



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Asropudin (2013:23), “Komputer adalah alat bantu pemrosesan data secara elektronik dan cara pemrosesan datanya berdasarkan urutan instruksi atau program yang tersimpan dalam memori masing-masing komputer.”

Hartono (2013:27), “Komputer adalah sebuah mesin yang dapat dikendalikan melalui perintah (*programmable machine*) yang dirancang secara otomatis melalukan serangkaian urutan perhitungan atau proses-proses yang diurutkan secara logis.

Kesimpulan, komputer adalah sebuah mesin atau alat bantu elektronik yang dikendalikan melalui perintah dan cara pemrosesan datanya berdasarkan urutan instruksi program yang tersimpan dalam masing-masing komputer.

2.1.2. Pengertian Basis Data (*Database*)

Raharjo (2011:3), “*database* didefinisikan sebagai kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil, dan dicari secara cepat.”

Risnandar et.al, (2013:90), “Basis data adalah kumpulan data yang tersimpan dalam tabel-tabel.”

Sujatmiko (2012:76), “Basis data adalah representasi kumpulan fakta yang saling berhubungan disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.”

Kesimpulan, basis data adalah kumpulan data yang terintegrasi dan tersimpan dalam tabel-tabel secara bersama sedemikian rupa tanpa pengulangan yang tidak perlu, untuk memenuhi berbagai kebutuhan.



2.1.3. Pengertian Aplikasi Web

Kadir (2009:2), “Aplikasi web adalah jenis aplikasi yang diakses melalui *browser*, biasanya Internet Explorer dan Mozilla Firefox .”

Sujatmiko (2012:23), “Aplikasi web adalah suatu aplikasi yang diakses menggunakan penjelajah web melalui suatu jaringan seperti internet atau intranet.”

Kesimpulan, aplikasi web adalah aplikasi yang di akses melalui browser melalui suatu jaringan internet atau intranet.

2.1.4. Pengertian Pengolahan Data

Kristanto (2008:8), “Pengolahan Data adalah waktu yang digunakan untuk menggambarkan perubahan bentuk data menjadi informasi yang memiliki kegunaan.”

2.2. Teori Khusus

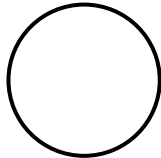


2.2.1. Pengertian DFD (*Data Flow Diagram*)

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

DFD dapat digunakan untuk merepresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. 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.

Sukamto et.al, (2014:71), menjelaskan notasi pada DFD adalah sebagai berikut :

Tabel 2.1. Simbol-simbol *Data Flow Diagram*

No	Simbol	Keterangan
1.		<p>Proses atau fungsi atau prosedur; pada permodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka permodelan 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>
2.		<p><i>File</i> atau basis data atau penyimpanan (<i>storage</i>); pada permodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka permodelan 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>
3.		<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan.</p> <p>catatan: nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) berupa kata benda.</p>

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

No	Simbol	Keterangan
4.		<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>

Sumber: Sukanto et.al (2014:71)

Kristanto (2008:68) menjelaskan beberapa petunjuk yang dapat dipakai dalam proses pembuatan simbol DFD. Petunjuk-petunjuk itu adalah sebagai berikut:

1. Penamaan yang jelas
 - a. Sebaiknya menggunakan nama yang mengacu pada fungsi, yaitu gabungan antara kata kerja yang spesifik.
 - b. Jangan menggunakan nama terlalu umum.
 - c. Gunakan nama yang mudah dimengerti oleh pemakai.
2. Memberi nomor pada proses
 - a. Nomor yang diberikan pada proses tidak harus menjadi nomor urut.
 - b. Penomoran dimaksudkan sebagai identifikasi proses dan memudahkan penurunan (level yang lebih rendah) ke proses berikutnya.
3. Penggambaran kembali
 - a. Ukuran dan bentuk lingkaran tetap sama.
 - b. Panah yang melengkung dan lurus tidak menjadi masalah.
4. Hindarilah proses yang mempunyai masukan tetapi tidak mempunyai keluaran begitu pula sebaliknya, hindarilah proses yang mempunyai keluaran tetapi tidak mempunyai masukan.



