



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

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

Sutarman (2012:3), komputer adalah alat yang dapat membaca input data dan mengolahnya sesuai dengan program yang ditetapkan untuk menghasilkan informasi yang merupakan output hasil pemrograman input data.

Hamacher dalam Mulyono (2010:1), komputer adalah mesin penghitung elektronik yang cepat dan dapat menerima informasi input digital, kemudian memprosesnya sesuai dengan program yang tersimpan di memorinya, dan menghasilkan output berupa informasi.

Dari pengertian diatas, dapat disimpulkan bahwa komputer adalah sekumpulan alat elektronik yang mampu melakukan beberapa tugas, yaitu menerima data (*input*), memproses data sesuai dengan instruksi yang diberikan, dan menghasilkan output berupa informasi.

2.1.2. Pengertian *Software*

Sukanto dan Shalahuddin (2013:2), *software* adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*).

Sutarman (2012:14), *software* adalah kumpulan program-program komputer yang memungkinkan *hardware* memproses data.

Dari pengertian diatas, dapat disimpulkan bahwa *software* adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak yang memungkinkan *hardware* memproses data dan mengoperasikan suatu pekerjaan sesuai dengan yang dikehendaki.



2.1.3. Pengertian Program

Sutarman (2012:3), program adalah barisan perintah atau instruksi yang disusun sehingga dapat dipahami oleh komputer dan kemudian dijalankan sebagai barisan perhitungan numerik, dimana barisan perintah tersebut berhingga, berakhir, dan menghasilkan *output*.

Siallagan (2009:3), program dapat dianalogikan sebagai instruksi atau perintah-perintah untuk mengoperasikan atau menjalankan *hardware*.

Dari pengertian diatas, dapat disimpulkan bahwa program adalah kumpulan instruksi atau perintah-perintah yang disusun sehingga dapat dipahami oleh komputer untuk mengoperasikan atau menjalankan *hardware*.

2.1.4. Pengertian Internet

Asropudin (2013:48), *internet* adalah istilah umum yang dipakai untuk menunjukkan *network* tingkat dunia yang terdiri dari komputer dan layanan servis atau sekitar 30 sampai 50 juta pemakai komputer dan puluhan sistem informasi termasuk *e-mail*, *Gopher*, *FTP* dan *World Wide Web*.

Ahmadi dan Dadang (2013:68), *internet* merupakan komunikasi jaringan komunikasi global yang menghubungkan seluruh komputer di dunia meskipun berbeda sistem operasi dan mesin.

Dari pengertian diatas, dapat disimpulkan bahwa *internet* adalah jaringan komunikasi global yang menghubungkan jaringan komputer dan fasilitas komputer yang terorganisasi melalui media telekomunikasi seperti telepon atau satelit.

2.2. Teori Judul

2.2.1. Pengertian Aplikasi

Asropudin (2013:7), Aplikasi dibuat untuk mengerjakan atau menyelesaikan masalah-masalah khusus.

Menurut Sutabri (2012:147), Aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya.



2.2.2. Pengertian Panduan

Berdasarkan definisi dari Kamus Bahasa Indonesia (2008:451), “Pemandu adalah penunjuk jalan atau pengiring, perintis”.

2.2.3. Pengertian Pemandu Wisata

Berdasarkan definisi dari Kamus Besar Bahasa Indonesia online, Pemandu wisata adalah orang yang pekerjaannya mendampingi wisatawan dengan mengatur perjalanan dan memberi penjelasan tentang tempat yang dikunjungi.

2.2.4. Pengertian Web

Ahmadi dan Dadang (2013:68), *web* adalah halaman informasi yang disediakan melalui jalur *internet* sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan *internet*”.

Asropudin (2013:109), *web* adalah sebuah kumpulan halaman (*webpage*) yang diawali dengan halaman muka (*homepage*) yang berisikan informasi, iklan, serta program interaksi.

Dari pengertian diatas dapat disimpulkan bahwa *web* adalah kumpulan halaman informasi yang disediakan melalui jalur *internet* dan dapat diakses di seluruh dunia selama terkoneksi dengan jaringan *internet*.

2.2.5. Pengertian Aplikasi Panduan Wisata Kota Palembang berbasis Web Mobile

Merupakan suatu perangkat lunak yang dikembangkan untuk difungsikan secara khusus dalam mengelolah data pariwisata di kota Palembang yang digunakan sebagai panduan informasi bagi para wisatawan yang berkunjung ke kota Palembang.

