

BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

Dalam bab ini akan diterapkan mengenai landasan teori yang bertujuan untuk membahas permasalahan yang diambil. Sesuai judul dalam pembuatan Laporan Akhir, yang akan dibuat penulis terdiri dari pengertian-pengertian sebagai berikut :

2.1.1. Pengertian Komputer

Pengertian Komputer menurut Hariyanto (2014:1), “Komputer adalah alat yang mampu melakukan komputasi dan membuat keputusan logis. Komputer melakukan dengan kecepatan cahaya, milyaran kali lebih cepat dibanding manusia. Komputer mengolah data dibawah kendali sekumpulan instruksi yang disebut program/aplikasi komputer. Program ini menuntun komputer melakukan serangkaian aksi-aksi yang dispesifikasikan”.

2.1.2. Pengertian Perangkat Lunak (*Software*)

Al Fatta (2007:10), “Perangkat Lunak yaitu program dan instruksi yang diberikan ke komputer”.

Dari pengertian tersebut, dapat disimpulkan bahwa “*Software* adalah kumpulan program komputer yang berisi perintah-perintah untuk melakukan pengolahan data”.

2.1.3. Pengertian Pengolahan Data

Menurut Kristanto (2011:8), “Pengolahan data adalah waktu yang digunakan untuk menggambarkan perubahan data menjadi informasi yang memiliki kegunaan”.



2.1.4. Pengertian Internet

Rafi'i (2008:283) menjelaskan, "Internet adalah jaringan komputer yang saling terhubung ke seluruh dunia tanpa mengenal batas teritorial, hukum, dan budaya".

Febrian (2006:2), "Internet merupakan tempat terhubungnya berbagai mesin komputer yang mengolah informasi di dunia ini, baik berupa server, komputer pribadi, *handphone*, komputer genggam, PDA, dan lain sebagainya".

2.1.5. Pengertian Program

Menurut Munir (2011:13), "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".

2.2. Teori Judul

Teori judul adalah teori yang menjelaskan judul aplikasi pembuatan Laporan Akhir.

2.2.1. Pengertian Aplikasi

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

2.2.2. Pengertian Pelayanan

Pengertian Pelayanan menurut pendapat Boediono (2003:60), "Pelayanan merupakan suatu proses bantuan kepada orang lain dengan cara-cara tertentu yang memerlukan kepekaan dan hubungan interpersonal agar terciptanya kepuasan dan keberhasilan".

(Sumber : <http://www.landasanteori.com/2015/10/pengertian-pelayanan-publik-dentuk.html>, diakses pada 25 April 2016 pukul 8:46 WIB)

2.2.3. Pengertian SKCK

Surat Keterangan Catatan Kepolisian (disingkat SKCK), sebelumnya dikenal sebagai Surat Keterangan Kelakuan Baik (disingkat SKKB) adalah surat keterangan yang diterbitkan oleh Polri yang berisikan catatan kejahatan seseorang. Dahulu, sewaktu bernama SKKB, surat ini hanya dapat diberikan yang tidak/belum pernah tercatat melakukan tindakan kejahatan hingga tanggal dikeluarkannya SKKB tersebut. Surat Keterangan Catatan Kepolisian atau SKCK adalah surat keterangan resmi yang diterbitkan oleh POLRI melalui fungsi Intelkam kepada seseorang pemohon/warga masyarakat untuk memenuhi permohonan dari yang bersangkutan atau suatu keperluan karena adanya ketentuan yang mempersyaratkan, berdasarkan hasil penelitian biodata dan catatan Kepolisian yang ada tentang orang tersebut.

(Sumber : <https://www.polri.go.id/layanan-skck.php>, diakses pada 25 April 2016 pukul 8:47 WIB)

2.2.4. Pengertian Web

Menurut Sibero (2013:11), Web adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia, dan lainnya pada jaringan internet”.

2.2.5. Pengertian Kepolisian Resort Kota Palembang

Polresta adalah struktur komando Kepolisian Republik Indonesia di daerah kabupaten/kota. Kepolisian Resort di wilayah perkotaan biasa disebut "Kepolisian Resort Kota" (Polresta) atau "Kepolisian Resort Kota Besar" (Polrestabes).

