



BAB II TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Aplikasi

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

Hendrayudi (2009:143) aplikasi ini adalah kumpulan data perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu.

2.1.2. Pengertian Data

Subtabri (2012:1) data merupakan bentuk jamak dari bentuk tunggal datum. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.

Yakub (2014:5) data adalah kenyataan yang menggambarkan adanya suatu kejadian (*event*), data terdiri dari fakta (*fact*) dan angka yang secara relatif tidak berarti bagi pemakai.

2.1.3. Pengertian Pengolahan Data

Sutabri (2012:6) menjelaskan, pengolahan data terdiri dari kegiatan-kegiatan penyimpanan data dan penanganan data. Penyimpanan data meliputi pengumpulan, pencarian, dan pemeliharaan. Sedangkan penanganan data meliputi berbagai kegiatan seperti pemeriksaan, perbandingan, pemilihan, peringkasan, dan penggunaan.

Kristanto (2012:6) menjelaskan bahwa pengolahan data adalah waktu yang digunakan untuk menggambarkan perubahan mentuk menjadi informasi yang memiliki kegunaan.

2.1.4. Pengertian Perpustakaan

Saputra dkk (2012:11) menyatakan, perpustakaan merupakan suatu tempat yang dibiayai dari dana umum, baik sebagian maupun seluruhnya dan terbuka



untuk masyarakat umum, serta memberikan layanan secara gratis kepada masyarakat umum.

Darmono Dalam Munawarah dan Euis (2011) menjelaskan, perpustakaan adalah salah satu bentuk organisasi sumber belajar yang menghimpun organisasi dalam bentuk buku dan bukan buku yang dapat dimanfaatkan oleh pemakai (guru, siswa, dan masyarakat) dalam upaya mengembangkan kemampuan dan kecakapannya.

2.1.5. Pengertian Aplikasi pengolahan data Perpustakaan pada Sekolah dasar (SD) Negeri 33 Palembang

Pengertian Aplikasi pengolahan data Perpustakaan pada Sekolah dasar (SD) Negeri 33 Palembang adalah sebuah aplikasi yang khusus dibangun untuk membantu mengolah data-data yang ada pada perpustakaan Sekolah dasar (SD) Negeri 33 Palembang.

2.2. Teori Khusus

2.2.1. Analisis dan Desain Sistem

Sukamto dan Shalahuddin (2014:17) menjelaskan, analisis sistem adalah mendefinisikan kebutuhan terkait sistem yang dikembangkan.

Sukamto dan Shalahuddin (2014:23) menyatakan, desain atau perancangan dalam pembangunan perangkat lunak merupakan upaya untuk mengkonstruksi sebuah sistem.

2.2.2. Pengembangan Sistem

Sukamto dan Shalahuddin (2014:26) menjelaskan SDLC atau *Software Development Life Cycle* atau sering disebut *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik).



2.2.2.1. Model Pengembangan Sistem

Sukamto dan Shalahuddin (2014:28) menjelaskan SDLC memiliki beberapa model dalam pengembangan sistem, salah satunya adalah model air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Berikut adalah tahapan dalam model *waterfall* :

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memesifikasi kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean.

3. Pembuatan kode program

Desain harus ditranslasikan ke dalam program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus kepada perangkat lunak secara dari segi logic dan fungsional dan memastikan bahwa semua bagian suda diuji.

2.2.2.2. Pengertian *Black-Box Testing*

Sukamto dan Shalahuddin (2014:275) menjelaskan bahwa *Black-Box Testing* (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Mengenai pengertian *BlackboxTesting*, Simanjutak, dkk (2010) menjelaskan bahwa:

Black-box testing adalah metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja). Pengetahuan khusus dari kode aplikasi / struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Uji kasus



dibangun di sekitar spesifikasi dan persyaratan, yakni, aplikasi apa yang seharusnya dilakukan. Menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus. Tes ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional. Perancang uji memilih input yang valid dan tidak valid dan menentukan output yang benar.

