



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Kadir (2017:02), “Komputer merupakan alat elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia.”

Paramytha (2016:06), “Komputer adalah alat hitung dengan konstruksi elektronik, yang dapat menerima *input* dan mengolah data, memberikan informasi, mempunyai *internal storage*, (untuk menyimpan program dan data yang diolah) dan bekerja secara otomatis dengan bantuan operating sistem menurut program yang disimpan didalamnya.”

Dari kedua kutipan diatas dapat disimpulkan bahwa komputer adalah alat elektronik yang dapat mengolah data digital dengan mengikuti serangkaian perintah atau program untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia.

2.1.2 Pengertian Internet

Abdulloh (2016:01), “*Internet* adalah jaringan yang digunakan untuk mengirim informasi kepada *website*.”

Sofana (2015:05), “*Internet* adalah interkoneksi jaringan komputer skala besar (mirip WAN), yang dihubungkan menggunakan protokol khusus.”

Dari kedua kutipan diatas dapat disimpulkan bahwa *Internet* adalah jaringan *global* yang menghubungkan berjuta-juta komputer di seluruh dunia melalui jalur telepon kabel maupun satelit yang terhubung tanpa batas waktu maupun tempat.



2.1.3 Pengertian Perangkat Lunak

Kadir (2017:02), “Perangkat lunak adalah instruksi-instruksi yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai. Sistem operasi seperti Windows, Mac OS, dan Linux, dan aplikasi seperti Microsoft Word dan Microsoft Excel adalah contoh 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*).”

Dari kedua kutipan di atas dapat disimpulkan bahwa perangkat lunak adalah instruksi-instruksi yang ditujukan kepada program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain dan cara penggunaan (*user manual*). agar dapat melaksanakan tugas sesuai kehendak pemakai.

2.1.4 Pengertian Sistem

Rosa dan Shalahudin (2018:02), “Sistem berarti kumpulan komponen yang saling terkait dan mempunyai satu tujuan yang ingin dicapai.”

Pratama (2014:07), “Sistem adalah sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama.”

Dari kedua kutipan di atas dapat disimpulkan bahwa sistem merupakan kumpulan dari prosedur-prosedur yang saling berhubungan untuk menyelesaikan satu tujuan yang ingin dicapai.

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 Tahapan Perangkat Lunak

Rosa dan Shalahuddin (2018:11) menjelaskan, “Proses perangkat lunak (*software process*) adalah sekumpulan aktifitas yang memiliki tujuan untuk mengembangkan atau mengubah perangkat lunak. Secara umum proses perangkat lunak terdiri dari:

1. Pengumpulan Spesifikasi (*Specification*)

Mengetahui apa saja yang harus dapat dikerjakan sistem perangkat lunak dan batasan pengembangan perangkat lunak.

2. Pengembangan (*Development*)

Pengembangan perangkat lunak untuk menghasilkan sistem perangkat lunak.

3. Validasi (*Validation*)

Memeriksa apakah perangkat lunak sudah memenuhi keinginan pelanggan (*customer*).

4. Evolusi (*Evolution*)

Mengubah perangkat lunak untuk memenuhi perubahan kebutuhan pelanggan (*Customer*).”

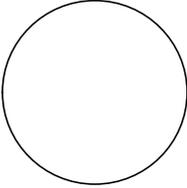
2.2.2 Tahapan Desain

Berikut ini merupakan tahapan desain aplikasi meliputi *Data Flow Diagram*, *Flowchart*, *Blockchart*, dan *Entity Relationship Diagram*.

2.2.2.1 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 (*output*). 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 : 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 : 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 memakain/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>) biasanya berupa kata benda</p> |

Lanjutan Tabel 2.1 Simbol-simbol *Data Flow Diagram*



| Notasi | Keterangan |
|---|---|
|  | <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: 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



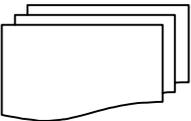
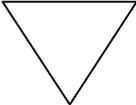
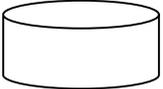
DFD Level 3,4,5 dan seterusnya merupakan breakdown dari modul pada DFD di atasnya. Breakdown pada level 3,4,5, dan seterusnya.

