



## **BAB II TINJAUAN PUSTAKA**

### **2.1. Teori Umum**

#### **2.1.1. Pengertian Aplikasi**

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

Sugiar (2014:83) menjelaskan, “Aplikasi adalah program yang dibuat untuk melaksanakan tugas tertentu yang dibutuhkan oleh pengguna komputer (*user*).”

#### **2.1.2. Pengertian Persediaan**

Indrajani (2015:63) menjelaskan, “Persediaan adalah aset yang tersedia untuk dijual dalam proses bisnis biasa atau aset yang ada dalam proses produksi untuk di jual kembali, atau aset dalam bentuk material atau bahan baku untuk di gunakan dalam proses produksi.”

#### **2.1.3. Pengertian Pendistribusian**

Yustisia (2015:66) menjelaskan, “Pendistribusian adalah penjualan, pengedaran, dan atau penyebaran ciptaan atau prodak hak terkait.”

#### **2.1.4. Pengertian Alat-alat Berat**

Kusuma (2013) menjelaskan, “Alat berat adalah alat bantu yang di gunakan oleh manusia untuk mengerjakan pekerjaan yang berat / susah untuk di kerjakan dengan tenaga manusia / membantu manusia dalam mengerjakan pekerjaan yang berat. misal untuk membuat sebuah danau, manusia menggunakan alat berat untuk mengerjakannya.”

#### **2.1.5. Pengertian Barang**

Juanda dan Lubis (2011:5) menjelaskan, “Barang berwujud yang menurut sifat atau hukumnya dapat berupa barang bergerak atau barang tidak bergerak, dan barang tidak berwujud.”

### **2.1.6. Pengertian Proyek**

Rusdi (2013:3) menjelaskan, “Proyek adalah serangkaian kegiatan-kegiatan yang diprakasai oleh pemilik untuk memenuhi kebutuhan akan fasilitas atau untuk meningkatkan sumberdaya yang dimilikinya.”

### **2.1.7. Pengertian PT**

Winarti dan Syahrizal (2012:1) menjelaskan, “PT adalah singkatan dari Perseroan Terbatas. Perseroan Terbatas adalah badan hukum yang merupakan persekutuan modal yang dilakukan oleh minimal dua orang dengan tanggung jawab yang hanya berlaku pada perusahaan saja, tanpa melibatkan harta pribadi atau perseorangan yang ada di dalamnya (para pemegang saham), didirikan berdasarkan perjanjian, melakukan kegiatan usaha dengan modal dasar yang seluruhnya terbagi dalam saham dan memenuhi persyaratan yang ditetapkan dalam Undang-Undang Republik Indonesia Nomor 40 Tahun 2007 Tentang Perseroan Terbatas serta peraturan pelaksanaannya.”

### **2.1.8. Pengertian PT Budi Bakti Prima**

PT Budi Bakti Prima adalah perusahaan persero terbatas yang bergerak di bidang kontraktor khusus nya pembangunan jalan. PT Budi Bakti terletak pada Jalan Letjen H.Alamsyah Ratu Prawira Negara No.51 Palembang.

### **2.1.9. Pengertian Aplikasi Persediaan dan Pendistribusian Alat-alat Berat dan Barang Proyek Berbasis Web pada PT Budi Bakti Prima Palembang**

Aplikasi Persediaan dan Pendistribusian Alat-alat berat dan Barang proyek berbasis web pada PT Budi Bakti Prima Palembang berfungsi mengolah persediaan dan distribusi alat-alat berat dan barang proyek, yang akan memudahkan pimpinan dan kepala gudang mengecek barang-barang milik PT Budi Bakti Prima dalam mengolah data-data dengan menggunakan bahasa pemrograman PHP dan *database* MySQL.

### 2.1.10. Metode Pengembangan Sistem

Kristanto (2008:41), menjelaskan siklus pengembangan system adalah kumpulan-kumpulan kegiatan dari analisis pendesain dan user dari system informasi yang dilaksanakan untuk dikembangkan dan diimplementasikan.

#### 1. Penyelidikan awal

Pada tahap penyelidikan awal, analis belajar dari pemakai mengenai apa yang diharapkan dari sebuah system informasi yang baru.

