

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari suatu unsur, komponen, atau variabel yang terorganisir, saling tergantung satu sama lain dan terpadu. Teori sistem secara umum yang pertama kali diuraikan oleh Kennet Boulding, terutama menekankan pentingnya perhatian terhadap setiap bagian yang membentuk sebuah sistem. Kecenderungan manusia yang mendapat tugas memimpin suatu organisasi adalah terlalu memusatkan perhatian pada salah satu komponen saja dari sistem organisasi.

Teori sistem melahirkan konsep-konsep futuristik, antara lain yang terkenal adalah konsep sibernetika (*cybernetics*). Konsep atau dibidang kajian ilmiah ini berkaitan dengan upaya menerapkan berbagai ilmu yaitu ilmu perilaku, fisika, biologi, dan teknik. Oleh karena itu sibernetika biasanya berkaitan dengan usaha-usaha otomasi tugas-tugas yang dilakukan manusia, sehingga melahirkan studi-studi tentang robotika, kecerdasan buatan (*artificial intelegence*). Unsur-unsur yang mewakili suatu sistem secara umum adalah masukan (*input*), pengolahan (*processing*), dan keluaran (*output*). (Tata Sutabri, 2012)

II.2. Informasi

Informasi adalah data yang telah diklasifikasikan diolah atau diinterpretasi untuk digunakan dalam proses pengambil keputusan. Sistem pengolahan

informasi mengolah data menjadi informasi atau tepatnya mengolah dari bentuk tak berguna menjadi berguna bagi penerimanya.

Informasi adalah data yang telah diklasifikasikan atau diinterpretasi untuk digunakan dalam pengambil keputusan (Tata Sutabri, 2012).

II.3. Sistem Informasi

Sistem informasi adalah berupa suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan data transaksi harian yang mendukung operasi yang bersifat manajerial dengan kegiatan strategi suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan (Tata Sutabri, 2012).

II.4. Sistem Pakar

II.4.1 Pengertian Sistem Pakar

Secara umum, sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti layaknya seorang pakar. Dengan sistem pakar ini, orang awam pun dapat menyelesaikan masalahnya atau hanya sekedar mencari suatu informasi berkualitas yang sebenarnya hanya dapat diperoleh dengan bantuan para ahli di bidangnya. Sistem pakar ini juga akan dapat membantu aktivitas para pakar sebagai asisten yang berpengalaman dan mempunyai asisten yang berpengalaman dan mempunyai pengetahuan yang dibutuhkan. (Muhammad Arhami, 2015)

Dalam penyusunannya, sistem pakar mengkombinasikan kaidah-kaidah penarikan kesimpulan (*inference rules*) dengan basis pengetahuan tertentu yang diberikan oleh satu atau lebih pakar dalam bidang tertentu. Kombinasi dari kedua hal tersebut disimpan dalam komputer, yang selanjutnya digunakan dalam proses pengambilan keputusan untuk penyelesaian masalah tertentu. (Muhammad Arhami, 2015)

Alasan Pengembangan Sistem Pakar, sistem pakar sendiri dikembangkan lebih lanjut dengan alasan :

1. Dapat menyediakan kepakaran setiap waktu dan di berbagai lokasi.
2. Secara otomatis mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar.
3. Seorang pakar akan pensiun atau pergi.
4. Seorang pakar adalah mahal.
5. Kepakaran dibutuhkan juga pada lingkungan yang tidak bersahabat.(Muhammad Arhami, 2015)

II.4.2 Ciri-ciri Sistem Pakar

Sistem pakar yang baik harus memenuhi ciri-ciri sebagai berikut :

1. Memiliki informasi yang handal
2. Mudah dimodifikasi
3. Dapat digunakan dalam berbagai jenis *computer*
4. Memiliki kemampuan untuk belajar beradaptasi.(Muhammad Arhami, 2015)

II.4.3 Keuntungan Sistem Pakar

Secara garis besar, banyak manfaat yang dapat diambil dengan adanya sistem pakar, antara lain :