5. Hati-hati dengan aliran data dan proses yang tidak dinamai. Aliran proses yang tidak diberi nama dapat mengakibatkan elemen data yang saling tidak berhubungan akan menjadi satu.

Sukamto et.al, (2014:72) menjelaskan tentang tahapan-tahapan perancangan dengan menggunakan Data Flow Diagram yaitu:

1. Membuat DFD Level 0 atau sering disebut juga *Context Diagram*

DFD Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD Level 0 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 yang sebelumnya sudah dibuat.

3. Membuat DFD Level 2

Modul-modul pada DFD Level 1 dapat di-*breakdown* menjadi DFD Level 2. Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu di-*breakdown* lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 yang di-*breakdown*.

4. Membuat DFD Level 3 dan seterusnya

DFD 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 2.



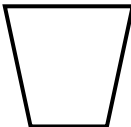

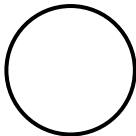

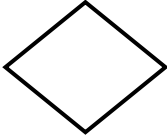

2.2.2. Pengertian *Blockchart*

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.”




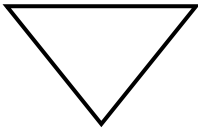
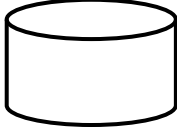

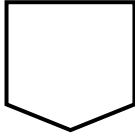
Kristanto (2008:75) juga menjelaskan tentang Simbol-simbol yang sering digunakan dalam *blockchart* dapat dilihat pada tabel berikut ini:

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.
4.		Proses dilakukan oleh komputer.
5.		Terminal yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.
6.		Terminal yang menandakan awal dan akhir dari suatu aliran.
7.		Pengambilan keputusan (<i>Decision</i>).
8.		Layar peraga (<i>monitor</i>).



Lanjutan Tabel 2.2 Simbol-simbol dalam *Blockchart*

No.	Simbol	Keterangan
9.		Pemasukkan data secara manual.
10.		Menandakan dokumen yang diarsipkan (arsip manual).
11.		Data penyimpanan (<i>Storage</i>).
12.		Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik.
13.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain.

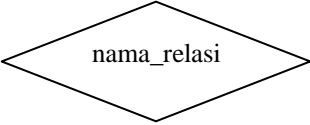
Sumber: Kristanto (2008:75)

2.2.3. Pengertian ERD (*Entity Rational Diagram*)

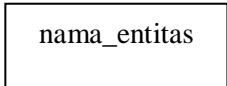
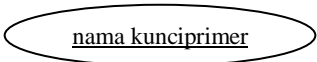
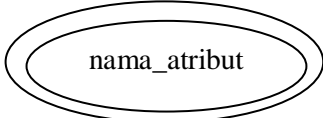
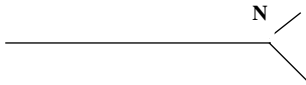
Sukamto et.al, (2013:50), “ERD adalah permodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk permodelan basis data relasional.”

Sukamto et.al, (2013:50), menjelaskan simbol-simbol yang digunakan dalam ERD, yaitu:

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

No.	Simbol	Keterangan
1.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.

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

No.	Simbol	Keterangan
2.	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 komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
3.	Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).
4.	Atribut multivalai / <i>multivalve</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5.	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. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan onetomany menghubungkan entitas A dan entitas B.