#### 2. Studi kelayakan

Studi kelayakan merupakan tahap yang paling penting, karena di dalamnya menyangkut berbagai aspek system baru yang diusulkan. Laporan mengenai studi kelayakan harus disampaikan kepada manajemen, yang pada gilirannya akan memberikan beberapa perubahan, menyarankan untuk diadakan penelitian lebih mendalam atau memutuskan untuk segera dilaksanakan. Ada tiga aspek dalam kelayakan yang perlu diperhatikan, yaitu:

- a. Kelayakan teknik merupakan jika tim penyusunan system dapat menyelesaikan masalah tersebut dengan menggunakan *hardware* dan *software* yang tersedia.
- b. Kelayakan operasi merupakan jika tim penyusunan system dapat menyelesaikan masalah tersebut dengan menggunakan personel dan prosedur yang tersedia.
- c. Kelayakan ekonomis merupakan jika tim penyusunan system dapat menyelesaikan masalah tersebut dalam waktu dan anggaran biaya yang masuk akal.

#### 3. Penentuan kebutuhan-kebutuhan system

Kebutuhan-kebutuhan system yang akan dikembangkan meliputi input, output, operasi dan *resources*, untuk memenuhi kebutuhan organisasi masa kini dan masa mendatang.

#### 4. Desain system

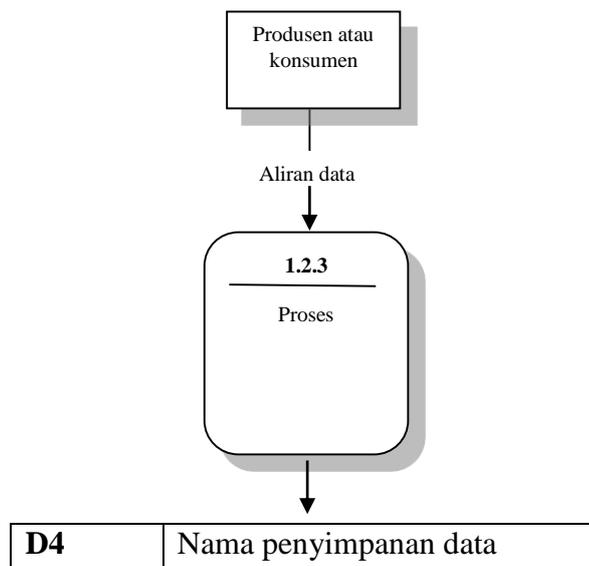
Untuk melakukan perbaikan terhadap system informasi, terlebih dahulu harus dipahami dengan jelas kondisi system yang ada sekarang dan yang dihadapi,

setelah itu sasaran dan kebutuhan system di masa yang akan datang. Kemudian baru dapat dimasukkan ide-ide secara bersama-sama ke dalam suatu desain yang akan memenuhi tujuan-tujuan yang telah ditetapkan.

## 2.2. Teori Khusus

### 2.2.1. Pengertian DFD (*Data Flow Diagram*)

Sukanto dan Shalahuddin (2014:69) menjelaskan, “*Data Flow Diagram* (DFD) awalnya dikembangkan oleh Chris Gane dan Trish Sarson pada tahun 1979 yang termasuk dalam *Structured Systems Analysis and Design Methodology* (SSADM) yang ditulis oleh Chris Gane dan Trish Sarson. System yang dikembangkan ini berbasis pada dekomposisi fungsional dari sebuah system.” Berikut adalah contoh DFD yang dikembangkan oleh Chris Gane dan Trish Sarson:



**Gambar 2.1.** Contoh DFD yang dikembangkan Chris Gane dan Trish Sarson

Edward Yourdon dan Tom DeMarco memperkenalkan metode yang lain pada tahun 1980-an di mana mengubah persegi dengan sudut lengkung (pada DFD Chris Gane dan Trish Sarson) dengan lingkaran untuk menotasikan. DFD

Edward Yourdon dan Tom DeMarco populer digunakan sebagai model analisis system perangkat lunak untuk system perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur.

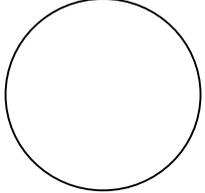
Informasi yang ada di dalam perangkat lunak dimodifikasi dengan beberapa transformasi yang dibutuhkan. *Data Flow Diagram* (DFD) atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).

