satu baris tertentu, tambahkan tanda **//** pada bagian awal baris program.
   b. Untuk komentar lebih dari satu baris, tambahkan tanda **/*** pada bagian awal baris dan tambahkan tanda ***/** pada bagian akhir baris program.
5. Penulisan suatu teks diapit dalam tanda **‘** atau **“**
6. Penulisan suatu tanda baca pada suatu teks seperti (‘, “, \, /, dll), gunakan tanda **** kemudian diikuti dengan tanda baca.

Contoh penulisan tata bahasa dasar *PHP* adalah sebagai berikut:

Tabel 2.6. Tata Bahasa Dasar *PHP*

```
<?php
        Echo “Katanya: \”PHP adalah bahasa yang mudah digunakan\””;
        //Hasil dari program diatas
        //Katanya: “PHP adalah bahasa yang mudah digunakan”
?>
```

(**Sumber**: Sibero, 2013:52)

2.4.2. Basis Data *(Database)*

Sujatmiko (2012:76) mengemukakan, “*database* (basis data) adalah representasi kumpulan fakta yang saling berhubungan disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundansi)* yang tidak perlu, untuk memenuhi berbagai kebutuhan”.

Menurut Sukamto dan Shalahuddin (2015:43) “basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat”.

Yakub (2012:51-52) mengemukakan, "basis data *(database)* merupakan kumpulan data yang saling berhubungan (punya relasi)".

Jadi, basis data (*database*) adalah tempat penyimpanan dari data yang saling berhubungan (berelasi).

2.4.3. *MySQL*

2.4.3.1. Pengertian *MySQl*

Sibero (2013:97) mengemukakan, "*MySQL* atau dibaca "*My Sekuel*" dengan (sic!) adalah suatu *RDBMS* (*Relational Database Management System*) yaitu aplikasi sistem yang menjalankan fungsi pengolahan data".

Hidayatullah dan Kawistara (2014:180) mengemukakan, "*MySQL* adalah salah satu aplikasi *DBMS* yang sudah sangat banyak digunakan oleh pemrogram aplikasi *web*".

Nugroho (2014:31) mengemukakan, "*MySQL* adalah *software* atau program aplikasi *database*, yaitu *software* yang dapat kita pakai untuk menyimpan data berupa informasi teks dan juga angka".

Jadi, *MySQL* adalah aplikasi *database* yang digunakan untuk proses pengolahan dan penyimpanan data.

2.4.3.2. Fungsi-Fungsi *MySQL*

Tabel 2.7. Fungsi-Fungsi *MySQL*

No.	Nama Fungsi	Kegunaan
1.	mysql_affected_rows()	Mengambil jumlah *record* hasil eksekusi
2.	mysql_close()	Menutup koneksi *database*
3.	mysql_connect()	Membuka koneksi *database*
4.	mysql_create_db()	Membuat *database* baru
5.	mysql_db_name()	Mengambil nama *database* hasil dari **mysql_list_dbs()**
6.	mysql_drop_db()	Menghapus suatu *database*
7.	mysql_errno()	Mengambil nomor *index* pesan kesalahan
8.	mysql_error()	Mengambil pesan kesalahan

Lanjutan **Tabel 2.7.** Fungsi-Fungsi *MySQL*

No.	Nama Fungsi	Kegunaan
9.	mysql_fetch_array()	Mengambil hasil *record* dan mendefinisikan dalam bentuk *array*
10.	mysql_fetch_assoc()	Mengambil hasil *record* seperti **mysql_fetch_array** dengan tipe **MYSQL_ASSOC**
11.	mysql_free_result()	Mengosongkan memori hasil proses
12.	mysql_insert_id()	Mengambil nomor ID terakhir setelah eksekusi. Fungsi ini berlaku untuk tabel yang memiliki kolom ID dengan tipe AUTO_INCREMENT
13.	mysql_list_dbs()	Mengambil daftar *database* aktif pada *server*
14.	mysql_num_rows()	Mengambil jumlah baris data hasil *query*
15.	mysql_pconnect()	Membuka koneksi persisten *database*
16.	mysql_ping()	Menguji respon *server* atau membuka kembali koneksi *database* yang terputus
17.	mysql_query()	Mengirimkan perintah *query* pada *server*
18.	mysql_real_escape_ string()	Mengubah karakter tanda baca menjadi bentuk yang aman pada *database*
19.	mysql_result()	Mengambil data hasil
20.	mysql_select_db()	Memilih dan mengaktifkan *database*
21.	mysql_unbuffered_ query()	Mengirimkan perintah pada *server* dan hasilnya tidak disimpan dalam *buffer memory*