(Sumber : <http://infopalembang.id/polresta-palembang/>, diakses pada 25 April 2016 pukul 8:48 WIB)

2.2.6. Pengertian Aplikasi Pelayanan SKCK Berbasis Web Pada Kepolisian Resort Kota Palembang

Pengertian Aplikasi Pelayanan SKCK Berbasis Web Pada Kepolisian Resort Kota Palembang adalah suatu proses pelayanan pihak kepolisian dalam melayani pemohon yang akan membuat SKCK secara *online*.

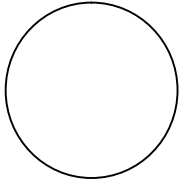
2.3. Teori Khusus

2.3.1. Pengertian *Data Flow Diagram* (DFD)

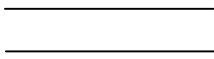
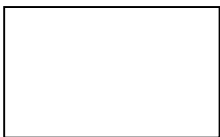
Menurut Kristanto (2011:55), “DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data yang disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut”.

Rosa A.S et.al (2014:70) menjelaskan, “*Data Flow Diagram* (DFD) atau dalam bahasa Indonesia Diagram Aliran Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*)”.


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

No	Simbol	Keterangan
1.		<p>Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program.</p> <p>Catatan :</p> <p>Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>

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 basisdata (<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 system lain yang terkait dengan aliran data dari 11 system yang dimodelkan.</p> <p>catatan : nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) berupa kata benda.</p>

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

No.	Simbol	Keterangan
4.		Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>). catatan : nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”

Sumber : Rosa A.S et.al (2014:71-72)

Rosa A.S et.al (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 system yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun system lain. DFD Level 0 digunakan untuk menggambarkan interaksi antara system yang akan dikembangkan dengan entitas luar.

2. Membuat DFD Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam system 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

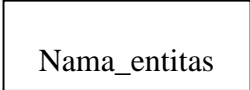
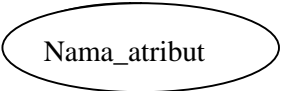
DFD Level3, 4, 5,dan seterusnya merupakan *breakdown* dari modul pada DFD Level di-atasnya. *Breakdown* pada level3, 4, 5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau 2.

2.3.2. Entity Relationship Diagram (ERD)

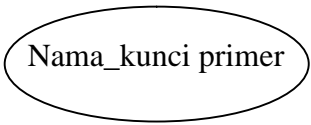
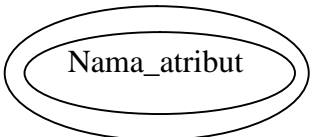
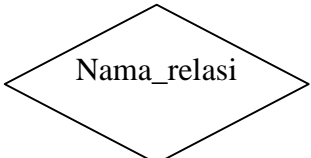
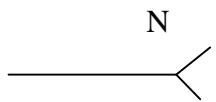
Rosa A.S et.al (2014:50) menjelaskan, “*Entity Relationship Diagram (ERD)* adalah pemodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basisdata relasional”.

Menurut Al Fatta (2007:121) menjelaskan, “*ERD (Entity Relationship Diagram)* adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam system 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 2.2 Simbol-Simbol *Entity Relationship Diagram (ERD)*

No.	Simbol	Keterangan
1.	Entitas/ <i>Entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal table pada basisdata; 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

Tabel 2.2 Simbol-Simbol *Entity Relationship Diagram* (ERD)

No.	Simbol	Keterangan
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>mutivalue</i> 	Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
4.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
5.	Asosiasi / <i>association</i> 	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B.



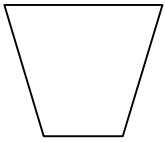

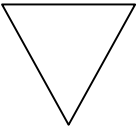
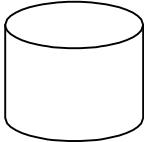

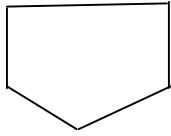
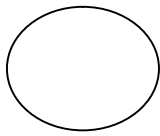
Sumber : Rosa A.S et.al (2014:50-51)

2.3.3. Block Chart

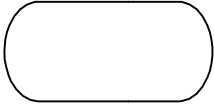
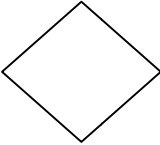
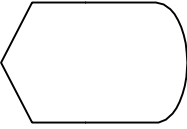
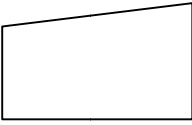
Menurut Kristanto (2011:68) menjelaskan, “*Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan

simbol-simbol tertentu. Pembuatan *Blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi”.

