



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

##### 2.1.1 Pengertian Komputer

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

Irma (2016:1), “Komputer adalah sekumpulan alat elektronik yang saling bekerja sama, dapat menerima data (*input*), mengolah data (*proses*) dan memberikan informasi (*output*) serta terkoordinasi di bawah kontrol program yang tersimpan di memorinya.”

Jadi, dapat penulis simpulkan bahwa pengertian komputer adalah sekumpulan alat elektronik yang saling bekerja sama, dapat menerima data (*input*), mengolah data (*proses*) dan memberikan informasi (*output*) untuk melaksanakan berbagai pekerjaan yang di lakukan oleh manusia.

##### 2.1.2 Pengertian Perangkat Lunak

Utami dan Feri (2015:2), “Software adalah perangkat lunak yang berisikan sebuah instruksi yang di perintahkan dan di proses dengan bantuan perangkat keras sehingga tanpa perangkat lunak, perangkat keras tidak bisa di pakai sehingga *software* dan *hardware* tidak bisa dipisahkan.”

Rosa dan 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*).”

Jadi, dapat penulis simpulkan bahwa pengertian perangkat lunak adalah kumpulan beberapa instruksi yang di perintahkan dan diproses dengan bantuan mesin komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan.

##### 2.1.3 Basis Data

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



Yanto (2016:11), “Basis data adalah kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan membuat informasi tersedia untuk memenuhi berbagai kebutuhan”.

Jadi, dapat penulis simpulkan bahwa pengertian basis data adalah kumpulan data yang sudah ada yang di olah sedemikian rupa dan membuat informasi teredia untuk memenuhi berbagai kebutuhan”.

## **2.2 Teori Judul**

### **2.2.1 Pengertian Aplikasi**

Sholicin (2016:1), menjelaskan bahwa “Aplikasi atau perangkat lunak (*software*) merupakan bagian yang tidak terpisahkan dari suatu sistem komputer, disamping keberadaan pengguna (*brainware*), perangkat keras (*hardware*) dan jaringan (*networking*).”

Indrajani (2018:3) menjelaskan bahwa “Aplikasi adalah program yang menentukan aktivitas pemrosesan informasi yang di butuhkan untuk penyelesaian tugas-tugas khusus dari pemakaian komputer”.

### **2.2.2 Pengertian Kepangkatan**

Zamzam (2015:34), mengatakan bahwa bahwa “Kepangkatan merupakan penghargaan pemerintah atas prestasi pegawai”.

Jadi dapat penulis simpulkan bahwa kepangkatan adalah penghargaan pemerintah atas prestasi pegawai dan tingkat-tingkat pada pangkat yang dimiliki oleh para Pegawai Negeri Sipil yang secara berkala dan berjenjang akan meningkat setiap 4 (empat) tahun sekali. Dan khusus bagi pegawai fungsional, pangkatnya dapat naik setiap 2(dua) tahun sekali”.

### **2.2.3 Pengertian Jaksa**

Mulanro (2007:254) dalam undang-undang Republik Indonesia No 16 Tahun 2004 pada pasal 1 ayat 1 berbunyi “Jaksa adalah pejabat fungsional yang di beri wewenang oleh undang-undang untuk bertindak sebagai penuntut umum dan pelaksanaan putusan pengadilan yang telah memperoleh kekuatan hukum tetap serta wewenang lain berdasarkan undang-undang”.



## 2.2.4 Pengertian Aplikasi Kepangkatan Jaksa pada Kejaksaan Tinggi Sumatera Selatan

Aplikasi Kepangkatan Jaksa pada Kejaksaan Tinggi Sumatera Selatan adalah sebuah Aplikasi yang mengelola data daftar urut kepangkatan jaksa untuk mempermudah dalam penyimpanan dan pencarian data pada Kejaksaan Tinggi Sumatera Selatan.

## 2.3 Teori Khusus

### 2.3.1 Kamus Data

Maniah dan Dini H. (2017:59) mengungkapkan mengenai pengertian kamus data yaitu “Sebagai katalog data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi.