(**Sumber**: Sibero, 2013:71-79)

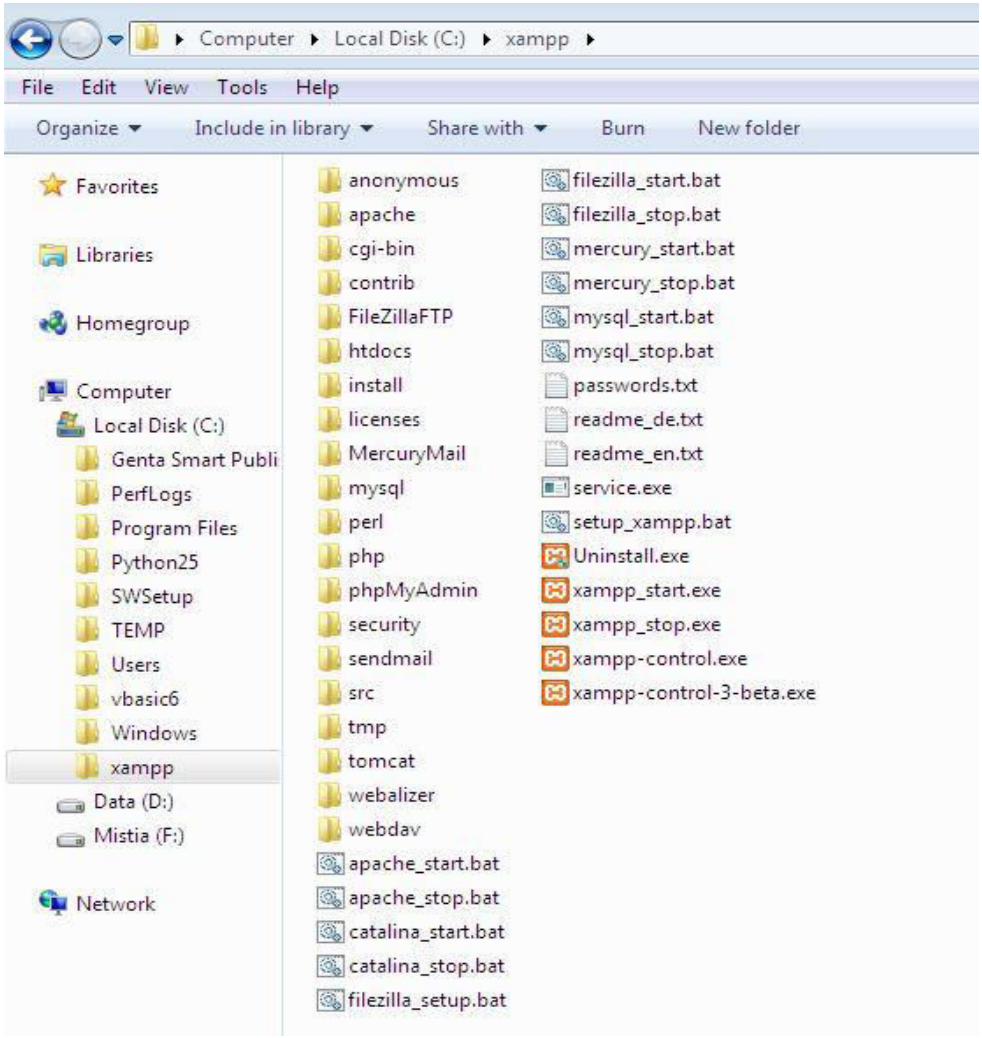
2.4.4. *XAMPP*

2.4.4.1. Pengertian *XAMPP*

Menurut Sidik (2014:72) "*XAMPP (X(Windows/Linux) Apache MySQL PHP* dan *Perl*) merupakan paket *server web PHP* dan *database MySQL* yang paling popular dikalangan pengembang *web* dengan menggunakan *PHP* dan *MySQL* sebagai *database*-nya".

2.4.4.2. Memahami Folder *XAMPP*

Nugroho (2013:6-9) menyatakan bahwa program *XAMPP* yang telah di-instalasi memiliki folder standar yang terletak pada **C:\xampp**. Susunan folder hasil instalasi dapat dilihat pada gambar di bawah ini.



