



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Teori Umum

##### 2.1.1. Pengertian Aplikasi

Asropudin (2013:6), aplikasi adalah software yang di buat oleh suatu perusahaan computer untuk mengerjakan tugas-tugas tertentu seperti *microsoft word* dan *microsoft excel*.

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

Kamus Besar Bahasa Indonesia Pusat Bahasa (2012:51),“aplikasi adalah program komputer atau perangkat lunak yang didesain untuk mengerjakan tugas tertentu”.

##### 2.1.2. Pengertian Data

Sutabri (2012:2), data adalah bahan mentah yang di proses untuk menyajikan informasi.

Asropudin (2013:22), data merupakan representasi dunia nyata yang mewakili suatu objek seperti manusia, hewan, peristiwa, konsep, keadaan, dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya.

Kristanto (2011:7), data adalah penggambaran dari sesuatu dan kejadian yang kita hadapi.

Sutabri (2012:1), data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.

##### 2.1.3. Pengertian Pengolahan Data

Kristanto (2011:8), pengolahan data adalah waktu yang digunakan untuk menggambarkan perubahan bentuk data menjadi informasi yang memiliki kegunaan.



#### **2.1.4. Pengertian Pegawai**

Kamus Besar Bahasa Indonesia mengemukakan bahwa, pegawai adalah orang yang bekerja pada pemerintahan (perusahaan, dan sebagainya).

#### **2.1.5. Pengertian Aplikasi Pengolahan Data Pegawai Pada Badan Kepegawaian Daerah Provinsi Sumatera Selatan**

Aplikasi Pengolahan Data Pegawai pada Badan Kepegawaian Daerah Provinsi Sumatera Selatan adalah suatu aplikasi yang mengelolah data pegawai yang terdapat pada Badan Kepegawaian Daerah Provinsi Sumatera Selatan yang dilakukan secara terkomputerisasi.

### **2.2. Teori Khusus**

#### **2.2.1. Metode Pengembangan Sistem**

Sukamto dan Shalahuddin (2014:28), menjelaskan metode pengembangan sistem yang sering digunakan dalam tahapan pengembangan sistem yaitu model SDLC air terjun (*waterfall*) sering juga disebut model *sekuensial* atau alur hidup klasik (*classic life cycle*) dengan tahapan sebagai berikut:

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 *user*.

b. Desain

Desain perangkat lunak adalah proses multi langkah yang focus pada desain pembuatan orogram perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean.

c. Pembuatan kode program

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

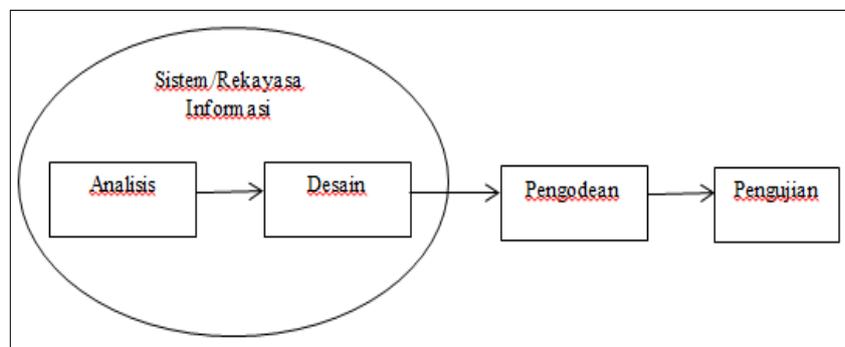


d. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logika 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 lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak yang baru.



**Gambar 2.1.** Gambar Ilustrasi Model *Waterfall*

(Sumber: Sukamto dan Shalahuddin, 2014:29)

### 2.2.2. Pengertian DataBase

Kadir (2013:411), database adalah kumpulan data yang disimpan dalam bentuk yang mudah diakses oleh pemakai.

sutabri (2012:47), Database adalah sekumpulan data yang saling berhubungan secara logis dan terorganisir dengan baik. Basis data merupakan salah satu komponen utama pendukung program aplikasi.

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

Supardi (2013:5) menjelaskan bahwa, “DFD sering juga disebut DAD (Diagram Aliran Data) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*inpu*) dan keluaran (*output*).”

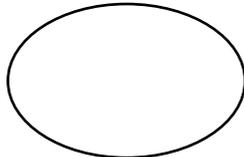


Kristanto (2011:55) menjelaskan bahwa, “DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem.”