DFD dapat digunakan untuk merepresentasikan sebuah system atau perangkat lunak pada beberapa level abstraksi. DFD dapat dibagi menjadi beberapa level yang lebih detail untuk merepresentasikan aliran informasi atau fungsi yang lebih detail. DFD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi. Oleh karena itu, DFD lebih sesuai digunakan untuk memodelkan fungsi-fungsi perangkat lunak yang akan diimplementasikan menggunakan pemrograman terstruktur karena pemrograman terstruktur membagi-bagi bagiannya dengan fungsi-fungsi dan prosedur-prosedur.

DFD tidak sesuai untuk memodelkan system perangkat lunak yang akan dibangun menggunakan pemrograman berorientasi objek. Paradigma pemrograman terstruktur dan pemrograman berorientasi objek merupakan hal yang berbeda. Jangan mencampuradukkan pemrograman terstruktur dan pemrograman berorientasi objek.

Rosa dan Shalahuddin (2014:71) menjelaskan, “notasi- notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut:

**Tabel 2.1.** Notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco)

Notasi	Keterangan
	<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: Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>
	<p><i>File</i> atau basisdata atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (<i>Entity Relationship Diagram (ERD)</i>, <i>Conceptual Data Model (CDM)</i>, <i>Physical Data Model (PDM)</i>)</p> <p>Catatan: Nama yang diberikan pada sebuah penyimpanan biasanya kata benda</p>
	<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 system yang dimodelkan.</p> <p>Catatan: Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda</p>

**Lanjutan Tabel 2.1.** Notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco)

Notasi	Keterangan
	<p>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>)</p> <p>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”</p>

Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD:

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

#### 4. Membuat DFD Level 3 dan seterusnya

DFD Level 3, 4, 5 dan seterusnya merupakan *breakdown* dari modul pada DFD Level di-atasnya. *Breakdown* pada level 3, 4, 5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.

Pada satu diagram DFD sebaiknya jumlah modul tidak boleh lebih dari 20 buah.

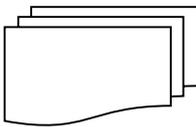
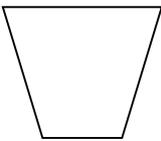
Jika lebih dari 20 buah modul, diagram akan terlihat rumit dan susah untuk dibaca sehingga menyebabkan system yang dikembangkan juga menjadi rumit.

#### 2.2.2. Pengertian *Blockchart*

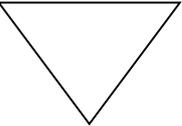
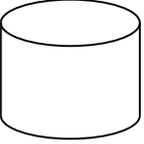
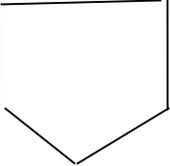
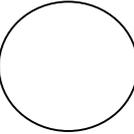
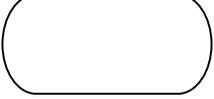
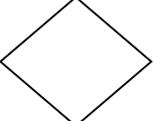
Kristanto (2008:75) menjelaskan, “*Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *Blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.”

Kristanto (2008:75) menjelaskan, “simbol-simbol yang sering digunakan dalam *blockchart* dapat dilihat pada tabel berikut ini:

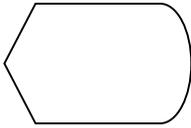
**Tabel 2.2.** Simbol-simbol dalam *Blockchart*

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

**Lanjutan Tabel 2.2.** Simbol-simbol dalam *Blockchart*

No	Simbol	Keterangan
4.		Proses dilakukan oleh komputer.
5.		Menandakan dokumen yang diarsipkan (arsip manual)
6.		Data penyimpanan (Storage)
7.		Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik.
8.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain.
9.		Terminal yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.
10.		Terminal yang menandakan awal dan akhir dari suatu aliran.
11.		Pengambilan keputusan ( <i>Decision</i> ).

**Lanjutan Tabel 2.2.** Simbol-simbol dalam *Blockchart*

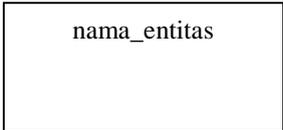
No	Simbol	Keterangan
12.		Layar peraga ( <i>monitor</i> ).