Black-Box Testing adalah salah satu cara pengujian perangkat lunak yang digunakan untuk melihat sistem kerja perangkat lunak.

Jadi, *black-Box Testing* adalah salah satu cara pengujian perangkat lunak yang digunakan untuk melihat sistem kerja perangkat lunak.

2.2.3. Pengertian Diagram Konteks

Ladjamudin(2005:64) menyatakan diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem.

Saputra (2014:7) menjelaskan bahwa diagram konteks merupakan diagram alir tingkat tinggi yang menggambarkan seluruh jaringan, masukan, dan keluaran untuk menggambarkan sistem yang sedang berjalan. Diagram ini berisi siapa saja yang memberikan data (inputan) ke sistem serta kepada siapa data informasi itu diberikan (melalui sistem).

Sunyoto (2014:130) menjelaskan bahwa diagram konteks merupakan tingkatan tertinggi dalam *data flow diagram*, menggambarkan satu lingkaran besar yang dapat mewakili seluruh proses yang terdapat di dalam suatu sistem.

Jadi, diagram konteks adalah diagram level tertinggi yang menggambarkan proses keseluruhan suatu sistem.

2.2.4. Pengertian DFD (*Data Flow Diagram*)

Sutabri (2012:117) menjelaskan, *Data Flow Diagram (DFD)* adalah suatu *network* yang menggambarkan suatu sistem automat/komputerisasi, manualisasi, atau gabungan dari keduanya, yang penggambarannya disusun dalam bentuk kumpulan komponen sistem yang saling berhubungan sesuai dengan aturan mainnya. Keuntungan penggunaan *DFD* adalah memungkinkan untuk menggambarkan sistem dari level yang paling tinggi kemudian menguraikannya menjadi level lebih rendah (dekomposisi).




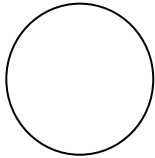
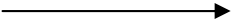
Indrajani (2015:27) menyatakan, *Data Flow Diagram (DFD)* adalah sebuah alat yang menggambarkan aliran data sampai sebuah sistem selesai, dan kerja atau proses dilakukan dalam sistem tersebut.

Sukanto dan Shalahuddin (2014:70) menyatakan, *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*).

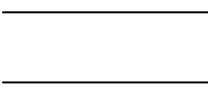
Sutabri (2012:116) menyatakan bahwa pendekatan analisis terstruktur diperkenalkan oleh DeMarco (1978) dan Gane Sarson (1979) melalui buku metodologi struktur analisis dan desain sistem informasi.

Simbol atau lambang yang digunakan dalam membuat *data flow diagram* versi Yourdan, De Marco ada 4 (empat) buah yaitu sebagai berikut:

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

No.	Notasi	Keterangan
1.		Entitas Luar
2.		Proses
3.		Arus Data

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

No.	Notasi	Keterangan
4.		Penyimpanan Data

(Sumber : Ladjamudin, 2005:72)

Sutabri (2012:120) menjelaskan tentang tahapan pada *Data Flow Diagram* yaitu:

a. Diagram Konteks

Diagram ini dibuat untuk menggambarkan sumber serta tujuan data yang akan diproses atau dengan kata lain diagram tersebut digunakan untuk menggambarkan sistem secara umum/ global dari keseluruhan sistem yang ada.

b. Diagram Nol



Diagram ini dibuat untuk menggambarkan tahapan proses yang ada di dalam diagram konteks, yang penjabarannya lebih terperinci.

2.2.5. Pengertian *BlockChart*

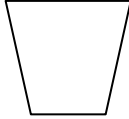

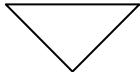
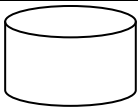

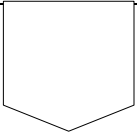
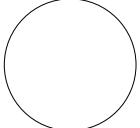

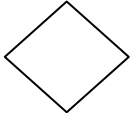

Kristanto (2008:68) menjelaskan, *blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.

