

BAB II

TINJAUAN PUSTAKA

II.1. Kecerdasan Buatan

Kecerdasan buatan atau *Artificial Intelligence* adalah suatu ilmu yang mempelajari cara membuat komputer melakukan sesuatu seperti yang dilakukan oleh H. A Simon [1987]. Kecerdasan buatan (*Artificial Intelligence*) merupakan kawasan penelitian, aplikasi dan instruktur yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas.

Rich and Knht [1991] mendefenisikan kecerdasan buatan (AI) sebagai studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia.

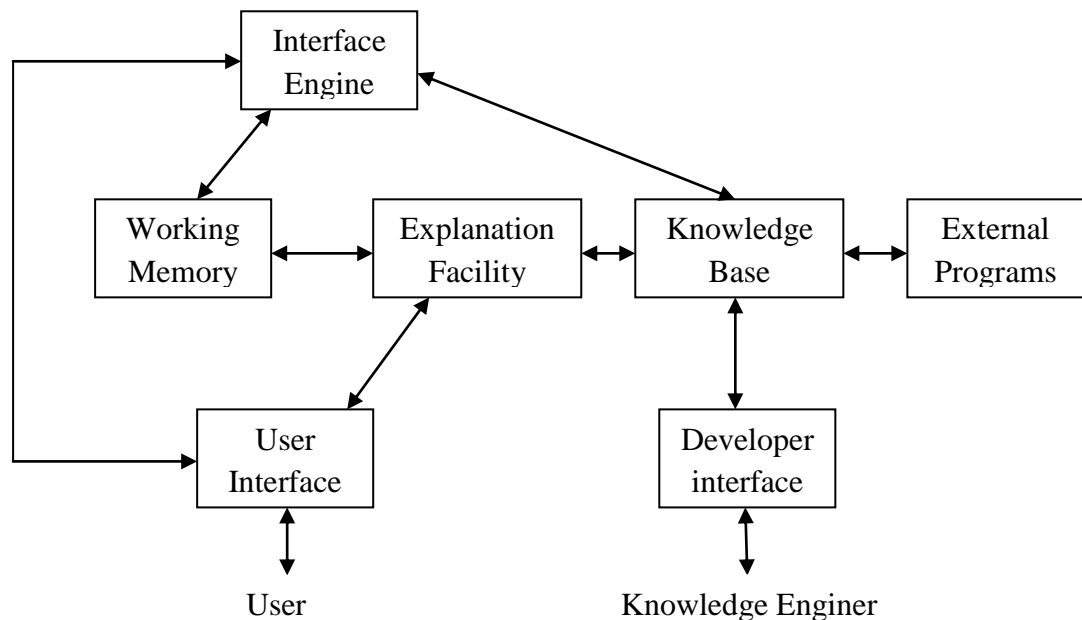
Sementara *ensiklopedia Britnnica* mendefenisikan kecerdasan buatan (AI) sebagai cabang dari ilmu komputer yang dalam merepresentasikan pengetahuan lebih banyak menggunakan bentuk simbol-simbol dari pada bilangan, dan memproses informasi berdasarkan metode *heuristic* atau dengan berdasarkan sejumlah aturan.

Ada tiga tujuan kecerdasan buatan, yaitu: membuat komputer lebih cerdas, mengerti tentang kecerdasan, dan membuat mesin lebih berguna. Yang dimaksud kecerdasan adalah kemampuan untuk belajar atau mengerti dari pengalaman, memahami pesan yang *kontradiktif* menanggapi dengan cepat dan baik atas situasi yang baru, menggunakan penalaran dalam memecahkan masalah serta

menyelesaikannya dengan efektif (Wisnton da Prendergast, 1994). (Sumber: Kusrini, 2006 Hal: 3-4).

II.2. Arsitektur Sistem Pakar

Arsitektur sistem pakar dapat dilihat pada gambar II.1 di bawah ini dimana sebuah sistem pakar terdiri dari tiga modul utama, yaitu: *knowledge base*, *working memory* dan *inference engine* yang merupakan bagian utama dari sebuah sistem pakar. Sedangkan bagian-bagian selain ketiga komponen utama itu adalah : *user interface*, *developer interface*, *explanation facility*, dan *external programs*.(Jurnal Informatika, Andreas Handojo, M. Isa Irwan ,Vol.5 No.1. Mei 2004: 33 - 34)



Gambar II.1 Arsitektur Sistem Pakar

(Sumber : Jurnal Informatika Universitas Kristen Petra Hal: 34)

Keterangan :