2.2.2.2 Pengertian *Blockchart*

Kristanto (2018:75) 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.

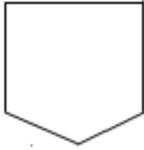
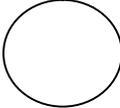
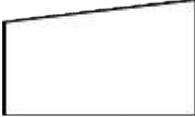
Adapun simbol-simbol yang sering digunakan dalam *block chart* dapat dilihat pada tabel berikut ini:

Tabel 2.2 Simbol-simbol *Block chart*

| Simbol | Arti |
|---|--|
|  | Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/berkas atau cetakan. |
|  | Multi Dokumen |
|  | Proses Manual |
|  | Proses yang dilakukan oleh komputer |
|  | Menandakan dokumen yang diarsipkan (arsip manual) |
|  | Data penyimpanan (<i>data storage</i>) |
|  | Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik |

Lanjutan Tabel 2.2 Simbol-simbol *Block chart*



| Simbol | Arti |
|---|---|
|  | Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain |
|  | 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>). |
|  | Pemasukkan data secara manual. |

(Sumber : Kristanto, 2018:75-77)

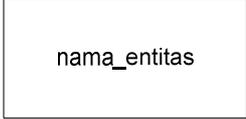
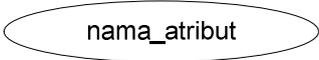
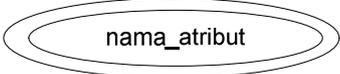
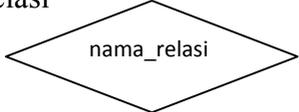
2.2.2.3 Pengertian ERD (*Entity Relational Diagram*)

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 |
| Asosiasi / <i>association</i> N  | 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. |

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

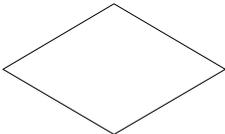
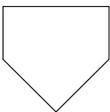
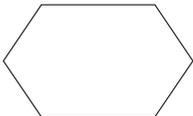
2.2.2.4 Pengertian *Flowchart*



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:

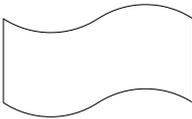
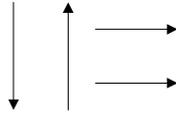
Tabel 2.4 Simbol-simbol *Flowchart*

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

Lanjutan Tabel 2.4 Simbol-simbol *Flowchart*

| No. | Simbol | Nama | Fungsi |
|-----|--------|------|--------|
|-----|--------|------|--------|



| | | | |
|----|---|--------------------|---|
| 8 |  | <i>Fanned Card</i> | Menyatakan <i>input</i> berasal dari kartu atau output data ke kartu. |
| 9 |  | <i>Punch Tape</i> | |
| 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)

2.2.2.5 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



2.3.1 Pengertian Aplikasi

Indrajani (2018:03) menyatakan “Aplikasi adalah program yang menentukan aktivitas pemrosesan informasi yang dibutuhkan untuk penyelesaian tugas-tugas khusus dari pemakai komputer.”

Sujatmiko (2012:23) menyatakan bahwa “Aplikasi adalah program komputer yang dibuat oleh suatu perusahaan komputer untuk membantu manusia dalam mengerjakan tugas tertentu, misalnya Ms-Word, Ms-Excel.”

Jadi berdasarkan pengertian diatas dapat disimpulkan bahwa aplikasi adalah program yang dikembangkan dalam menentukan aktivitas pemrosesan informasi yang dibutuhkan untuk penyelesaian tugas-tugas khusus dari pemakai komputer.

2.3.2 Pengertian Rekap

Waskito (2017:431), “Rekap adalah ringkasan isi atau ikhtisar pada akhir laporan atau akhir hitungan.”