Sukanto dan Shalahuddin (2014:288) menjelaskan bahwa, “DFD atau Diagram Alir Data (DAD) adalah representasi grafik untuk perancangan pemrograman terstruktur yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).”

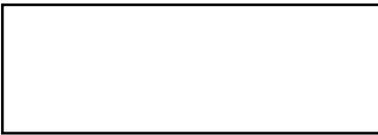
Menurut Supardi (2013:8), menjelaskan tentang notasi atau simbol pada DFD adalah sebagai berikut:

**Tabel 2.1.** Simbol-simbol *Data Flow Diagram* (DFD)

NO.	Simbol	Keterangan
1.		<b>Proses</b> ( <i>Process</i> ) atau <b>fungsi</b> ( <i>Function</i> ) atau <b>prosedur</b> ( <i>Procedure</i> ). Pada pemrograman terstruktur, notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program. Catatan: Nama proses biasanya berupa kata kerja.
2.		<b>File</b> atau <b>basis data</b> atau <b>penyimpanan</b> ( <i>storage</i> ) pada pemrograman terstruktur. Notasi inilah yang harusnya menjadi tabel-tabel basis data yang dibutuhkan. Tabel-tabel ini harus sesuai dengan ERD ( <i>Entity Relationship Diagram</i> ), CDM ( <i>Conceptual Data Model</i> ), PDM ( <i>Physical Data Model</i> ). Catatan: Nama penyimpanan biasanya berupa kata benda.



**Lanjutan Tabel 2.1.** Simbol-simbol *Data Flow Diagram* (DFD)

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>
4.		<p><b>Aliran data</b> merupakan data yang dikirim antar-proses 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.</p> <p>Misalnya “data Siswa” atau tanpa kata data, misalnya “Siswa”.</p>

(Sumber: Supardi, 2013:8)

Supardi (2013:9) menjelaskan tentang tahapan-tahapan perancangan dengan menggunakan *Data Flow Diagram* yaitu:

1. Membuat DFD level 0 atau sering juga disebut CD (*Context Diagram*)
2. Membuat DFD level 1 merupakan hasil uraian (breakdown) DFD level 0 yang sebelumnya sudah dibuat, dan seterusnya.

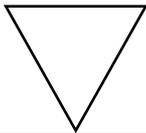
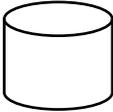
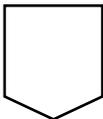
#### 2.2.4. Pengertian *Block Chart*

Kristanto (2011:68) menjelaskan bahwa, “*Block chart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *block chart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.”

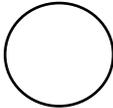
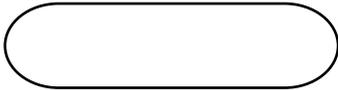
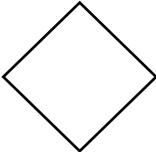


Kristanto (2011:68), menjelaskan simbol-simbol yang sering digunakan dalam block chart dapat dilihat pada tabel berikut ini:

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

No.	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/benda/berkas/cetakan.
2.		Multi dokumen
3.		Proses manual
4.		Proses yang dilakukan oleh computer
5.		Menandakan dokumen yang diarsipkan (arsip manual)
6.		Data penyimpanan (data storage)
7.		Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik.
8.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain

Lanjutan Tabel 2.2. Simbol-simbol *Block Chart*

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 (decision)
12.		Layar peraga (monitor)
13.		Pemasukan data secara manual

(Sumber: Kristanto, 2011:68)

### 2.2.5. Pengertian *Flow Chart*

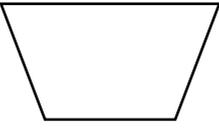
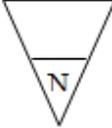
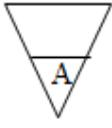
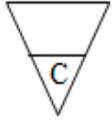
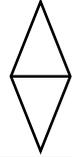
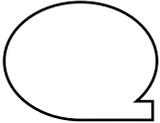
Supardi (2013:51) menjelaskan bahwa, “*Flowchart* merupakan Diagram Alur yang sering digunakan sistem analis dalam membuat atau menggambarkan logika program.”

Menurut Supardi (2013:53), menjelaskan tentang beberapa simbol yang digunakan dalam *flowchart* yaitu sebagai berikut:

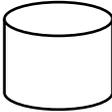
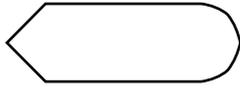
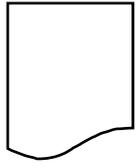
Tabel 2.3. Simbol-simbol *Flowchart*

No.	Simbol	Keterangan
1.		Simbol <b>dokumen</b> yang menunjukkan dokumen input dan <i>output</i> baik untuk proses manual, mekanik atau komputer.