Tabel 2.3 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 terdefiniskan termasuk aktivitas fisik
8.		Terminasi yang mewakili symbol tertentu untuk digunakan pada aliran lain padahalamanyanglain.
9.		Terminal yang mewakili symbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.

Lanjutan **Tabel 2.3** Simbol-Simbol *Blockchart*

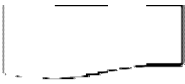

No.	Simbol	Keterangan
10.		Terminal yang menandakan awal dan akhir dari suatu aliran.
11.		Pengambilan keputusan (<i>Decision</i>).
12.		Layar peraga (monitor)
13.		Pemasukkan data secaramanual.

Sumber : Kristanto (2011:68-70)

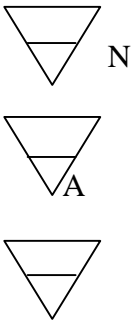
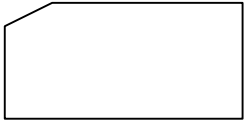
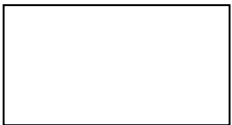

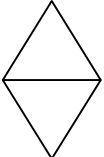
2.3.4. *Flow Chart*

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

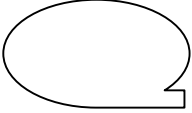
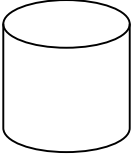
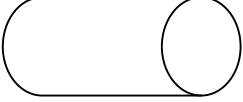




Tabel 2.4 Simbol-simbol dalam *FlowChart*

No.	Simbol	Keterangan
1.	Simbol Dokumen 	Menunjukkan dokumen <i>Input</i> dan <i>output</i> baik untuk proses manual, mekanik, atau komputer
2.	Simbol kegiatan manual 	Menunjukkan pekerjaan manual


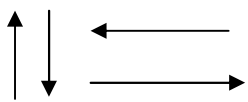
Lanjutan **Tabel 2.4** Simbol-simbol dalam *FlowChart*

No.	Simbol	Keterangan
3.	Simbol simpanan offline 	File non-komputer yang diarsipurut angka (numerical) File non-komputer yang diarsipurut huruf (alphabetical) File non-komputer yang diarsipurut tanggal (cronological)
4.	Simbol kartu plong 	Menunjukkan <i>input/output</i> yang menggunakan kartu plong (<i>punched card</i>).
5.	Simbol proses 	Menunjukkan kegiatan proses dari operasi program komputer
6.	Simbol operasi luar 	Menunjukkan operasi yang dilakukan diluar proses operasi komputer
7.	Simbol pengurutan offline 	Menunjukkan proses pengurutan data diluar proses komputer

Lanjutan **Tabel 2.4** Simbol-simbol dalam *FlowChart*

No.	Simbol	Keterangan
8.	Simbol pita magnetik 	Menunjukkan <i>input/output</i> menggunakan pita magnetik
9.	Simbol <i>hard disk</i> 	Menunjukkan <i>input/output</i> menggunakan <i>hard disk</i> .
10.	Simbol drum magnetik 	Menunjukkan <i>input/output</i> menggunakan drum magnetik
11.	Simbol pita kertas berlubang 	Menunjukkan <i>input/output</i> menggunakan pita kertas berlubang
12.	simbol keyboard 	Menunjukkan <i>input/output</i> menggunakan <i>online keyboard</i>
13.	Simbol <i>display</i> 	Menunjukkan <i>output</i> yang ditampilkan di monitor
14.	Simbol pita kontrol 	Menunjukkan penggunaan pita kontrol (<i>control tape</i>) dalam <i>batch control total</i> untuk pencocokan di proses <i>batch processing</i> .

