



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

##### 2.1.1 Pengertian Komputer

Menurut Sujatmiko (2012:70), “Komputer adalah mesin yang dapat mengolah data digital dengan mengikuti serangkaian perintah atau program”. Sedangkan, menurut Kadir (2013:2), “Komputer merupakan peralatan elektronik yang biasa dipakai orang untuk membantu pelaksanaan pekerjaan”.

Disimpulkan bahwa komputer adalah alat elektronik yang dapat mengolah data digital, dengan mengikuti serangkaian program yang digunakan untuk membantu pelaksanaan pekerjaan.

##### 2.1.2 Pengertian Perangkat Lunak

Menurut Rosa & Shalahuddin (2018:2), “Perangkat lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*)”. Sedangkan menurut Kadir (2013:2), “Perangkat lunak berupa instruksi-instruksi yang ditujukan kepada komputer dapat melaksanakan tugas sesuai dengan kehendak pemakai”.

Berdasarkan beberapa definisi perangkat lunak diatas, dapat disimpulkan bahwa perangkat lunak adalah instruksi-instruksi atau program komputer yang terasosiasi dengan dokumentasi perangkat lunak yang ditujukan untuk melaksanakan tugas sesuai pemakai.

##### 2.1.3 Pengertian Internet

Menurut Fauziah dalam bukunya Konsep Dasar Perancangan Web (2014:1), “Internet merupakan jaringan global yang menghubungkan suatu jaringan yang satu dengan jaringan yang lainnya di seluruh dunia”. Sedangkan, menurut Lado & Bimantara (2018:72), “Internet merupakan jaringan komputer yang saling terkoneksi. Tiap jaringan komputer dapat mencakup puluhan, ratusan bahkan ribuan komputer, dan memungkinkan mereka untuk berbagi informasi satu dengan yang lain dan untuk berbagi sumber-sumber daya komputerisasi seperti super komputer yang kuat dan *database* informasi”.

---



Berdasarkan beberapa definisi di atas maka dapat disimpulkan bahwa internet adalah jaringan komputer global yang saling terkoneksi di seluruh dunia.

#### **2.1.4 Pengertian Data**

Indrajani (2015:69) mengemukakan bahwa, “Data merupakan fakta mentah tentang orang, tempat, kejadian, dan apapun yang penting bagi perusahaan, di mana data itu sendiri tidak memiliki arti”. Sedangkan menurut Asropudin (2013:22) mengatakan bahwa, “Data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti”. Menurut Wangdra dan Husda (2017:13) “Data merupakan sesuatu yang belum mempunyai arti bagi penerimanya dan masih memerlukan adanya suatu pengolahan”.

Dari beberapa definisi di atas maka penulis menyimpulkan bahwa data merupakan kumpulan fakta yang tidak memiliki arti sehingga harus diolah agar dapat menghasilkan suatu informasi.

#### **2.1.5 Pengertian Basis Data (*Database*)**

Menurut Sujatmiko (2012:40), “Basis data (*database*) adalah kumpulan informasi yang disimpan didalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut”. Disamping itu, Kadir (2013:264) menyatakan bahwa *database* adalah suatu metode pengorganisasian data agar data dapat dimanipulasi, juga dapat diperoleh.

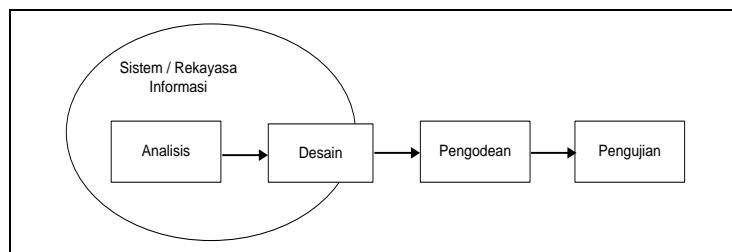
Berdasarkan pengertian diatas dapat disimpulkan bahwa basis data (*database*) adalah kumpulan data yang saling berhubungan, diorganisasi secara sistematis, menggunakan suatu program komputer agar dapat dimanipulasi dan diperoleh dengan cepat dan mudah.

## 2.2 Teori Khusus

### 2.2.1 Metode Pengembangan Sistem

Berdasarkan Rosa dan Shalahuddin (2018:28) dijelaskan bahwa model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linier*) 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 adalah gambar model air terjun :



**Gambar 2.1** Model *Waterfall*

#### 1. Analisis kebutuhan perangkat lunak

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

#### 2. 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.

#### 3. Pembuatan Kode Program

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.