Lanjutan Tabel 2.3. *Flowchart*

2.		Simbol <b>kegiatan manual</b> yang menunjukkan pekerjaan manual.
3.		Simbol <b>simpanan offline</b> yang berupa file non komputer yang diarsip urut angka (numerical)
4.		Simbol <b>simpanan offline</b> yang berupa file non-komputer yang diarsip urut huruf (alphabetical).
5.		Simbol <b>simpanan offline</b> yang berupa file non-komputer yang diarsip urut tanggal ( <i>chronological</i> ).
6.		Simbol <b>kartu plong</b> yang menunjukkan <i>input/output</i> yang menggunakan kartu plong ( <i>punched card</i> ).
7.		Simbol <b>proses</b> yang menunjukkan kegiatan proses dari operasi program komputer.
8.		Simbol <b>operasi luar luar</b> yang menunjukkan operasi yang dilakukan di luar porses operasi komputer.
9.		Simbol <b>pengurutan offline</b> yang menunjukkan proses pengurutan data dari luar proses komputer.
10.		Simbol <b>pita magnetik</b> yang menunjukkan bahwa <i>input/output</i> menggunakan pita magnetik.

Lanjutan Tabel 2.3. Simbol-simbol *Flowchart*

11.		Simbol <b>hard disk</b> yang menunjukkan <i>input/output</i> menggunakan <i>hard disk</i> .
12.		Simbol diskette yang menunjukkan <i>input/output</i> menggunakan diskette.
13.		Simbol drum magnetik yang menunjukkan bahwa <i>input/output</i> menggunakan drum magnetik.
14.		Simbol <b>pita kertas terhubung</b> yang menunjukkan bahwa <i>input/output</i> menggunakan pita kertas terhubung.
15.		Simbol <b>keyboard</b> yang menunjukkan <i>input/output</i> menggunakan <i>on-line keyboard</i> .
16.		Simbol <b>display</b> yang menunjukkan <i>output</i> yang ditampilkan di monitor.
17.		Simbol <b>pita kontrol</b> yang menunjukkan penggunaan pita kontrol ( <i>control tape</i> ) dalam <i>batch control total</i> untuk pencocokan di proses <i>batch processing</i> .
18.		Simbol <b>hubungan komunikasi</b> yang menunjukkan proses transmisi data melalui <i>channel</i> komunikasi.
19.		Simbol <b>penjelasan</b> yang menunjukkan penjelasan dari suatu proses.

Lanjutan Tabel 2.3. Simbol-simbol *Flowchart*

20.		Simbol <b>penghubung</b> yang menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain.
21.		Simbol input/output yang digunakan untuk mewakili data input/output.
22.		Simbol garis alir yang digunakan untuk menunjukkan arus dari proses
23.		Simbol <b>keputusan</b> ( <i>decision symbol</i> ) yang digunakan untuk suatu penyeleksian kondisi di dalam program
24.		Simbol <b>proses terdefinisi</b> yang digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain.
25.		Simbol <b>persiapan</b> yang digunakan untuk memberi nilai awal suatu besaran.
26.		Simbol <b>titik terminal</b> yang digunakan untuk menunjukkan awal dan akhir dari suatu proses.

(Sumber: Supardi 2013:53)

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

Supardi (2013:13), “ERD merupakan model konseptual yang mendeskripsikan hubungan antar-penyimpanan (dalam DFD).“

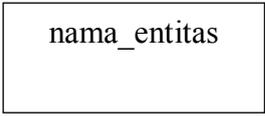
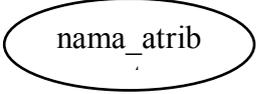
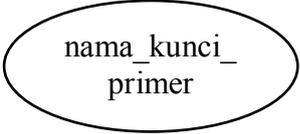
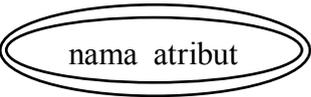
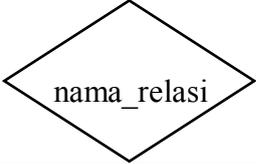
Ladjamudin (2013:142), “ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam istem yang abstrak.”



Sukamto dan Shalahuddin (2014:289), “*Entity Relationship Diagram* (ERD) adalah pemodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basis data relasional.”