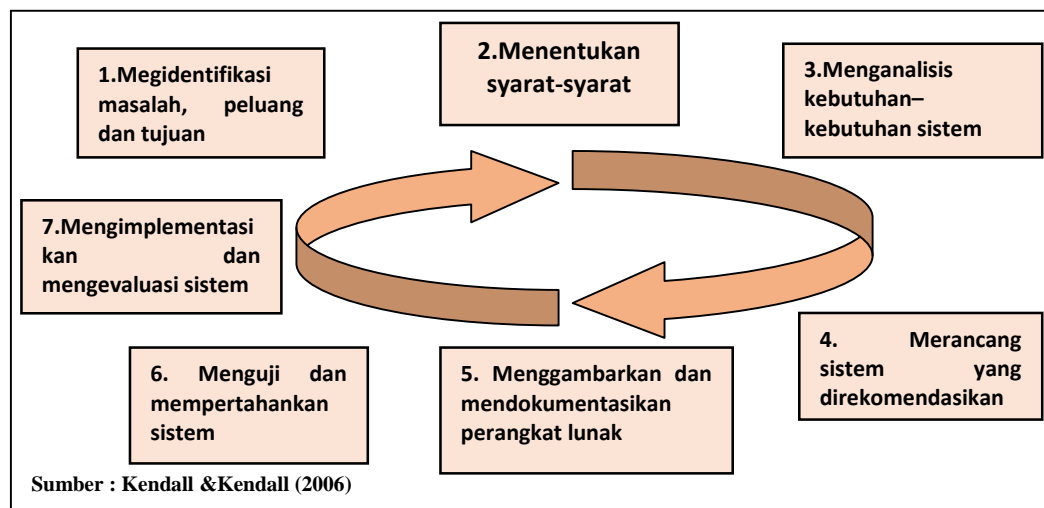
2.3. Teori Khusus

2.3.1. Metodologi SDLC (*System Development Life Cycle*)

System Development Life Cycle (SDLC) adalah pendekatan melalui beberapa tahap untuk menganalisis dan merancang sistem yang dimana sistem tersebut telah dikembangkan dengan sangat baik melalui penggunaan siklus kegiatan penganalisis dan pemakai secara spesifik.

2.3.2. Tahapan – Tahapan Metode SDLC (*System Development Life Cycle*)

Dalam Metode SDLC terdapat tujuh tahapan pengembangan seperti yang ditunjukkan pada gambar berikut :



Gambar 2.1. Tahapan Metode SDLC

Keterangan dari tahapan SDLC (*System Development Life Cycle*) adalah sebagai berikut :

1. Mengidentifikasi Masalah, Peluang dan Tujuan

Dalam tahapan pertama dari SDLC ini, penganalisis melakukan identifikasi masalah, peluang, dan tujuan yang ingin dicapai.

- Identifikasi masalah, dilakukan dengan melihat kenyataan yang terjadi dalam suatu perusahaan.
- Identifikasi peluang, dilakukan untuk mengetahui proses apa saja yang dapat diubah menjadi lebih baik dengan adanya sistem terkomputerisasi.
- Identifikasi tujuan, dilakukan untuk mengetahui proses apa saja yang dicapai perusahaan.



2. Menentukan Syarat-syarat informasi.

Tahap ini, penganalisis memasukkan apa saja yang menentukan syarat-syarat informasi untuk para pemakai yang terlibat. Diantara perangkat-perangkat yang dipergunakan untuk menetapkan syarat-syarat informasi di dalam bisnis diantaranya ialah menentukan sampel, dan memeriksa data mentah, wawancara mengamati perilaku pembuat keputusan dan lingkungan kantor dan prototyping.

3. Menganalisis Kebutuhan Sistem.

Tahap ini, perangkat dan teknik-teknik tertentu akan membantu penganalisis menentukan kebutuhan. Perangkat yang dimaksud adalah penggunaan diagram aliran data, untuk menyusun daftar input, proses, dan output fungsi bisnis dalam bentuk grafik terstruktur. Pada poin ini, penganalisis sistem menyiapkan suatu proposal sistem yang berisikan ringkasan apa saja yang ditemukan, analisis biaya atau keuntungan alternatif yang tersedia serta rekomendasi atas apa saja (bila ada) yang harus dilakukan.

4. Merancang Sistem yang Direkomendasikan.

Tahap ini, penganalisis merancang prosedur data entry sedemikian rupa, sehingga data yang dimasukkan ke dalam sistem informasi benar-benar akurat. Tahap perancangan juga mencakup perancangan file-file atau basis data yang bisa menyimpan data-data yang diperlakukan oleh pembuat keputusan. Dalam tahap ini penganalisis juga bekerja sama dengan pemakai untuk merancang output.

5. Mengembangkan dan Mendokumentasikan Perangkat Lunak.

Tahap ini, penganalisis bekerja sama dengan pemrogram untuk mengembangkan suatu perangkat lunak awal yang diperlukan.

6. Menguji dan Mempertahankan Sistem.

Tahap ini, sebelum sistem informasi dapat digunakan, maka harus dilakukan pengujian terlebih dahulu.