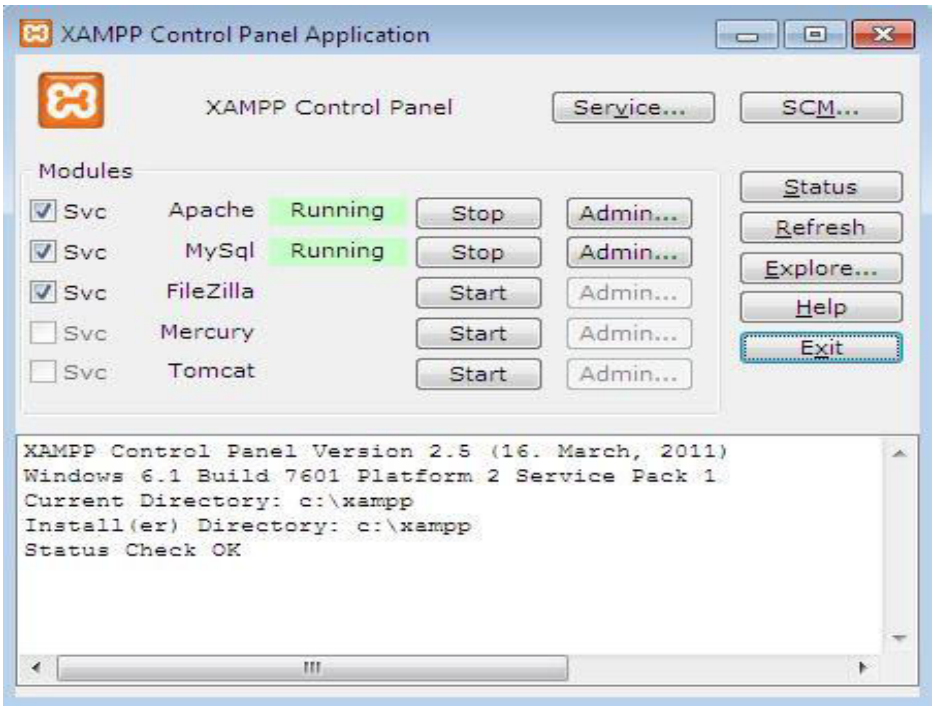
Gambar 2.1. Susunan *Folder XAMPP* di drive **C:**

Di bawah *folder* utama ***xampp***, terdapat beberapa *folder* penting. Berikut ini adalah penjelasan mengenai *folder-folder* penting tersebut.

1. *Apache*, merupakan *folder* utama dari *Apache Webserver*.
2. *Htdocs*, merupakan *folder* utama untuk menyimpan data-data latihan *web*, baik *PHP* maupun *HTML* biasa. Semua *folder* dan *file* program di ***htdocs*** bisa diakses dengan mengetikkan alamat http://localhost/ di *browser*.

3. *Mysql*, merupakan *folder* utama untuk *database MySQL server*.
4. *Php*, merupakan *folder* utama untuk program *PHP*.

Untuk mengakses *XAMPP* yang telah di-instalasi, dapat dilakukan dengan mengklik tombol ***Start -> All Programs -> Apache Friends -> XAMPP -> XAMPP Control Panel***. Adapun tampilan dari *XAMPP Control Panel* adalah sebagai berikut:



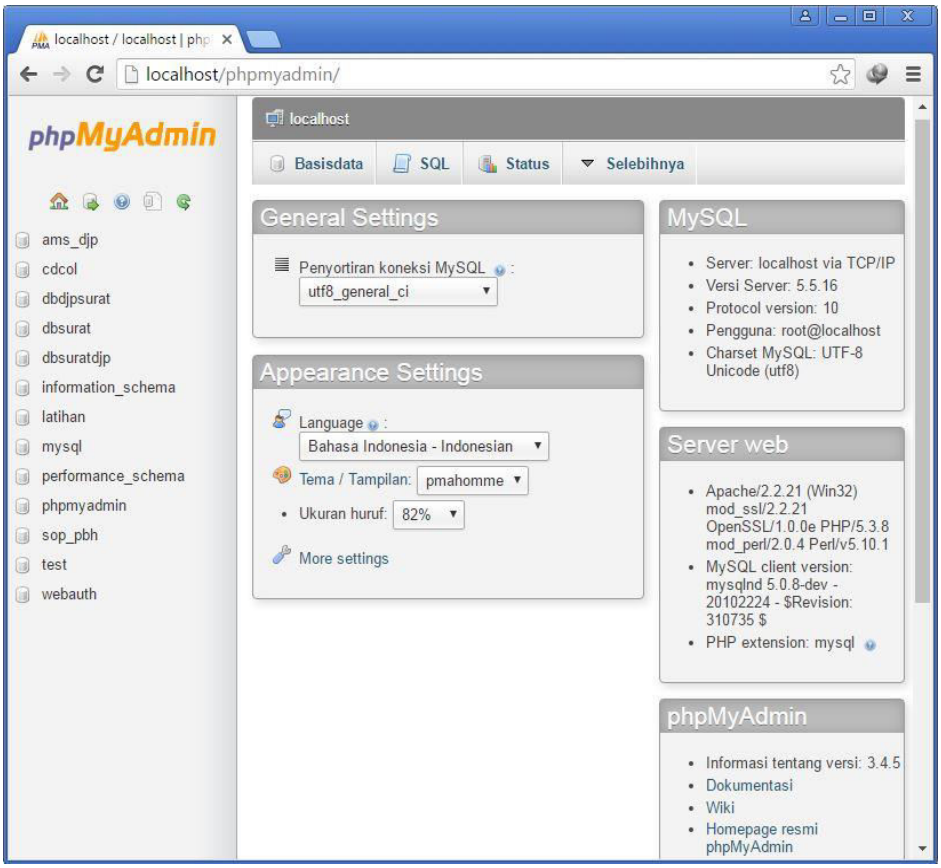
Gambar 2.2. Tampilan *XAMPP Control Panel*

2.4.5. *PhpMyAdmin*

Nugroho (2013:15) mengemukakan bahwa "*phpMyAdmin* adalah aplikasi manajemen *database server MySQL* berbasis *web*". Aplikasi *phpMyAdmin* memungkinkan *programmer* untuk mengelola *database* sebagai *root* (pemilik *server*) atau *user* biasa, membuat *database* baru, dan melakukan operasi perintah-perintah *database* secara lengkap. Berikut ini cara mengakses *phpMyAdmin* di *browser*.

1. Buka *browser*.
2. Pada kotak alamat, ketikkan http://localhost/phpmyadmin/ (tulis dalam huruf kecil semua).

3. Lalu tekan ***Enter.***



Gambar 2.3. Tampilan *phpMyAdmin*

2.4.6. *Macromedia Dreamweaver 8*

2.4.6.1. Pengertian *Macromedia Dreamweaver 8*

Saleh, dkk (2007:1) mengemukakan bahwa "*macromedia dreamweaver* merupakan *software web design* yang berguna untuk mendesain atau merancang *web* dan *layout* halaman *web*".

2.4.6.2. Ruang Kerja *Macromedia Dreamweaver 8*

Saleh, dkk (2007:3-13) menjelaskan tentang ruang kerja *Dreamweaver 8* adalah sebagai berikut.

1. Konsep Ruang Kerja *Dreamweaver 8*

   Saat pertama kali memulai program *Dreamweaver 8*, maka akan muncul tampilan pertama *Dreamweaver 8*, kategori atau jenis *web* yang diinginkan dapat langsung dipilih.

Gambar 2.4. Tampilan Awal *Macromedia Dreamweaver 8*

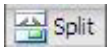
Dreamweaver 8 mempunyai tiga macam tampilan untuk *layout* ruang kerja, yaitu *Code View, Design View*, dan *Code and Design View*

. Adapun kegunaan dari tiga macam tampilan *layout* dapat dilihat pada tabel berikut ini.