1. Memungkinkan orang awam bisa mengerjakan pekerjaan para ahli
2. Bisa melakukan proses secara berulang secara otomatis.
3. Menyimpan pengetahuan dan keahlian para pakar.
4. Meningkatkan output dan produktivitas.
5. Meningkatkan kualitas.
6. Mampu mengambil dan melestarikan keahlian para pakar (terutama yang termasuk keahlian langka).
7. Memiliki kemampuan untuk mengakses pengetahuan.
8. Memiliki reabilitas.
9. Meningkatkan kapabilitas sistem komputer.
10. Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dan mengandung ketidakpastian.
11. Sebagai media pelengkap dalam pelatihan.
12. Meningkatkan kapabilitas dalam penyelesaian masalah.
13. Menghemat waktu dalam pengambilan keputusan. (Muhammad Arhami, 2015)

II.4.4 Kelemahan Sistem Pakar

Di samping memiliki beberapa keuntungan, sistem pakar juga memiliki beberapa kelemahan, antara lain :

1. Biaya yang diperlukan untuk membuat dan memeliharanya sangat mahal.

2. Sulit dikembangkan. Hal ini tentu saja erat kaitannya dengan ketersediaan pakar di bidangnya.
3. Sistem Pakar tidak 100% bernilai benar.(Muhammad Arhami, 2015)

II.4.5 Modul Penyusun Sistem Pakar

Menurut Staugaard (1987) suatu sistem pakar disusun oleh tiga modul utama yaitu :

1. Modul Penerimaan Pengetahuan (*Knowledge Acquisition Mode*) Sistem berada pada modul ini, pada saat ia menerima pengetahuan dari pakar. Proses mengumpulkan pengetahuan-pengetahuan yang akan digunakan untuk pengembangan sistem, dilakukan dengan bantuan *knowledge engineer*. Peran *knowledge engineer* adalah sebagai penghubung antara suatu sistem pakar dengan pakarnya.
2. Modul Konsultasi (*Consultation Mode*)
Pada saat sistem berada pada posisi memberikan jawaban atas permasalahan yang diajukan oleh user, sistem pakar berada dalam modul konsultasi.Pada modul ini, user berinteraksi dengan sistem dengan menjawab pertanyaan-pertanyaan yang diajukan oleh sistem.
3. Modul Penjelasan (*Explanation Mode*)
Modul ini menjelaskan proses pengambilan keputusan oleh sistem bagaimana suatu keputusan dapat diperoleh.(Muhammad Arhami, 2015)

Struktur Sistem Pakar, komponen utama pada struktur sistem pakar menurut Hu et al (1987) meliputi:

1. Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan merupakan inti dari suatu sistem pakar, yaitu berupa representasi pengetahuan dari pakar. Basis pengetahuan tersusun atas fakta dan kaidah. Fakta adalah informasi tentang objek, peristiwa, atau situasi. Kaidah adalah cara untuk membangkitkan suatu fakta baru dari fakta yang sudah diketahui.

2. Mesin Inferensi (*Inference Engine*)

Mesin inferensi berperan sebagai otak dari sistem pakar. Mesin inferensi berfungsi untuk memandu proses penalaran terhadap suatu kondisi, berdasarkan pada basis pengetahuan yang tersedia. Di dalam mesin inferensi terjadi proses untuk memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan dalam rangka mencapai solusi atau kesimpulan. Dalam prosesnya, mesin inferensi menggunakan strategi penalaran dan strategi pengendalian. Strategi penalaran terdiri dari strategi penalaran pasti (*Exact Reasoning*) dan strategi penalaran tak pasti (*Inexact Reasoning*). *Exact reasoning* dilakukan jika semua data yang dibutuhkan untuk menarik suatu kesimpulan tersedia, sedangkan *inexact reasoning* dilakukan pada keadaan sebaliknya. Strategi pengendalian berfungsi sebagai panduan arah dalam melakukan proses penalaran.

3. Basis Data (*Data Base*)

Basis data terdiri atas semua fakta yang diperlukan, dimana fakta-fakta tersebut digunakan untuk memenuhi kondisi dari kaidah-kaidah dalam sistem. Basis data menyimpan semua fakta, baik fakta awal pada saat sistem mulai beroperasi, maupun fakta-fakta yang diperoleh pada saat proses penarikan

kesimpulan sedang dilaksanakan. Basis data digunakan untuk menyimpan data hasil observasi dan data lain yang dibutuhkan selama pemrosesan.