Sumber: Sukamto et.al (2014:50)

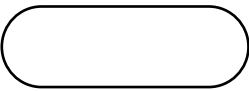




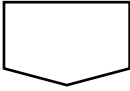

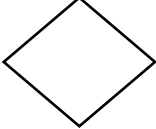


2.2.4. Pengertian *Flowchart*

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

Fungsi *Flowchart* adalah untuk mengetahui alur sistem yang akan dilakukan oleh *user* dalam hal yang bertindak sebagai *user* adalah admin unit kerja, manajer departemen, admin departemen akuntansi, admin departemen anggaran, admin departemen akuntansi dan administrator sehingga kita dapat mengetahui proses yang dilakukan admin unit kerja untuk memasukkan data permintaan kode aset tetap dan proses yang dilakukan admin departemen akuntansi, admin departemen anggaran dan manajer departemen akuntansi untuk mendapatkan data permintaan kode aset tetap pada aplikasi yang telah kami buat.

Tabel 2.4 Simbol-simbol data *Flowchart*

No.	Simbol	Keterangan
1.		Terminal menyatakan awal atau akhir dari suatu algoritma.
2.		Menyatakan proses.
3.		Proses yang terdefinisi atau sub program.
4.		Menyatakan masukan dan keluaran (<i>input/output</i>).
5.		Menyatakan penyambung ke simbol lain dalam satu halaman.
6.		Menyatakan penyambung ke halaman lainnya.
7.		Menyatakan pencetakan (dokumen) pada kertas.
9.		Menyatakan desicion (keputusan) yang digunakan untuk penyeleksian kondisi di dalam program.

Lanjutan Tabel 2.4 Simbol-simbol data *Flowchart*

No.	Simbol	Keterangan
10.		Menyatakan media penyimpanan drum magnetik.
11.		Menyatakan <i>input/output</i> menggunakan disket.
12.		Menyatakan operasi yang dilakukan secara manual.
13.		Menyatakan arah aliran pekerjaan (proses).
14.		Multidocument (banyak dokumen).
15.		Delay (penundaan atau kelambatan).

Sumber: Indrajani (2015:37)

2.2.5. Pengertian Kamus Data

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

Sukanto et.al, (2014:74), menjelaskan simbol-simbol yang di gunakan dalam kamus data, yaitu :

Tabel 2.5 Simbol-simbol dalam Kamus Data

No.	Simbol	Arti
1.	=	disusun atau terdiri atas
2.	+	Dan
3.	[]	baik ...atau...
4.	{ } ⁿ	n kali diulang/ bernilai banyak
5.	()	data operasional



Lanjutan Tabel 3.5 Simbol-simbol dalam Kamus Data

No.	Simbol	Arti
6.	*...*	batas komentar

Sumber: Sukamto et.al (2014:74)

2.2.6. Metode Pengembangan Sistem

Sukamto et.al, (2014:28), menjelaskan metode pengembangan sistem yang sering digunakan dalam tahapan pengembangan sistem yaitu model SDLC 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 perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*). Berikut ini tahapan-tahapan model air terjun :

a. Analisis kebutuhan perangkat lunak

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

b. Desain

Desain perangkat lunak adalah proses multi langkah yang focus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prsedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak dihasilkan pada tahap ini juga perlu didokumentasikan.

c. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program computer sesuai dengan desain yang telah dibuat pada tahap desain.



d. Pengujian

Pengujian focus pada perangkat lunak secara segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji.

e. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.3. Teori Judul

2.3.1. Pengertian Aplikasi

Sujatmiko (2012:23), “Aplikasi adalah program komputer yang dibuat oleh sebuah perusahaan komputer untuk membantu manusia dalam mengerjakan tugas-tugas tertentu.”

2.3.2. Pengertian Pengolahan Data

Kristanto (2008:8), “Pengolahan Data adalah waktu yang digunakan untuk menggambarkan perubahan bentuk data menjadi informasi yang memiliki kegunaan.”

2.3.3. Pengertian Permohonan

Menurut Kamus Besar Bahasa Indonesia (1990:251), “Permohonan adalah permintaan kepada orang yang lebih tinggi kedudukannya. Mohon adalah minta dengan hormat; berharap supaya mendapatkan sesuatu.”