1. *Knowledge base* adalah representasi pengetahuan dari seorang atau beberapa pakar yang diperlukan untuk memahami, memformulasikan dan memecahkan masalah. Dalam hal ini digunakan untuk memecahkan masalah-masalah yang terjadi pada komputer. *Knowledge base* ini terdiri dari dua elemen dasar, yaitu fakta dan *rules*.
2. *Inference engine* merupakan otak dari sistem pakar yang mengandung mekanisme fungsi berpikir dan pola-pola penalaran sistem yang digunakan oleh seorang pakar. Mekanisme ini yang menganalisis suatu masalah tertentu dan kemudian mencari solusi atau kesimpulan yang terbaik.
3. *Working Memory* merupakan tempat penyimpanan fakta-fakta yang diketahui dari hasil menjawab pertanyaan.
4. *User/ Developer interface*. Semua *software* pengembangan system pakar memberikan *interface* yang berbeda bagi *user* dan *developer*. *User* akan berhadapan dengan tampilan yang sederhana dan mudah sedangkan *developer* akan berhadapan dengan *editor* dan *source code* waktu mengembangkan program.
5. *Explanation facility* memberikan penjelasan saat mana user mengetahui apakah alasan yang diberikan sebuah solusi.
6. *External programs*. Berbagai program seperti *database*, *spreadsheets*, *algorithms*, dan lainnya yang berfungsi untuk mendukung sistem.

II.3. Aplikasi Sistem Pakar

Sistem pakar (*expert system*) adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para ahli, dengan sistem pakar orang awampun dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli.

1. Ciri-ciri Sistem Pakar.

Suatu sistem dikatakan pakar apabila mempunyai ciri-ciri sebagai berikut:

- a. Terbatas pada domain tertentu.
- b. Dapat memberikan penalaran untuk data-data yang tidak pasti.
- c. Dapat mengemukakan rangkaian alasan-alasan yang diberikan dengan cara yang dapat dipahami.
- d. Berdasarkan pada kaidah atau *rule* tertentu.
- e. Dirancang untuk dikembangkan secara bertahap.
- f. Keluarannya atau *output* bersifat anjuran.

2. Keuntungan Sistem Pakar.

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

- a. Memungkinkan orang awam bisa mengerjakan pekerjaan para ahli.
- b. Bisa melakukan proses secara berulang secara otomatis.
- c. Menyimpan pengetahuan dan keahlian para pakar.
- d. Meningkatkan *output* dan produktivitas.

- e. Meningkatkan kualitas.
- f. Mampu mengambil dan melestarikan keahlian para pakar (termasuk keahlian yang langka).
- g. Mampu beroperasi dalam lingkungan yang berbahaya.
- h. Memiliki pengetahuan untuk mengakses pengetahuan.
- i. Memiliki reabilitas.
- j. Meningkatkan kapabilitas sistem komputer.
- k. Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dan mengandung ketidakpastian.
- l. Sebagai media pelengkap dalam penelitian.

3. Kelemahan Sistem Pakar.

Disamping memiliki beberapa keuntungan, sistem pakar juga memiliki beberapa kelemahan antara lain:

- a. Biaya yang diperlukan untuk membuat dan memeliharanya mahal.
- b. Sulit dikembangkan.
- c. Sistem pakar tidak 100% bernilai benar.

4. Komponen Sistem Pakar.

Untuk membangun sistem pakar komponen-komponen dasar yang harus dimiliki paling sedikit adalah sbagai berikut:

- a. Antar muka pemakai (*User Interface*).
- b. Basis Pengetahuan (*Knowledge Base*).
- c. Mesin Inferensi.

5. Basis Pengetahuan.

Basis Pengetahuan berisi pengetahuan-pengetahuan dalam penyelesaian masalah, tentu saja dalam domain tertentu. Ada dua bentuk pendekatan basis pengetahuan yang sangat umum digunakan yaitu:

a. Penalaran berbasis aturan (*Rule Base Reasoning*)

Pengetahuan direpresentasikan dengan menggunakan aturan berbentuk IF-THEN. Bentuk ini digunakan apabila kita memiliki sejumlah pengetahuan pakar pada permasalahan tertentu, dan pakar dapat menyelesaikan masalah tersebut secara berurutan.

b. Penalaran berbasis kasus (*Case Base Reasoning*)

Basis pengetahuan akan berisi solusi-solusi yang telah dicapai sebelumnya. Kemudian akan diturunkan suatu solusi untuk keadaan yang terjadi sekarang (fakta yang ada).

II.3.1. Mesin Inferensi

Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace*, dan untuk memformalisasikan kesimpulan (*Turban, 1995*).

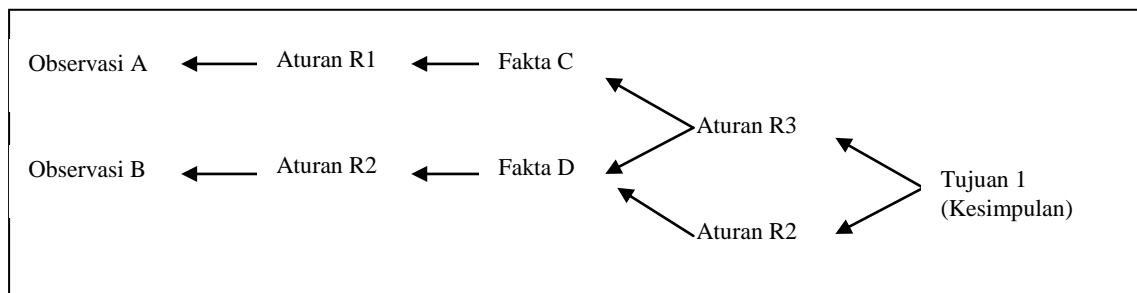
Kebanyakan sistem pakar berbasis aturan menggunakan strategi inferensi yang dinamakan modus ponens. Berdasarkan strategi ini, jika terdapat aturan “IF A