7. Mengimplementasikan dan Mengevaluasi Sistem.



Tahap terakhir ini, penganalisis membantu untuk mengimplementasikan sistem informasi. Tahap ini melibatkan pelatihan bagi pemakai untuk mengendalikan system.

2.3.3. Unified Modelling Language (UML)

Menurut Sukamto dan Shalahuddin (2013:133), *UML (Undefined Modelling Language)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

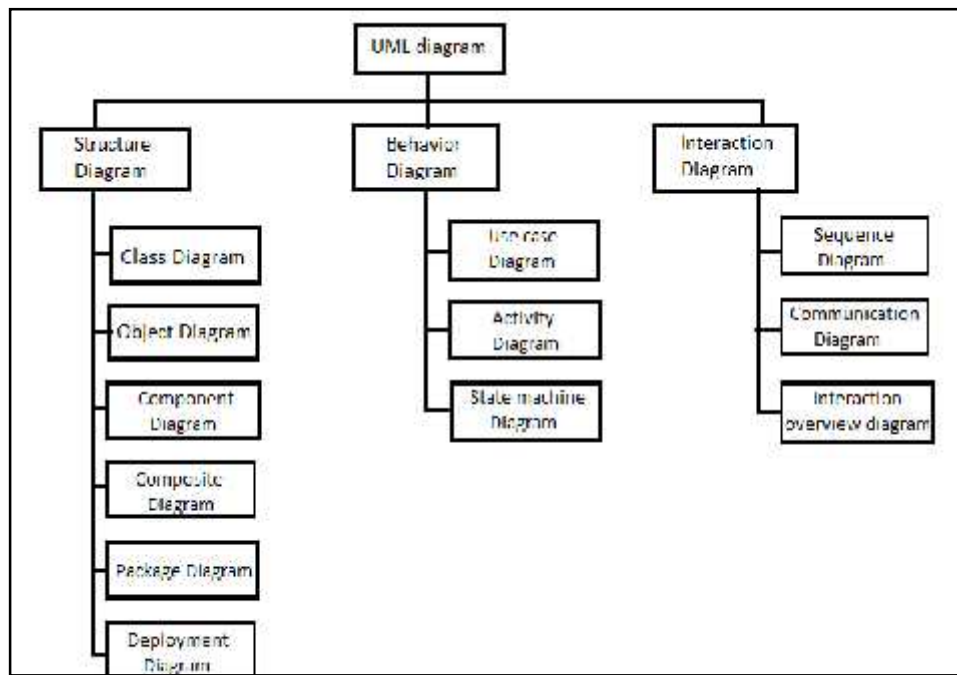
Menurut Prabowo dan Herlawati (2011:6), “*UML* singkatan dari *Unified Modeling Language* yang berarti bahasa permodelan standar.”

UML menyediakan bahasa pemodelan visual yaitu proses penggambaran informasi-informasi secara grafis dengan notasi-notasi baku yang telah disepakati sebelumnya. Dengan menggunakan pemodelan *UML* ini, pengembang dapat melakukan:

- a. tinjauan umum bagaimana arsitektur sistem secara keseluruhan.
- b. Penelaahan bagaimana objek-objek dalam sistem saling mengirimkan pesan (*message*) dan saling bekerjasama satu sama lain.
- c. Menguji apakah sistem/perangkat lunak sudah berfungsi seperti yang seharusnya.
- d. Dokumentasi sistem/perangkat lunak untuk keperluan-keperluan tertentu di masa yang akan datang.

2.3.3.1. Macam-macam Diagram UML

Pada *UML* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini:



Gambar 2.2. Macam-macam Diagram UML

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

a. *Structure Diagram*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

b. *Behavior Diagram*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

c. *Interaction Diagram*

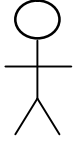
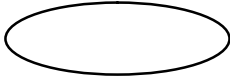
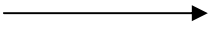
Yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

2.3.4. Use Case Diagram

Menurut Sukanto dan Shalahuddin (2013:155), *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Berikut adalah simbol-simbol yang ada pada diagram *use case*:

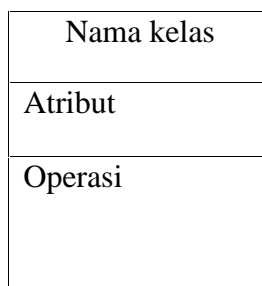
Tabel 2.1 Simbol-simbol Diagram Use Case

No.	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Merupakan seseorang atau sesuatu yang berinteraksi dengan sistem
2.		<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar antar unit atau aktor
3.		<i>Generalization</i>	Menggambarkan hubungan generalisasi dan spesialisasi (umum-khusus) antara interaksi dalam objek

2.3.5. ClassDiagram

Widodo dan Herlawati (2011:3) menjelaskan, *Class diagram* adalah penggambaran satu set objek yang memiliki atribut dan *behavior* yang sama.