### 2.2.3. Pengertian ERD (*Entity Relationship Diagram*)

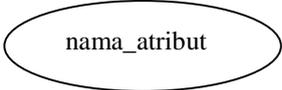
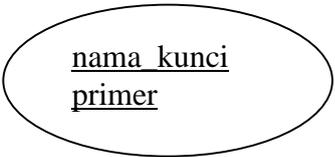
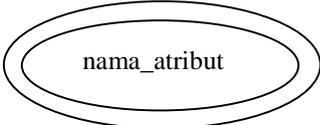
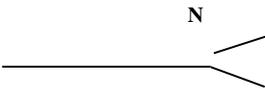
Sukanto dan Shalahuddin (2014:50) menjelaskan, “Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan DBMS maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow’s Foot, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen.”

Berikut adalah symbol-simbol yang digunakan pada ERD dengan notasi Chen:

**Tabel 2.3.** Symbol-simbol ERD dengan notasi Chen

Simbol	Deskripsi
Entitas / <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi computer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.

**Lanjutan Tabel 2.3.** Symbol-simbol ERD dengan notasi Chen

Simbol	Deskripsi
Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
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)
Atribut multivalai / <i>multivalue</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
Asosiasi / <i>association</i> 	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian  Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas

ERD biasanya memiliki hubungan *binary* (satu relasi menghubungkan dua buah entitas). Beberapa metode perancangan ERD menoleransi hubungan relasi *ternary* (satu relasi menghubungkan tiga buah relasi) atau *N-ary* (satu relasi menghubungkan banyak entitas), tapi banyak metode perancangan ERD yang tidak mengizinkan hubungan *ternary* atau *N-ary*.

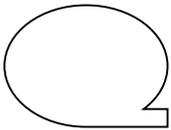
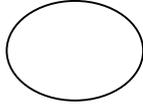
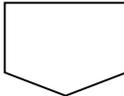
#### 2.2.4. Pengertian *Flowchart*

Indrajani (2015:36), “*Flow chart* adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program.”

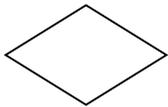
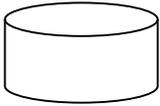
eWolf Community (2011:16), “*flowchart* adalah simbl-simbol pekerjaan yang menunjukkan bagan aliran proses yang saling terhubung.”

eWolf Community (2011:17), menjelaskan simbol-simbol dalam *flowchart* adalah sebagai berikut:

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

No.	Simbol	Arti
1.		Simbol Start atau End yang mendefinisikan awal atau akhir dari sebuah <i>flowchart</i> .
2.		Simbol pemrosesan yang terjadi pada sebuah alur kerja.
3.		Simbol yang menyatakan bagian dari program (sub program).
4.		Simbol masukan atau keluaran dari atau ke sebuah pita magnetic.
5.		Simbol Input/Output yang mendefinisikan masukan dan keluaran proses.
6.		Simbol konektor untuk menyambung proses pada lembar kerja yang sama.
7.		Simbol konektor untuk menyambung proses pada lembar kerja yang berbeda.
8.		Simbol masukan atau keluaran dari atau ke sebuah dokumen.

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

No.	Simbol	Arti
9.		Simbol untuk memutuskan proses lanjutan dari kondisi tertentu.
10.		Simbol database atau basis data.
11.		Simbol yang menyatakan piranti keluaran, seperti layar monitor, <i>printer</i> , dll.
12.		Simbol yang mendefinisikan proses yang dilakukan secara manual.

### 2.2.5. Pengertian Kamus Data

Sukamto dan Shalahuddin (2014:73) menjelaskan, “Kamus data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada DFD. Kamus data 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).” Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur. Kamus data biasanya berisi:

- a. Nama – nama dari data
- b. Digunakan pada – merupakan proses-proses yang terkait data
- c. Deskripsi – merupakan deskripsi data
- d. Informasi tambahan – seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data

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

**Tabel 2.5.** Simbol-simbol dalam Kamus Data

<b>Simbol</b>	<b>Keterangan</b>
=	disusun atau terdiri dari
+	Dan
[]	baik ...atau...
{ <sup>n</sup> }	n kali diulang/ bernilai banyak
()	data opsional
*...*	batas komentar

Kamus data pada DFD nanti harus dapat dipetakan dengan hasil perancangan basis data yang dilakukan sebelumnya. Jika ada kamus data yang tidak dapat dipetakan pada tabel hasil perancangan basis data berarti hasil perancangan basis data dengan perancangan dengan DFD masih belum sesuai, sehingga harus ada yang diperbaiki baik perancangan basis datanya, perancangan DFD-nya, atau keduanya.

