



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Teori Umum

##### 2.1.1. Pengertian Komputer

Tim EMS (2014:1), “Komputer adalah alat yang dipakai untuk mengolah data menggunakan sebuah program, program sendiri adalah prosedur yang dirumuskan.”

Kadir (2017:2), “Komputer merupakan peralatan elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia”.

Kesimpulannya, Komputer adalah sebuah peralatan elektronik yang digunakan untuk mengolah data dengan sebuah program yang bermanfaat untuk melaksanakan program.

##### 2.1.2. Pengertian Internet

Priyanto dan Jauhari (2014:1), “*Intenet* adalah jaringan global yang menghubungkan komputer-komputer seluruh dunia, dengan internet sebuah komputer bisa mengakses data yang terdapat pada komputer lain di benua yang berbeda.”

Sofana (2015:5), “*Internet* adalah interkoneksi jaringan komputer skala besar (miri WAN), yang dihubungkan menggunakan protokol khusus.”

Kesimpulannya, *Internet* adalah interkoneksi jaringan komputer global dengan skala besar sehingga dapat mengakses data yang dihubungkan menggunakan protokol khusus.

##### 2.1.3. Pengertian Perangkat Lunak

Rosa dan Shalahuddin (2018:02), “Perangkat Lunak (*Software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain dan cara penggunaan (*user manual*).”

Kadir (2017:2), “Perangkat lunak (*software*) adalah instruksi-instruksi yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai.”



Kesimpulannya, Perangkat Lunak adalah kumpulan beberapa intruksi-intruksi yang dieksekusi oleh mesin komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan.

#### **2.1.4. Pengertian Sistem**

Fathansyah (2018:12), “Sistem adalah sebuah tatanan (keterpaduan) yang terdiri dari atas sejumlah komponen fungsional (dengan satuan fungsi dan tugas khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses tertentu.”

Suryantara (2017:01), “Sistem terdiri atas komponen-komponen yang saling berhubungan satu sama lain dan bekerja sama untuk mencapai suatu tujuan.”

Kesimpulannya, sistem adalah sebuah tatanan (keterpaduan) yang terdiri dari komponen-komponen fungsional yang saling berhubungan satu sama lain dan bekerja sama untuk memenuhi dan mencapai suatu tujuan tertentu.

#### **2.1.5. Metode Pengembangan Sistem**

Rosa dan Shalahudin (2018:28) menjelaskan tentang metode pengembangan sistem yaitu *waterfall*. Metode air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup terurut mulai dari analisis, desain, pengkodean, pengujian dan pemeliharaan.

##### **a. Analisis Kebutuhan Perangkat Lunak**

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan sistem agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

##### **b. Desain**

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



### c. Pembuatan Kode Barang

Pada tahap pengkodean, desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

### d. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (error) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. Dalam penelitian ini pengujian sistem akan menggunakan pengujian Black-Box. Pengujian Black-box berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian metode ini memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

### e. Pendukung (*support*) dan pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah mengirimkan ke user. Perubahan bisa terjadi adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.

## 2.2. Teori Khusus

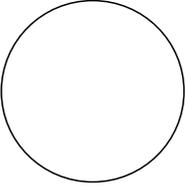
### 2.2.1 Pengertian DFD (*Data Flow Diagram*)

Rosa dan Shalahuddin (2018:70) menjelaskan, “*Data Flow Diagram* (DFD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*ouput*).DFD dapat digunakan untuk mempresentasikan sebuah system atau perangkat lunak pada beberapa level yang lebih detail untuk mempresentasikan aliran informasi atau fungsi yang lebih detail.DFD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi.



Adapun notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut :

**Tabel 2.1** Simbol-simbol *Data Flow Diagram*

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 :</p> <p>Nama yang diberikan pada sebuah proses biasanya berupa kata kerja</p>
	<p>File 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 harusnya 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 :</p> <p>Nama yang diberikan pada sebuah</p>



	penyimpanan biasanya kata benda
Notasi	Keterangan
	<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 sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan</p> <p>Catatan :</p> <p>Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda</p>
	<p>Aliran data: merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses masukan (<i>input</i>) atau keluaran (<i>output</i>)</p> <p>Catatan:</p> <p>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>

(sumber : Rosa dan Shalahuddin, 2018:71-72)



Rosa dan Shalahuddin (2018:72-73) menjelaskan, Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD:

1. Membuat DFD Level 0 (nol) atau sering disebut juga Context Diagram

DFD Level 0 (nol) menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD Level 0 (nol) digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

2. Membuat DFD Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. DFD Level 1 merupakan hasil *breakdown* DFD Level 0 (nol) yang sebelumnya sudah dibuat.