4. Antarmuka Pemakai (*User Interface*)

Fasilitas ini digunakan sebagai perantara komunikasi antara pemakai.dengan komputer.(Muhammad Arhami, 2015)

II.5 Forward Chaining

Forward Chaining adalah pencarian atau teknik pelacakan ke depan yang dimulai dengan informasi yang ada dan penggabungan *rule* untuk menghasilkan suatu kesimpulan atau tujuan. (Russel S,Norvig P, 2003). Pelacakan maju ini sangat baik jika bekerja dengan permasalahan yang dimulai dengan rekaman informasi awal dan ingin dicapai penyelesaian akhir, karena seluruh proses akan dikerjakan secara berurutan maju.

Contoh :

Terdapat 10 aturan yang tersimpan dalam basis pengetahuan yaitu :

R1 : if A and B then C

R2 : if C then D

R3 : if A and E then F

R4 : if A then G

R5 : if F and G then D

R6 : if G and E then H

R7 : if C and H then I

R8 : if I and A then J

R9 : if G then J

R10 : if J then K

II.6. Metode *Certainty Factor*

Faktor kepastian (*certainty factor*) diperkenalkan oleh Shortliffe Buchanan dalam pembuatan MYCIN pada tahun 1975 untuk mengakomodasi ketidakpastian pemikiran (*inexact reasoning*) seorang pakar. Teori ini berkembang bersamaan dengan pembuatan sistem pakar MYCIN. Team pengembang MYCIN mencatat bahwa pakar sering kali menganalisa informasi yang ada dengan ungkapan seperti misalnya: mungkin, kemungkinan besar, hampir pasti. Untuk mengakomodasi hal ini tim MYCIN menggunakan *certainty factor* (CF) guna menggambarkan tingkat keyakinan pakar terhadap permasalahan yang sedang dihadapi. Secara umum, *rule* direpresentasikan dalam bentuk sebagai berikut (John Durkin, 1994) :

$$\begin{array}{l} \text{IF } E1 \text{ [AND / OR] } E2 \text{ [AND / OR] } \dots E_n \\ \text{THEN } H \text{ (CF = CF}_i \text{)} \end{array}$$

Dimana:

E1 ... En : fakta – fakta (*evidence*) yang ada.

H : hipotesa atau konklusi yang dihasilkan.

CF : tingkat keyakinan (*Certainty Factor*) terjadinya hipotesa H akibat adanya fakta – fakta E1 s/d En .

Definisi menurut David McAllister *Certainty Factor* adalah suatu metode untuk membuktikan apakah suatu fakta itu pasti ataukah tidak pasti yang

berbentuk *metric* yang biasanya digunakan dalam sistem pakar. Metode ini sangat cocok untuk sistem pakar yang mendiagnosis sesuatu yang belum pasti.

II.6.1 Aturan metode *Certainty Factors*:

Metode yang digunakan dalam penelitian ini adalah *certainty factor*. Faktor kepastian (*certainty faktor*) menyatakan kepercayaan dalam sebuah kejadian (fakta atau hipotesis) berdasarkan bukti atau penilaian pakar (Kusumadewi, 2003). Faktor kepastian diperkenalkan oleh Shortliffe Buchanan dalam pembuatan MYCIN pada tahun 1975 untuk mengakomodasi ketidakpastian pemikiran seorang pakar. Team pengembang MYCIN mencatat bahwa seorang dokter sering kali menganalisa informasi yang ada dengan ungkapan seperti : mungkin, kemungkinan besar, hampir pasti. Oleh sebab itu team MYCIN menggunakan metode *certainty factor* (CF) guna menggambarkan tingkat keyakinan pakar terhadap permasalahan yang sedang dihadapi.

Untuk membantu menyelesaikan masalah ketidakpastian terhadap hipotesis diagnosa, maka digunakan metode *Certainty Factor*. Faktor kepastian (*Certainty Factor*) menyatakan kepercayaan dalam sebuah kejadian (atau fakta atau hipotesis) berdasarkan bukti atau penilaian pakar. *Certainty factor* menggunakan suatu nilai untuk mengasumsi derajat keyakinan seorang pakar terhadap suatu data.