THEN B”, dan jika diketahui bahwa A benar, maka dapat disimpulkan bahwa B juga benar, Strategi inferensi modus ponens dinyatakan dalam bentuk:

$$[A \text{ AND } (A \rightarrow B)] \rightarrow B$$

Dengan A dan $A \rightarrow B$ adalah proposisi-proposisi dalam basis pengetahuan.

Terdapat dua pendekatan untuk mengontrol inferensi dalam sistem pakar berbasis aturan, yaitu runut balik (*backward chaining*) dan runut maju (*forward chaining*). Runut balik adalah pendekatan yang dimotori tujuan (*goal-driven*). Dalam pendekatan ini pelacakan dimulai dari tujuan, selanjutnya dicari aturan yang memiliki tujuan tersebut untuk kesimpulannya. Selanjutnya proses pelacakan menggunakan premis untuk aturan tersebut sebagai tujuan baru sebagai kesimpulan. Proses berlanjut sampai semua kemungkinan ditemukan. Gambar II.2 menunjukkan proses *backward chaining*.

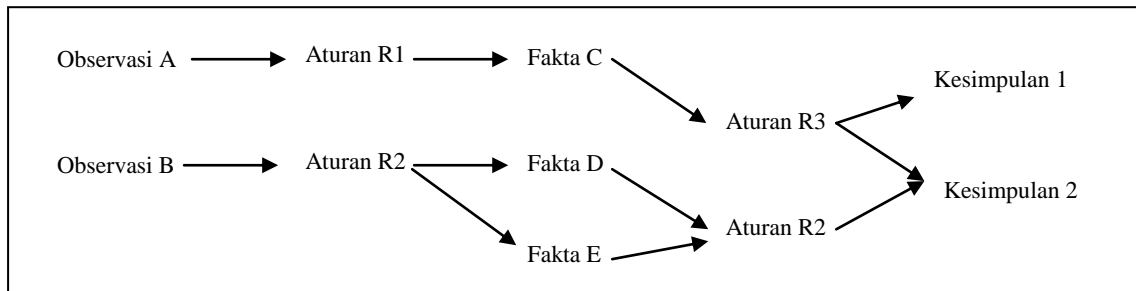


Gambar II.2 Proses Backward Chaining

(Sumber : Muhammad Arhami, 2005 Hal: 19)

Runut maju adalah pendekatan yang dimotori data (*data-driven*). Dalam pendekatan ini pelacakan dimulai dari informasi masukan, dan selanjutnya mencoba menggambarkan kesimpulan. Runut maju mencari fakta yang sesuai

dengan bagian IF dari aturan IF-THEN. Gambar II.3 menunjukkan proses *forward chaining*.



Gambar II.2 Proses Forward Chaining

(Sumber : Muhammad Arhami, 2005 Hal: 20)

II.3.2. Teknik Representasi Pengetahuan

Representasi pengetahuan adalah suatu teknik untuk merepresentasi basis pengetahuan yang diperoleh ke dalam suatu skema/ diagram tertentu sehingga dapat diketahui relasi/ keterhubungan antara suatu data dengan data yang lain. Teknik ini membantu *knowledge engineer* dalam memahami struktur pengetahuan yang akan dibuat sistem pakarnya.

Terdapat beberapa teknik representasi pengetahuan yang bisa digunakan dalam pengembangan suatu sistem pakar, yaitu:

1. *Rule-Based Knowledge*

Pengetahuan direpresentasikan dalam suatu bentuk fakta (*facts*) dan aturan (*rules*). Bentuk representasi ini terdiri atas premis dan kesimpulan.

2. *Frame-Based Knowledge*

Pengetahuan direpresentasikan dalam bentuk hirarki atau jaringan *frame*.

3. *Object-Based Knowledge*

Pengetahuan direpresentasikan sebagai jaringan dari objek-objek. Objek adalah elemen data yang terdiri dari data dan metoda (proses).

4. *Case-Based Knowledge*

Pengetahuan direpresentasikan dalam bentuk kesimpulan kasus (*cases*).

II.3.3 Kategori Masalah Sistem Pakar

Saat ini sistem pakar dibuat untuk memecahkan berbagai macam masalah dalam berbagai bidang. Secara umum ada beberapa kategori dan area permasalahan sistem pakar, yaitu (*Kusrini, 2006*).