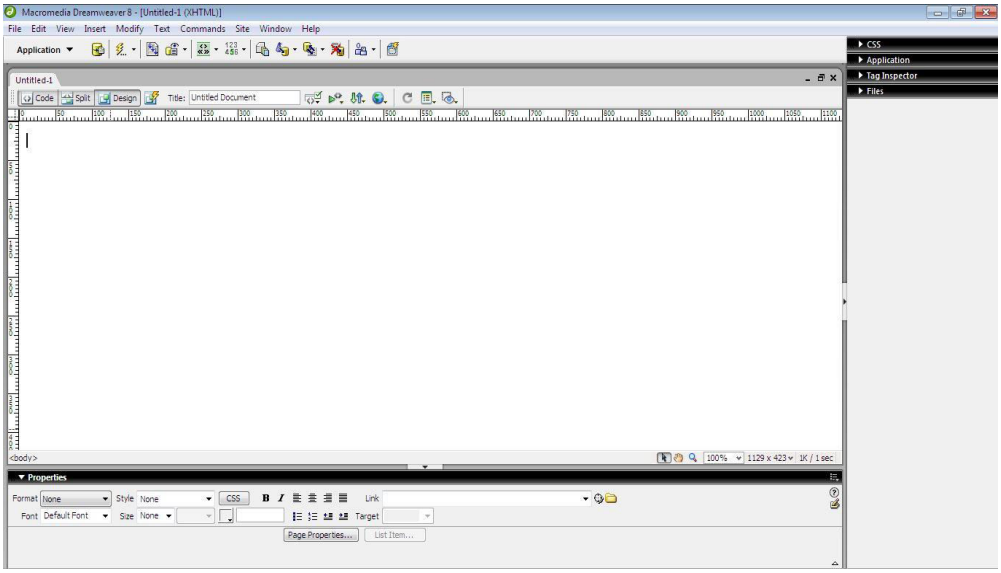
Tabel 2.8. Kegunaan Tampilan *Layout*

	Pada *Code View*, berguna untuk mengedit langsung pada halaman *web* dengan menuliskan kode *HTML*, *javascript*, *PHP*, *ASP*, dan berbagai macam kode-kode yang lain.
	Pada *Design View*, berguna untuk menampilkan dan mengedit dokumen secara *visual*. Tampilan dari *Design View* seperti anda menampilkannya dengan *web browser* seperti *Microsoft Internet Explorer.*

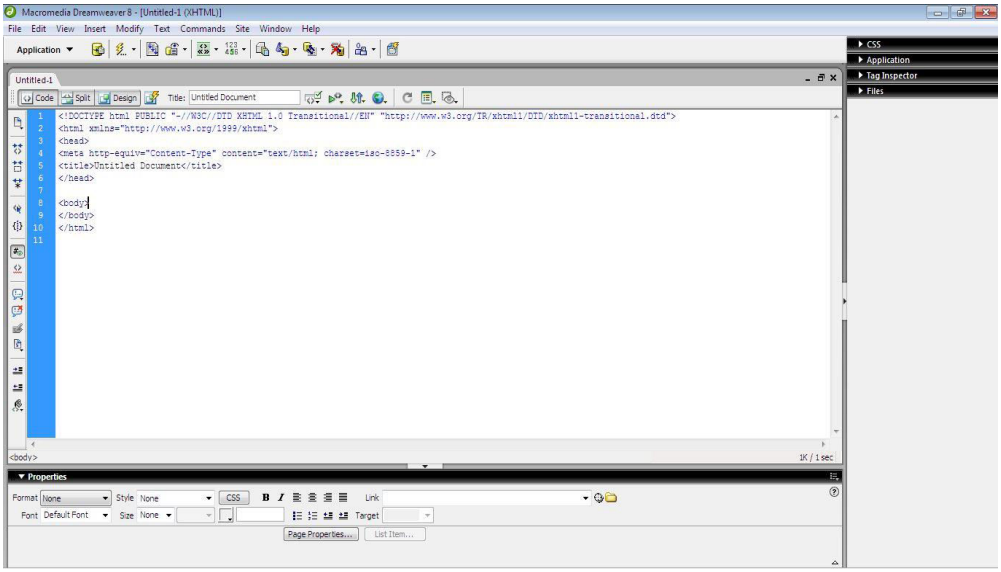
Lanjutan **Tabel 2.8.** Kegunaan Tampilan *Layout*

	Pada *Code and Design View*, untuk membuat halaman *web* dengan *Code View* dan *Design View* secara bersamaan.

