

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Penelitian tentang Sistem Informasi Klinik sudah banyak dilakukan, Bahasa pemrograman dan *software* yang digunakan juga bervariasi. Penelitian sistem informasi klinik yang menggunakan *Borland Delphi 7* pernah dilakukan Wijanto (2014). Pada penelitian ini, Sistem Informasi Klinik diharapkan bisa memberikan kemudahan dalam segala proses pengolahan data pasien, data dokter, data obat, data penyakit, data pemeriksaan, data resep, dari laporan registrasi pasien, laporan data dokter, laporan registrasi pasien, laporan pembayaran dan laporan rekam medis serta. Tujuan lain dibangunnya Sistem Informasi Klinik Pratama Mitra Sehat akan memberikan kemudahan, kelancaran, dan penghematan waktu.

Penelitian pembuatan sistem informasi pada klinik yang menggunakan *NetBeans* pernah dilakukan (Fitriana, 2014). Penelitian tersebut fokus pada apotek yang mampu mengolah data pasien, data pembelian, penjualan, retur, pembayaran serta perhitungan laba. Penelitian yang menggunakan *NetBeans* juga pernah dilakukan (Wahyono, 2014) yang mampu mengelola data pasien, obat dan transaksi agar lebih cepat dan mudah serta penelitiannya lebih meluas ke rekam medis pada poli umum, poli gigi, dan poli bidan.

Kedua penelitian di atas menggunakan *software* yang sama yaitu *NetBeans*, serta Bahasa pemrograman yang digunakan adalah Bahasa pemrograman *Java*. Pada kedua penelitian tersebut letak perbedaan lebih pada batasan masalahnya, dimana penelitian yang dilakukan (Fitriana, 2014) batasan masalahnya ada pada pengolahan data apotek sedangkan penelitian yang dilakukan (Wahyono, 2014) batasan masalahnya pada pengolahan data klinik dan apotek.

Berkaitan dengan penelitian ini, dilakukan tinjauan pustaka juga pada penelitian (Sarwoko, 2011) tentang Sistem Informasi Hasil Belajar Siswa Berbasis *Client-Server* dengan tujuan penelitian sistem informasi dapat melakukan pengolahan

dan pelaporan nilai siswa dengan cepat dan akurat kebenarannya. Berbasis *client-server* juga diterapkan dalam pembuatan sistem informasi ini sehingga dapat memberikan landasan teori mengenai *client-server* untuk penelitian yang akan dilakukan.

Beberapa penelitian yang telah disebutkan di atas dapat dijadikan sebagai acuan dalam melakukan penelitian ini. Beberapa kelebihan dan kekurangan yang ada sebelumnya akan dijadikan referensi sehingga pada penelitian selanjutnya didapatkan hasil yang lebih baik dan melengkapi kekurangan sebelumnya. Berikut ringkasan penelitian yang disajikan dalam Tabel 2.1

Tabel 2.1 Perbandingan Sistem yang Dibuat.

No	Peneliti	Implementasi Program	Isi
1.	Wijayanto (2014)	Menggunakan <i>Borland Delphi 7</i>	Pembuatan Sistem Informasi ini diharapkan bisa mempermudah dalam pencatatan pasien pada Klinik Pratama Mitra Sehat.
2.	Fitriana (2014)	Menggunakan <i>Netbeans</i>	Pembuatan Sistem Informasi ini mampu mengolah data pasien, data pembelian, penjualan, retur, pembayaran serta perhitungan laba.
3.	Wahyono dkk (2014)	Menggunakan <i>Netbeans</i>	Pembutan Sistem Informasi ini dapat mengelola data pasien, obat dan transaksi pada poli umum, poli gigi, dan poli bidan agar lebih cepat dan mudah.
4.	Sarwoko (2011)	Menggunakan <i>Netbeans</i> dan Berbasis <i>Client-Server</i>	Penelitian ini Sistem Informasinya dapat melakukan pengolahan dan pelaporan nilai siswa bisa dilakukan dengan cepat dan akurat kebenarannya yang berbasis <i>client-server</i> .
5.	Yeni Anggraheny (2018)	Menggunakan <i>NetBeansIDE 7.4</i> dan <i>MySQL</i>	Aplikasi Rekam Medik ini dapat digunakan untuk penyimpanan data obat, tindakan, pasien, dokter dan petugas serta pengolahan data registrasi, rekam tindakan, pemberian obat dan pembayaran.

## 2.2 Kerangka Berfikir

Kerangka berfikir yang di gunakan dalam pengembangan dapat di lihat pada Gambar 2.4.