1. Interpretasi, yaitu pengambilan keputusan atau deskripsi tingkat tinggi dari sekumpulan data mentah, termasuk di antaranya juga pengawasan, pengenalan pengucapan, analisis citra, interpretasi sinyal, dan beberapa analisis kecerdasan.
2. Prediksi, yaitu memprediksi akibat-akibat yang memungkinkan dari situasi-situasi tertentu, di antaranya peramalan, prediksi demografis, peramalan ekonomis, prediksi lalu lintas, estimasi hasil, militer, pemasaran, atau peramalan keuangan.
3. Diagnosis, yaitu menentukan sebab malfungsi dalam situasi kompleks yang didasarkan pada gejala-gejala yang teramati, di antaranya medis, elektronik, mekanis dan diagnosis perangkat lunak.

4. Desain, yaitu menentukan konfigurasi komponen-komponen sistem yang cocok dengan tujuan-tujuan kinerja tertentu yang memenuhi kendala-kendala tertentu, di antaranya *layout* sirkuit dan perancangan bangunan.
5. Perencanaan, yaitu merencanakan serangkaian tindakan yang akan dapat mencapai sejumlah tujuan dengan kondisi awal tertentu, di antaranya perencanaan keuangan, komunikasi, militer, pengembangan produk, routing dan manajemen proyek.
6. Monitoring, yaitu membandingkan tingkah laku suatu sistem yang teramati dengan tingkah laku yang diharapkan darinya, di antaranya *Computer Aided Monitoring System*.
7. *Debugging* dan *repair*, yaitu menentukan dan mengimplementasikan cara-cara untuk mengatasi malfungsi, di antaranya memberikan resep obat terhadap suatu kegagalan.
8. Instruksi, yaitu mendeteksi dan mengoreksi defisiensi dalam pemahaman domain subjek, di antaranya melakukan instruksi untuk diagnosis, *debugging* dan perbaikan kinerja.
9. Pengendalian, yaitu mengatur tingkah laku suatu *environment* yang kompleks seperti kontrol terhadap interpretasi-interpretasi, prediksi, perbaikan dan monitoring kelakuan sistem.
10. Seleksi, yaitu mengidentifikasi pilihan terbaik dari sekumpulan (*list*) kemungkinan.
11. Simulasi, yaitu pemodelan interaksi antara komponen-komponen sistem.

II.4. Sistem Pemanas Air Listrik (*Water Heater*)

Water heater adalah perangkat elektronika rumah tangga yang digunakan untuk mengubah air dingin menjadi air panas melalui rangkaian yang dirancang sebaik mungkin dalam sebuah perangkat elektronika water heater yang digunakan sebagai kebutuhan rumah tangga seperti mandi.

Dalam pemanas air dikontrol hydraulical dua tahap tankless, berjalan dengan single-phase power listrik. Tangki berisi tembaga pemanas dengan 18 kilowatt daya maksimum, tankless pemanas air disebut juga aliran kontinu, flash, on-demand atau instant-on pemanas air. Pemanas air langsung memanaskan air ketika mengalir melalui perangkat dan tidak menahan air internal apapun kecuali apa yang ada dalam kumparan penukaran panas. Bagaimana tankless pemanas air berkerja, yaitu ketika ada permintaan untuk air (misal keran air panas dibuka untuk wastafel, shower, bak, atau mesin cuci) air tankless pemanas air induksi turbin aliran memulai proses pemanasan. Turbin aliran air mengirim sinyal ke panel kontrol yang terlihat pada beberapa faktor yaitu suhu air masuk, suhu air yang diinginkan sebagaimana pada kontroler suhu, dan perbedaan dihitung antara dua suhu, tergantung pada suhu air yang masuk. Aliran listrik/ gas ke dalam perakitan burner dimodulasi dan urutan pengapian dimulai. Air dipanaskan ke suhu yang diinginkan karena beredar melalui penukar panas tembaga menyediakan air panas terus menerus, dan ketika keran air panas dimatikan, tankless pemanas air mati dan ditempatkan dalam modus siaga menunggu panggilan berikutnya untuk air panas. Ada beberapa masalah yang sering terjadi dalam mesin pemanas air yaitu :

1. Pemanas listrik gagal panas terkirim di mesin pemanas air
2. Kekuatan pemanas terganggu dengan kontrol/ elemen pemanas

Cara penanganannya adalah :

1. Pastikan arus listrik mengalir,periksa saklar utama pada pemanas air dan pemutus sirkuit (atau sekring) yang berfungsi sebagai pemanas air, bila saklar utama terputus reset dengan membalik jalan *OFF* dan kemudian *ON*, ganti sekering dengan yang baru dengan ranting yang sama. Periksa cutoff suhu tinggi dalam pemanas air dan buka panel sambil menekan tombol reset bila tombol tidak bersuara klik dan tidak memiliki kekuatan untuk menekan, maka suhu tinggi memburuk.
2. Bila air dalam kompartemen menyebabkan thermostat tidak berfungsi maka kebocoran elemen perlu diganti.(*Sumber : Jurnal Reeves, Hal: 3 - 4*)