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


Tabel 2.2. Simbol-simbol dalam *Block Chart*

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

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

No	Simbol	Nama Simbol
3.		Proses manual.
4.		Proses yang dilakukan oleh komputer.
5.		Menandakan dokumen yang diarsifkan (arsif manual)
6.		Data penyimpanan (data storage)
7.		Proses apa saja yang tidak terdefinisi termasuk aktifitas fisik.
8.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang ain.
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 (<i>decision</i>).
12.		Layar peraga (<i>monitor</i>).

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

No	Simbol	Nama Simbol
13.		Pemasukan data secara manual.

(Sumber : Kristanto, 2008:68)

2.2.6. Pengertian ERD (*Entity Relational Diagram*)

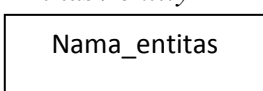
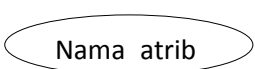
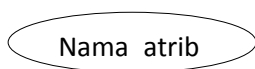
Ladjamudin (2005:143) menyatakan, diagram hubung atau dikenal dengan sebutan E-R Diagram adalah notasi grafik dari sebuah model data atau sebuah model jaringan yang menjelaskan tentang data yang tersimpan (*stored data*) dalam sistem secara abstrak.

Sutanta (2011:91) menjelaskan, *Entity Relationship Diagram (ERD)* adalah bentuk diagram untuk menggambarkan *Entity Relationship Model (ERM)* yang digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara *logic*.

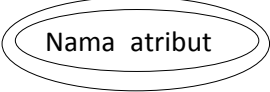
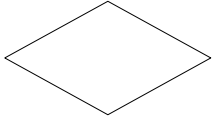
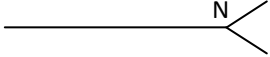
Sukamto dan Shalahuddin (2014:53) menyatakan, ERD adalah bentuk paling awal dalam melakukan perancangan basis data relasional.

Berikut adalah simbol-simbol yang digunakan pada ERD:

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

No	Simbol	Keterangan
1.	Entitas / <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data;
2.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
3.	Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan.

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

No	Simbol	Keterangan
4.	Atribut multivalai / <i>multivalued</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali kata kerja;
6.	Asosiasi / <i>association</i> 	Penghubung antar relasi dan entitas dimana ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian.

(Sumber : Sukanto dan Shalahuddin, 2014:50)

Adapun jenis-jenis Kerelasiaan Antar Entitas (*Relationship*), yaitu:

1. Kerelasiaan jenis 1-ke-1/satu ke satu (*one to one*)

Kerelasiaan jenis ini terjadi jika kejadian atau transaksi di antara dua entitas yang berhubungan hanya memungkinkan terjadi sebuah kejadian atau transaksi pada kedua entitas.

2. Kerelasiaan jenis n-ke-1/banyak ke satu (*many to one*) atau 1-ke-n/satu ke banyak (*one to many*).

Kerelasiaan jenis ini terjadi jika kejadian atau transaksi di antara dua entitas yang berhubungan hanya memungkinkan terjadi satu kali dalam entitas pertama dan dapat terjadi lebih dari satu kali kejadian atau transaksi pada entitas kedua.

3. Kerelasiaan jenis n-ke-n/banyak ke banyak (*many to many*)

Kerelasiaan jenis ini terjadi jika kejadian atau transaksi di antara dua entitas yang berhubungan memungkinkan terjadi lebih dari satu kali dalam entitas pertama dan entitas kedua.



2.2.7. Pengertian *Flowchart*

Indrajani (2015:31) menjelaskan, *flowchart* merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program,

Saputra (2014:14) menjelaskan, *flowchart* merupakan suatu diagram yang menggambarkan alur kerja suatu sistem.