Keterangan Diagram Kerangka Pemikiran (Gambar 2.4) :

### 1. Latar Belakang Masalah

Mahasiswa mengembangkan aplikasi rekam medik Klinik Tirta Amarta Boyolali.

### 2. Penguasaan Dasar (JAVA, NETBEANS IDE 7.4 dan MySQL)

Mahasiswa melakukan beberapa percobaan pembuatan aplikasi sederhana dengan JAVA, NETBEANS IDE 7.4 dan MySQL agar lebih menguasai bahasa pemrograman tersebut dan mendapatkan hasil yang maksimal.

### 3. Observasi Aplikasi

Mahasiswa mengamati beberapa aplikasi yang sudah ada, baik dari karya ilmiah, buku atau internet yang dapat dijadikan referensi untuk membangun aplikasi.

### 4. Analisis dan Perancangan Sistem

Mahasiswa menganalisa dan merancang sistem yang akan dibangun seperti apa bagaimana desainnya, dan apa saja isinya, sehingga aplikasi ini dapat membantu memecahkan permasalahan yang ada pada Klinik Tirta Amarta Boyolali.

### 5. Implementasi dan Pengembangan Sistem

#### a. Membuat *database* dengan MySQL

Mahasiswa membuat *database* dari data-data yang telah didapatkan sesuai dengan kebutuhan aplikasi.

#### b. Perancangan aplikasi pengolahan data

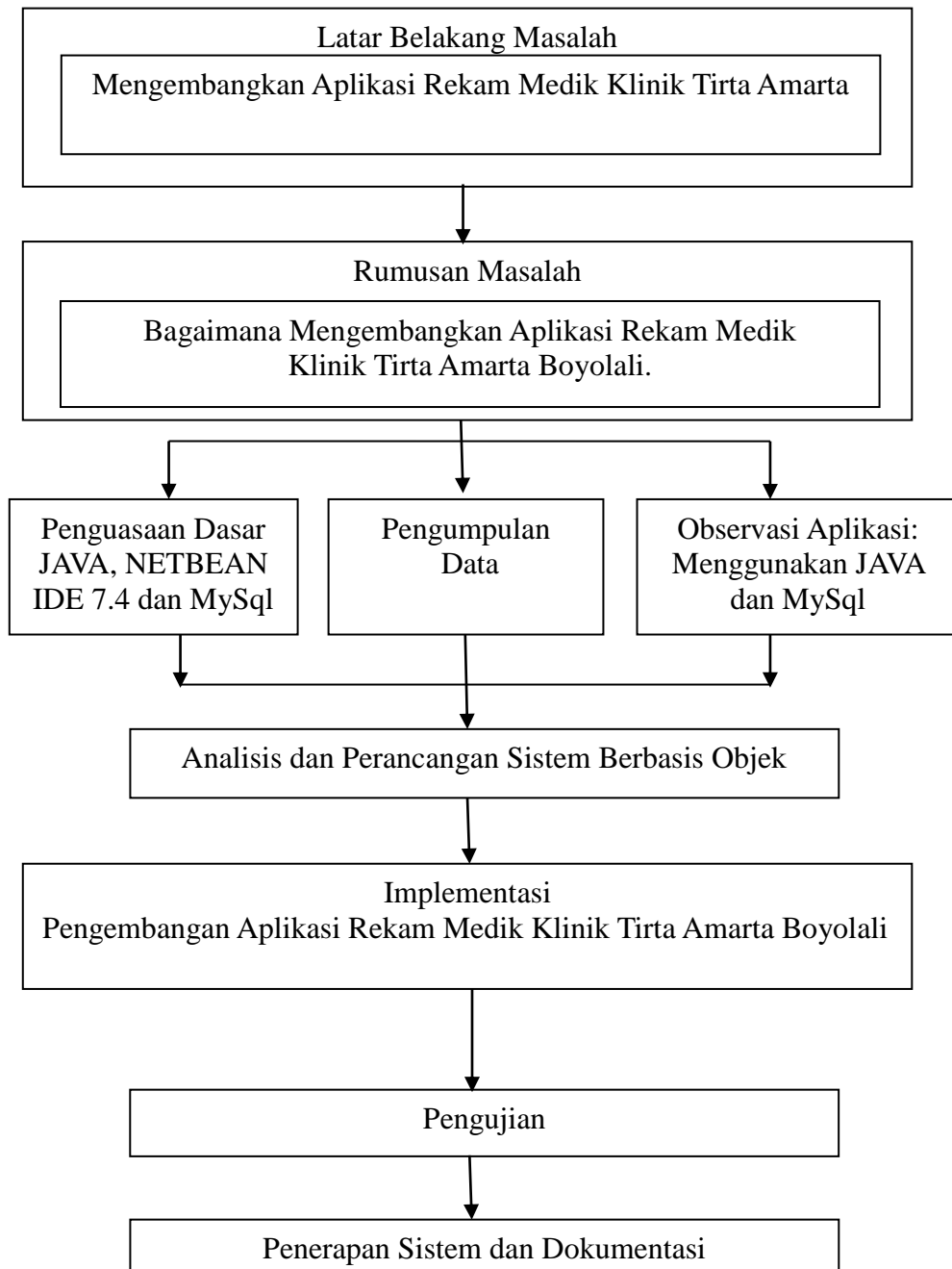
Mahasiswa membuat aplikasi dengan dasar *database* yang telah selesai dibuat.