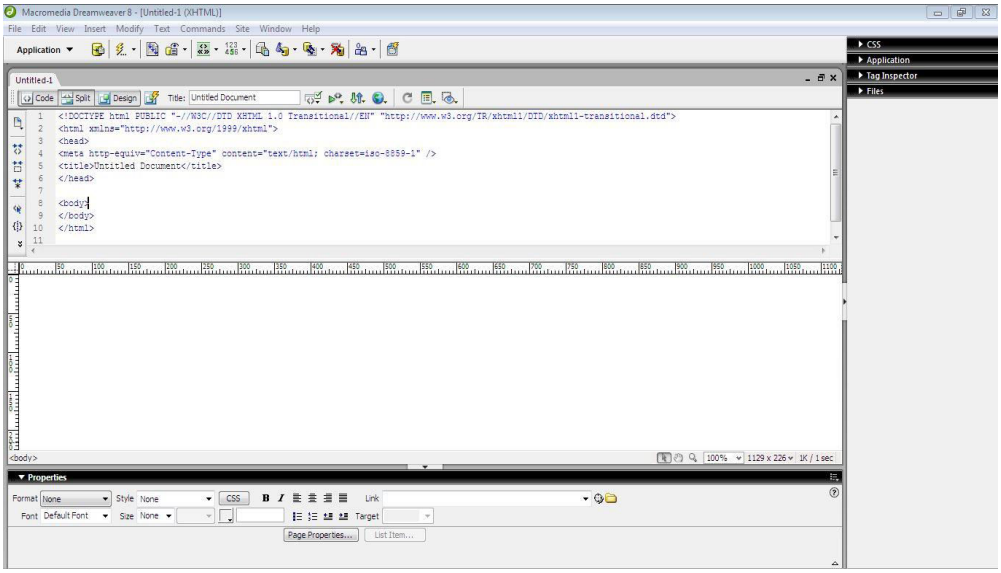
Tiga macam tampilan *layout dreamweaver 8* adalah sebagai berikut:



Gambar 2.5. Tampilan *Layout Design View*



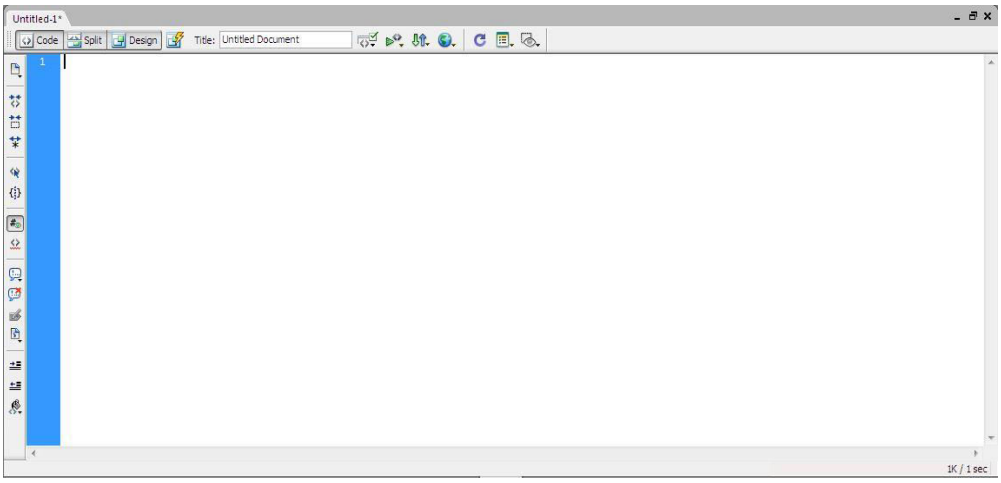
Gambar 2.6. Tampilan *Layout Code View*



Gambar 2.7. Tampilan *Layout Code and Design View*

2. *Document Window*

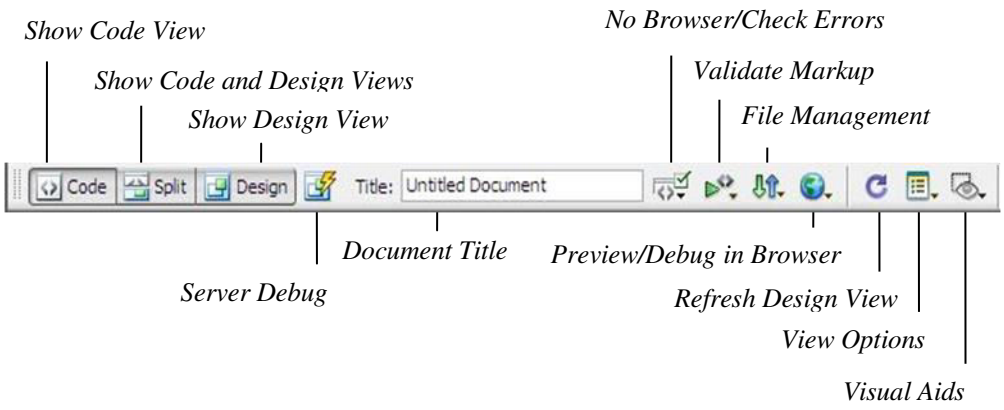
Document Window merupakan komponen dalam *dreamweaver* yang berguna untuk menyisipkan *image*, teks, animasi atau objek yang lain dan hal tersebut bisa dibuat melalui *document window*.