2.3.4. Pengertian Perizinan

Menurut Kamus Besar Bahasa Indonesia (1990:239), “Perizinan adalah hal pemberian izin. Izin adalah pernyataan mengabdulkan, persetujuan membolehkan.



2.3.5. Pengertian Investasi

Menurut Kamus Besar Bahasa Indonesia (1990:142), “Investasi adalah penanaman uang atau modal di suatu perusahaan atau proyek untuk tujuan memperoleh keuntungan.”

2.3.6. Pengertian Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu Kabupaten Muara Enim

Dinas Penanaman Modal dan Perizinan Terpadu Satu Pintu Kabupaten Muara Enim (DPM-PTSP) merupakan lembaga teknis daerah yang mempunyai tugas melaksanakan urusan pemerintahan bidang penanaman modal yang menjadi kewenangan daerah dan tugas pembantuan

2.3.7. Pengertian Aplikasi Pengolahan Data Permohonan Perizinan Investasi Pada Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu Kabupaten Muara Enim

Aplikasi Pengolahan Data Permohonan Perizinan Investasi Pada Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu Kab. Muara Enim adalah suatu aplikasi yang dibuat untuk mempermudah calon pemohon izin investasi dalam hal pendaftaran perizinan online.

2.4. Teori Program

2.4.1. Pengertian HTML

Asropudin (2013:44), “*Hypertext Markup Language* adalah bahasa komputer yang digunakan untuk membuat sebuah halaman *web page* .”

Sujatmiko (2012:128), “*HTML (Hypertext Markup Language)* adalah bahasa komputer yang digunakan untuk membuat sebuah halaman *web page* atau program yang digunakan untuk menulis (membuat) halaman web di internet.”

Kesimpulannya, *HTML* adalah bahasa pemrograman untuk membuat halaman web yang berfungsi memberi perintah ke browser untuk melakukan manipulasi tampilan melalui *tag-tag* yang ditulis dalam *HTML*.



2.4.2. Pengertian MySQL

Raharjo (2011:21), “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*).”

2.4.3. Pengertian CSS (*Cascading Style Sheets*)

Risnandar et.al, (2013:39), “CSS (*Cascading Style Sheets*) adalah kumpulan aturan-aturan pemformatan yang mengontrol tampilan konten dalam sebuah halaman web.”

2.4.4. Pengertian PHP (*Hypertext Preprocessor*)

Risnandar et.al, (2013:57), “PHP (*Hypertext Preprocessing*) merupakan bahasa pemrograman yang biasa digunakan untuk membuat halaman HTML.”

Sujatmiko (2012:213), “PHP adalah bahasa pemrograman yang bisa kita gunakan untuk membuat aplikasi web.”

Kesimpulan, PHP adalah bahasa pemrograman yang digunakan untuk membuat aplikasi web.

2.4.4.1. Tipe Data PHP

Risnandar et.al, (2013:60), PHP mempunyai tipe data dasar yaitu:

1. Integer

Integer merupakan tipe data yang mencakup semua bilangan bulat. Range bilangan *integer* adalah antara -2.147.483.647 sampai dengan 2.147.483.647.

Contoh *integer* :

\$biaya = 1000;

\$a = 1234;

\$a = -123;



2. *FloatingPoint*

Floatingpoint merupakan tipe data yang mencakup semua bilangan desimal (bilangan yang memiliki angka dibelakang koma). Range bilangan *floatingpoint* antara $1e-308$ sampai dengan $1e308$ (10^{-308} s.d. 10^{308}).

Contoh *floatingpoint* :

```
$a = 1.234;
```

```
$a = 1.2e3;
```

3. *Character*

Character merupakan tipe yang digunakan untuk menyimpan data-data yang berupa karakter (1 huruf). Penulisan biasanya diapitnya dengannya tandanya kutipnya satunya ('...').