3. Membuat DFD Level 2

Modul-modul pada DFD Level 1 dapat di-breakdown menjadi DFD Level 2 (dua) Modul mana saja yang harus di-breakdown lebih 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 sistem, jumlah DFD Level 2 sama dengan jumlah modul ada DFD Level 1 (satu) yang di-breakdown.

4. Membuat DFD Level 3 dan seterusnya

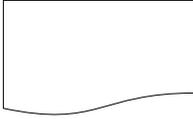
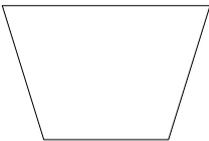
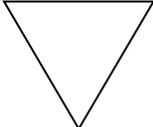
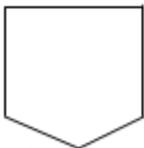
DFD Level 3,4,5 dan seterusnya merupakan breakdown dari modul pada DFD diatasnya. Breakdown pada level 3,4,5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.

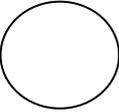
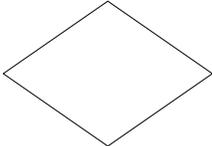
### **2.2.2. Pengertian *Block chart***

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

Adapun simbol-simbol yang sering digunakan dalam block chart dapat dilihat pada tabel berikut ini:

Tabel 2.2 Simbol-simbol *Block chart*

No.	Simbol	Arti
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/berkas atau cetakan.
2.		Multi Dokumen
3.		Proses Manual
4.		Proses yang dilakukan oleh komputer
5.		Menandakan dokumen yang diarsipkan (arsip manual)
6.		Data penyimpanan ( <i>data storage</i> )
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

No.	Simbol	Arti
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> ).
13.		Pemasukkan data secara manual.

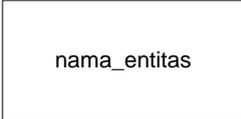
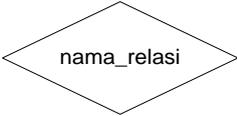
(Sumber : Kristanto, 2008:75-77)

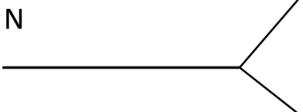
### 2.2.3. Pengertian ERD (*Entity Relational Diagram*)

Rosa dan Shalahuddin (2018:50) menjelaskan, “*Entity Relational Diagram* (ERD) digunakan untuk pemodelan basis data relasional. 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.

Adapun simbol-simbol yang digunakan pada ERD dengan notasi Chen adalah sebagai berikut:

**Tabel 2.3** Simbol-simbol *Entity Relational Diagram*

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

Simbol	Deskripsi
Asosiasi / <i>association</i>  	Penghubung antar 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. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B maka

(Sumber: Rosa dan Shalahuddin , 2018:50-51)

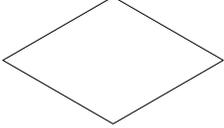
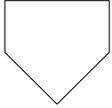
#### 2.2.4. Pengertian *Flowchart*

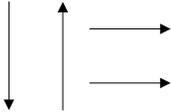
Lamhot (2015:14) menjelaskan, “*Flowchart* adalah langkah-langkah menyelesaikan masalah yang dituliskan dalam simbol-simbol tertentu.”

Menurut Lamhot (2015:14-16), Gambaran simbol-simbol *Flowchart* sebagai berikut:

**Tabel 2.4** Simbol-simbol *Flowchart*

No.	Simbol	Nama	Fungsi
1		<i>Terminal</i>	Menyatakan permulaan atau akhir suatu program.
2		<i>Input/output</i>	Menyatakan proses <i>input</i> atay output tanpa tergantung jenis peralatannya.
3		<i>Proccess</i>	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.

No.	Simbol	Nama	Fungsi
4		<i>Decision</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya atau tidak.
5		<i>Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.
6		<i>Offline Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.
7		<i>Predefined process</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
8		<i>Fanned Card</i>	Menyatakan <i>input</i> berasal dari kartu atau output data ke kartu.
9		<i>Punch Tape</i>	

No.	Simbol	Nama	Fungsi
10		<i>Document</i>	Mencetak keluaran dalam bentuk dokumen (melalui Printer)
11		<i>Flow</i>	Menyatakan jalannya arus suatu proses

(Sumber: Lamhot, 2015:14-15)

### 2.2.5. Pengertian Kamus Data

Rosa dan Shalahuddin (2018:73) menjelaskan, “Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*ouput*) dapat dipahami secara umum (memiliki standar cara penulisan).”