2.3.3 Pengertian *Monitoring*

Pramukti dan Chahyaningsih (2018:14), “Segala usaha dan kegiatan untuk mengetahui dan menilai kenyataan yang sebenarnya mengenai pelaksanaan tugas dan kegiatan, apakah sesuai dengan yang semestinya atau tidak.”

2.3.4 Pengertian Hasil Kerja

Veltia (2015:03), “Hasil kerja adalah objek berwujud atau tak berwujud yang merupakan hasil pelaksanaan proyek, sebagai bagian dari suatu kewajiban atau obligasi.”

2.3.5 Pengertian Karyawan

Waskito (2017:183), “Karyawan adalah orang yang bekerja pada suatu lembaga dengan mendapat upah.”

2.3.6 Pengertian Harian



Waskito (2017:136), “Waktu dari pagi sampai kepada pagi lagi, yaitu satu putaran bumi pada sumbunya.”

2.3.7 Pengertian Lepas

Waskito (2017:192), “Lepas adalah bebas dari ikatan.”

2.3.8 Pengertian Aplikasi Rekap dan *Monitoring* Hasil Kerja Karyawan Harian Lepas pada PT Sawit Selatan Divisi IV Kebun Limau Kesturi Berbasis *Android*

Aplikasi Rekap dan *Monitoring* Hasil Kerja Karyawan Harian Lepas pada PT Sawit Selatan Divisi IV Kebun Limau Kesturi Berbasis *Android* adalah suatu aplikasi yang berisi informasi-informasi hasil kerja karyawan harian lepas yang dibawah naungan PT Sawit Selatan Divisi IV Kebun Limau Kesturi dan akan digunakan oleh mandor, kerani, dan manajer kebun untuk menunjang kegiatan pendistribusian informasi yang datanya terkoneksi pada *internet* sehingga penyebaran informasi hasil kerja karyawan harian lepas dapat lebih cepat dan efisien.

2.4 Teori Program

2.4.1 Pengertian Basis Data

Indrajani (2018:02), “Basis data adalah kumpulan terpadu dari data logis yang saling berhubungan. Basis data mengonsolidasi banyak catatan yang sebelumnya disimpan dalam file terpisah.”

Rosa dan Shalahuddin (2018:43), “Basis data adalah sistem komputerisasi yang tujuan utamanya adalah memelihara data yang sudah ada yang diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.”

Dari kedua kutipan diatas dapat disimpulkan bahwa Basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis dan saling berhubungan satu dengan yang lain yang tujuan utamanya adalah memelihara data yang sudah ada yang diolah.



2.4.2 Pengertian XAMPP

Enterprise (2018:03), “*XAMPP* merupakan server yang paling banyak digunakan untuk keperluan belajar PHP secara mandiri, terutama bagi programmer pemula.”

Nugroho (2013:01), “*Xampp* adalah program web lengkap yang dapat Anda pakai untuk belajar pemrograman web, khususnya *PHP* dan *MySQL*, paket ini dapat didownload secara gratis dan legal.”

Dari kedua kutipan diatas dapat disimpulkan bahwa *XAMPP* adalah suatu program 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*.

2.4.3 Pengertian MySQL

Enterprise (2018:02), “*MySQL* merupakan server yang melayani database. Untuk membuat dan mengolah database, kita dapat mempelajari pemrograman khusus yang disebut *query* (perintah) *SQL*.”

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*).”

Dari kedua kutipan diatas dapat disimpulkan bahwa *MySQL* adalah bahasa yang digunakan untuk mengelola data pada *RDBMS* bersifat *open source* sehingga banyak digunakan di dunia.

2.4.4 Pengertian PHP

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

Subagia (2018:01), “*PHP* merupakan sebuah bahasa pemrograman yang berjalan dalam sebuah web server (server side).”

Enterprise (2018:01), “*PHP* merupakan bahasa pemrograman yang digunakan untuk membuat website dinamis dan interaktif.”