Kelas, seperti juga objek, adalah sesuatu yang membungkus (*encapsulation*) informasi (baca:atribut) dan perilaku (baca:operasi) dalam dirinya. Berdasarkan pernyataan di atas, diketahui bahwasanya notasi kelas dalam *UML* adalah sebagai berikut:


Gambar 2.3. Notasi kelas dalam UML



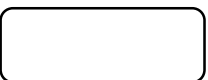
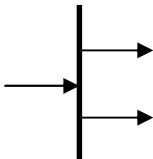
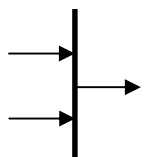
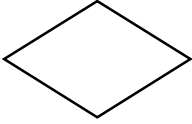


2.3.6. Activity Diagram

Menurut Nugroho (2005:61), *Activity diagram* adalah salah satu cara untuk memodelkan *event-event* yang terjadi dalam suatu *use case*.

Berikut simbol-simbol yang sering digunakan dalam *activity diagram*:

Tabel 2.2. Simbol-simbol Activity Diagram

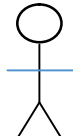
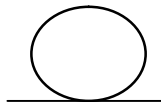
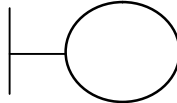

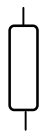

No.	Simbol	Nama	Keterangan
1.		<i>Start Point</i> (titik awal)	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2.		<i>End Point</i> (titik akhir)	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
3.		<i>Activities</i> (aktivitas)	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
4.		<i>Fork</i> (Percabangan)	Asosiasi percabangan dimana jika ada pilihan aktivitas yang lebih dari satu
5.		<i>Join</i> (penggabungan)	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
6.		<i>Decision</i> (keputusan)	Pilihan untuk mengambil keputusan



2.3.7. Sequence Diagram

Menurut Nugroho (2005:91), *Sequence diagram* adalah *interaction diagram* yang memperlihatkan *event-event* yang berurutan sepanjang berjalannya waktu.

Tabel 2.3. Simbol-simbol Sequence Diagram



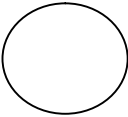

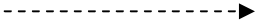
No.	Simbol	Nama	Keterangan
1.		<i>Actor</i> (aktor)	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2.		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan.
3.		<i>Boundary Class</i>	Menggambarkan sebuah penggambaran dari form.
4.		<i>Control Class</i>	Menggambarkan penghubung boundary dengan tabel.
5.		<i>A Focus of Control and A Life Line</i>	Tempat menggambarkan tempat mulai dan berakhirnya sebuah <i>message</i> .
6.		<i>Specification</i>	Spesifikasi komunikasi dari antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi.



2.3.8. Collaboration Diagram

Sequence diagram, collaboration diagram juga digunakan untuk memperlihatkan aliran-aliran pada *use case*. Sementara *sequence diagram* berurutan menurut waktu, *collaboration diagram* berfokus pada relasi-relasi yang terjadi antara objek yang satu dengan objek-objek yang lainnya.” Nugroho (2005:93). “ Diagram kolaborasi adalah diagram interaksi yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan (*message*).” Nugroho (2005:19).

Tabel 2.4. Simbol-simbol Collaboration Diagram

No.	Simbol	Keterangan
1		<i>Note</i>
2		<i>Class</i>
3		<i>Interface</i>
4		<i>Association</i>
5		<i>Dependency & Realize</i>



2.3.9. Kamus Data

Sukamto dan Shalahuddin (2013:73), “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 penulisan)”.

Sukamto dan Shalahuddin (2013:74), menjelaskan bahwa kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan, yaitu sebagai berikut:

Tabel 2.5. Simbol-simbol dalam kamus data

No.	Simbol	Keterangan
1.	=	Artinya adalah disusun atau terdiri atas
2.	+	Artinya adalah dan
3.	[]	Artinya adalah baik...atau...
4.	{ } ⁿ	Artinya adalah n kali diulang/ bernilai banyak
5.	()	Artinya data operasional
6.	*...*	Artinya adalah batas komentar

(Sumber : Sukamto dan Shalaluddin, 2013:73)

2.4. Teori Program

2.4.1. Basis Data (*Database*)

Priyadi (2014:2), “basis data adalah sekumpulan fakta berupa representasi tabel yang saling berhubungan dan disimpan dalam media penyimpanan secara digital”.

Sukamto dan Shalaludin (2013:43), “basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”.

Priyadi (2014:3) menjelaskan beberapa elemen basis data, di antaranya adalah sebagai berikut:



1. Tabel

Pada suatu basis data, tabel di representasikan menjadi suatu bentuk segiempat berupa matriks, yang terdiri dari kolom dan baris.

2. *Field*

Pada tabel basis data, kolom merupakan representasi untuk sebuah nama *field* yang pembacaan datanya dilakukan secara vertikal.