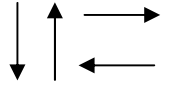

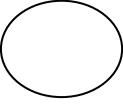
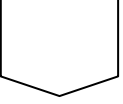
Ladjamudin(2005:263) menyatakan, *flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah.

Simbol-simbol yang digunakan dapat dibagi menjadi 3 (tiga) kelompok, yakni sebagai berikut:

1. *Flow direction symbols*

Digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Berikut simbol *Flow direction symbols* seperti pada tabel dibawah ini:

Tabel 2.4. Simbol-simbol *Flow direction symbols*

No	Simbol	Nama Simbol	Keterangan
1.		Arus <i>(Flow)</i>	Menyatakan jalannya arus suatu proses.
2		Communication Link	Menyatakan bahwa adanya transisi suatu data/informasi dari satu lokasi ke lokasi lainnya.
3		Connector	Menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang sama.
4		Offline Connector	Menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda.

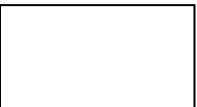

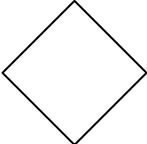

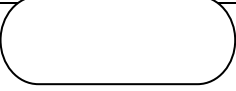
(Sumber: Ladjamudin, 2005:266)



2. Processing symbols

Menunjukkan jenis operasi pengolahan dalam suatu proses / prosedur. Berikut simbol *Processing symbols* seperti pada tabel dibawah ini:

Tabel 2.5. Simbol-simbol *Processing symbols*

No	Simbol	Nama Simbol	Keterangan
1.		<i>Offline Connector</i>	Menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda.
2.		<i>Manual</i>	Menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer.
3.		<i>Decision</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya/tidak.
4.		<i>Predefined Process</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
5.		<i>Terminal</i>	Menyatakan permulaan atau akhir suatu program.



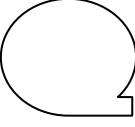


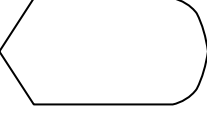

(Sumber: Ladjamudin, 2005:267)



3. Input / Output symbols

Menunjukkan jenis peralatan yang digunakan sebagai media *input* atau *output*.


Tabel 2.6. Simbol-simbol *Input / Output symbols*

No.	Simbol	Nama Simbol	Keterangan
1.		<i>Input/Output</i>	Menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya.
2.		<i>Punched Card</i>	Menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu.
3.		<i>Magnetic Tape</i>	Menyatakan <i>input</i> berasal dari pita magnetic atau <i>output</i> disimpan ke pita magnetic.
4.		<i>Disk Storage</i>	Menyatakan input berasal dari dari disk atau output disimpan ke disk.
5.		<i>Document</i>	Mencetak keluaran dalam bentuk dokumen (melalui printer).
6.		<i>Display</i>	Menyatakan peralatan output yang digunakan berupa layar (video, komputer).
7.		<i>Keying Operation</i>	Menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard.



8.		<i>OfflineStorage</i>	Menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.
----	---	-----------------------	--

Lanjutan Tabel 2.6. Simbol-simbol *Input / Output symbols*

No.	Simbol	Nama Simbol	Keterangan
9.		<i>Manual Input</i>	Memasukkan data secara manual dengan menggunakan <i>onlinekeyboard</i> .

(Sumber: Ladjamudin, 2005:268)

2.2.8. Pengertian Kamus Data

Indrajani(2015:30) menyatakan, kamus data adalah katalog fakta tentang data dan kebutuhan informasi suatu sistem informasi.

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

Ladjamudin (2005:70) menyatakan, kamus data sering disebut juga dengan sistem *datadictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi.