**Tabel 2.1** simbol kamus data:

No	Simbol	Uraian
1	=	Terdiri dari, mendefinisikan, di uraikan menjadi
2	+	Dan
3	()	Menunjukkan suatu elemen yang bersifat pilihan (opsional). Elemen-elemen yang bersifat pilihan ini bisa dikosongkan pada layar masukan atau bisa juga dengan memuat spasi atau nol untuk field-field numeric pada struktur file.
4	{ }	Menunjukkan elemen-elemen repetitive, juga di sebut kelompok berulang atau tabel-tabel. Kemungkinan bisa ada satu atau beberapa elemen berulang di dalam kelompok tersebut. Kelompok berulang bisa mengandung keadaan-keadaan tertentu, seperti misalnya, jumlah pengulangan yang pasti atau batas tertinggi dan batas terendah untuk jumlah pengulangan.

**Lanjutan Tabel 2.1** simbol kamus data:

No	Simbol	Uraian
5	[ ]	Menunjukkan salah satu dari dua situasi tertentu. Satu elemen bisa ada sedangkan elemen lainnya juga ada, tetapi tidak bisa kedua duanya ada secara bersamaan. Elemen-elemen yang ada di dalam tanda kurung ini saling terpisah satu sama lain. (dengan kata lain, memilih salah satu dari sejumlah alternatif, seleksi)
6		Pemisah sejumlah alternatif pilihan antara symbol [ ]
7	@	Identifikasi atribut kunci
8	**	Komentar

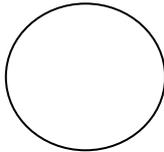
Sumber: Maniah dan Dini H, 2017:59

### 2.3.2 Data Flow Diagram (DFD)

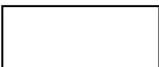
Rosa dan Shalahuddin (2018:68), menjelaskan bahwa, “Diagram Alir Data adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang di aplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).

Rosa dan Shalahuddin (2018:69), menjelaskan 4 (empat) simbol DFD adalah sebagai berikut :

**Tabel 2.2** Simbol-Simbol *Data Flow Diagram* (DFD)

No	Notasi	Nama Simbol	Keterangan
1		Proses	pada pemodelan perangkat lunak yang akan di implementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program. Catatan: Kata Kerja.

Lanjutan Tabel 2.2 Simbol-Simbol *Data Flow Diagram* (DFD)

No	Notasi	Nama Simbol	Keterangan
2		Data Store	<p>File atau basis data atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan di implementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang di butuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (<i>Entity realitionship Diagram (ERD)</i>, <i>Conceptual Data Model (CDM)</i>, <i>Physical Data Model (PDM)</i>)</p> <p>Catatan: Nama yang di berikan pada sebuah penyimpanan biasanya kata benda</p>
3		Entitas Eksternal	<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 di modelkan atau sistem lain yang terkait dengan aliran data dari sistem yang di modelkan</p> <p>Catatan: Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda.</p>

Lanjutan Tabel 2.2 Simbol-Simbol *Data Flow Diagram* (DFD)

No	Notasi	Nama Simbol	Keterangan
4		Aliran Data	<p>Merupakan data yang di kirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>)</p> <p>Catatan: Nama yang di gunakan 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, 2018:68-69

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 di kembangkan . DFD level 1 merupakan hasil *breakdown* DFD Level 0 yang sebelumnya sudah di buat.

3. Membuat DFD Level 2

Modul-Modul Pada DFD Level 1 (satu) dapat di-*breakdown* menjadi DFD Level 2 (dua). Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada tingkat ke detailan 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 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 sam persis dengan DFD Level 1 atau Level 2.

### 2.3.3 Blockchart

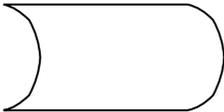
indrawan (2015), mengatakan bahwa “*Block chart* berfungsi memodelkan masukan, keluaran, referensi, master, proses ataupun transaksi dalam simbol-simbol tertentu. Pada dasarnya tidak berorientasi pada fungsi, waktu ataupun aliran data tetapi lebih ke arah proses (saling melengkapi dengan PS)”.