#### 4. 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.

#### 5. Pendukung (*support*) atau Pemeliharaan (*maintenance*)

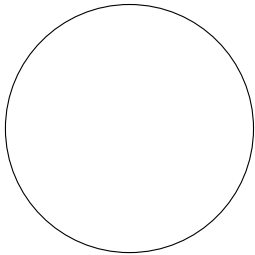
Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah mengirimkan ke user. Perubahan bisa terjadi adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.

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



Menurut Rosa dan Shalahuddin (2018:69), “*Data Flow Diagram* adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*)”.

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


**Tabel 2.1** Notasi *Data Flow Diagram* (DFD)

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>



Notasi	Keterangan
	<p>File basisdata atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemograman 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</i> (ERD), <i>Conceptual Data Model</i> (CDM), <i>Physical Data Model</i> (PDM)).</p> <p>Catatan :</p> <p>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 atau 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>



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>

(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 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 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 yang di-*breakdown*.

4. Membuat DFD Level 3 dan seterusnya



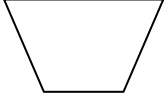

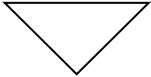
DFD Level 3,4,5 dan seterusnya merupakan *breakdown* dari modul pada DFD di atasnya. *Breakdown* pada level 3,4,5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.


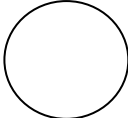
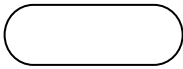
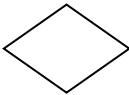


### 2.2.2 Pengertian *Blockchart*

Kristanto (2008:68), *block chart* 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.

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

**Tabel 2.2.** Simbol-Simbol *Block Chart*

Simbol	Keterangan
	Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan.
	Multi dokumen.
	Proses manual.
	Proses yang dilakukan oleh komputer.
	Menandakan dokumen yang diarsipkan (arsip manual).

Simbol	Keterangan
	Data penyimpanan ( <i>data storage</i> ).
	Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.
	Terminasi yang menandakan awal dan akhir dari suatu aliran.
	Pengambilan keputusan ( <i>decision</i> ).
	Layar peraga ( <i>monitor</i> ).
	Pemasukan data secara manual.

(Sumber : Kristanto, 2008:68-70)

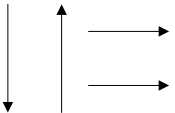

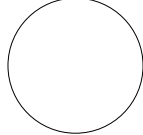
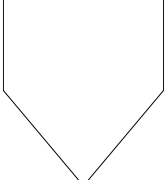

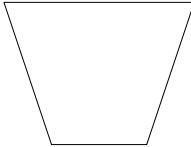
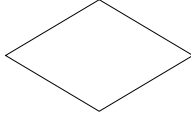
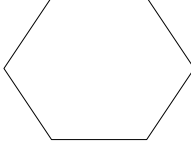
### 2.2.3 Pengertian *Flowchart*

Menurut Ladjamudin dalam Aulia (2018:14), "*Flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu masalah. Simbol yang menghubungkan antara simbol yang satu dengan sistem yang lain. Simbol ini disebut juga *connecting line*".



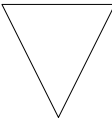
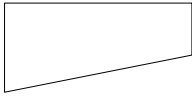
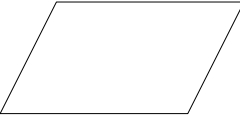

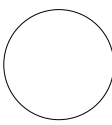
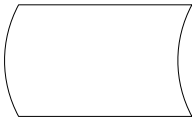

Adapun simbol-simbol *flowchart* sebagai berikut :



**Tabel 2.3** Simbol *Flowchart*

Simbol	Keterangan
	Simbol arus/flow Untuk menyatakan jalannya arus suatu proses
	Simbol Communication Link Untuk menyatakan bahwa adanya transisi suatu data/informasi dari satu lokasi ke lokasi lainnya.
	Simbol Connector Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang sama.
	Simbol Offline Connector Untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman/lembar yang berbeda.
	Simbol Offline Connector Untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman/lembar yang berbeda.
	Simbol Manual Untuk menyatakan suatu tindakan (proses) ke yang tidak dilakukan oleh komputer manual.
	Simbol decision / logika Untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban. ya / tidak.
	Simbol Predefined Proses Untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.