Menurut Sukamto dan Shalahuddin (2014:50), menjelaskan tentang beberapa simbol-simbol pada *Entity Relational Diagram* (ERD) yaitu sebagai berikut:

**Tabel 2.4.** Simbol-simbol pada *Entity Relational Diagram*

no.	Simbol	Keterangan
1.	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
2.	Atribut 	Field atau kolom data yang butuh disimpan dalam suatu entitas
3.	Atribut kunci primer 	Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record 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>multivalued</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5.	Relasi 	Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.

**Lanjutan Tabel 2.4.** Simbol-simbol pada *Entity Relational Diagram*

6.	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 disebut kardinalitas.
----	---	--

(Sumber: Sukamto dan Shalahuddin 2014:50)

### 2.2.7. Kamus Data

Sukamto dan Shalahuddin (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). Kamus data biasanya berisi:

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

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan. Simbol–simbol yang ada dalam kamus data adalah sebagai berikut:

**Tabel 2.5** Simbol–simbol dalam kamus data

No.	Simbol	Arti
1	=	Disusun atau terdiri atas
2	+	Dan
3	[   ]	Baik ...atau...
4	{ } <sub>n</sub>	n kali diulang/ bernilai banyak
5	( )	Data opsional
6	*...*	Batas komentar

Sumber: Sukamto dan Shalahuddin (2013:74)



## 2.3 Teori Program

### 2.3.1 Pengenalan PHP

Nugroho (2013:153) PHP (*PHP : Hypertext Preprocessor*) adalah bahasa program yang digunakan untuk membuat aplikasi berbasis *web* (*website*, *blog*, atau aplikasi *web*).

Dalam membuat *website* ataupun aplikasi berbasis *web*, bukan hanya kode PHP saja yang dibutuhkan, tapi juga akan menggunakan kode **HTML** (*Hyper Text Markup Language*) untuk desain tampilan, yaitu untuk mengatur teks, tabel dan juga membuat form. Selain itu akan dibutuhkan juga **CSS** (*Cascading Style Sheets*) sebagai kode pemanis *web*, juga bisa jadi pengganti HTML.

Macdoms (2010:341), PHP (*PHP Hypertext Preprocessor*) adalah bahasa pemrograman yang bekerja dalam sebuah *webserver*. *Script-script* PHP dibuat harus tersimpan dalam sebuah server dan dieksekusi atau diproses dalam server tersebut. Dengan menggunakan program PHP, sebuah *website* akan lebih interaktif dan dinamis.

#### 2.3.1.1 Skrip Dasar PHP

Dalam penulisannya, skrip PHP tidak harus berdiri sendiri namun dapat disisipkan diantara kode HTML. Skrip PHP harus selalu diawali dengan `<? atau <?php` dan diakhiri dengan `?>`.

Contoh penulisan skrip PHP sebagai berikut:

```
<?
..... [tempat penulisan skrip PHP]
?>
```

Atau

```
<?php
..... [tempat penulisan skrip PHP]
?>
```

Sedangkan untuk menampilkan ke dalam browser, digunakan fungsi `echo"data";`



**<?php**

**Echo “laporan akhir”;**

**?>**

Semua teks yang diketik setelah tanda buka skrip (<?) dan tanda tutup skrip (?>) akan dieksekusi sebagai suatu skrip PHP. Anda dapat membuat keterangan atau komentar di dalam skrip PHP dan komentar tersebut tidak akan dieksekusi sebagai sebuah skrip.

Beberapa cara untuk memberi keterangan di dalam skrip PHP adalah:

- a) Gunakan tag /\* dan diakhiri tag \*/ apabila jumlah keterangan lebih dari 1 baris.
- b) Gunakan tag //, tag ini digunakan untuk keterangan yang hanya terdiri dari satu baris saja.
- c) Gunakan tag #, tag ini juga digunakan untuk 1 baris komentar saja.

Dalam penulisannya, baris perintah skrip PHP selalu diakhiri dengan menuliskan tanda titik koma (;).