II.5. Aplikasi Web

Yang dimaksud dengan aplikasi *web* atau aplikasi berbasis *web* (*Web-Jubased application*) adalah aplikasi yang dijalankan melalui *browser*. Aplikasi seperti ini pertama kali dibangun hanya dengan menggunakan bahasa yang disebut HTML (*HyperText Markup Language*) dan protokol yang digunakan dinamakan HTTP (*HyperText Transfer Protocol*). Namun, tentu saja hal seperti ini memiliki kelemahan. Semua perubahan harus dilakukan pada level aplikasi. Pada perkembangan berikutnya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan HTML. Pada saat ini, banyak skrip seperti itu, antara

lain yaitu PHP, ASP, ASP.NET, sedangkan contoh yang berupa objek antara lain adalah *applet* (Java).

Dengan memperluas kemampuan HTML, yakni dengan menggunakan perangkat lunak tambahan, perubahan informasi dalam halaman-halaman *web* dapat ditangani melalui perubahan data, bukan melalui perubahan program. Sebagai implementasinya, aplikasi *web* dapat dikoneksikan ke *database*. Dengan demikian, perubahan informasi dapat dilakukan oleh *operator* atau yang bertanggung jawab terhadap kemutakhiran data, dan tidak menjadi tanggung jawab pemrogram atau *web master*.

Konsep yang mendasari aplikasi *web* sebenarnya sederhana. Operasi yang melatarbelakanginya melibatkan pertukaran informasi antara komputer yang meminta informasi, yang disebut klien, dan komputer yang memasok informasi disebut *server*. Secara lebih detail, *server* yang melayani permintaan dari klien sesungguhnya berupa perangkat lunak yang dinamakan *web server*. Secara internal, *web sever* inilah yang berkomunikasi dengan perangkat lunak lain yang disebut *middleware*, daqn *middleware* inilah yang berhubungan dengan *database*. Model seperti inilah yang mendukung *web* dinamis.

Web server adalah *server* yang melayani permintaan klien terhadap halaman *web*. Apache, IIS (*Internet Information Server*), dan Xitami merupakan contoh perangkat lunak *web server*.

Middleware adalah perangkat lunak yang bekerja sama dengan *web server* dan berfungsi menerjemahkan kode-kode tertentu, menjalankan kode-kode

tersebut, dan memungkinkan berinteraksi dengan basis data. PHP, ASP, dan Perl adalah beberapa contoh *middleware*.

Browser atau *web browser* adalah perangkat lunak di sisi klien yang digunakan untuk mengakses informasi *web*. *Internet Explorer*, *Netscape*, dan *Mozilla* merupakan contoh *browser*. (Sumber : Abdul Kadir, 2008, hal: 356-357)

II.6. Database

Database sering didefinisikan sebagai kumpulan data yang terkait. Secara teknis, yang berada dalam sebuah *database* adalah sekumpulan tabel atau objek lain (*indeks*, *view*, dan lain-lain). Tujuan utama pembuatan *database* adalah untuk memudahkan dalam *mengakses* data. Data dapat ditambahkan, diubah, dihapus, atau dibaca dengan relatif mudah dan cepat.

Sebuah tabel (atau kadang disebut relasi) berisi sejumlah baris dan kolom menyatakan sebuah data. Saat ini tersedia banyak perangkat lunak yang ditujukan untuk mengelola *database*. Perangkat lunak seperti itu biasa dinamakan DBMS (*Database Management System*). *Access*, *Ms SQL Server*, dan *MySQL* merupakan kelas *database server*, yaitu jenis yang secara aktif memantau permintaan akses terhadap data. Dalam hal ini, *database server* akan segera menanggapi permintaan data. Adapun yang bukan termasuk *database server* adalah *Access*.

Pengaksesan data dalam *database* dapat dilakukan dengan mudah melalui SQL (*Structured Query Language*). Data dalam *database* bisa diakses melalui aplikasi non-*web* (misalnya dengan *Visual Basic*) maupun aplikasi *web* (misalnya dengan PHP). (Sumber : Abdul Kadir, 2009, hal:14-15)

Secara umum, *database* berarti koleksi data yang saling terkait. Secara praktis, basis data dapat dianggap sebagai suatu penyusunan data yang terstruktur yang disimpan dalam media penganal (*hard disk*) yang tujuannya adalah agar data tersebut dapat diakses dengan mudah dan cepat.

Ada beberapa macam *database*, antara lain yaitu *database* hierarkis, *database* jaringan, dan *database* relasional. *Database* relasional merupakan *database* yang populer saat ini dan telah diterapkan pada berbagai *platform*, dari PC hingga mini komputer.

Sebuah *database* relasional tersusun atas sejumlah tabel. Sebagai contoh, *database* akademis mencakup tabel-tabel seperti dosen, mahasiswa, KRS, nilai, dan lain-lain. Basis data tentang bintang film bisa mencakup info pribadi (nama, jenis kelamin, tanggal lahir, dan sebagainya) dan film-film yang telah dibintangi.