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

**Tabel 2.3** Simbol-simbol *Blockchart*

No	Simbol	Keterangan
1		<b>Form</b> , digambarkan dengan kombinasi persegi panjang dan garis lengkung umumnya mendefinisikan dokumen masukan (baca formulir) dan dokumen keluaran (baca laporan).
2		<b>Papan kunci</b> , digambarkan dengan segitiga dan segiempat umumnya mendefinisikan fungsi pemasukan data ( <i>key in</i> ) dapat berarti masukan untuk di rekam ataupun tidak untuk direkam (kedalam <i>storage</i> )
3		<b>Proses</b> , digambarkan dengan persegi panjang, umumnya mendefinisikan mekanisme perekaman, proses dan pelaporan.

Lanjutan Tabel 2.3 Simbol-simbol *Blockchart*

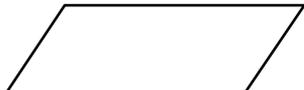
No	Simbol	Keterangan
4		<b>File</b> , digambarkan dengan kombinasi garis lengkung dan lurus umumnya mendefinisikan file referensi, file master atau file temporer yang di gunakan dalam proses.
5		<b>Monitor</b> , digambarkan dengan kombinasi garis lengkung umumnya mendefinisikan keluaran dalam bentuk layar ( <i>screen</i> ).

Sumber: <http://divaindrawantkj.blogspot.com/2015/12/simbol-simbol-pada-flowchart-dan.html>

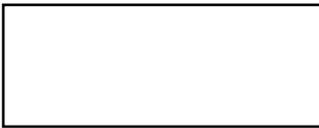
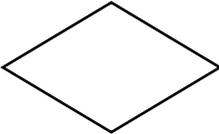
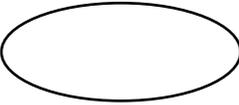
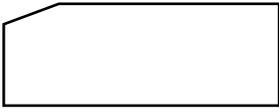
#### 2.3.4 *Flowchart*

Sitorus (2015:14), “Flowchart merupakan alat bantu berbentuk diagram alir. Flowchart menggambarkan urutan dari suatu prosedur pemecah masalah, sehingga flowchart merupakan langkah-langkah penyelesaian masalah yang di tuliskan dalam simbol-simbol tertentu”.

Tabel 2.4 Simbol *Flowchart*

No	Simbol <i>Flow Chart</i>	Nama	Fungsi <i>Flow Chart</i>
1		Terminal	Terminal merupakan permulaan atau akhir dari suatu program.
2		Input/ Output	Menyatakan Proses Input/ Output tanpa tergantung jenis peralatannya

Lanjutan Tabel 2.4 Simbol *Flowchart*

No	Simbol <i>Flow Chart</i>	Nama	Fungsi <i>Flow Chart</i>
3		Proses	Menyatakan suatu tindakan (proses) yang di lakukan oleh komputer.
4		Desicion	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jabatan ya/tidak.
5		Connector	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
6		Offline connector	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
7		Predefined proses	Menyatakan persediaan tempat menyediakan suatu pengolahan untuk memberi harga awal
8		Punched card	Menyatakan input berasal dari kartu atau output ditulis ke kartu

Sumber: Sitorus, 2015:14

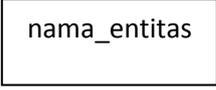
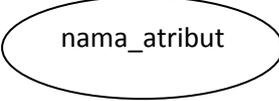


### 2.3.5 Entity Relationship Diagram (ERD)

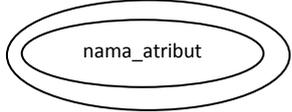
Rosa dan Shalahuddin (2018:53), mengungkapkan bahwa “*Entity Relationship Diagram* (ERD) adalah bentuk paling awal dalam melakukan perancangan basis data relasional. Jika menggunakan OODBMS maka perancangan *Entity Relationship Diagram* (ERD) tidak perlu di lakukan.