Contoh *character* :

```
$abjad = 'a';
```

```
$abjad = 'b';
```

Jika karakter-karakter berkumpul membentuk suatu kata, kalimat atau paragraf maka tipe tadi tidak bisa lagi disebut sebagai *character* tapi bertipe *string*. *String* merupakan tipe data tersendiri dan tidak dapat dikelompokkan menjadi tipe data dasar. Penulisan biasa diapit dengan tanda kutip 2 ("...").

Contoh *string* :

```
$nama = "gus";
```

```
$alamat = "Jalan Tanjung";
```

4. *Boolean*

Boolean menyimpan data yang bernilai logika (benar atau salah, 0 atau 1, ya atau tidak).

2.4.4.2. Skrip *PHP*

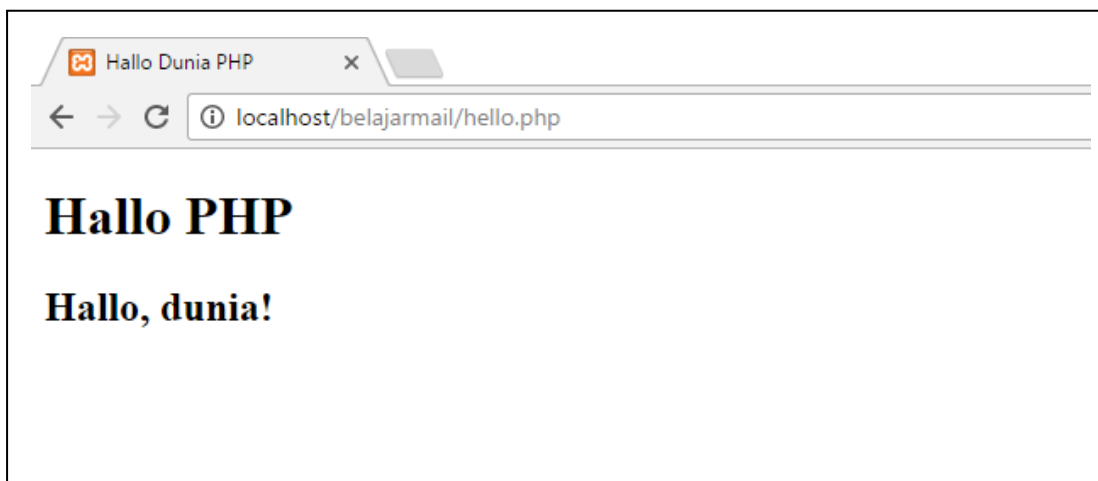
Winarno (2014:49), "*PHP* adalah bahasa pemrograman web berbasis server (server-side) yang mampu memarsing kode *PHP* dari kode web dengan ekstensi .php, sehingga menghasilkan tampilan website yang dinamis di sisi client (browser)."

```
<!DOCTYPE html>
```



```
<html>
<head>
  <title>Hallo Dunia PHP</title>
</head>
<body>
<h1>Hallo PHP</h1>
<?php
echo "<h2>Hallo, dunia!</h2>";
phpInfo();
?>
</body>
</html>
```

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



Gambar 2.1. Tampilan *PHP* sederhana

2.4.5. Hubungan *HTML*, *CSS*, *PHP*, dan *MySQL*

Ketika *user* akan mengakses data perusahaan maka *PHP* akan berusaha untuk mengambil data dari *MySQL* menggunakan salah satu fungsi yang ada di



PHP yaitu *mysqli_query()*. Setelah data diambil, maka *PHP* akan mengirimkan hasilnya ke *web server* yang berbasis *HTML* dengan menggunakan salah satu fungsi *PHP* yaitu *echo()*. Selanjutnya *HTML* bertugas untuk menampilkan halaman web ke browser sesuai kebutuhan menggunakan tag-tag *HTML* dan dipercantik oleh *CSS*.