(Sumber : Abdul Kadir, 2008, hal: 3)

II.7. Perangkat Lunak Pendukung

Perangkat lunak pendukung digunakan sebagai alat untuk membantu penulis dalam proses pembuatan program sistem pakar diagnosis kerusakan *water heater*. Perangkat lunak yang digunakan oleh penulis diantaranya yaitu :

II.8. PHP (Hypertext Preprocessor)

PHP merupakan singkatan dari *Hypertext Preprocessor* serta merupakan bahasa pemrograman skrip yang ditempatkan dalam *server* dan diproses di server. Hasilnya dikirimkan ke klien, tempat pemakai menggunakan *browser*.

PHP dirancang untuk membuat aplikasi *web* yang bersifat dinamis. Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini, menampilkan isi *database* kehalaman *web* pada prinsipnya PHP mempunyai fungsi yang sama dengan skip-skip seperti ASP (*Active Server Page*) . PHP bermula saat Ramus Lerdorp membuat sejumlah skrip *Perl* yang dapat mengamati siapa saja yang meliha t-lihat daftar riwayat hidupnya, yakni pada tahun 1994. Skrip-skip ini selanjutnya dikemas menjadi tool yang disebut “Personal Home Page” Paket inilah yang menjadi cikal-bakal PHP. Pada tahun 1995, Rasmus Lerdorp menciptakan PHP/FI Versi 2. Pada versi inilah pemrograman dapat menempelkan kode terstruktur didalam tag HTML, kode PHP bias berkomunikasi dengan *database* dan melakukan perhitungan-perhitungan yang kompleks. Pada awalnya, PHP dirancang untuk diintegrasikan dengan web server Apache. Namun, belakangan PHP juga dapat berkerja dengan web server seperti PWS (*Personal Web Server*), ISS (*Internet Information Server*), dan *Xitami*.

Skrip PHP berkedudukan sebagai tag dalam bahasa HTML, sebagaimana diketahui, HTML (*Hypertext Markup Language*) adalah bahasa standart untuk membuat halaman-halaman web. (*Sumber : Abdul Kadir, 2008, Hal: 2 - 3*)

PHP adalah program aplikasi yang bersifat *Server Slide*, artinya hanya dapat berjalan pada sisi server saja dan tidak dapat berfungsi tanpa adanya sebuah

server didalamnya dan merupakan sebuah bahasa pemrograman scripting berlisensi *Open Source*. Script ini dapat bercampur dengan Script Tag HTML sehingga karena kemampuannya tersebut, ia disebut sebagai bahasa yang *embeded* pada Tag HTML. (Sumber : *Bunafit Nugroho, 2005, Hal: 369 - 370*)

II.9. MySQL (My Structure Query Language)

MySQL adalah salah satu jenis *database server software* yang sangat terkenal, kepopulerannya disebabkan *MSQL* menggunakan *SQL* sebagai bahasa dasar untuk mengakses *databasenya* dan bersifat *Open Source* (Anda tidak perlu membayar untuk menggunakannya) perangkat lunak sendiri bisa di-download dari <http://www.mysql.com>. *MYSQL* termasuk jenis *RDBMS (Relational Database Management System)*, itu sebabnya, istilah seperti table, baris dan kolom digunakan pada *MYSQL*. Pada *MYSQL* sebutan database mengandung satu atau sejumlah table. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

MYSQL menyediakan program *mysql* yang berfungsi untuk mengakses database server *MYSQL* dari sisi klien.(Sumber: *Abdul Kadir, 2008, Hal: 348*)

II.10. Apache

Web server Apache berbasiskan *open source* dan mulai populer di *internet* semenjak tahun 1996 karena *open source*, *Apache* bebas didistribusikan oleh siapa saja dan ke siapa saja.

Software ini dapat diunduh pada situs <http://www.apache.org> dan tersedia untuk berbagai *platform* (*Windows, Linux, dan UNIX*). Supaya dokumen-dokumen *web*, baik berupa HTML ataupun PHP bisa diakses oleh *browser* maka dokumen-dokumen tersebut perlu diletakkan dalam direktori khusus yang diatur oleh Apache. (*Sumber : Abdul Kadir, 2008, hal. 360*)