3. *Record*

Pada tabel basis data, baris merupakan suatu representasi untuk sebuah *record* yang pembacaan datanya dilakukan secara horizontal. Satu baris pada sebuah tabel merupakan data yang dimiliki oleh satu *record*. Nilai-nilai yang dimiliki oleh sebuah *record* merupakan gabungan dari semua *field* yang terdapat dalam tabel tersebut.

4. Kardinalitas

Kardinalitas merupakan batasan dari banyaknya hubungan, yang dapat dilakukan oleh himpunan entitas dalam melakukan relasi dengan himpunan entitas lainnya. Variasi kemungkinan untuk melakukan relasi yang dimiliki oleh kardinalitas terdiri dari empat macam, yaitu:

- a. Satu ke satu \longrightarrow (1:1).
- b. Satu ke banyak \longrightarrow (1:N).
- c. Banyak ke satu \longrightarrow (N:1).
- d. Banyak ke banyak \longrightarrow (N:N).

Database berfungsi untuk menyimpan informasi atau data. Untuk mengenal *database* diperlukan *software* yang sering disebut dengan *DBMS (Database Management System)*. Dengan *DBMS* pengguna dapat membuat, mengolah, mengontrol, dan mengakses *database* dengan mudah, praktis dan efisien.

2.4.2. Pengertian HTML

Asropudin (2013:44), *HTML* atau *Hyper Text Markup Language* adalah bahasa komputer yang digunakan untuk membuat sebuah halaman *webpage*.



2.4.3. Sekilas Tentang PHP

2.4.3.1. Pengertian Pemrograman PHP (*Hypertext Preprocessor*)

Master.com (2012:5), PHP merupakan singkatan dari PHP *Hypertext Preprocessor* yang digunakan sebagai bahasa *script serverside* dalam pengembangan Web yang disisipkan pada dokumen HTML.

2.4.3.2. Sintaks PHP

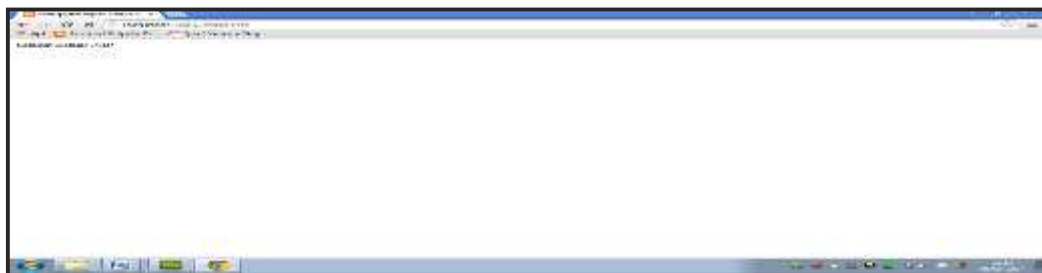
Badiyanto (2013:32), *PHP* yang merupakan sebuah bahasa *scripting* yang terpasang pada *HTML* dan skrip *HTML* adalah bahasa standar untuk membuat halaman *web*.

Sintaks *PHP* berkedudukan sebagai tag dalam bahasa *HTML*. Sebagaimana diketahui, *HTML* adalah bahasa standar untuk membuat halaman-halaman *web*.

Berikut adalah sintaks *PHP* yang ditulis didalam skrip *HTML*.

```
<html>
<head>
<title> Menggabungkan PHP dan HTML </title>
</head>
<body>
    <?php
    echo "Selamat Belajar PHP";
    ?>
</body>
</html>
```

Bila dijalankan melalui browser, skrip tersebut membentuk tampilan seperti berikut:



Gambar 2.4. Tampilan PHP Sederhana



2.4.4. Sekilas Tentang MySQL

2.4.4.1. Pengertian MySQL

Kadir (2013:15), MySQL adalah nama database server, dengan menggunakan MySQL kita bisa menyimpan data dan kemudian data bisa diakses dengan cara yang mudah dan cepat.

MySQL memiliki beberapa kelebihan, antara lain:

1. MySQL dapat digunakan oleh beberapa *user* dalam waktu bersamaan tanpa mengalami masalah.
2. MySQL memiliki kecepatan yang bagus dalam menangani *query* sederhana.
3. MySQL memiliki operator dan fungsi secara penuh dan mendukung perintah *select* dan *where* dalam perintah *Query*.
4. MySQL memiliki keamanan yang bagus karena beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perjanjian yang mendetail serta sandi terenkripsi.
5. MySQL mampu menangani basis data dalam skala besar dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta kurang lebih 5 milyar baris. Selain itu batas indeks yang dapat ditampung menvcapai 32 indeks pada setiap tabelnya.
6. MySQL dapat melakukan koneksi dengan *client* menggunakan *protocol TCP/IP*, *Unix Soket (UNIX)* atau *Named Pipes (NT)*.
7. MySQL dapat mendeteksi pesan kesalahan pada *client* dengan menggunakan lebih dari dua puluh bahasa.
8. MySQL dapat berjalan stabil dari berbagai sistem operasi seperti *Windows*, *Linux*, *Freebsd*, *Mac Os X Server*, *Solaris*, *Amiga*, dan masih banyak lagi.
9. MySQL didistribusikan secara *open source*, dibawah lisensi *GPL* sehingga dapat digunakan secara gratis