Sukanto dan Shalahuddin (2014:74) menjelaskan bahwa kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan, yaitu sebagai berikut:

Tabel 2.7.Simbol-simbol dalam Kamus Data

No	Notasi	Arti
1.	=	Terdiri atas
2.	+	AND atau dan
3.	()	Opsional
4.	[]	Memilih salah satu alternatif
5.	**	Komentar



6.	@	identifikasi atribut kunci
7.		Pemisah alternatif simbol []

(Sumber: Sukamto dan Shalahuddin, 2014:74)

2.3. Teori Program

2.3.1. Basis Data (*Database*)

Sukamto dan Shalahuddin (2014:43) menjelaskan bahwa basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

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

Kadir (2003:70) mengatakan bahwa basis data (*database*) merupakan sekumpulan tabel, hubungan, dan lain-lain yang berkaitan dengan penyimpanan data.

Dari pengertian tersebut dapat disimpulkan bahwa basis data adalah media untuk menyimpan sekumpulan dari data-data atau *record* terhadap obyek tertentu agar dapat diakses dengan mudah dan cepat.

2.3.2. Pengertian *HTML*

Jubilee Enterprise (2015:135) menyatakan, *HTML* adalah *script* pemrograman yang mengatur bagaimana kita menyajikan informasi di dunia internet dan bagaimana informasi itu membawa kita dari satu tempat ke tempat lainnya.

Sibero (2014:19) berpendapat, *HTML* adalah bahasa yang digunakan pada dokumen *web* sebagai bahasa untuk pertukaran dokumen *web*.

Saputra (2012:1) menyatakan, *HTML* (*Hypertext Markup Language*), yaitu suatu bahasa pemrograman hypertext.

Struktur dokumen *HTML* sebagai berikut:

```
<html>
```



```

<head>
  <title></title>
</head>
</html>

```

Dari pengertian tersebut, dapat disimpulkan bahwa *HTML* adalah bahasa pemrograman yang digunakan untuk menyajikan informasi atau dokumen ke dunia internet.

2.3.3. *Cascading Style Sheet (CSS)*

Madcoms (2012:54) menyatakan bahwa *Cascading Style Sheet (CSS)* kumpulan kode-kode yang digunakan untuk mengendalikan tampilan isi suatu halaman *web*.

Sibero (2014:112) berpendapat bahwa *Cascading Style Sheet (CSS)* dikembangkan untuk menata gaya pengaturan halaman *web*.

Saputra (2012:6) menjelaskan bahwa *CCS* atau yang memiliki kepanjangan *Cascading Style Sheet*, merupakan suatu bahasa pemrograman web yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam web sehingga tampilan web akan lebih rapi, terstruktur, dan seragam.

Dari pengertian tersebut, dapat disimpulkan bahwa *Cascading Style Sheet (CSS)* adalah kumpulan kode-kode yang digunakan untuk pengaturan halaman *web*.

2.3.4. *JavaScript*

Sibero (2014:150) menyatakan, *Java Script* adalah suatu bahasa pemrograman yang dikembangkan untuk dapat berjalan di *webbrowser*.

Sibero (2014:150) juga berpendapat bahwa *Java Script* adalah bahasa skrip, yaitu kumpulan instruksi perintah yang digunakan untuk mengendalikan beberapa bagian dari sistem operasi.

Dari pengertian diatas dapat disimpulkan bahwa *JavaScript* adalah bahasa pemrograman yang memiliki sekumpulan perintah khusus yang digunakan untuk membuat sebuah halaman *web* yang lebih responsif dan interaktif.