2.4.6. *Javascript*

Wahana Komputer (2016:2), “*Javascript* adalah bahasa yang digunakan untuk membuat program yang digunakan supaya dokumen *HTML* yang ditampilkan dalam browser menjadi lebih interaktif, tidak sekedar indah saja.”

Winarno (2014:129), “*Javascript* adalah bahasa *scripting client side* yang digunakan untuk memberi efek pemrograman di halaman web.”

Kesimpulan, *Javascript* adalah bahasa *scripting client side* yang digunakan untuk memberi efek pemrograman di halaman web ditampilkan dalam browser menjadi lebih interaktif.”

2.4.6.1. Tipe Data, Variabel dan Operator *JavaScript*

Sidik (2011:31) menjelaskan variabel dan operator yang biasa digunakan *javascript*:

1. Tipe Data

Tipe data yang dapat diproses dalam *Javascript*, dibedakan menjadi :

a. *Numerik*

Data *numerik* adalah data berupa bilangan atau angka yang dapat dikenai operasi matematika, seperti penjumlahan, pengurangan, perkalian, atau pembagian.

b. *String*

Data *string* adalah semua data berupa huruf, tanda baca, angka, dan rangkaian huruf dan angka.

c. *Boolean*

Data *boolean* adalah data yang terdiri atau dua nilai, benar atau salah, *true* atau *false*.



d. *Date*

Data *date* adalah data khusus yang menyatakan tentang waktu – waktu tanggal dan jam.

e. *Objek*

Tipe data objek adalah tipe data yang mendefinisikan objek.

f. *RegExp*

Tipe data *RegExp* adalah tipe data yang digunakan untuk mendefinisikan pola atau *regular expression (regex)*.

g. *Null dan undefine*

Tipe data *null* dan *undefine* adalah tipe data khusus. *Null* digunakan untuk mendefinisikan data kosong (*null*), sedangkan *undefined* didefinisikan untuk data yang belum didefinisikan untuk data yang belum didefinisikan, bernilai *null*.

2. *Variabel*

Variabel adalah tempat untuk menyimpan data atau informasi yang akan diproses, berada dalam memori komputer.

a. *Variabel Tunggal*

Variabel tunggal adalah variabel yang digunakan untuk menyimpan satu data.

b. *Array*

Array adalah variabel jamak, yaitu satu nama variabel yang dapat memiliki banyak data; beberapa variabel yang diacu dengan satu nama.

c. *Lingkup dan Umur Variabel*

Tempat mendeklarasi suatu variabel akan menentukan lingkup atau ruang variabel itu dikenali dalam program.

3. *Operator*

Operator adalah alat yang digunakan untuk melakukan pemrosesan data.

a. *Operator Berdasar Jumlah Operand.*

Operator menurut jumlah *operand* yang dimanipulasinya :

- i. *Unary* adalah operator yang beroperasi ada *operand* tunggal.
- ii. *Binary operand* yang beroperasi untuk dua *operand*.



b. Operator Berdasar Tipe Data

1. Operator *Aritmatika*

Operator *aritmatika* digunakan untuk melakukan pemrosesan data numerik hasilnya adalah data numerik.

2. Operator *Assignment*

Operator *assignment* adalah operator yang digunakan untuk menyatakan suatu pernyataan perintah *assignment* dan ekspresi *aritmatika* dalam bentuk yang disingkat.

3. Operator Bit (*Bitwise Operator*)

4. Operator *String*

Operator *string* adalah operator yang digunakan untuk melakukan pemrosesan data *string* hasilnya berupa data *string*.

5. Operator Perbandingan

Operator perbandingan digunakan untuk melakukan perbandingan dua data, hasilnya berupa nilai *boolean* benar atau salah.

6. Operator Logika

Operator logika (*boolean*) digunakan untuk melakukan pemrosesan data *boolean* hasilnya adalah data *boolean*.