Gambar 2.8. Tampilan *Document Window*

3. *Document Toolbar*

*Document toolb*ar mengandung tombol-tombol yang menyediakan tampilan berbeda dari *document window*.



Gambar 2.9. Tampilan *Document Toolbar*

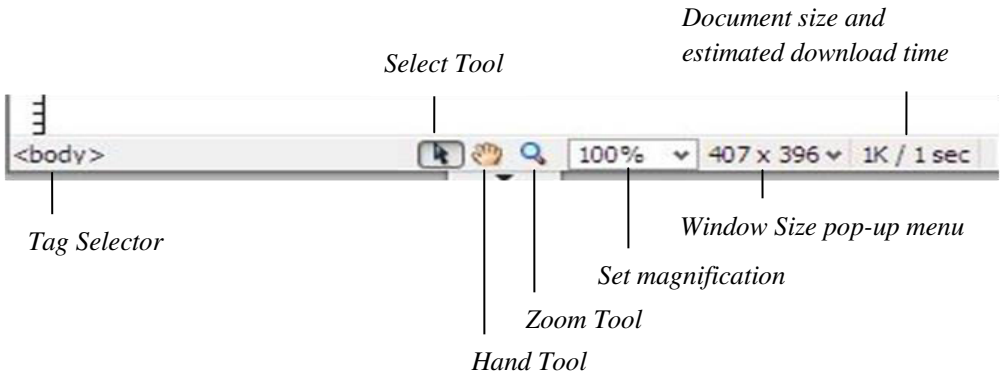
a. *Show Code View*, berguna untuk menampilkan kode-kode pada *document window*.
b. *Show Code and Design Views,* berguna untuk menampilkan kode-kode dan desain *visual* secara bersamaan pada *document window*.
c. *Show Design View*, berguna untuk menampilkan desain secara *visual* pada *document window*.
d. *Server Debug*, berguna untuk menampilkan laporan *debug* pada *Coldfusion* pada halaman. Laporan *debug* tersebut meliputi kesalahan yang ada pada halaman.
e. *Document Title*, berguna untuk memberikan nama atau judul pada dokumen yang dibuat yang nantinya akan ditampilkan pada *browser*.
f. *No Browser/Check Errors*, berguna untuk memeriksa kecocokan atau *kompatibel* pada *browser* yang digunakan.
g. *Validate Markup*, berguna untuk mengesahkan dokumen pada saat sekarang atau dokumen yang terpilih.
h. *File Management*, berguna untuk menampilkan manajemen *file* pada *menu*.
i. *Preview/Debug in Browser*, berguna untuk melihat atau menampilkan *debug* dokumen pada *browser* sebelum dokumen tersebut di publikasikan.
j. *Refresh Design View*, berguna untuk melihat ulang suatu dokumen setelah dilakukan perubahan (penyegaran dokumen).

k. *View Option*, pilihan *setting* untuk *layout design* yang sekarang ditampilkan (yang tampak pada *document window*) seperti *Head Content, Rulers, Grid, Guides*.

l. *Visual Aids*, memberikan pilihan menggunakan alat *visual* yang berbeda untuk mendesain halaman.

4. Status Bar

Komponen ini berfungsi memberikan informasi atau menampilkan hierarki *tag* pada dokumen yang sedang aktif, dimana *status bar* ini terletak di bawah *document window*.



Gambar 2.10. Tampilan *Status Bar*

a. *Tag Selector*, berfungsi menampilkan hirarki *tag-tag HTML* pada *document window*.

b. *Hand Tool*, berfungsi untuk mengklik dan menyeret dokumen pada *document window* dan biasanya bergambar tangan, untuk me-nonaktifkan *hand tool* tersebut dapat dilakukan dengan cara klik *Select tool*.

c. *Zoom Tool and Set Magnification*, berfungsi untuk mengeset perbesaran pada dokumen.

d. *Window Size Pop-Up Menu*, berfungsi untuk mengeset besar kecilnya ukuran dokumen.

e. *Document Size and Estimated Download Time*, berfungsi untuk menentukan ukuran *file* pada dokumen dan menentukan waktu untuk *mendownload*.