Simbol	Keterangan
	Simbol Terminal Untuk menyatakan permulaan atau akhir suatu program
	Simbol Keying Operating Untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i> .
	Simbol Off-line Storage Untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.
	Simbol Manual Input Untuk memasukkan data secara manual dengan menggunakan <i>online keyboard</i>
	Simbol Input-Output Untuk menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya.
	Simbol Punched Card Untuk menyatakan bahwa adanya transisi suatu data/informasi dari satu lokasi ke lokasi lainnya.
	Simbol Magnetic-tape unit Untuk menyatakan bahwa adanya transisi suatu data/informasi dari satu lokasi ke lokasi lainnya.
	Simbol Disk Storage Untuk menyatakan input berasal dari disk atau output disimpan ke disk
	Simbol Document Untuk mencetak laporan ke printer



Simbol	Keterangan
	<p>Simbol Display</p> <p>Untuk menyatakan peralatan output yang digunakan berupa layar (video, komputer)</p>

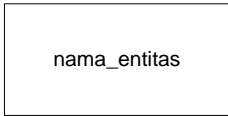
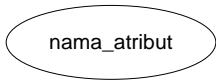
(Sumber: Aulia, 20018:14-15)

#### 2.2.4 Pengertian *Entity Relationship Diagram* (ERD)

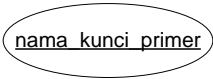
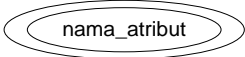
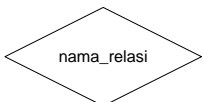

Rosa dan Shalahuddin (2018:50), *Entity Relationship Diagram* (ERD) digunakan untuk pemodelan basis data relasional. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot, dan beberapa notasi lain. Namun yang paling banyak digunakan adalah notasi dari Chen.

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

**Tabel 2.4** Simbol *Entity Relationship Diagram* (ERD)

Simbol	Deskripsi
<p>Entitas / <i>entity</i></p> 	<p>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.</p>
<p>Atribut</p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.</p>



Simbol	Deskripsi
Atribut kunci primer 	<i>Field</i> atau kolom data data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa <i>id</i> ; 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 dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas lain yang disebut kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B maka ERD biasanya memiliki hubungan <i>binary</i> (satu relasi menghubungkan dua buah entitas).

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



### 2.2.5 Pengertian Kamus Data

Menurut Rosa dan Shalahuddin (2018:73), “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 penulisan)”.

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

**Tabel 2.5** Simbol Kamus Data

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

(Sumber: Rosa dan Shalahuddin, 2018:74)

## 2.3 Teori Judul

### 2.3.1 Pengertian Aplikasi

Menurut Sujatmiko (2012:23), “Aplikasi adalah kumpulan beberapa perintah yang dieksekusi oleh mesin komputer dalam menjalankan pekerjaannya dan menyimpan perintah atau dokumen”. Sedangkan, menurut Sutabri (2014:10), “Program aplikasi adalah perangkat lunak yang dirancang khusus untuk kebutuhan tertentu, misalnya program pengolah kata, mengelola lembar kerja, program presentasi, desain grafis, dan lain-lain.”

Berdasarkan definisi-definisi diatas, dapat disimpulkan bahwa aplikasi adalah perangkat lunak atau kumpulan beberapa perintah yang dieksekusi mesin komputer untuk kebutuhan tertentu dalam menjalankan dan menyimpan perintah atau dokumen.



### 2.3.2 Pengertian *E-Pharmacy*

Menurut S.V. Chordiya dan B. M. Garge dalam jurnalnya *E-pharmacy vs conventional pharmacy* (2018:121), “*E-Pharmacy* adalah apotek yang beroperasi melalui internet dengan menjual obat-obatan dan mengirimkannya kepada pelanggan”.

### 2.3.3 Pengertian Apotek

Berdasarkan Peraturan Menteri Kesehatan Republik Indonesia Nomor 35 Tahun 2014 Tentang Standar Pelayanan Kefarmasian di Apotek, “Apotek adalah sarana pelayanan kefarmasian tempat dilakukan praktik kefarmasian oleh Apoteker”.

### 2.3.4 Pengertian *Web Mobile*

Menurut Usmento *et al.* (2018: 34), “*Website mobile* adalah situs yang dirancang khusus untuk perangkat *mobile* yang dirancang menggunakan standar protokol yang sama dengan *desktop web*”.

### 2.3.5 Pengertian Aplikasi *E-Pharmacy* Berbasis *Web Mobile* pada Apotek Siguntang Palembang

Aplikasi *E-Pharmacy* Berbasis *Web Mobile* pada Apotek Siguntang Palembang merupakan suatu perangkat lunak berupa *website* penjualan obat online di Apotek Siguntang Palembang yang beroperasi pada *browser* perangkat *mobile*.