Kamus data memiliki beberapa simbol sebagai berikut :

**Tabel 2.5** Simbol-simbol Kamus Data

Simbol	Keterangan
=	Disusun atau terdiri dari
+	Dan
[]	Baik...atau...
{ }n	N kali diulang/bernilai banyak
()	Data opsional
*...*	Batas komentar

(Sumber: Rosa dan Shalahuddin, 2018:74)



### **2.3. Teori Judul**

#### **2.3.2. Pengertian Aplikasi**

Kamus Besar Bahasa Indonesia menjelaskan aplikasi adalah penerapan, penggunaan, lamaran, tambahan.

Indrajani (2018:3) menyatakan “Aplikasi adalah program yang menentukan aktivitas pemrosesan informasi yang dibutuhkan untuk penyelesaian tugas-tugas khusus dari pemakai komputer.”

Kesimpulannya, Aplikasi adalah suatu penerapan program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas tertentu.

#### **2.3.3. Pengertian Pengolahan**

Kamus Besar Bahasa Indonesia menjelaskan pengolahan adalah proses, perbuatan, cara mengolah;.

#### **2.3.4. Pengertian Data**

Kamus Besar Bahasa Indonesia menjelaskan data adalah keterangan yang benar dan nyata; bahan nyata yang dapat dijadikan sebagai dasar kajian (analisis dan kesimpulan).

Indrajani (2018:02) bahwa “Data adalah fakta atau observasi mentah yang biasanya mengenai fenomena atau transaksi bisnis.”

Kesimpulannya, Data adalah keterangan fakta atau observasi yang benar dan nyata mengenai fenomena yang dijadikan sebagai dasar kajian.

#### **2.3.5. Pengertian Pegawai**

Kamus Besar Bahasa Indonesia menjelaskan pegawai adalah karyawan, orang yang bekerja disuatu instansi dan mendapatkan upah (baik tetap maupun borongan, dsb): yang bekerja pada pekerjaan; alat, perkakas.

#### **2.3.6. Pengertian Aplikasi Pengolahan Data Absensi Pegawai pada PT Musi Banyuasin Indah (MBI) Sei-Selabu berbasis *website*.**

Aplikasi Pengolahan Data Pegawai pada PT Musi Banyuasin Indah (MBI) Sei-Selabu berbasis *website* adalah suatu aplikasi yang berisi informasi-informasi data pegawai yang dibawah naungan PT Musi Banyuasin Indah (MBI) Sei-Selabu dan akan diberikan kepada pegawai untuk menunjang kegiatan dan kinerja kepegawaian yang datanya terkoneksi pada *internet* sehingga memudahkan



melakukan pengolahan data pegawai yang ada pada PT Musi Banyuasin Indah (MBI) Sei-Selabu.

## 2.4. Teori Program

### 2.4.1 Pengertian Basis Data

Rosa dan Salahuddin (2018: 43) menyatakan “Basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.”

Kristanto (2018:79), “Basis data adalah kumpulan data, yang dapat digambarkan sebagai aktifitas dari satu atau lebih organisasi yang berelasi.”

Kesimpulannya, Basis data adalah media yang berisi kumpulan data yang disimpan dikomputer secara sistematis untuk memperoleh informasi dari organisasi yang berelasi.

#### 2.4.1.1 Istilah-istilah dalam Basis Data

Kadir (2017:103), menjelaskan beberapa istilah yang sering digunakan pada pembahasan *database* (basis data), yaitu sebagai berikut :

1. **DBMS**, yaitu sistem perangkat lunak yang menyediakan layanan bagi user untuk membuat, mengontrol dan mengakses database.
2. **Table**, merupakan kumpulan data yang diorganisasikan dalam baris (record) dan kolom (field). Dalam satu database biasanya terdiri dari beberapa tabel.
3. **Field**, merupakan kolom pada tabel yang memiliki ukuran dan tipe data tertentu.
4. **Record**, merupakan baris data yang memiliki nilai untuk masing-masing field pada tabel. Satu tabel biasanya terdiri dari banyak record.