### 2.3.1.2. Fungsi-Fungsi PHP

Beberapa fungsi-fungsi dasar PHP yang sering digunakan adalah sebagai berikut:

#### Fungsi Date

PHP menyediakan fungsi date untuk menampilkan tanggal sekarang. Bentuk fungsi : date(format penulisan tanggal); Di bawah ini adalah keterangan format penulisan tanggal:

**Tabel 2.6** Format Penulisan Tanggal

Argumen	Hasil
D	Tanggal dengan format 2 digit. Contoh: 01-31
D	Nama hari dengan panjang 3 karakter. Contoh: Sun, Mon, Tue.
F	Nama bulan dengan format lengkap. Contoh: January
M	Nama bulan dengan format 2 digit. Contoh: 01
M	Nama bulan dengan format 3 karakter. Contoh: “Jan”.
N	Nama bulan tanpa 0 di depan. Contoh: 1




---

**Lanjutan Tabel 2.6** Format Penulisan Tanggal

Y	Tahun dalam format 4 digit. Contoh: 2008
Y	Tahun dalam format 2 digit. Contoh: 08

Sumber: Madcoms (2010:357)

### Fungsi Time

Fungsi ini digunakan untuk membaca data waktu dalam hitungan detik dari tanggal 1 Januari 1970 jam 00:00:00 sampai waktu skrip ini diakses atau dijalankan. Bentuk fungsi : `time()`;

### Fungsi Include

Fungsi ini digunakan untuk memanggil file yang sudah dibuat. Bentuk fungsi : `include("nama_file.php")`;

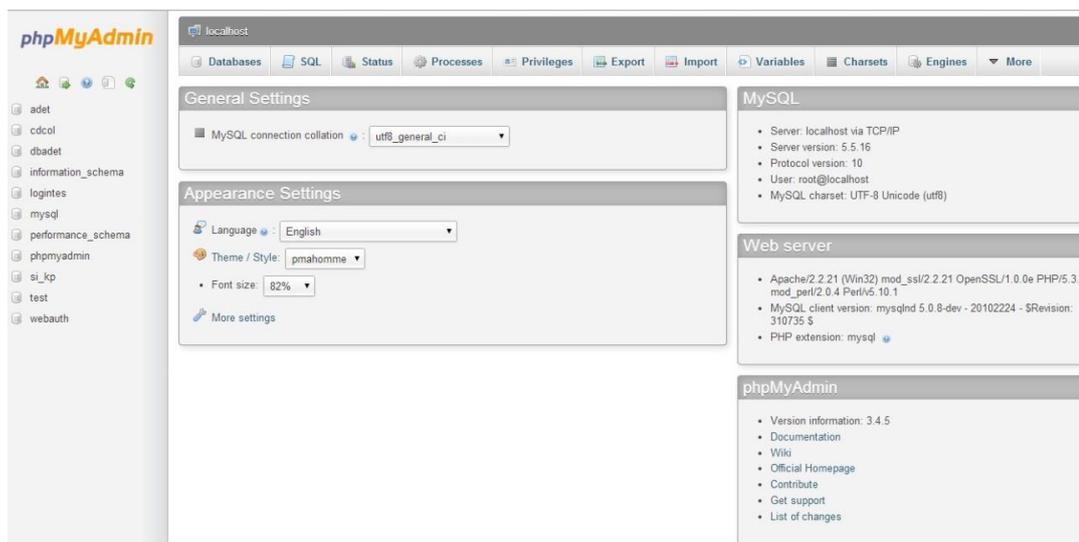
### 2.3.2 Apache

Nugroho (2013:1), Apache adalah folder utama dari websrever.

### 2.3.3 PHPMyAdmin

Nugroho (2013:71), *PhpMyAdmin* merupakan sebuah program bebas yang berbasis web yang dibuat menggunakan aplikasi *PHP*, *tools MySQL Client* berlisensi *Freeware*, anda dapat mengunduhnya secara gratis pada situs resminya di [www.phpmyadmin.net](http://www.phpmyadmin.net).

*PhpMyAdmin* dapat digunakan dengan mudah untuk memanajemen *database MySQL* secara visual, dan *Server MySQL*, sehingga tidak lagi menulis *query SQL* setiap ingin melakukan perintah operasi *database*. *PhpMyAdmin* harus dijalankan di sisi server *web* (misalnya: *Apache web server*) dan pada komputer harus tersedia *PHP*, karena berbasis *web*.



**Gambar 2.2 PHPMYAdmin**

### 2.3.4. Pengenalan *MySQL*

#### 2.3.4.1. Pengertian *MySQL*

Komputer (2014:73) menjelaskan bahwa, “*MySQL* merupakan sistem manajemen database yang bersifat *relational*. Artinya, data yang dikelola dalam database akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan menjadi jauh lebih cepat. *MySQL* dapat digunakan untuk mengelola database mulai dari yang kecil sampai dengan yang sangat besar.”