Lanjutan **Tabel 2.4** Simbol-simbol dalam *FlowChart*

No.	Simbol	Keterangan
15.	Simbol hubungan komunikasi 	menunjukkan proses transmisi data melalui channel komunikasi
14.	Simbol garis alir 	Menunjukkan arus dari proses

Sumber: Supardi (2013:53-59)

3.3.5 Pengertian KamusData

Menurut Rosa A.S et.al (2014:73) menjelaskan, “Kamusdata adalah kumpulan daftar elemen data yang mengalir pada system perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan)”.

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 : Rosa A.S et.al (2014:74)

2.4. Teori Program

Teori program adalah teori yang menjelaskan bahasa pemrograman yang diaplikasikan penulis dalam pembuatan Laporan Akhir.

2.4.1. Pengertian *HTML*

Menurut Suryana & Koesharyatin (2014:29), “*HyperText Markup Language (HTML)* adalah bahasa yang digunakan untuk menulis halaman web. *HTML* merupakan pengembangan dari standar pemformatan dokumen teks, yaitu *Standard Generalized Markup Language (SGML)*. *HTML* pada dasarnya merupakan dokumen ASCII atau teks biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu”.

2.4.2. Pengertian *PHP*



Gambar 2.1. Logo *PHP*

Menurut Nugroho (2004:140), “*PHP* merupakan bahasa standar yang digunakan dalam dunia *web site*. *PHP* adalah bahasa program yang berbentuk *script* yang diletakan di dalam server web. *PHP* diciptakan dari ide Rasmus Lerdof yang membuat sebuah *script perl*”.

2.4.2.1. *Script PHP*

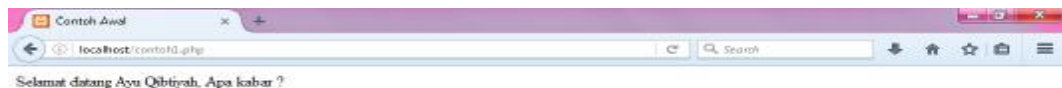
Menurut Nugroho (2004:151), “*PHP* adalah program yang fleksibel, artinya *script-script PHP* dapat anda tuliskan pada sela-sela tag *HTML*. Karena *PHP* memiliki sifat yang dapat berkonteraksi dengan program lain, maka kita tidak usah bingung dalam menuliskan skriptnya. Jadi, kita dapat membuat sebuah program *PHP* yang di dalamnya berupa tag-tag *HTML* murni yang kemudian disisipi *script PHP*. Akan tetapi, *PHP* juga dapat berdiri sendiri sehingga tidak memerlukan *HTML* murni untuk meletakkan *script-scriptnya*”.

Contoh script

```

<HTML>
<HEAD>
<TITLE>Contoh Awal</TITLE>
</HEAD>
<?php
Print (“Selamat datang Ayu Qibtiyah, Apa kabar ?”);
?>
<BODY>
</BODY>
</HTML>
  
```

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



Gambar 2.2. Contoh Script PHP

2.4.2.2. Teknik Penulisan PHP

Menurut Nugroho (2004: 144-147), “Ada beberapa cara menuliskan *script PHP*:

- a. `<?... ScriptPHP disini...?>` atau
- b. `<?php ... ScriptPHPdisini...?>` atau
- c. `<%... ScriptPHPdisini... %>` atau
- d. `<SCRIPT language=”php”>... ScriptPHP disini... </SCRIPT>`



Jadi, semua kata dan *script* yang diletakkan pada daerah *script* akan dianggap sebagai perintah *PHP* sehingga jika terjadi kesalahan atau kata-katanya tidak sesuai dengan program akan dianggap salah dan akan mengakibatkan program yang kita buat menjadi *error*.