#### 2.4.1.2 Atribut (*attribute*)

Atribut merupakan semua informasi yang berkaitan dengan entitas. Yanto (2016:34-38), menjelaskan macam-macam bentuk atribut :

1. **Atribut Sederhana** (*Simple Attribute*), adalah atribut yang nilainya tidak dapat dibagi lagi menjadi banyak atribut yang lebih kecil.
2. **Atribut Komposit** (*Composite Attribute*), adalah atribut gabungan yang nilainya dapat dipecah menjadi bagian yang lebih kecil. Atau sering disebut atribut yang terdiri dari beberapa atribut kecil di dalamnya.



3. **Atribut Bernilai tunggal** (*single Value Attribute*), adalah jenis atribut yang nilainya hanya satu dari suatu entitas.
4. **Atribut Bernilai Banyak** (*multivalued Attribute*), adalah jenis atribut yang nilainya lebih dari satu dalam suatu entitas tertentu.
5. **Atribut Turunan** (*Derived Attribute*), adalah jenis atribut yang nilainya diperoleh dari atribut lain.
6. **Atribut Identitas** (*Key Attribute*), adalah atribut yang dijadikan sebagai kunci pada suatu tabel. Sifat atribut identitas ini unik, tidak ada yang menyamai, atribut identitas terdiri dari beberapa jenis yaitu :
  - a. **Super Key**, adalah satu atribut atau kumpulan atribut yang secara unik mengidentifikasi sebuah baris di dalam relasi atau himpunan dari satu atau lebih entitas yang dapat digunakan untuk mengidentifikasi secara unik sebuah entitas dalam set entitas.
  - b. **Candidate Key**, adalah atribut yang menjadi determinan yang dapat dijadikan identitas baru pada sebuah relasi.
  - c. **Primary Key**, adalah kandidat key yang dipilih untuk mengidentifikasi baris data secara unik dalam relasi.
  - d. **Alternative Key**, adalah candidate key yang tidak terpilih sebagai primary key atau atribut untuk menggantikan kunci utama.
  - e. **Foreign Key**, adalah atribut dengan domain yang sama yang menjadi kunci utama sebuah relasi, tetapi pada relasi lain atribut tersebut sebagai atribut biasa.
  - f. **Composite Key**, adalah kunci yang terdiri dari dua atribut atau lebih.

#### 2.4.2 Pengertian XAMPP

Wahana Komputer (2014:72), “XAMPP merupakan singkatan dari X (empat sistem operasi apa pun), Apache, MySQL, PHP, dan Perl. XAMPP adalah tool yang menyediakan paket perangkat lunak dalam satu buah paket.”

Riyanto (2015:1) menyatakan “XAMPP merupakan paket PHP dan MySQL berbasis *open source*, yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis PHP. XAMPP mengombinasikan beberapa paket perangkat lunak berbeda ke dalam satu paket.”

Kesimpulannya, *XAMPP* adalah tool berbasis *open source* yang terdiri dari *Apache*, *MySQL*, *PHP* dan *Perl* dan membantu pengembangan aplikasi berbasis *PHP*.

### 2.4.3 Pengertian *MySQL*

Raharjo (2015:16), “*MySQL* merupakan software RDBMS (atau server database) yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user (multi-user), dan dapat melakukan suatu proses secara sinkron atau berbarengan (multi-threaded).”

Enterprise (2018: 2), “*MySQL* merupakan server yang digunakan untuk melayani database.”

Kesimpulannya, *MySQL* adalah sebuah software yang dapat mengelola dan melayani database dengan sangat cepat yang dapat diakses oleh banyak user.

### 2.4.4 Pengertian *PHP*

Raharjo (2016:38), “Bahasa Pemrograman *PHP* adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web*.”

Wahana Komputer (2014:33), “*PHP* merupakan bahasa berbentuk script yang ditempatkan di dalam server baru kemudian diproses. Kemudian hasil pemrosesan dikirimkan kepada web klien. Bahasa pemrograman ini dirancang untuk membentuk web dinamis.”

Kesimpulannya, *PHP* adalah salah satu bahasa pemrograman yang diproses dan dirancang untuk membuat aplikasi berbasis *web*.