2.4.4.2. Fungsi-Fungsi MySQL

Kadir (2008:360), menjelaskan tentang sejumlah fungsi yang berawalan *mysql_* yang digunakan untuk mengakses *database server MySQL* sebagai berikut:

Tabel 2.6. Fungsi-Fungsi *MySQL*

No	Nama Fungsi	Kegunaan
1	<i>MySQL_connect()</i>	Membuka koneksi ke <i>server MySQL</i>
2	<i>MySQL_close()</i>	Menutup koneksi <i>MySQL</i>
3	<i>MySQL_select_db()</i>	Membuka/mengaktifkan suatu basis data
4	<i>MySQL_query()</i>	Mengeksekusi <i>string query</i>
5	<i>MySQL_db_query()</i>	Melakukan <i>query</i>
6	<i>MySQL_num_fields()</i>	Menghitung jumlah <i>field</i> dalam tabel tertentu
7	<i>MySQL_fetch_assoc()</i>	Mengambil hasil <i>query</i> sebelumnya
8	<i>MySQL_fetch_result()</i>	Membebaskan <i>resource</i> dari hasil <i>query</i>
9	<i>MySQL_fetch_row()</i>	Menampilkan hasil <i>query</i> ke dalam <i>array</i>
10	<i>MySQL_fetch_array()</i>	Menampilkan hasil <i>query</i> sebagai <i>array</i>
11	<i>MySQL_fetch_field()</i>	Menampilkan informasi <i>field</i> termasuk tipe datanya
12	<i>MySQL_affected_rows()</i>	Menampilkan jumlah baris yang dikenai operasi <i>SQL</i>
13	<i>MySQL_errno</i>	Menampilkan kode kesalahan (<i>error number</i>) dari perintah <i>SQL</i>
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 seluruh daftar <i>database MySQL</i>
17	<i>MySQL_list_tables()</i>	Menampilkan seluruh tabel dalam <i>database</i> tertentu

(Sumber : Kadir, 2008:360)

2.4.5. Mengenal tentang *Adobe Dreamweaver CS5*

2.4.5.1. Pengertian *Adobe Dreamweaver CS5*

Madcoms (2011:2), *Adobe Dreamweaver CS5* adalah sebuah *HTML editor profesional* untuk mendesain *web* secara visual dan mengelola situs atau halaman *web*. *Dreamweaver* merupakan software utama yang digunakan oleh *Web Designer* maupun *Web Programmer* dalam mengembangkan suatu situs *web*.

2.4.5.2. Area Kerja *Adobe Dreamweaver CS5*

Area kerja *Adobe Dreamweaver CS5* adalah lingkungan yang digunakan untuk merancang halaman *web*.



Gambar 2.5. Tampilan lembar kerja *Dreamweaver CS5*

Area kerja yang ada pada *Adobe Dreamweaver CS5* antara lain:

1. **Application Bar**, berada di bagian paling atas jendela aplikasi *Dreamweaver CS5*. Baris ini berisi tombol *workspace (workspace switcher)*, menu, dan aplikasi lainnya.



Gambar 2.6. Tampilan *Application Bar*

2. **Toolbar Document**, berisi tombol-tombol yang digunakan untuk mengubah tampilan jendela dokumen, sebagai contoh tampilan *Design* atau tampilan *Code*.