Rosa dan Shalahuddin (2018:50), menjelaskan simbol-simbol yang digunakan dalam ERD dengan menggunakan notasi Chen, yaitu

**Tabel 2.5** Simbol-Simbol *Entity Relationship Diagram* (ERD)

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 lebih ke kata benda dan belum merupakan nama tabel.
2.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
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)

Lanjutan Tabel 2.5 Simbol-Simbol *Entity Relationship Diagram* (ERD)

No	Simbol	Keterangan
4.	Atribut multinilai / multivalue 	<i>Field</i> atau kolom data butuh disimpan dalam satu entitas yang dapat dimiliki nilai lebih dari satu.
5.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
6.	Asosiasi / <i>association</i> 	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian.  lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang

Sumber: Rosa dan Shalahuddin, 2018:50-53

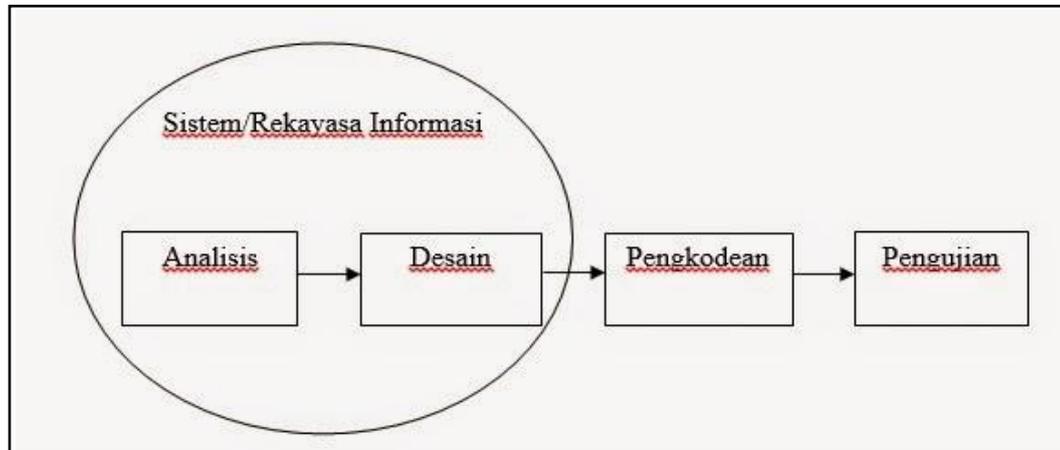
### 2.3.6 Metode Pengembangan Sistem

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan Model SLC Air Terjun (*waterfall*) atau sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life*). Menurut sukamto dan Shalahuddin (2018:28), “Metode 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:



Sumber: Sukamto dan Shalahuddin (2018)

**Gambar 2.1** Ilustrasi Model *Waterfall*

- a. Analisis kebutuhan perangkat lunak  
Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat di pahami perangkat lunak seperti apa yang di butuhkan 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 perangkat lunak termasuk struktur, data, arsitektur perangkat lunak representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat di implimentasikan menjadi program pada tahap selanjutnya, desain perangkat lunak yang di hasilkan pada tahap ini juga perlu di dokumentasikan.



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 logik dan fungsional dan memastikan bahwa semua bagian sudah di uji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

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

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak baru.

## 2.4 Teori Program

### 2.4.1 Pengenalan PHP

#### 2.4.1.1 Pengertian PHP (*Hypertext Preprocessor*)

Ardhana (2014:65), “PHP *Hypertext Preprocessor* atau sering di sebut PHP merupakan bahasa pemrograman berbasis *server-side* yang dapat melakukan parsing script PHP menjadi script web sehingga dari sisi client menghasilkan suatu tampilan yang menarik.”

Madcoms (2016:2), “ PHP (*Hypertext Preprocessor*) adalah bahasa script yang dapat ditanamkan atau di sisipkan ke dalam HTML.”