### 6. Pengujian Sistem

Pengujian sistem ini dilakukan bertujuan untuk mengetahui jika ternyata masih ada kesalahan atau kekurangan pada aplikasi yang telah selesai dibuat.

### 7. Penerapan Sistem dan Dokumentasi

Pada tahap akhir, dimana aplikasi telah siap digunakan setelah melewati tahap pengujian dan membuat dokumentasi dari keseluruhan kegiatan penyusunan tugas akhir.



Gambar 2.1 Diagram Kerangka Pemikiran

## **2.3 Landasan Teori**

### **2.3.1 Pengertian Aplikasi**

Menurut (Anisyah, 2000), aplikasi adalah penerapan, penggunaan atau penambahan, dapat disimpulkan bahwa aplikasi merupakan *software* yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data.

### **2.3.2 Pengertian Rekam Medis**

Menurut PERMENKES No.269/MenKes/PER/111/2008 rekam medis adalah berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain kepada pasien pada sarana pelayanan kesehatan.

Menurut Depkes RI (1997) rekam medik adalah untuk menunjang tercapainya tertib administrasi rumah sakit akan berhasil sebagaimana yang diharapkan. Tanpa didukung suatu sistem pengelolaan rekam medis yang baik dan benar, mustahil tertib administrasi rumah sakit akan berhasil sebagaimana yang diharapkan. Rekam Medis adalah berkas yang berisikan catatan dan dokumen tentang identitas, pemeriksaan, diagnosis pengobatan, tindakan dan pelayanan lain yang diberikan kepada pasien selama dirawat di rumah sakit baik yang dilakukan di unit rawat jalan, rawat inap dan unit gawat darurat. Oleh karenanya rekam medis harus diisi langsung oleh dokter dan tenaga kesehatan lain seperti perawat, bidan fisioterapi.

Rekam medis harus diisi langsung pada setiap tindakan yang dilakukan, sehingga dari catatan tersebut dapat setiap saat diketahui dan diperoleh gambaran secara jelas mengenai pelayanan atau tindakan yang telah dilakukan terhadap pasien. Informasi yang terdapat pada rekam medis atau isi dari rekam medis bersifat rahasia karena hal ini menjelaskan hubungan yang khusus antara pasien dan dokter yang wajib dilindungi dari pembocoran sesuai dengan kode etik kedokteran dan peraturan perundangan yang berlaku.

### 2.3.3 Pengertian Klinik

Menurut (Oktaviani, 2008), klinik adalah suatu lembaga kesehatan dimana orang melakukan pemeriksaan terhadap kesehatannya. Klinik disebut juga suatu tempat yang memberikan pelayanan kesehatan kepada masyarakat.

Klinik adalah fasilitas pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan yang menyediakan pelayanan medis dasar dan atau spesialisik, diselenggarakan oleh lebih dari satu jenis tenaga kesehatan dan dipimpin oleh seorang tenaga medis. Tenaga medis adalah dokter, dokter spesialis, dokter kandungan atau dokter kandungan spesialis, sedangkan tenaga kesehatan adalah setiap orang yang mengabdikan diri dalam bidang kesehatan serta memiliki pengetahuan dan atau keterampilan melalui pendidikan di bidang kesehatan yang untuk jenis tertentu memerlukan kewenangan untuk melakukan upaya kesehatan.

Berdasarkan jenis pelayanannya, klinik dibagi menjadi Klinik Pratama dan Klinik Utama menurut Peraturan Menteri Kesehatan Republik Indonesia Nomor 028/Menkes/Per/I/2011.