**Gambar 2.1** Tampilan Logo *PHP*

#### 2.4.4.1 Sintaks Dasar *PHP*

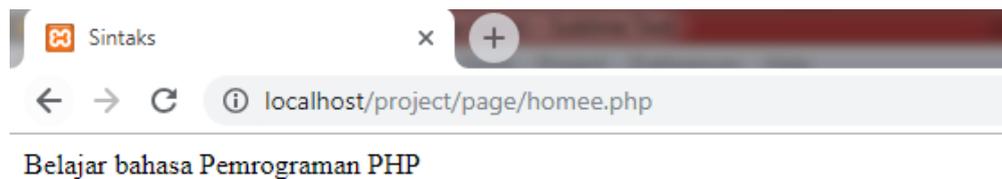
Kode (Script) *PHP* yang sering disebut dengan istilah *embedded script* yaitu script *PHP* yang disisipkan di antara script *HTML*. Jadi dapat dikatakan script *PHP* hanya ditulis atau disisipkan ketika dibutuhkan saja, seperti

menampilkan data dari *database* meng-*upload* file, *delete* data, edit data dan lain sebagainya. Contoh script :

```

<HTML>
  <HEAD>
    <TITLE>Sintaks</TITLE>
  </HEAD>
<BODY>
  <?php
    echo "Belajar bahasa Pemrograman PHP";
  ?>
</BODY>
</HTML>

```



**Gambar 2.2** Contoh Script PHP

#### 3.4.4.2 Tipe Data PHP

Tipe data PHP digunakan untuk menentukan jenis data yang akan disimpan dalam suatu variabel. Raharjo (2015:37-40), menjelaskan PHP mempunyai empat tipe data dasar sebagai berikut :

1. **Integer (Bilangan Bulat)** adalah tipe data yang merepresentasikan bilangan utuh atau bilangan bulat. Rentang nilai yang diperbolehkan pada tipe data integer adalah dari -2,147,483,648 sampai +2,147,483,647.
2. **Floating-point (Bilangan Riil)** adalah tipe data yang merepresentasikan nilai-nilai numerik dalam bentuk pecahan atau mengandung angka desimal



di belakang koma. Tipe *floating-point* berada dalam rentang 1.7E-308 dan 1.7E+308 dengan 15 digit akurasi.

3. **Karakter** merupakan tipe data yang digunakan untuk merepresentasikan data dengan nilai karakter tunggal, misal 'A', 'z', 'o', dan sebagainya. Karakter dalam PHP diapit dengan menggunakan tanda petik tunggal ('...').
4. **String** adalah tipe data yang merepresentasikan data yang berupa teks (kumpulan karakter). Penulisan tipe *string* diapit oleh petik ganda ("...") dan tunggal ('...').
5. **Boolean** merupakan tipe data yang digunakan untuk menyatakan nilai true (benar) atau false (salah) dan digunakan dalam proses pemilihan atau pengulangan statement, di bagian kondisi.
6. **Array** adalah variabel yang menyimpan sekelompok nilai, yang dapat diidentifikasi atau diakses berdasarkan posisinya atau dengan nama yang telah didefinisikan sebelumnya. Dalam PHP, indeks *array* dimulai dari 0 bukan 1.

#### 2.4.5 Pengertian JavaScript

Kadir (2017:122), "*JavaScript* adalah bahasa pemrograman berbentuk skrip yang dapat dipakai untuk mengakses elemen-elemen di dokumen *HTML*."

Wahana Komputer (2014:15), "*JavaScript* adalah bahasa pemrograman script pada browser, atau biasa disebut dengan istilah *client side programming*."

Kesimpulannya, *JavaScript* adalah bahasa pemrograman berbentuk skrip yang dapat dipakai untuk mengakses elemen dokumen html pada browser.

##### 2.4.5.1 Dasar JavaScript

Cara menggunakan *JavaScript* adalah dengan dimasukkan di antara kode *HTML* menggunakan tag `<script>` dan `</script>`. *Javascript* bisa diletakkan di tag `<body>` ataupun tag `<head>` dari kode *HTML*. Untuk memasukkan *javascript* anda harus menggunakan tag `<script>`, tag `<script>` dan `</script>` menentukan dimana *javascript* harus dimulai dan diakhiri.

Baris diantara tag `<script>` dan `</script>` ini berisi data *Javascript* contohnya seperti berikut :



```
<script>
```

```
Alert (“kode javascript pertama”);
```

```
</script>
```

#### **2.4.6. Pengertian Website**

Abdulloh (2016: 1), “*Website* atau disingkat *web*, dapat diartikan sekumpulan halaman yang terdiri atas beberapa laman yang berisi informasi dalam bentuk data digital baik berupa teks, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet.”

##### **2.4.6.1. Bahasa Pemrograman Website**

Abdulloh (2016: 1), “Bahasa pemrograman merupakan bahasa yang dapat dipahami oleh computer.”

Terdapat beberapa jenis bahasa pemrograman dalam membangun *website*, Abdulloh (2016: 2) menjelaskan ada 4 jenis bahasa pemrograman dalam membangun *website*, yaitu HTML, CSS, PHP dan Javascript.