Adapun kriteria yang harus diperhatikan dalam penulisan *script PHP* adalah sebagai berikut:

- a. Setiap halaman yang mengandung *script PHP* harus disimpan dengan EXTENSI *PHP* sesuai dengan program *PHP* yang mendukungnya (nama_file.php, atau nama_file.php3, nama_file.php4).
- b. Setiap *script PHP* harus didahului dengan pembuka *PHP* (<?php dan lain-lain) dan kemudian diakhiri dengan penutup (?>).
- c. Setiap baris *script* harus didahului pernyataan cetak atau dalam program pascal kita sebut *write* atau *REM*, sedangkan dalam *PHP* pernyataan cetak dibedakan menjadi dua yaitu *Print* dan *Echo*. Adapun kriteria penulisannya sebagai berikut:

Print (“*IsiPerintah*”)

Printf (“*IsiPerintah*”)

Echo “*IsiPerintah*”

- d. Setiap akhir baris perintah harus diakhiri dengan titik koma (;) misalnya kita akan menuliskan “Hello, Nama Saya Ayu Qibtiyah”, maka *script* yang harus anda tuliskan adalah sebagai berikut:

```
<?php
```

```
Echo “Hello, Nama Saya Ayu Qibtiyah”;
```

```
?>
```

- e. Semua variable harus diberi tanda string dolar (\$) pada penulisan awalnya. Sebagai contoh, kita akan menuliskan nama sebagai variabel yang isinya AyuQibtiyah, maka kita harus menuliskan sebagai berikut:

```
$nama = AyuQibtiyah;
```

- f. Penulisan *comment* atau keterangan didahului dengan pembuka /* dan diakhiri dengan */. Biasanya kita menggunakan ini untuk memberikan

comment yang berbentuk kalimat. Sebagai contoh, kita akan memberi keterangan penulisan nama diatas.

Contoh *Script*:

```
<?php
/* Untukmenuliskan komentar, biasanya komentar yang berbentuk
kalimat*/
Echo "Hello, Nama Saya Ayu Qibtiyah";
?>
```

- g. Selain menggunakan tanda */**, penulisan *comment* juga dapat menggunakan tanda *slash* ganda(*//*). Tanda ini hanya digunakan untuk menuliskan pesan yang hanya berisi satu baris saja.

Contoh*Script*:

```
<?php
//Untukmenuliskankomentarsatu baris
Echo "Hello, Nama Saya Ayu Qibtiyah";
?>
```

- h. Semua *script HTML* yang akan digabungkan dalam *script PHP* harus dihilangkan tanda petiknya (“ ”). Sebagai contoh, jika kita ingin memberi warna *background* pada halaman kita, maka perintah yang kita gunakan dalam *HTML* adalah `<BODY bgcolor="#000000">`. Akan tetapi, jika kita letakkan dalam *PHP*, maka *script* tersebut harus diubah menjadi `<BODY bgcolor=#000000>`.

2.4.3. Pengertian *PhpMyAdmin*

Menurut Nugroho (2004:65), “*PhpMyAdmin* adalah suatu program *open source* yang berbasis web yang dibuat menggunakan aplikasi *PHP*, program ini digunakan untuk mengakses *database MySQL*”.



2.4.4. Pengertian Basis Data (*Database*)

Menurut Nugroho (2004:69), “*Database / Basis Data* merupakan sekumpulan data yang sangat kompleks yang memiliki hubungan satu dengan yang lainnya”.

Menurut Rosa A.S dan Shalahuddin (2014:43), “*Basis Data* adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”.

2.4.5. Pengertian *MySQL*

Menurut Andi (2016:145), “*MySQL* adalah salah satu jenis *database* server yang sangat terkenal. *MySQL* menggunakan bahasa *SQL* untuk mengakses *database*-nya. Lisensi *MySQL* adalah *FOSS License Exception* dan ada juga yang versi komersial”.

Menurut Nugroho (2004: 29), “*MySQL (My Structure Query Language)* atau yang biasa dibaca “mai-se-kuel” adalah sebuah program pembuat *database* yang bersifat *open source*, artinya siapa saja boleh menggunakannya dan tidak dicekal”.

2.4.6. Fungsi-fungsi *MySQL*

Menurut Nugroho (2004:163-166), ”Untuk dapat menggunakan semua perintah *SQL* pada *PHP*, kita memerlukan suatu *MySQL*. Fungsi ini berguna untuk mengantarkan perintah *SQL* pada *PHP* menuju ke server sehingga perintah tersebut dapat di eksekusikan oleh server *MySQL*”.