Jadi, dapat penulis simpulkan bahwa pengertian PHP adalah bahasa pemrograman berbasis *server-side* yang bisa kita gunakan untuk membuat aplikasi web yang disisipkan pada HTML.



**Gambar 2.2** Lambang *PHP*

#### **2.4.1.2 Sintaks Dasar PHP**

Yuana (2015:2), menjelaskan kode-kode PHP dituliskan diantara tanda berikut ini:

```
<?php
```

```
...
```

```
...
```

```
...
```

```
?>
```

Atau

```
<?
```

```
...
```

```
...
```

```
...
```

```
?>
```

Apabila membuat kode php dan berencana akan mendistribusikan ke pihak/orang lain, maka usahakan menggunakan sintaks `<?php ... ?>`. Hal ini dikarenakan untuk penggunaan kode yang menggunakan `<? ... ?>` terkadang tidak bisa dijalankan dalam server tertentu.



### 2.4.1.3 Tipe Data PHP

Tipe data merupakan jenis dari suatu data yang akan di proses oleh bahasa pemrograman. Murya (2014:26), menjelaskan beberapa tipe data dalam PHP, sebagai berikut:

1. **Integer** merupakan tipe data yang berguna untuk menyimpan bilangan bulat. Range bilangan integer adalah antara -2.147.4833.647 sampai dengan 2.147.483.647
2. **Double Floating** adalah tipe data yang berguna untuk menyimpan bilangan desimal. Range bilangan floating point antara 1e308 sampai dengan 1e308.
3. **Boolean** adalah tipe data yang paling sederhana, hanya berupa **TRUE** dan **FALSE**.
4. **String** adalah tipe data yang terdiri dari kata, bias berupa kata tunggal maupun kalimat. Penulisan string harus diapit dengan tanda petik, baik berupa petik tunggal ( ' ... ' ) maupun petik ganda ( " ... " ).
5. **Objek** adalah tipe data dibuat dengan tujuan agar para programmer terbiasa dengan OOP. Tipe data ini bisa berupa bilangan.
6. **Array** merupakan **Tipe Compound Primitif**, terdapat pada bahasa pemrograman lain.
7. **Null** adalah tipe data yang tidak memuat apapun. Setiap variable yang diset menjadi tipe data Null, ini akan menjadikan variabel tersebut kosong.
8. **Resources** tipe data spesial yang satu ini dikhususkan untuk menyimpan *resources*, sumber atau alamat.

### 2.4.2 Pengertian MySQL

Rosa dan Shalahddin (2018:46), "SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya di kembangkan berdasarkan teori aljabar relasional dan kalkulus."

Ardhana (2014:46), "MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang *multithread*, *muti-user*, dengan sekitar 6 juta instalasi di seluruh dunia."



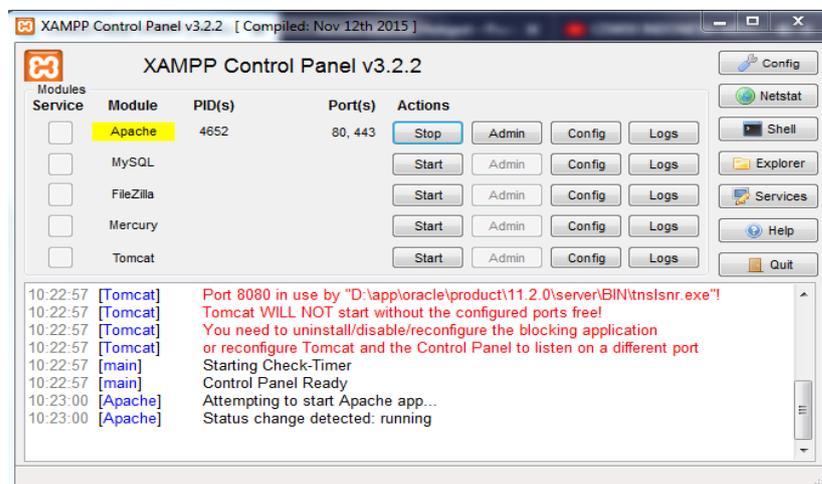
Jadi, dapat penulis simpulkan bahwa pengertian MySQL adalah bahasa yang digunakan untuk mengelola data SQL (*database management system*) atau DBMS yang *multithread, multi-user*.