- a) Klinik Pratama merupakan klinik yang menyelenggarakan pelayanan medik dasar.
- b) Klinik Utama merupakan klinik yang menyelenggarakan pelayanan medik spesialisik atau pelayanan medik dasar dan spesialisik. Klinik Pratama atau Klinik Utama dapat mengkhususkan pelayanan pada satu bidang tertentu berdasarkan disiplin ilmu, golongan umur, organ atau jenis penyakit tertentu.

### 2.3.4 Pengertian Database

Menurut Silberschatz, dkk (2002) yang dikutip dalam buku karangan Simarmata, dkk (2005), mendefinisikan basisdata (*database*) sebagai kumpulan data berisi informasi yang sesuai untuk sebuah perusahaan.

Menurut Stephens, dkk (2000) yang dikutip dalam buku karangan Simarmata, dkk (2005), basisdata (*database*) adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Dengan basisdata, pengguna dapat menyimpan data secara terorganisasi.

### 2.3.5 Pengujian *Black Box*

Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk program. Pengujian *black-box* bukan merupakan alternatif dari teknik *white-box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan daripada *white-box* (Pressman, 2002).

### 2.3.6 Analisis dan Perancangan Berbasis Objek

*Object Oriented Programming* atau Pemrograman Berorientasi Objek adalah konsep pemrograman yang difokuskan pada penciptaan kelas yang merupakan abstraksi/*blueprint/prototype* dari suatu objek (Hermawan, 2004). Analisis dan perancangan metodologi berorientasi objek menggunakan menggunakan diagram *Unified Modeling Language* (UML).

#### 2.3.11.1 *Use Case Diagram*

*Use case* diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*. Simbol-simbol yang digunakan pada *use case diagram* disajikan pada Tabel 2.1.

#### 2.3.11.2 *Class Diagram*

*Class* diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class* diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class* diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

*Class* memiliki tiga area pokok :

- 1) Nama (dan *stereotype*)

- 2) Atribut
- 3) Metoda

Simbol-simbol yang digunakan pada *class diagram* disajikan pada Tabel 2.2.

#### **2.3.11.3 Sequence Diagram**

*Sequence* diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram. Simbol-simbol yang digunakan pada *sequence* diagram ditampilkan pada Tabel 2.3

#### **2.3.11.4 Activity Diagram**

*Activity* diagram menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas. Simbol-simbol yang digunakan pada *activity diagram* disajikan pada Tabel 2.4.

#### **2.3.11.5 Component Diagram**







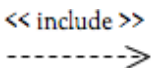
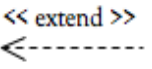
*Component* diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan. Simbol-simbol yang digunakan pada *component diagram* disajikan pada Tabel 2.5.

#### **2.3.11.6 Deployment Diagram**


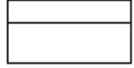
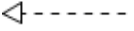


*Deployment* diagram memberikan gambaran dari arsitektur fisik perangkat lunak, perangkat keras, dan artefak dari sistem. *Deployment* diagram dapat dianggap sebagai ujung spektrum dari kasus penggunaan, menggambarkan bentuk fisik dari sistem yang bertentangan dengan gambar konseptual dari pengguna dan perangkat berinteraksi dengan sistem. Simbol-simbol yang digunakan pada *deployment diagram* disajikan pada Tabel 2.6.





Tabel 2.1 Simbol *Use Case* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Unidirectional association</i>	Menggambarkan relasi antara aktor dengan <i>use case</i>
3		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
4		<i>Boundary</i>	Boundary biasanya berupa tepi dari sistem, seperti user interface, atau suatu alat yang berinteraksi dengan sistem lain.
5		<i>Control element</i>	Control element mengatur aliran dari informasi untuk sebuah scenario.
6		<i>Entity</i>	Entity biasanya elemen yang bertanggung jawab menyimpan data atau informasi.
7.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit
8.		<i>Extend</i>	Menspesifikasi bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.




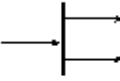

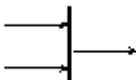
Tabel 2.2 Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Menggambarkan relasi generalisasi
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
5		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
6		<i>Association</i>	Menggambar relasi asosiasi

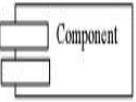

Tabel 2.3 Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	<i>Message</i> ditampilkan sebagai anak panah dari lifeline dari satu objek ke objek yang lain.