Tabel 2.6. Fungsi-fungsi *MySQL*

No.	Fungsi <i>MySQL</i>	Penggunaan
1	<i>Mysql_change_user()</i>	<i>Change logged in user of the active connection</i>
2	<i>Mysql_close()</i>	Digunakan untk menutup koneksi dengan <i>MySQL</i>
3	<i>Mysql_affected_rows()</i>	<i>Get number of affected rows in previous MySQL operation</i>



Lanjutan Tabel 2.6. Fungsi-fungsi MySQL

No.	Simbol	Keterangan
4	<i>Mysql_connect()</i>	Untuk membuka koneksi dengan <i>database MySQL server</i>
5	<i>Mysql_create_db()</i>	Untuk membuat <i>database</i>
6	<i>Mysql_data_seek()</i>	-
7	<i>Mysql_db_name()</i>	Untuk menghasilkan hasil data
8	<i>Mysql_db_query()</i>	Untuk mengantarkan <i>Query MySQL</i>
9	<i>Mysql_drop_db()</i>	Untuk menghapus <i>database</i>
10	<i>Mysql_errno()</i>	Untuk menampilkan pesan kesalahan dalam bentuk no dari <i>server MySQL</i>
11	<i>Mysql_error()</i>	Untuk menampilkan pesan kesalahan dalam bentuk teks dari <i>server MySQL</i>
12	<i>Mysql_fetch_array()</i>	Menghasilkan data berupa array dalam bentuk angka dari isi tabel <i>MySQL</i>
13	<i>Mysql_fetch_assoc()</i>	Menghasilkan data berupa array dalam bentuk field dari tabel <i>MySQL</i>
14	<i>Mysql_fetch_field()</i>	Menghasilkan informasi kolom dari hasil yang kemudian dikembalikan sebagai suatu objek
15	<i>Mysql_fetch_lengths()</i>	Menghasilkan besar file dari hasil tabel <i>MySQL</i>
16	<i>Mysql_fetch_object()</i>	Menghasilkan sebuah baris sebagai suatu objek
17	<i>Mysql_fetch_row()</i>	Menghasilkan array/baris dengan keluaran nama field pada tabel <i>MySQL</i>
18	<i>Mysql_field_name()</i>	Menghasilkan nama field khusus pada <i>database MySQL</i>
19	<i>Mysql_field_len()</i>	<i>Returns the length of the specified field</i>
20	<i>Mysql_field_seek()</i>	<i>Set result pointer to a specified offset</i>
21	<i>Mysql_field_table()</i>	<i>Get name of the table the specified fields is in</i>
22	<i>Mysql_field_type()</i>	<i>Get the type of the specified field in a result</i>
23	<i>Mysql_free_result()</i>	<i>Free result memory</i>



Lanjutan Tabel 2.6. Fungsi-fungsi MySQL

No.	Simbol	Keterangan
24	<i>Mysql_insert_id()</i>	<i>Get the id generated from the previous INSERT operation</i>
25	<i>Mysql_list_dbs()</i>	<i>List database available on a MySQL server</i>
26	<i>Mysql_list_fields()</i>	<i>List MySQL result fields</i>
27	<i>Mysql_list_tables()</i>	<i>List tables in a MySQL database</i>
28	<i>Mysql_num_fields()</i>	<i>Get number of fields in result</i>
29	<i>Mysql_num_rows()</i>	<i>Get number of rows in result</i>
30	<i>Mysql_pconnect()</i>	Membuka koneksi langsung dengan MySQL
31	<i>Mysql_query()</i>	Digunakan untuk mengirimkan perintah SQL
32	<i>Mysql_result()</i>	Untuk menghasilkan data
33	<i>Mysql_select_db()</i>	Digunakan untuk masuk pada database MySQL
34	<i>Mysql_tablename()</i>	<i>Get table name of field</i>

Sumber : Bunafit Nugroho (164-166)

2.4.7. Pengertian XAMPP

Menurut Wahana (2009:30), “XAMPP adalah salah satu paket instalasi Apache, PHP dan MySQL secara instan yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut sama seperti *PHPTriad*”.