## 2.3. Teori Program

### 2.3.1. Pengertian Basis Data (*Database*)

Indrajani (2015:70), “Basis data adalah sekumpulan elemen data terintegrasi yang secara logika saling berhubungan.”

Sukanto dan Shalahudin (2014:43), “Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.”

Sutanta (2011:35), “Basis data adalah sekumpulan dari berbagai macam-tipe *record* yang memiliki hubungan antar-*record* dan rincian data terhadap obyek tertentu.”

Jadi berdasarkan pengertian diatas, dapat disimpulkan bahwa basis data adalah kumpulan informasi yang disimpan didalam computer secara sistematis untuk memperoleh informasi dari basis data tersebut.

### 2.3.2. HTML ( *Hyper Text Markup Language* )

Winarno (2014:1), “*Hypertext Markup Language* (HTML) adalah sebuah bahasa untuk menampilkan konten di web.”

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

Jadi berdasarkan pengertian diatas, dapat disimpulkan bahwa *Hypertext Markup Language* (HTML) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web.

### 2.3.3. Pengertian *MySQL*

Winarno (2014:101), “*MySQL* adalah sebuah *software database*. *MySQL* merupakan tipe data relasional yang artinya *MySQL* menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan.”

Kadir (2013:15) menjelaskan, “*MySQL* adalah nama *database server*. *Database server* adalah *server* yang berfungsi untuk menangani *database*. *Database* adalah suatu pengorganisasian data dengan tujuan memudahkan penyimpanan dan pengaksesan data. Dengan menggunakan *MySQL*, kita bisa menyimpan data dan kemudian data bisa diakses dengan cara yang mudah dan cepat.”

Saputra dkk (2012:7), “*MySQL* merupakan *database* yang bekerja menggunakan *SQL Language* (*Structure Query Language*).”

Jadi berdasarkan pengertian diatas, dapat disimpulkan bahwa *MySQL* adalah sistem manajemen database SQL yang bersifat *open source* dan paling populer saat ini.

### 2.3.4. Tentang *PHP*

#### 2.3.4.1. Pengertian *PHP* (*Hypertext Preprocessor*)

Nugroho (2013:153) menjelaskan, “*PHP* kepanjangan dari *Hypertext Preprocessor*) itu bahasa pemrograman berbasis web. *PHP* itu adalah bahasa

program yang digunakan untuk membuat aplikasi berbasis web (website, blog, atau aplikasi web).”

Kadir (2013:120) menjelaskan, “PHP merupakan bahasa pemrograman yang ditujukan untuk membuat aplikasi web. Ditinjau dari pemrosesannya, PHP tergolong berbasis *server side*. Artinya, pemrosesan dilakukan di *server*.”

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

#### 2.3.4.2. Tipe Data *PHP*

Sidik (2014:106) menjelaskan tipe data *PHP* yaitu :

1. Tipe Data Integer

Tipe Data Integer adalah tipe data numerik yang digunakan untuk menyatakan bilangan bulat. Tipe data integer dapat dituliskan dengan notasi desimal, oktal atau hexadesimal.

2. Tipe Data *Floating Point*

Tipe Data *Floating Point* adalah tipe data yang digunakan untuk menyatakan data numerik berupa bilangan pecahan.

3. Tipe Data *String*

Tipe Data *String* adalah tipe data yang berjenis *string* atau teks. Teks atau tulisan adalah semua data yang dinyatakan dengan huruf dan tanda baca, termasuk didalamnya ada data berupa angka. Data angka yang ada didalam suatu teks atau tulisan akan dianggap karakter, karena data tersebut tidak dapat dioperasikan dalam operasi matematika

4. Tipe Data *Boolean*

Tipe Data *Boolean* adalah tipe data yang digunakan untuk menyatakan nilai benar atau salah, atau nilai *Boolean*.