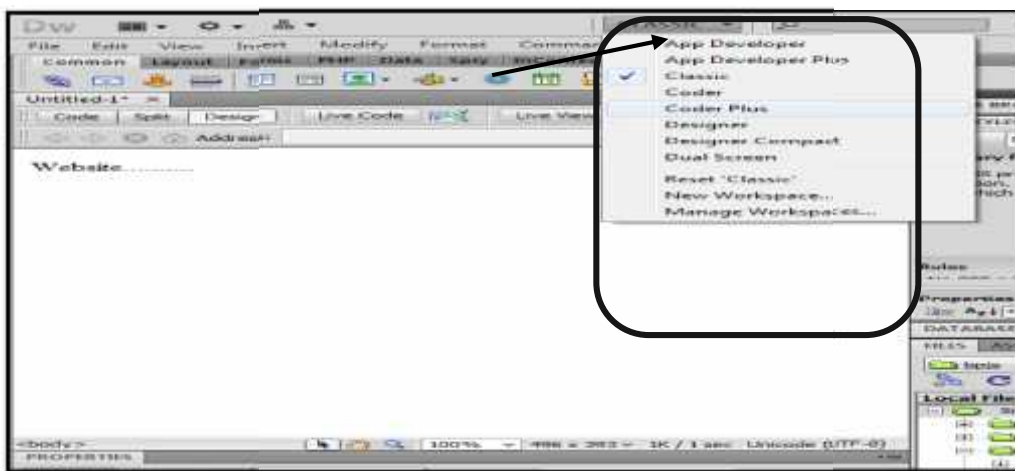
Gambar 2.7. Tampilan *Toolbar Document*

3. **Jendela Dokumen**, adalah lembar kerja tempat membuat dan mengedit *design* halaman *web*.



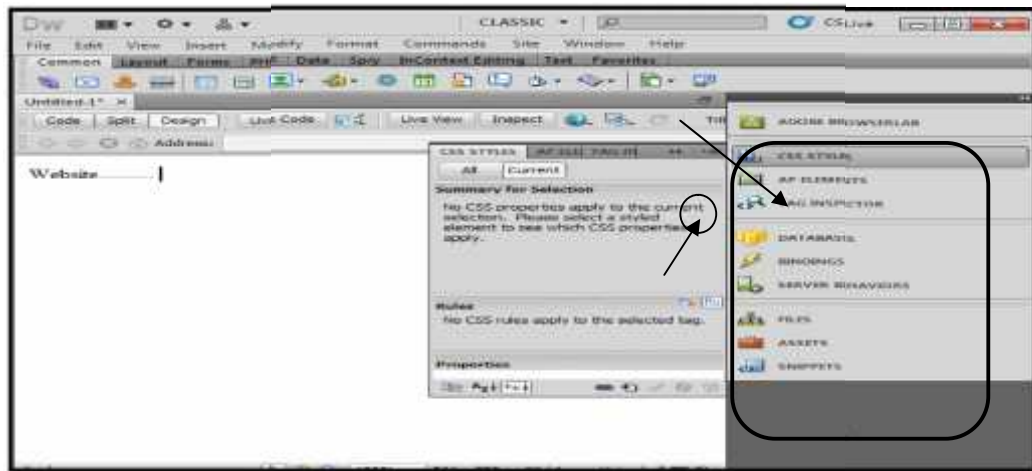
Gambar 2.5. Jendela Dokumen

4. **Workspace Switcher**, digunakan untuk mengubah tampilan ruang kerja (*workspace*) *Dreamweaver CS5*. Sebagai contoh, mengubah tampilan menjadi tampilan *classic*, yaitu tampilan ruang kerja *Dreamweaver* versi sebelumnya.



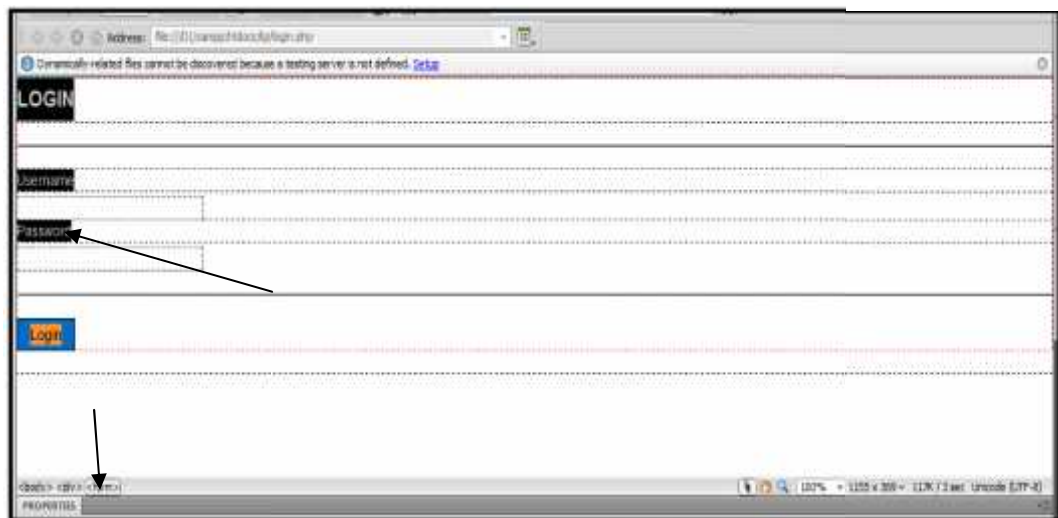
Gambar 2.6. Tombol perintah untuk memilih jenis tampilan ruang kerja

5. **Panel Groups**, adalah kumpulan panel yang saling berkaitan, panel-panel di kelompokkan pada judul-judul tertentu berdasarkan fungsinya. Panel ini digunakan untuk memonitor dan memodifikasi pekerjaan. Secara *default*, panel group berisi panel *Insert*, *CSS Style*, *Asset*, *AP Element* dan *Files*.