2.4.8. Pengertian Dreamweaver cs5

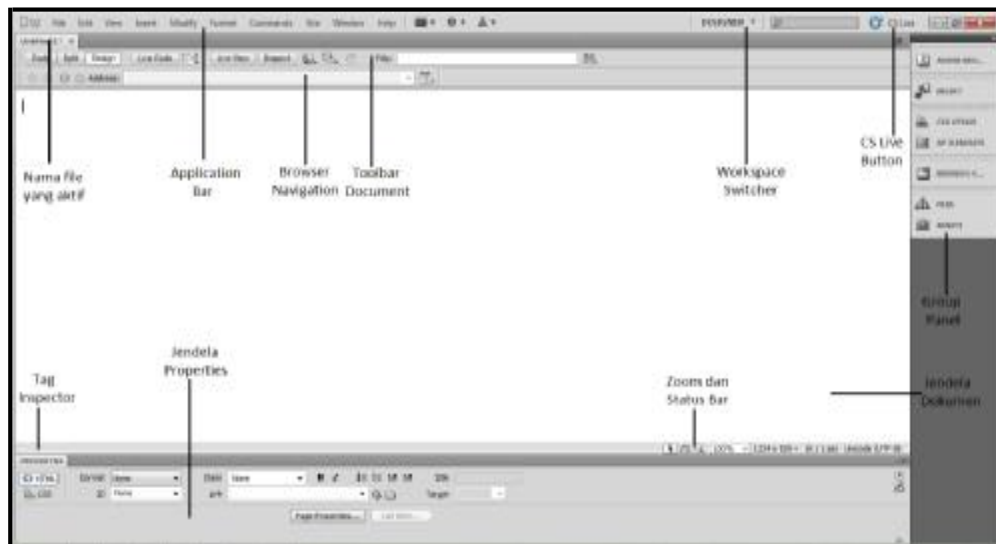
Wahana (2011:2), “*dreamweaver* adalah salah satu aplikasi yang dapat dimanfaatkan untuk merancang desainweb secara visual atau aplikasi editor web”.

Pada *dreamweaverCS5*, 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 2.3. Logo *Dreamweaver CS5*

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



Gambar 2.4. Tampilan Ruang Kerja *Adobe Dreamweaver CS5*

1. *Application Bar*

Berada dibagian paling atas jendela aplikasi *Dreamweaver CS5*. Baris ini berisi tombol *workspace (workspaces witcher)*, *CSLive*, menudan aplikasi lainnya.



Gambar 2.5. 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 2.6. Tampilan *Toolbar Document*

3. *Jendela Dokumen*

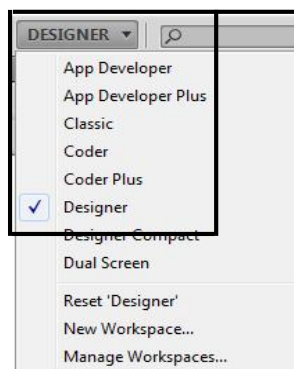
Adalah lembar kerja tempat membuat dan mengedit desain halaman *web*.



Gambar 2.7. *Jendela Dokumen*

4. *Workspace Switcher*

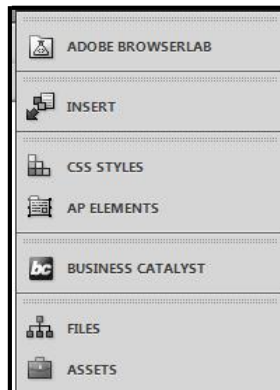
Digunakan untuk mengubah tampilan ruang kerja (*workspace*) *Adobe Dreamweaver CS5*.



Gambar 2.8. 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 2.9. Tampilan *Panel Groups*

6. Tag Selector

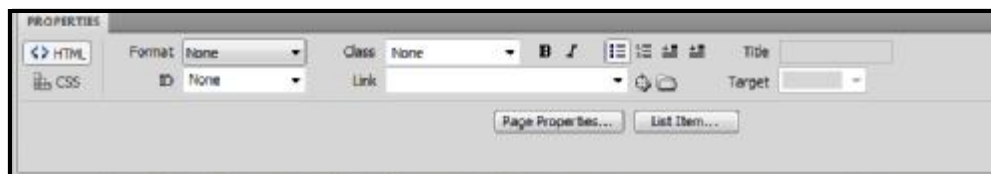
Bagian ini menampilkan hirarki pekerjaan yang sedang terpilih pada jendela dokumen, dapat juga digunakan untuk memilih objek pada jendela desain berdasarkan jenis atau kategori objek tersebut.