Notasi faktor kepastian :

$$CF[H, E] = MB[H, E] - MD[H, E]$$

$$MB[h, e1 \wedge e2] = \begin{matrix} 0 & MD[h, e1 \wedge e2] = 1 \\ MB[h, e1] + MB[h, e2].(1 - MB[h, e1]) & \text{lainnya} \end{matrix}$$

$$MD[h, e1 \wedge e2] = \begin{matrix} 0 & MB[h, e1 \wedge e2] = 1 \\ MD[h, e1] + MD[h, e2].(1 - MD[h, e1]) & \text{lainnya} \end{matrix}$$

Keterangan:

CF = *Certainty Factor* dalam hipotesis H yang dipengaruhi oleh fakta E

MB = *Measure of Believe*, merupakan nilai kenaikan dari kepercayaan hipotesis H dipengaruhi oleh fakta E.

MD = *Measure of Disbelieve*, merupakan nilai kenaikan dari ketidakpercayaan hipotesis H dipengaruhi oleh fakta E.

H = Hipotesis (Dugaan)

E = *Evidence* (Peristiwa/fakta)

$$CF(H,E) = MB(H,E) - MD(H, E)$$

CF(H,E) : *certainty factor* dari hipotesis H yang dipengaruhi oleh gejala (*evidence*) E. Besarnya CF berkisar antara -1 sampai dengan 1.

Nilai -1 menunjukkan ketidakpercayaan mutlak sedangkan nilai 1 menunjukkan kepercayaan mutlak.

MB(H,E) : ukuran kenaikan kepercayaan (*measure of increased belief*) terhadap hipotesis H yang dipengaruhi oleh gejala E.

MD(H,E) : ukuran kenaikan ketidakpercayaan (*measure of increased disbelief*) terhadap hipotesis H yang dipengaruhi oleh gejala E

II.6.2 Kelebihan dan Kekurangan Metode *Certainty Factors*

1. Kelebihan metode *Certainty Factors* adalah:
 - a. Metode ini cocok dipakai dalam sistem pakar untuk mengukur sesuatu apakah pasti atau tidak pasti dalam mendiagnosis penyakit sebagai salah satu contohnya.
 - b. Perhitungan dengan menggunakan metode ini dalam sekali hitung hanya dapat mengolah dua data saja sehingga keakuratan data dapat terjaga.
2. Kekurangan metode *Certainty Factors* adalah:
 - a. Ide umum dari pemodelan ketidakpastian manusia dengan menggunakan numerik metode *certainty factors* biasanya diperdebatkan. Sebagian orang akan membantah pendapat bahwa *formula* untuk metode *certainty factors* diatas memiliki sedikit kebenaran.
 - b. Metode ini hanya dapat mengolah ketidakpastian/kepastian 5 data saja. Perlu dilakukan beberapa kali pengolahan data untuk data yang lebih dari 5 buah.

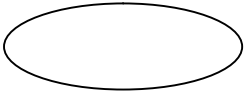
II.7. *Unified Modeling Language (UML)*

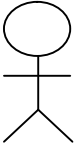
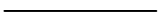

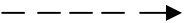
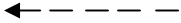
UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. Alat bantu yang digunakan dalam perancangan, berorientasi objek berbasis UML adalah sebagai berikut :

1. *Use Case Diagram*

Use Case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sistem informasi dan siapa saja yang berhak menggunakan fungsi - fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal nama <i>use case</i>.</p>



	<p>Aktor adalah abstraction dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi actor, harus ditentukan pembagian tenaga dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau system bisa muncul dalam beberapa peran. Perlu dicatat bahwa actor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki control terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p>
	<p><i>Include</i>, merupakan didalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.</p>
	<p><i>Extend</i>, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p>


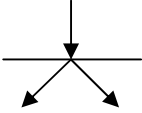
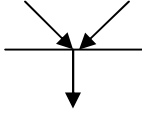
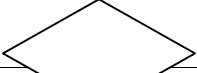

(Sumber :Panduan Praktis Membuat UML, 2012)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel II.2. Simbol *Activity Diagram*