5. *Insert Bar*

   Insert Bar ini berisi tombol-tombol yang berfungsi untuk menyisipkan berbagai macam objek seperti tabel, *layer, image,* ke dalam dokumen. Komponen *insert bar* mempunyai beberapa kategori elemen yang dapat disisipkan. Adapun kategori-kategori dari *insert bar* tersebut yaitu:

Gambar 2.11. Tampilan *Insert Bar*

a. *Common*, berfungsi untuk menyisipkan atau memasukkan objek pada halaman *web*, seperti objek gambar dan tabel.
b. *Layout*, berfungsi untuk menyisipkan elemen *layer* atau tabel dalam dokumen, pdan juga memberikan fasilitas untuk berpindah dari tampilan standar ke tampilan *layout* atau sebaliknya.
c. *Forms,* berfungsi untuk memasukkan objek atau *elemen form.*
d. *Text*, berfungsi untuk menyisipkan teks pada dokumen sesuai dengan format yang diinginkan seperti *italic, bold, list, heading*, dan lain-lain.
e. *HTML*, berfungsi untuk menyisipkan kode-kode *HTML.*
f. *Application*, berfungsi untuk menyisipkan objek yang mempunyai unsure dinamis seperti *record* dan memperbarui *format.*
g. *Flash Elements*, berfungsi untuk menyisipkan objek atau elemen *Macromedia Flash.*
h. *Favorites*, berfungsi untuk menggolongkan dan mengorganisir tombol yang digunakan dalam satu tempat.

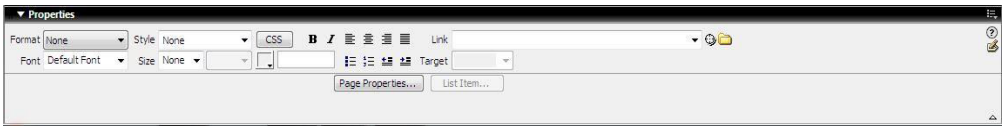
6. *Coding Toolbar*

   Coding toolbar ini berisi tombol-tombol yang memberikan fasilitas untuk dapat menggunakan banyak operasi pada *tag-tag HTML.*

Gambar 2.12. Tampilan *Coding Toolbar*

7. *Property Inspector*

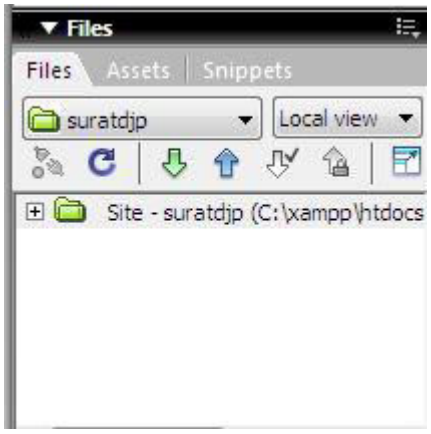
   Property Inspector merupakan komponen yang memberikan fasilitas untuk melihat atau mengedit berbagai *property* elemen yang dipilih seperti teks, tabel dan *layer*.



Gambar 2.13. Tampilan *Property Inspector*

8. *Files Panel*

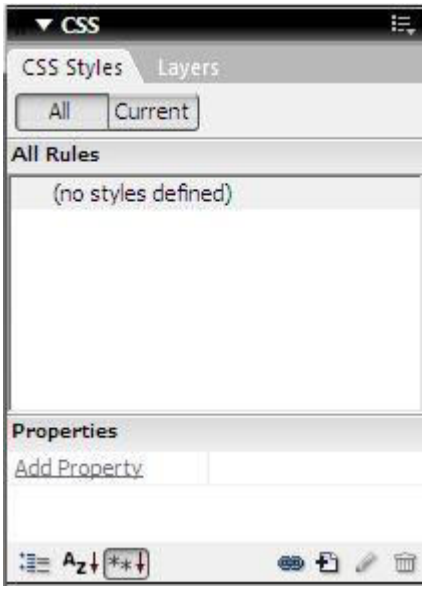
   Files Panel berfungsi untuk mengatur *file* dan *folder* yang dibuat dalam *dreamweaver*. *Files Panel* ini juga dapat melakukan koneksi ke *remote server*.



Gambar 2.14. Tampilan *Files Panel*

9. *CSS Styles Panel*

   CSS Styles Panel memberikan fasilitas untuk membuat, mengedit, melihat, menghapus, atau mengekspor *style sheet* dengan mudah. *Panel* ini memiliki dua modus yaitu *modus All* dan *Current*.



Gambar 2.15. Tampilan *CSS Styles Panel*