Gambar 2.7. Tampilan *panel groups*

6. **Tag Selector**, diletakkan di bagian bawah jendela dokumen, satu baris dengan status bar. *Tag selector* juga menampilkan informasi format dari bagian yang sedang aktif pada lembar kerja *design*.



Gambar 2.8. Contoh tampilan *tag selector*

7. **Panel Property**, digunakan untuk melihat dan mengubah berbagai properti objek atau teks pada jendela *design*. Properti-properti dalam jendela panel property sebagian ada yang tidak tampil. Untuk menampilkan atau menyembunyikan properti dari objek terpilih, dapat dilakukan dengan cara, *klik panah di pojok kanan bawah dari jendela panel property*.

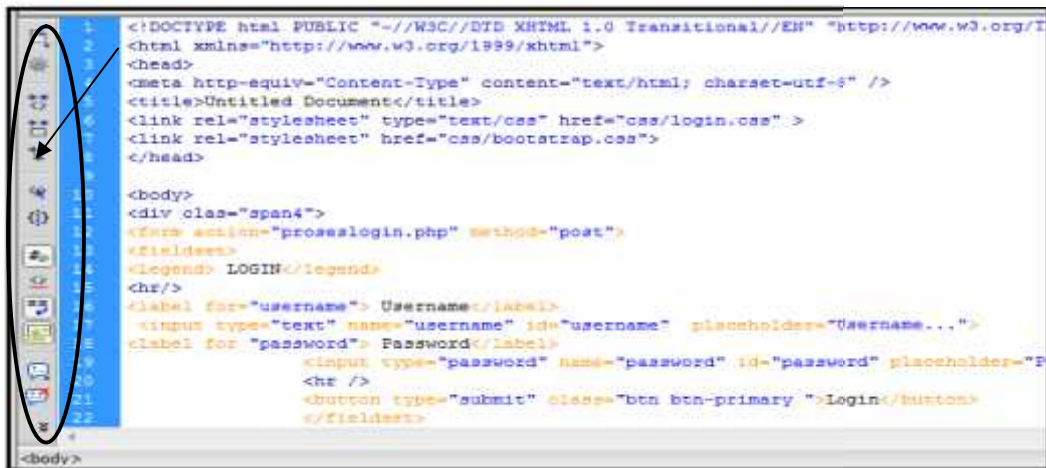


Gambar 2.9. Tombol panah pada *property*



Gambar 2.10. Contoh tampilan jendela *properties* untuk objek *form*

8. **Toolbar Coding**, berisi tombol-tombol yang digunakan untuk melakukan operasi kode-kode standar. *Toolbar* ini hanya tampil pada jendela *code*.



Gambar 2.11. Tampilan *toolbar Coding*

9. **Panel Insert**, berisi tombol-tombol untuk menyisipkan berbagai jenis objek, seperti *image*, tabel, atau objek media ke dalam jendela dokumen.



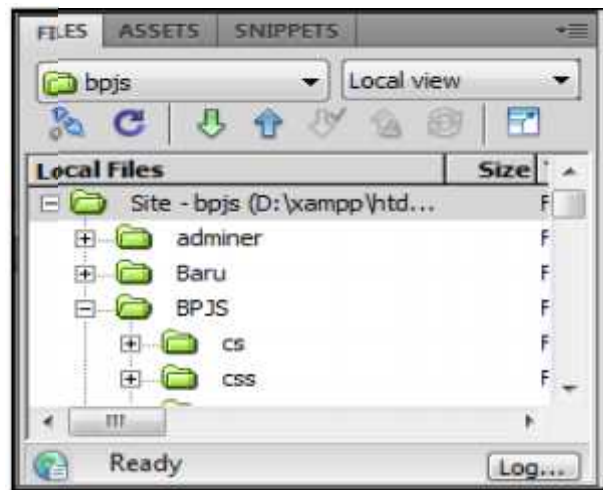
Gambar 2.12. Tampilan *panel Insert*

10. **Panel Groups**, adalah kumpulan panel yang saling berkaitan satu sama lainnya yang dikelompokkan di bawah satu judul, seperti *Insert*, *Files*, *CSS Styles* dan sebagainya.



Gambar 2.13. Group panel

11. Panel *Files*, digunakan untuk mengatur *file-file* dan *folder-folder* yang membentuk situs *Web*. Panel *File* ditampilkan dalam panel *grup*.



Gambar 2.14. Contoh tampilan panel *Files*

2.4.7. CSS

Badiyanto (2013:25), CSS adalah kepanjangan dari *Cascading Style Sheets*, CSS skrip yang berisi rangkaian intruksi yang memantau suatu teks akan tertampil dihalaman *web browser*.