Gambar	Keterangan
	<p><i>Start point</i>, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.</p>
	<p><i>End point</i>, akhir aktifitas.</p>

	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> , (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Panduan Praktis Membuat UML, 2012)

3. *Class Diagram*(Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi Kelas (*Class*), Relasi, *Associations*, *Generalizations* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operation/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.3. Simbol *Class Diagram*

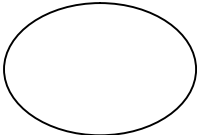
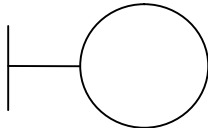
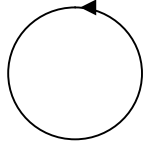
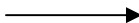
Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

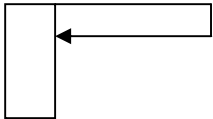


(Sumber : Panduan Praktis Membuat UML, 2012)

4. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* yaitu :

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih faktor dengan sistem, seperti tampilan form entry dan <i>form</i> cetak.
	<i>Control Class</i> , objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .

	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
	<p><i>Activation</i>, <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(Sumber : Panduan Praktis Membuat UML, 2012)

II.8. Bahasa Pemrograman *PHP*

PHP adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan *web*. Selain itu, *PHP* juga bisa digunakan sebagai bahasa pemrograman umum. *PHP* dikembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh *The PHP Group*. Situs resmi *PHP* beralamat di <http://www.php.net>.

PHP disebut bahasa pemrograman *server side* karena *PHP* diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti *JavaScript* yang diproses pada *web browser (client)*.

Pada awalnya *PHP* merupakan singkatan dari *Personal Home Page*. Sesuai dengan namanya, *PHP* digunakan untuk membuat website pribadi. Dalam beberapa tahun perkembangannya, *PHP* menjelma menjadi bahasa pemrograman web yang *powerful* dan tidak hanya digunakan untuk membuat

halaman *web* sederhana, tetapi juga *website* populer yang digunakan oleh jutaan orang seperti *wikipedia*, *wordpress*, *joomla*, dll. (Ir. Yuniar Supardi, 2007 : 12)

II.9. *MySQL*

MySQL adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan multi *user*. *MySQL* memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*. Penulis sendiri dalam menjelaskan buku ini menggunakan *MySQL* yang *free software* karena bebas menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensi, yang berada di bawah lisensi GNU/ GPL (*General Public License*), yang dapat anda *download* pada alamat resminya <http://www.mysql.com>. (Wahana Komputer, 2010).

Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. *MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya. *SQL* (*Structured Query Language*) adalah seakuntansi konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

II.10. *XAMPP*

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai

server yang berdiri sendiri (*localhost*), yang terdiri atas *program Apache HTTP Server, MySQLdatabase*, dan penerjemahbahasa yang ditulis dengan bahasa pemrograman *PHP dan Perl*.

Nama *XAMPP* merupakan singkatan dari *X* (empat sistem operasi apapun), *Apache, MySQL, PHP dan Perl*. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis. Untuk mendapatkannya dapat *download* langsung dari *web* resminya.

II.11. Adobe Dreamweaver CS5

Adobe Dreamweaver CS5 adalah versi terbaru dari *Dreamweaver* yang merupakan bagian dari *Adobe Creative Suite 5*. *Dreamweaver* sendiri merupakan aplikasi yang digunakan sebagai *HTML editor profesional* untuk mendesain web secara visual. Aplikasi ini juga bias dikenal dengan istilah *WYSIWYG (What You See Is What You Get)*, yang intinya adalah anda tidak harus berurusan dengan tag-tag *HTML* untuk membuat sebuah site dan dapat melihat hasil desainnya secara langsung.

Dengan Kemampuan fasilitas yang optimal dalam jendela *Design* akan memberikan kemudahan untuk mendesain *web* meskipun untuk para web desainer pemula sekalipun. Kemampuan *Dreamweaver* untuk berinteraksi dengan beberapa bahasa pemrograman seperti *PHP, ASP, JavaScript*, dan yang lainnya juga memberikan fasilitas maksimal kepada *desainer web* dengan menyertakan bahasa pemrograman didalamnya. (Andi Offset, Yogyakarta, 2011 : 2).