2.3.5. Sekilas Tentang *MySql*

1. Pengertian *MySql*

Sibero (2014:97) menjelaskan, *MySQL* atau dibaca 'My Sekuel' dengan adalah suatu *RDBMS (Relation Database Management System)* yaitu aplikasi sistem yang menjelaskan fungsi pengolahan data. Berikut tipe-tipe data pada *MySQL*:

Tabel 2.8. Tipe Data pada *MySQL*

Tipe Data	Ukuran (Bytes)
TINYINT	1
SMALLINT	2
MEDIUM	3
INT	4
BIGINT	8
FLOAT(p)	4
FLOAT	4
DOUBLE	8
REAL	8
DECIMAL (L,D)	L + 1
NUMERIC (L,D)	L + 1
BIT (L)	8
DATE	3
TIME	3
DATETIME	8
TIMESTAMP	4
YEAR	1
CHAR (L)	L



BINARY(L)	L + 1
VARCHAR (L)	L + 1
VARBINARY (L)	L + 1
TINYBLOB	L + 1

Lanjutan Tabel 2.8. Tipe Data pada *MySQL*

Tipe Data	Ukuran (Bytes)
TINYTEXT	L + 1
BLOB	L + 2
TEXT	L + 2
MEDIUMBLOB	L + 3
MEDIUMTEXT	L + 3
LOBLOB	L + 4
LONGTEXT	L + 4
ENUM ('nilai1',...)	1 – 2
SET ('nilai1',...)	1 – 8

Saputra et. all (2012:8), berpendapat bahwa *MySQL* memiliki beberapa kelebihan, antara lain:

1. Bersifat *open source*, yang memiliki kemampuan untuk dapat dikembangkan lagi.
2. Menggunakan bahasa *SQL (StructureQueryLanguage)*, yang merupakan standar bahasa dunia dalam pengolahan data.
3. *Super performance* dan *reliable*, tidak bisa diragukan, pemrosesan database-nya sangat cepat dan stabil.
4. Sangat mdah dipelajari (*easyofuse*).
5. Memiliki dukungan support (group) pengguna *MySQL*.
6. Mampu lintas *Platform*, dapat berjalani berbagai sistem informasi.
7. *Multiuser*, di mana *MySQL* dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami konflik.



2.3.6. Sekilas Tentang *PHP*

2.3.6.1. Pengertian *PHP* (*PHP HypertextPreprocessor*)

Madcoms (2012:206) menyatakan, “*PHP* adalah salah satu bahasa pemrograman yang berjalan dalam sebuah *webserver* dan berfungsi sebagai pengolah data pada sebuah *server*.”

Sibero (2014:49) berpendapat, “*PHP* adalah suatu bahasa dengan hak cipta terbuka atau yang juga dikenal dengan istilah *opensource*, yaitu pengguna dapat mengembangkan kode-kode fungsi *PHP* sesuai kebutuhannya.”

Wahana Komputer (2012:76) menjelaskan bahwa *PHP* dapat dikatakan sebagai sebuah *server-side embedded script language*, artinya sintak-sintak dan perintah program yang ditulis akan sepenuhnya dijalankan oleh *server* tetapi dapat disertakan pada halaman *HTML* biasa.

Dari ketiga pengertian di atas, dapat disimpulkan bahwa *PHP* adalah salah satu bahasa pemrograman yang dirancang untuk bekerja atau membangun sebuah aplikasi berbasis *web*.

2.3.6.2. Sintaks *PHP*

Wahana Komputer (2012:76) menjelaskan bahwa untuk menuliskan sintaks *PHP*, harus diawali dengan *tag*<? dan diakhiri dengan *tag* ?>/ sedangkan sintaks untuk menampilkan dalam *web browser* dapat menggunakan perintah *print* atau *echo*. Selain itu *PHP* juga bisa dituliskan dalam skrip *HTML*. Lebih jelasnya perhatikan kode program berikut:

```
<html>
```