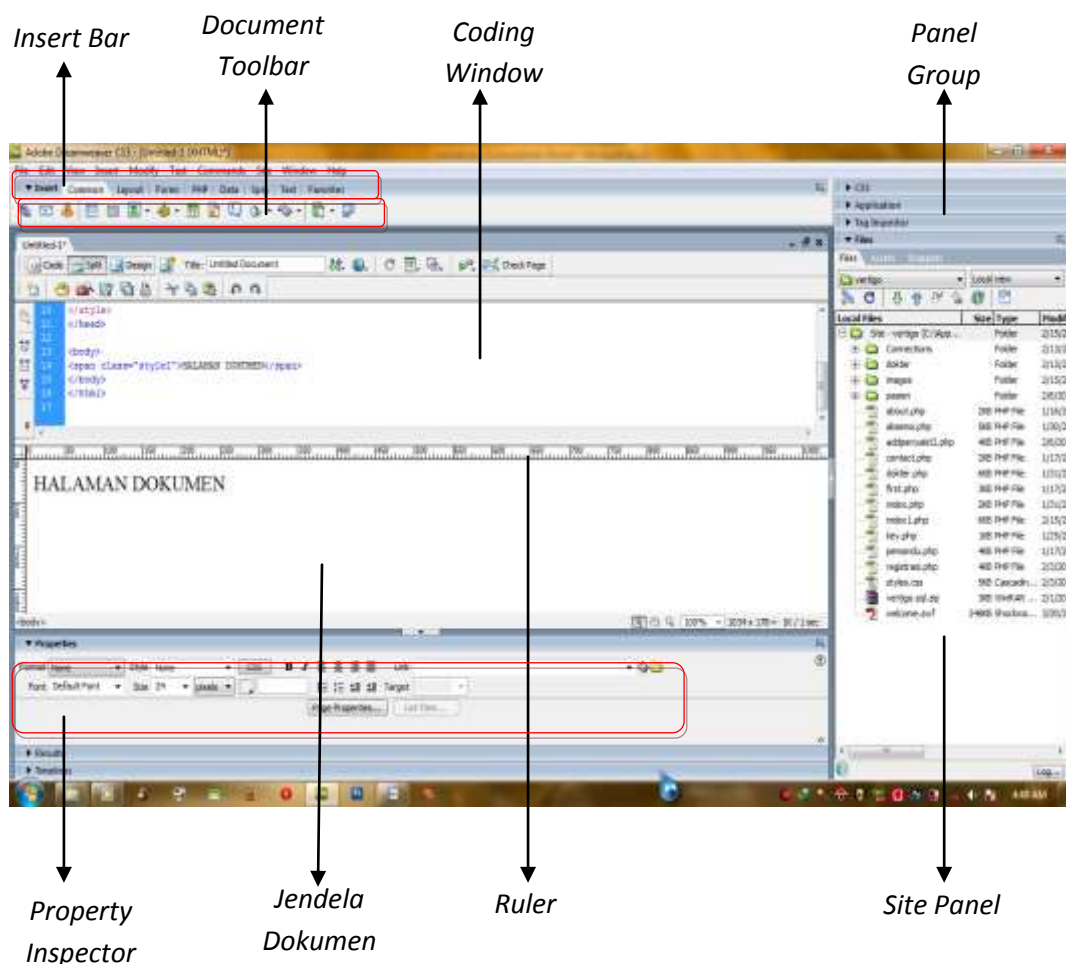
II.11. Dreamweaver CS3

Dreamweaver adalah sebuah HTML editor profesional untuk mendesain secara visual dan mengelola situs *web* maupun halaman *web*. Pada *Dreamweaver CS3*, terdapat beberapa kemampuan bukan hanya sebagai software untuk desain *web* saja tetapi untuk menyunting kode-kode serta pembuatan aplikasi web dengan menggunakan berbagai bahasa pemrograman *web* seperti: JPS, PHP, ASP, dan *ColdFusion*. *Dreamweaver* merupakan *software* utama yang digunakan oleh *web* desainer maupun web programmer dalam mengembangkan suatu situs *web*. Hal ini disebabkan ruang kerja, fasilitas dan kemampuan *Dreamweaver* yang mampu meningkatkan produktivitas dan efektivitas desain maupun suatu situs web.

Dreamweaver CS3 memiliki kemampuan *toolbar*, dimana *Dreamweaver CS3* dapat digunakan untuk memodifikasi tampilan *toolbar* atau menambahkan fungsi baru. Selain *user interface* baru, *Dreamweaver CS3* memiliki kemampuan untuk menyunting kode dengan lebih baik, *Dreamweaver CS3* juga dapat melakukan print kode pada jendela *code view*, selain itu juga memiliki fasilitas *code hints* yang membantu dalam urusan tag-tag serta tag *inspector* yang sangat berguna

dalam menangani tag-tag HTML. Ruang kerja *Dreamweaver CS3* memiliki komponen-komponen yang memberikan fasilitas dan ruang untuk menuangkan kreasi saat berkerja. Komponen-komponen yang tersedia oleh ruang kerja.

Window Dreamweaver CS3 ini dibagi menjadi 7 bagian, yaitu : *Insert Bar*, *Document Toolbar*, *Document Window*, *Panel Groups*, *Tag Selector*, *Property Inspector* dan *Site Panel*.