## 5. Tipe Data *Array*

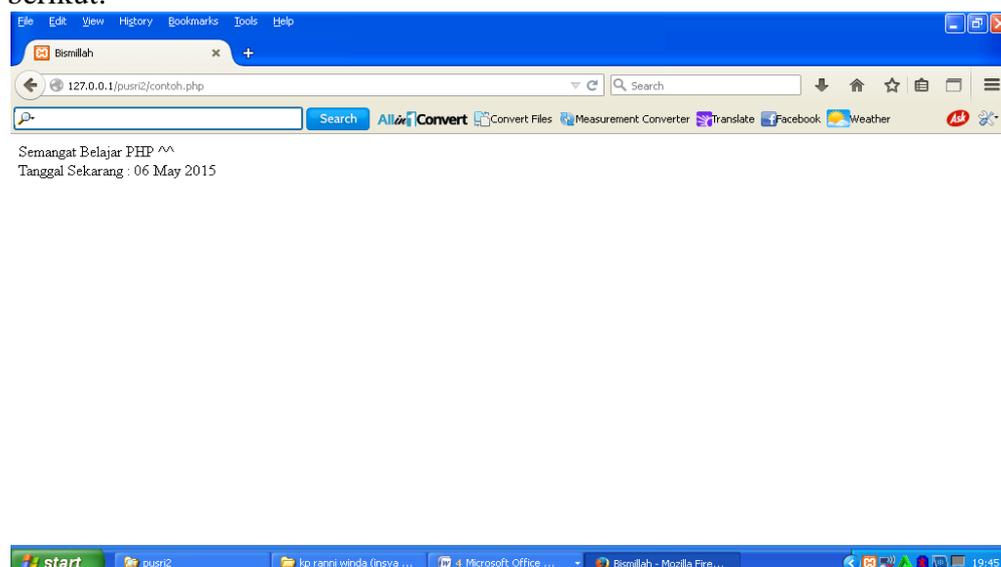
*Array* dalam *PHP* juga merupakan tipe data, bukan sekedar variabel jamak, variabel yang mempunyai banyak elemen yang diacu dengan satu nama yang sama. Setiap elemen variabel diacu dengan menggunakan nomor variabel.

### 2.3.4.3. Skrip *PHP*

Berikut contoh kode *PHP* yang berada di dalam kode *HTML* (disimpan dengan ekstensi *.php*).

```
<HTML>
<HEAD>
    <TITLE>Bismillah</TITLE>
</HEAD>
<BODY>
    Semangat Belajar PHP ^^<BR>
    <?php
    printf("Tanggal Sekarang : %s", Date("d F Y"));
    ?>
</BODY>
</HTML>
```

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



**Gambar 2.2** Tampilan *PHP* Sederhana

### 2.3.5. Pengertian Adobe Dreamweaver

Madcoms (2012:2), “*Dreamweaver* 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.”

Sadeli (2014:12), “*Dreamweaver* adalah suatu perangkat lunak web editor keluaran adobe sistem yang digunakan untuk mengembangkan dan mendesain suatu *website* dengan fitur-fitur yang menarik dan kemudahan dalam penggunaannya.”

Bekti (2015:2), “*Adobe Dreamweaver* adalah aplikasi design dan pengembangan web yang menyediakan editor WYSIWYG visual (bahasa sehari-hari yang disebut sebagai *Design view*) dan kode editor dengan fitur standar seperti *syntax highlighting*, *code completion*, dan *code collapsing* serta fitur lebih canggih seperti *real-time syntax checking* dan *code introspection*”

#### 2.3.5.1. Ruang Kerja Adobe Dreamweaver CS6

Madcoms (2012:3) menjelaskan, “Tampilan ruang kerja *Adobe Dreamweaver CS6* adalah sebagai berikut:

a. *Application Bar*

Berada di bagian paling atas jendela aplikasi *Dreamweaver CS5*. Baris ini berisi tombol *Workspce (Workspace Swithcer)*, *Menu* dan aplikasi-aplikasi dari *Dreamweaver*.

b. *Toolbar Document*

Berisi tombol-tombol yang digunakan untuk mengubah tampilan jendela dokumen, sebagai contoh tampilan jendela *design* atau tampilan jendela *code*.