## 2.4 Teori Program

### 2.4.1 Pengertian PHP



**Gambar 2.2** Logo PHP

Menurut Badiyanto (2013:32), “PHP (*Hypertext Preprocessor*) adalah bahasa skrip yang dapat ditanamkan ke dalam HTML/PHP banyak dipakai untuk membua situs *web* dinamis”. Sedangkan menurut Winarno dkk (2014:49), “PHP (*Hypertext Preprocessor*), adalah sebuah 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)”.

Berdasarkan pendapat para ahli diatas dapat disimpulkan bahwa, PHP adalah bahasa pemrograman berbasis *server-side* yang bisa kita gunakan untuk membuat aplikasi *web* yang ditanamkan kedalam HTML, yang dijalankan di dalam sebuah *webserver*, dan juga bisa menghasilkan tampilan *website* yang dinamis.

Contoh Script :

```
<HTML>
  <HEAD>
    <TITLE>Contoh Awal</TITLE>
  </HEAD>
  <BODY>
    <?php
      echo “Belajar bahasa Pemrograman PHP”;
    ?>
  </BODY>
</HTML>
```

## 2.4.2 Pengertian MySQL



Gambar 2.3 Logo MySQL

Badiyanto (2013:57), “*Database* MySQL merupakan sebuah *database server SQL multiuser* dan *multi-threaed*. MySQL dibuat oleh TcX dan telah dipercaya mengelola sistem dengan 40 buah *database* berisi 10,000 tabel dan 500 diantaranya memiliki 7 juta baris”. Sedangkan menurut Winarno dkk (2014:102), “MySQL adalah sebuah *software database*. MySQL merupakan tipe data



relasional yang artinya MySQL menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan”.

Berdasarkan pendapat para ahli diatas dapat disimpulkan bahwa, MySQL adalah bahasa yang digunakan untuk mengelola data pada RDBMS yang bertipe data relasional.

### 2.4.3 Pengertian XAMPP



Gambar 2.4 Logo XAMPP

Madcoms (2009:1), “XAMPP adalah salah satu paket *software web server* yang terdiri dari Apache, MySQL, PHP dan phpMyAdmin”. Sedangkan menurut Nugroho (2013:1), “XAMPP adalah paket program *web* lengkap yang dapat anda pakai untuk belajar pemograman web, khususnya PHP dan MySQL, paket ini dapat didownload secara gratis dan lengas”.

Berdasarkan pendapat para ahli diatas dapat disimpulkan bahwa, XAMPP adalah suatu *software web server* yang terdiri dari *Apache* (server), MySQL (database), dan PHP digunakan untuk mengakses fungsi yang ada dalam halaman *website* tersebut agar bisa diakses oleh *user*.

### 2.4.4 Pengertian HTML

Asropudin (2013:44), “HTML (*Hypertext Markup Language*) adalah bahasa komputer yang digunakan untuk membuat sebuah halaman *webpage*”. Sedangkan menurut Winarno dkk (2014:1), “HTML (*Hypertext Markup Language*) Adalah sebuah bahasa untuk menampilkan konten di *web*”.

Berdasarkan pendapat para ahli diatas dapat disimpulkan bahwa, HTML merupakan salah satu format yang digunakan aplikasi untuk menampilkan konten di *web*.





### 2.4.5 Pengertian JavaScript



**Gambar 2.5** Logo JavaScript

Menurut Winarno dkk (2014:129), “JavaScript adalah bahasa *scripting client side* yang sangat populer karena javascript bisa dipakai di HTML, *web*, untuk *server*, PC, laptop, tablet, ponsel dan lainnya serta menjadi dasar yang bisa digunakan untuk teknologi lainnya seperti Ajax, jQuery dan *jQuery Mobile*”.

### 2.4.6 Pengertian CSS

Menurut Madcoms (2012:54), “*Cascading Style Sheet (CSS)* adalah kumpulan kode yang digunakan untuk mengendalikan tampilan isi suatu halaman *web*”. Sedangkan menurut Abdulloh (2016:2), “CSS singkatan dari *Cascading Style Sheets*, yaitu skrip yang digunakan untuk mengatur desain *website*”.

Kesimpulannya, adalah CSS merupakan suatu kumpulan kode atau skrip yang digunakan untuk mengendalikan tampilan atau mengatur isi maupun desain suatu *website*.