### 2.4.3 Pengertian XAMPP

Dadan dan Karendi (2015:28), “XAMPP adalah salah satu web *server apache* yang terintegrasi dengan *mysql* dan *phpmyadmin*.”

Madcoms (2014:186), “XAMPP adalah sebuah paket kumpulan software yang terdiri dari Apache, MySQL, PhpMyAdmin, Perl, Filezilla dan lain-lain.”

Jadi, dapat penulis simpulkan bahwa pengertian XAMPP adalah program web lengkap yang terdiri dari *server apache MySQL, PhpMyAdmin, perl, Filezilla* dan lain lain.



Gambar 2.3 XAMPP Control Panel

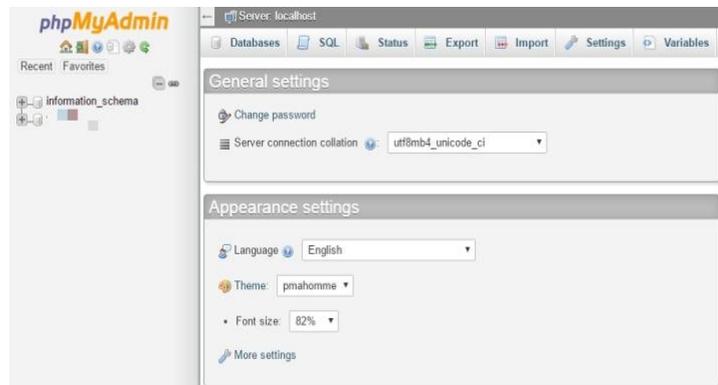
### 2.4.4 Pengertian PHPMYAdmin

Rahman (2014:12), “PHPMYAdmin adalah aplikasi PHP sebagai administrator MySQL. PHPMYAdmin mendukung berbagai aktivitas MySQL seperti pengelolaan data, *table*, relasi antar tabel, dan lain sebagainya.

Hikmah, dkk (2015:2), “PHPMYAdmin merupakan aplikasi yang dapat di gunakan untuk membuat *database*, pengguna (*user*), memodifikasi tabel, maupun mengirim *database* secara cepat dan mudah tanpa harus menggunakan perintah (*command*) SQL”.



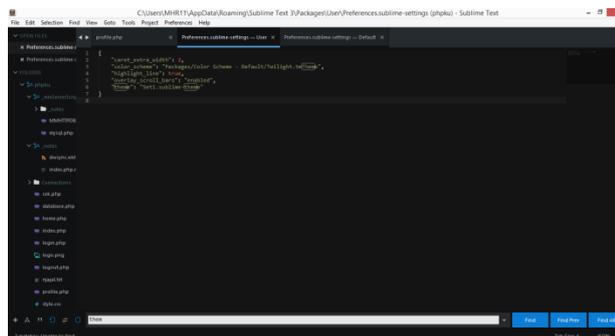
Jadi, dapat penulis simpulkan bahwa pengertian *PHPMysqlAdmin* adalah aplikasi yang di gunakan untuk membuat database secara cepat dan mudah, mendukung berbagai aktivitas *MySQL* seperti pengelolaan data, *table*, relasi antar tabel, dan lain sebagainya.



**Gambar 2.4** *PHPMysqlAdmin*

#### 2.4.5 Pengertian *Sublime Text*

Rerung (2018:25), “*Sublime Text* adalah aplikasi editor untuk kode dan teks yang dapat berjalan di berbagai platform *operating system* dengan menggunakan teknologi API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim. Aplikasi ini sangatlah fleksibel dan powerfull.



**Gambar 2.5** Tampilan *Sublime Text*