c. *Jendela Dokumen*

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

d. *Workspace Swithcer*

Digunakan untuk mengubah tampilan ruang kerja (*workspace*) *Dreamweaver CS6*. Sebagai contoh mengubah tampilan menjadi tampilan *Classic*, yaitu tampilan ruang kerja *dreamweaver* versi sebelumnya.

e. *Panel Groups*

Adalah kumpulan panel yang saling berkaitan, Panel ini dikelompokkan pada judul-judul tertentu berdasarkan fungsinya. Panel ini digunakan untuk memonitor dan memodifikasi pekerjaan. Secara default, *panel group* berisi *panel insert*, *CSS styles*, *Asset*, *AP Element*, dan *Files*.

f. *Tag Selector*

Diletakkan di bagian bawah jendela dokumen, satu baris dengan status bar. 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.

g. *Panel Property*

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

h. *Toolbar Coding*

Berisi tombol-tombol yang digunakan untuk melakukan operasi *code-code* standar.

i. *Panel Insert*

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

j. *Panel group*

Kumpulan *panel* yang saling berkaitan satu sama lainnya yang dikelompokkan di bawah satu judul, seperti: *insert*, *files*, *css styles*, dan sebagainya.

k. *Panel Files*

digunakan untuk mengatur file-file dan folder-folder yang membentuk situs *web* anda.

### 2.3.6. Pengertian XAMPP

Sidik (2014:72) menjelaskan, “XAMPP (X/(Windows/Linux) Apache MySQL PHP dan Perl) adalah paket *server web* PHP dan *database* MySQL yang paling populer dikalangan pengembang *web* dengan menggunakan PHP dan MySQL sebagai databasenya.”

Nugroho (2013:1) menjelaskan, “XAMPP adalah paket program web lengkap yang dapat Anda pakai untuk belajar pemrograman web, khususnya PHP dan MySQL, paket ini dapat didownload secara gratis dan legas.”

Dibawah folder utama *xampp*, terdapat beberapa folder penting yang perlu diketahui. Penjelasan fungsinya sebagai berikut:

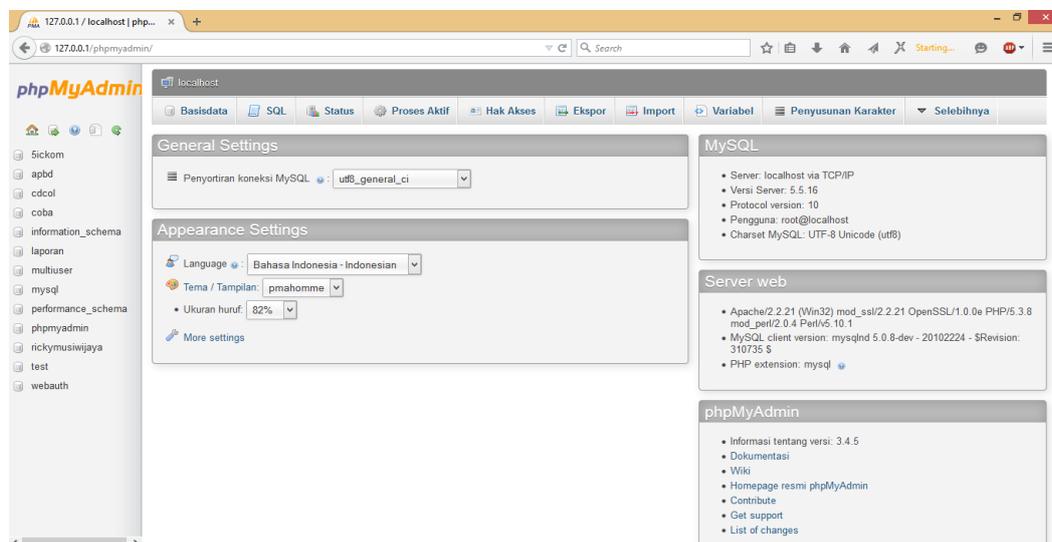
- *apache*: Folder utama dari *Apache Webserver*.
- *htdocs*: Folder utama untuk menyimpan data-data latihan *web*, baik PHP maupun HTML biasa. Pada folder ini, anda dapat membuat subfolder sendiri untuk mengelompokkan file latihannya. Semua

folder dan file program di htdocs bisa diakses dengan mengetikkan alamat <http://localhost/> di browser.