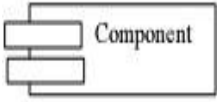
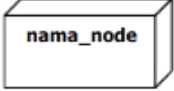

Tabel 2.4 Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
3		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
5		<i>Decision node</i>	Suatu titik/point pada activity diagram yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi
6		<i>Join Node</i>	Beberapa aliran masukan tertentu menjadi satu aliran

Tabel 2.5 *Component Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Component</i>	Sebuah komponen melambangkan sebuah entitas <i>software</i> dalam sebuah sistem. Sebuah komponen dinotasikan sebagai sebuah kotak segiempat dengan dua kotak kecil tambahan yang menempel disebelah kirinya.
2		<i>Dependency</i>	Sebuah <i>Dependency</i> digunakan untuk menotasikan relasi antara dua komponen. Notasinya adalah tanda panah putus-putus yang diarahkan kepada komponen tempat sebuah komponen itu bergantung.

Tabel 2.6 Simbol *Deployment Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Component</i>	Pada <i>deployment</i> diagram, komponen-komponen yang ada diletakkan di dalam node untuk memastikan keberadaan posisi mereka.
2		<i>Node</i>	Node menggambarkan bagian-bagian <i>hardware</i> dalam sebuah sistem. Notasi untuk node digambarkan sebagai sebuah kubus 3 dimensi.
3.		<i>Association</i>	Sebuah <i>association</i> digambarkan sebagai sebuah garis yang menghubungkan dua node yang mengindikasikan jalur komunikasi antara komponen-komponen <i>hardware</i> .

### 2.3.12 Pengertian *Java Script*

Menurut (Dhewiberta, 2006), *java script* adalah bahasa pemrograman yang sederhana karena bahasa ini tidak dapat digunakan untuk membuat aplikasi ataupun *applet*. Dengan *java script* kita dapat dengan mudah membuat sebuah halaman *web* yang interaktif. Program *java script* dituliskan pada file *HTML* (\*.htm\*.html\*).

### 2.3.13 Pengertian *Netbeans*

Suatu *IDE (Integrated Development Environment)* adalah lingkupan pemrograman yang mengintegrasikan kedalam suatu aplikasi perangkat lunak yang menyediakan pembangun *GUI*, suatu *text* atau *code editor*, suatu *compiler* atau *enterpreter* dan suatu *debugger*. (Java Education Network Indonesia, 2007) *NetBeans IDE* merupakan perangkat lunak yang digunakan untuk membangun perangkat lunak yang lain. *NetBeans IDE* dapat digunakan untuk membangun perangkat lunak berbasis *Java Standart Edition*, *Java Enterprise Edition*, *Java Micro Edition*, *Java FX*, *C/C++*, *Ruby*, *Groovy* dan *Phyton*. *Java* adalah nama untuk sekumpulan teknologi untuk membuat perangkat lunak pada komputer

*standalone* ataupun pada lingkungan jaringan. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di *Sun Microsystems* saat ini merupakan bagian dari *Oracle* dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada *C* dan *C++* namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin arah bawah yang minimal (Shalahuddin dan Rosa, 2010).

#### **2.3.14 Pengertian MySQL**

Menurut (Kadir, 2008), MySQL adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. MySQL termasuk jenis RDBMS (*Relational Database Management System*). Itulah sebabnya, istilah seperti *table*, baris, dan kolom digunakan pada MySQL. Pada MySQL, sebuah *database* mengandung satu atau sejumlah *table*. Selain itu, ia bersifat *Open Source* (Anda tidak perlu membayar untuk menggunakannya) pada berbagai platform (kecuali untuk jenis *Enterprise*, yang bersifat komersial).

#### **2.3.15 Pengertian Xampp**

*XAMPP* adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data MySQL di komputer lokal". *XAMPP* berperan sebagai *server web* pada komputer lokal. *XAMPP* juga dapat disebut sebuah *Cpanel server virtual*, yang dapat membantu melakukan *preview* sehingga dapat dimodifikasi *website* tanpa harus *online* atau terakses dengan *internet*.

Kepanjangan dari *XAMPP* sendiri yaitu Apache, PHP, MySQL dan perl. *XAMPP* merupakan *tool* yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan menginstall *XAMPP* maka tidak perlu lagi melakukan instalasi dan konfigurasi *web server* Apache, PHP dan MySQL secara manual. *XAMPP* akan menginstallasi dan mengkonfigurasikannya secara otomatis untuk anda atau auto konfigurasi.

Mengenal bagian XAMPP yang biasa digunakan pada umumnya : *htdocs* adalah folder tempat meletakkan berkas-berkas yang akan dijalankan, seperti berkas PHP, HTML dan skrip lain. phpMyAdmin merupakan bagian untuk mengelola basis data MySQL yang ada dikomputer. Kontrol Panel yang berfungsi untuk mengelola layanan *service* XAMPP. Seperti menghentikan *stop* layanan, ataupun memulai *start*.

\