Dari ketiga kutipan diatas dapat disimpulkan bahwa *PHP* adalah bahasa pemrograman berbasis *server-side* yang bisa kita gunakan untuk membuat aplikasi



web yang disisipkan pada HTML, yang dijalankan di *server*, dan juga bisa digunakan untuk membuat aplikasi dekstop.



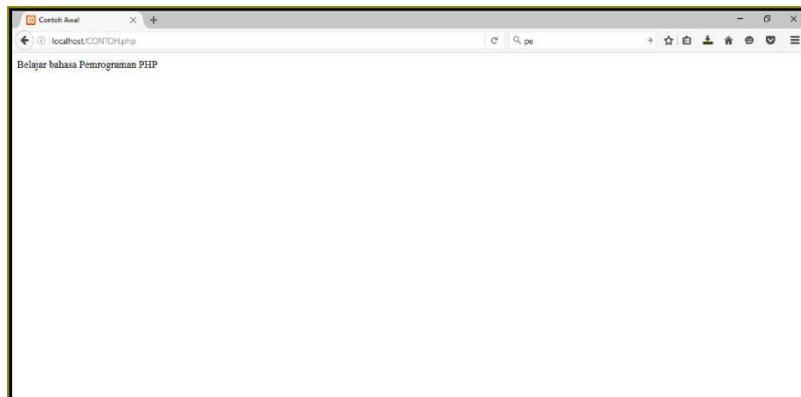
Gambar 2.1 Tampilan Logo PHP

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



Gambar 2.2 Contoh Script PHP

2.4.4.2 Tipe Data PHP

Winarno (2014:133) mengatakan bahwa variabel pada PHP dapat menyimpan berbagai macam tipe data. Beda tipe data dapat melakukan hal yang berbeda pula. Berikut tipe data yang didukung oleh PHP :

1. **String** yaitu tipe data berupa teks atau angka yang ditulis di antara petik ganda, seperti: “Selamat pagi!” atau “12345”.
2. **Integer**, yaitu tipe data numerik non desimal antara -2, 147, 483, 648 hingga 2, 147, 483, 647.
3. **Float**, yaitu tipe data numerik berupa angka desimal. Tanda koma pada angka desimal menggunakan titik, misal: 23.25.
4. **Boolean**, yaitu tipe data yang hanya dapat bernilai *true* atau *false*.
5. **Array**, yaitu tipe data yang menyimpan banyak nilai dalam satu variabel.
6. **Object**, yaitu tipe data yang menyimpan data beserta informasi bagaimana data diproses.

2.4.5 Pengertian JavaScript

Winarno (2014:129) mengatakan bahwa JavaScript adalah bahasa scripting client site yang sangat populer. JavaScript digunakan untuk memberikan efek pemrograman di halaman. JavaScript juga menjadi dasar yang bisa digunakan teknologi lainnya, seperti Ajax, jQuery dan jQuery Mobile.



Abdulloh (2016:03) menyatakan bahwa peran javascript dalam membangun website adalah memberikan efek animasi yang menarik dan interaktivitas dalam penanganan event yang dilakukan oleh pengguna website.

Dari kedua kutipan diatas dapat disimpulkan bahwa JavaScript adalah bahasa scripting client site yang digunakan untuk membuat program yang digunakan agar dokumen HTML yang ditampilkan dalam browser menjadi lebih interaktif.

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 Android

Safaat(2015:01), “*Android* adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem informasi, *middleware*, dan aplikasi.”

Wahana (2012:01), “*Android* adalah sistem operasi berbasis Linux untuk telepon seluler seperti telepon pintar atau komputer tablet. Sebagai aplikasi terbuka, *Android* adalah platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang bisa dipakai oleh berbagai macam peranti bergerak.”



Dari kedua kutipan diatas dapat disimpulkan bahwa *Android* adalah sistem operasi berbasis *Linux* yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer *tablet*.