- Manual: Berisi subfolder yang di dalam terdapat manual program dan *database*, termasuk manual PHP dan MYSQL.
- mysql: Folder utama untuk *database* MYSQL server. Di dalamnya terdapat subfolder data (lengkapnya: C:\xampp\mysql\data) untuk merekam semua nama *database*, serta subfolder bin yang berisi *tools* klien dan server MYSQL.
- php: Folder utama untuk program PHP.

### 2.3.7 Pengertian phpMyAdmin

Nugroho (2013:71) menjelaskan, “phpMyAdmin adalah tools *MySQL Client* berlisensi *Freeware*, Anda dapat mengunduhnya secara gratis pada situs resminya di [www.phpmyadmin.net](http://www.phpmyadmin.net).”



**Gambar 2.3** Tampilan phpMyAdmin

### 2.3.8 Sekilas Tentang CSS

Betha dan Husni (2014 : 132) menjelaskan, “Secara umum disusun oleh tiga bagian yaitu, selector (elemen yang akan didefinisikan), property (atribut yang akan diubah) dan nilai sebagaimana berikut:

---

Selector {property : value}

Antara property dan nilai disahkan dengan titik-dua (colon) seperti contoh di bawah ini:

Body {color : black}

Jika nilai berupa beberapa kata, gunakan tanda petik ganda:

p{font - family : "sans serif"}

Jika lebih dari satu property pisahkan dengan titik-koma (semi colon):

p{text – align : center; color : red}

Jika ingin lebih mudah dibaca sebagaimana berikut:

```
p
{
text – align : center ;
color : black ;
font – family : arial
}
```

Jika selector dikelompokkan:

```
H1, H2, H3, H4, H5, H6
{
color : green ;
}
```

Jika menggunakan atribut class (dalam definisi CSS):

P.Kanan {text – align : right}

P.Kiri {text – align : center}

Berikut ini sintaks yang ditulis dalam dokumen html menggunakan definisi di atas:

<P class="kanan"> This Paragraph will be right aligned <p>

<P class="kiri"> This Paragraph will be center aligned <p>

Jika menggunakan ID atribut:

#Kanan {text – align : right}

---

### 2.3.9 Pengertian JavaScript

Betha dan Husni (2014:267) menjelaskan, “JavaScript merupakan modifikasi dari bahasa c++ dengan pola penulisan yang lebih sederhana. Interpreter bahasa ini sudah disediakan asp ataupun internet explorer.” Secara khusus beberapa hal yang penting dalam JavaScript adalah:

- Menggunakan blok awal “{“ dan blok akhir “}”
- Automatic conversion dalam pengoperasian tipe data yang berbeda
- Sensitive case, sehingga programmer java harus ekstra hati-hati dalam menggunakan nama variable, fungsi dan lain-lain
- Extension umumnya menggunakan “\*.js”
- Setiap statement dapat diakhiri dengan “;” sebagaimana halnya c++ tetapi dapat juga tidak
- Jika tidak didukung oleh browser versi lama, scriptnya dapat disembunyikan di antara tag “<!--” dan “-->”
- Jika program dalam satu baris terlalu panjang dapat disambung ke baris berikut dengan karakter

Betha dan Husni (2014 : 272) menjelaskan, “Tipe data pada JavaScript umumnya sama pada setiap mesin, hal ini diakibatkan desain awal java memang dikhususkan untuk bisa berjalan pada semua mesin.”

**Tabel 2.6.** Tipe data JavaScript

Primitive Type	Size	Minimum	Maximum	Version Type
Boolean	1 bit	-	-	
Char	16 bit	Unicode 0	Unicode 2E16-1	
Byte	8 bit	-128	+127	Versi 1.1 keatas
Short	16 bit	-2E-15	+2E15-1	Versi 1.1 keatas

---

Int	32 bit	-2E-31	+2E31-1	
Long	64 bit	-2E-63	+2E63-1	
Float	32 bit	IEEE754	IEEE754	
Double	64 bit	IEEE754	IEEE754	
Void	-	-	-	Versi 1.1 keatas