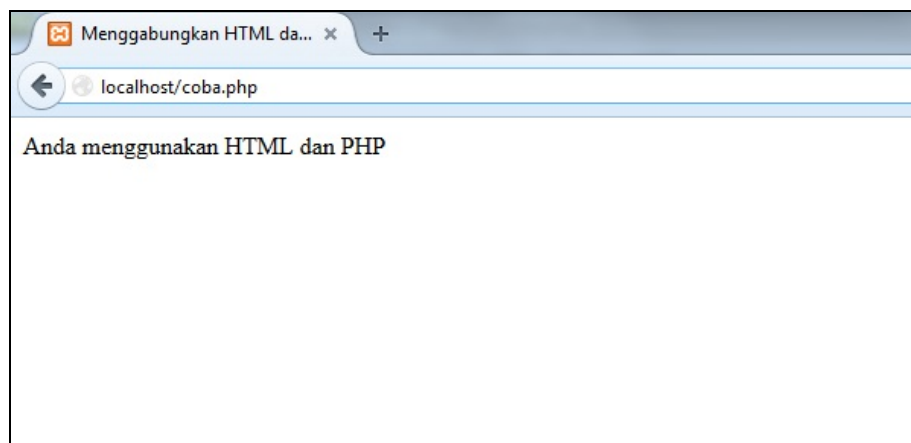
```
<head>
```

```
<title> Menggabungkan HTML dan PHP</title>
```



```
</head>
<body>
    <?php
        echo "Anda menggunakan HTML dan PHP";
    ?>
</body>
</html>
```

Pada saat dijalankan pada *browser*, tampilannya sebagai berikut:

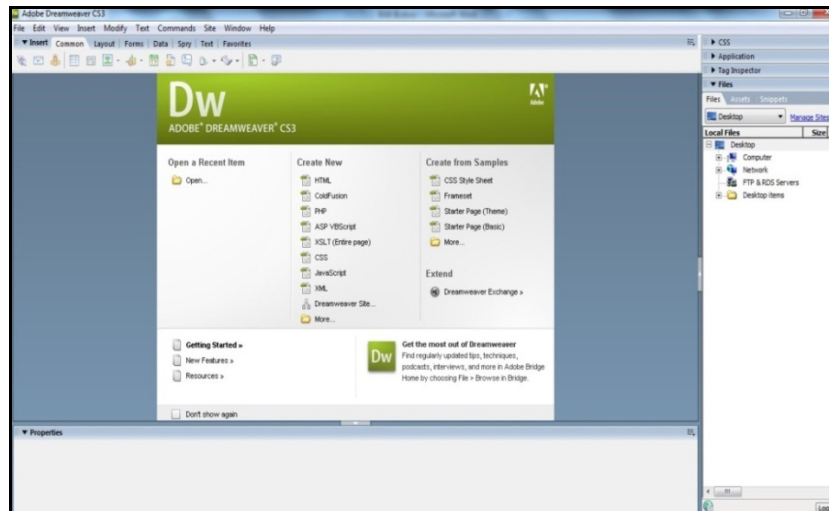


Gambar 2.1.Tampilan *PHP* sederhana

2.3.7. Sekilas Tentang *Adobe Dreamweaver*

2.3.7.1. Pengertian *Adobe Dreamweaver*

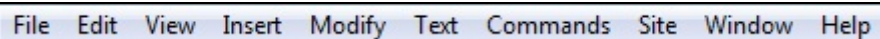
Sibero (2014:384) menjelaskan bahwa *Adobe Dreamweaver* adalah suatu produk *WebDeveloper* yang dikembangkan oleh *AdobeSystemInc.*, sebelumnya produk *Dreamweaver* dikembangkan oleh *Macromedia Inc.*, yang kemudian sampai dengan saat ini pengembangannya diteruskan oleh *AdobeSystemInc.* Setelah diambil alih oleh *AdobeSystemInc.*, *Dreamweaver* dikembangkan dan dirilis dengan kode nama *Creative Suit (CS)*.