Gambar 2.10. *Tag Selector*

7. Panel Property

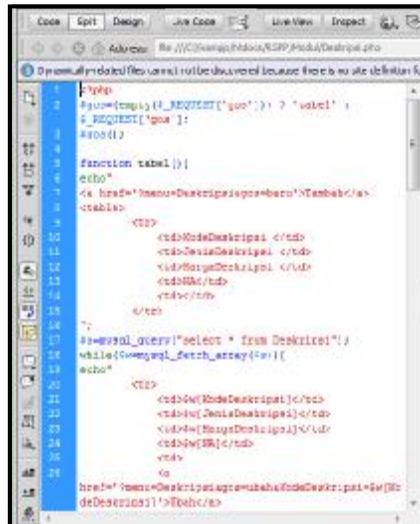
Digunakan untuk melihat dan mengubah berbagai *property* objek atau teks pada jendela *design*.



Gambar 2.11. Tampilan Jendela *Properties*

8. Toolbar Coding

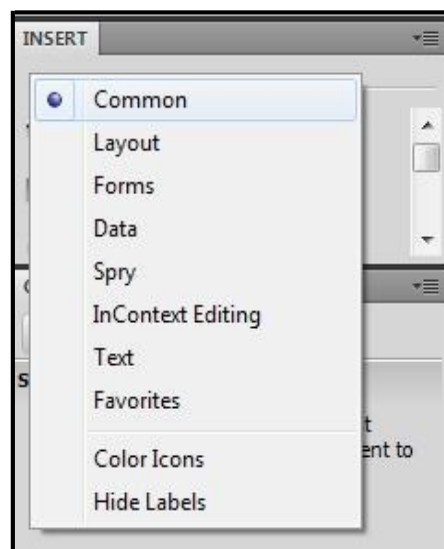
Berisi tombol-tombol yang digunakan untuk melakukan operasi kode-kode standar. Jendela ini hanya akan tampil pada jendela *Code* dan *Split*.



Gambar 2.12. Tampilan *Toolbar Coding*

9. Panel Insert

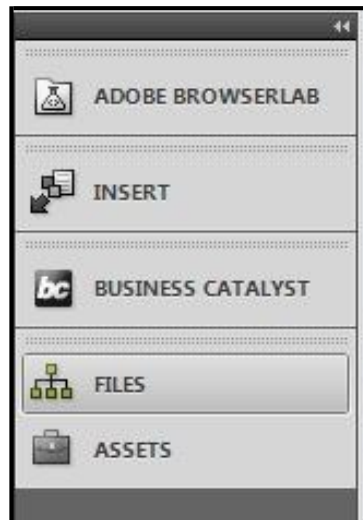
Berisi tombol-tombol untuk menyisipkan berbagai jenis objek, seperti *image*, tabel, atau objek media kedalam jendela dokumen.



Gambar 2.13. Tampilan *Panel Insert*

10. Panel Groups

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



Gambar 2.14. Tampilan *Panel Groups*

11. Panel Files

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



Gambar 2.15. Tampilan *Panel File*



2.4.9. Pengertian CSS (*Cascading Style Sheets*)

Menurut Sibero (2014:112), “*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”.

Menurut Suryana & Koesharyatin (2014:101), “*Cascading Style Sheet* (*CSS*) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan website, baik tata letaknya, jenis huruf, warna, dan semua yang berhubungan dengan tampilan. Pada umumnya *CSS* digunakan untuk memformat halaman web yang ditulis dengan *HTML* atau *XHTML*”.

2.4.10. Pengertian JavaScript

Menurut Suryana & Koesharyatin (2014:181), “JavaScript adalah bahasa script berdasar pada objek yang memperbolehkan pemakai untuk mengendalikan banyak aspek interaksi antara pemakai pada suatu dokumen *HTML*”.