Gambar 2.2.Tampilan Utama *Adobe Dreamweaver CS3*

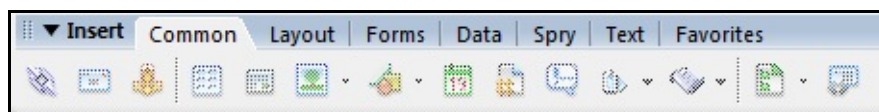
2.3.7.2. Ruang Kerja (*Workspace*) *Adobe Dreamweaver CS3*

- a. **WelcomeScreen**, adalah layar pada *workspace* yang digunakan untuk memudahkan pengguna membuat *file* yang pernah digunakan.
- b. **Menu**, adalah media yang digunakan untuk melakukan atau menjalankan perintah pada *Dreamweaver*. Menu berisi perintah untuk pembuatan, pengaturan, manipulasi dan konfigurasi pada *Dreamweaver*.



Gambar 2.3.Tampilan Menu *Adobe Dreamweaver CS3*

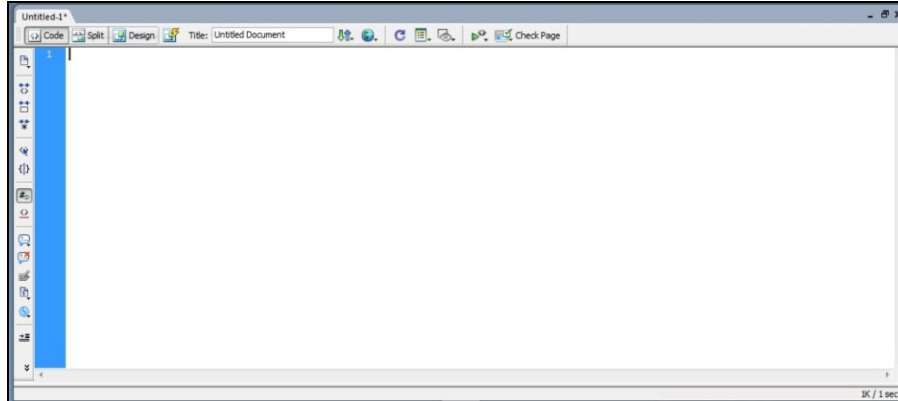
- c. **InsertBar**, adalah suatu bentuk *toolbar* yang berisi perintah untuk menambahkan suatu elemen pada dokumen.



Gambar 2.4.Tampilan Insert Bar *Adobe Dreamweaver CS3*

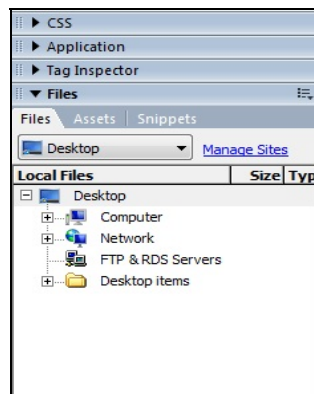


- d. **DocumentWindow**, adalah editor *text* atau visual pada ruang kerja. *DocumentWindow* menyajikan bentuk dokumen secara kode maupun visual.



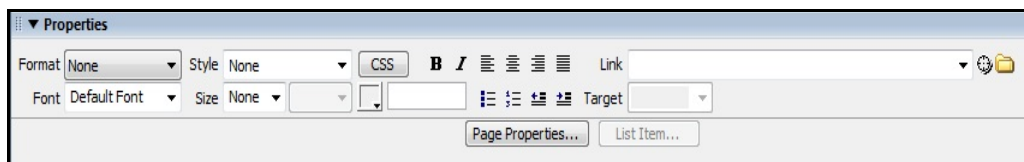
Gambar 2.5. Tampilan *DocumentWindow* Adobe Dreamweaver CS3

- e. **PanelGroup**, adalah bagian ruang kerja *Dreamweaver* yang berisikan kumpulan *panel*. *PanelGroup* terdiri dari kumpulan beberapa *panelcontrol*.



Gambar 2.6. Tampilan *Panel Group* Adobe Dreamweaver CS3

- f. **PropertyInspector**, digunakan untuk menampilkan informasi *property* elemen *HTML* tertentu yang sedang dipilih.



Gambar 2.7. Tampilan *PropertyInspector* Adobe Dreamweaver CS3