Gambar II.4 Tampilan ruang kerja tipe coder

(Sumber : Madcoms, 2008, Hal: 3-4)

Keterangan Gambar :

1. *Insert bar*, memuat tombol-tombol yang berfungsi untuk memasukkan/ menyisipkan berbagai jenis obyek seperti gambar, tabel dan layer ke dalam suatu dokumen. Setiap obyek yang dimasukkan dengan meng-klik tombol *insert* pada *insert bar* ini adalah seperti halnya memasukkan potongan tag HTML ke dalam halaman yang sedang dibuat.
2. *Document window*, berfungsi untuk menampilkan dokumen di mana kita sekarang bekerja.
3. *Document toolbar*, berisi tombol dan menu *pop-up* yang menyediakan tampilan yang berbeda-beda dari *Document Window*.
4. *Panel groups*, adalah kumpulan panel yang saling berkaitan satu sama lain, yang dikelompokkan di bawah satu judul.
5. *Tag selector*, berfungsi untuk menampilkan *hierarki tag* di sekitar pilihan yang aktif pada *Design View*.
6. *Property inspector*, berfungsi untuk melihat dan mengubah berbagai *property* obyek atau teks.
7. *Files panel*, berfungsi pengaturan file-file atau folder-folder yang membentuk situs web/ direktori kerja.

II.12. Data Flow Diagram (DFD)

Pendekatan analisis terstruktur diperkenalkan oleh *DeMarco (1978)* dan *Gane Sarson (1979)* melalui buku metodologi struktur analisis dan desain sistem

informasi. Mereka menyarankan untuk menggunakan data flow diagram (DFD) dalam menggambarkan atau membuat model sistem.

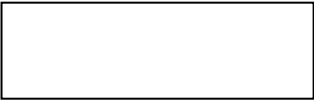
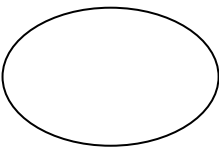
Data Flow Diagram (DFD) adalah suatu network yang menggambarkan suatu system automat/komputerisasi, manualisasi atau gabungan dari keduanya, yang penggambarannya di susun dalam bentuk kumpulan komponen sistem yang saling berhubungan sesuai dengan aturan mainnya. (*Tata Sutabri, S.Kom, 2004: 163*).


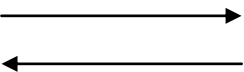
Ada beberapa tahapan dalam pembuatan DFD yaitu:

- a. Diagram Konteks; digunakan untuk menggambarkan sistem secara umum/global dari keseluruhan sistem yang ada.
- b. Diagram Level nol; digunakan untuk menggambarkan tahapan proses yang ada di dalam diagram konteks, yang penjabarannya lebih terperinci.
- c. Diagram Detil (level satu); digunakan untuk menggambarkan arus data secara lebih mendetail lagi dari tahapan proses yang ada di dalam diagram nol. (*Tata Sutabri, S.Kom, 2004: 166*).

Simbol-simbol yang digunakan dalam DFD adalah :

Tabel II.I Simbol-Simbol DFD

No	Simbol	Fungsi
1		External Entity, Simbol ini digunakan untuk menggambarkan asal atau tujuan data.
2		Atribut/ Proses, Simbol ini digunakan untuk proses pengolahan atau transformasi data.




3		Data Storage (penyimpanan data), Simbol ini digunakan untuk menggambarkan data flow yang sudah disimpan atau diarsipkan.
4		Data Flow(arus data), Simbol ini digunakan untuk menggambarkan aliran data yang berjalan.

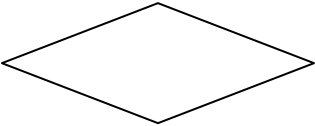

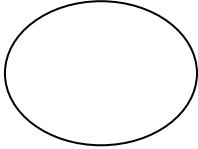
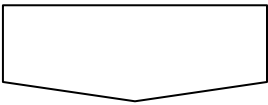
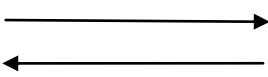
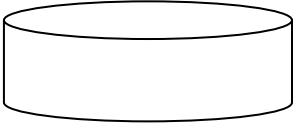
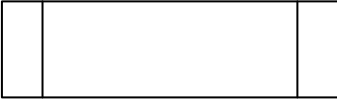

(Sumber : Tata Sutabri, S.Kom, 2004, Hal: 163)

II.13. Flowchart

Flowchart merupakan diagram alir yang menggambarkan urutan logika dari suatu prosedur pemecahan masalah. Dalam menggambarkan program *flowchart*, telah tersedia simbol-simbol standar, tetapi seperti pada sistem *flowchart*, pemrograman juga harus melengkapi penggambaran program *flowchart* dengan kamus simbol. Berikut ini adalah gambar dari simbol-simbol standar yang digunakan pada program *flowchart*. (Budi Sutedjo Dharma Oetomo, S.Kom; 2006: 128).

Tabel II.2 Simbol-Simbol *Flowchart*

No	Simbol	Fungsi
1		Terminal, untuk memulai dan mengakhiri suatu proses/ kegiatan
2		Proses, Suatu yang menunjukkan setiap pengolahan yang dilakukan oleh komputer
3		Input, untuk memasukan hasil dari suatu proses

4		Decision, Suatu kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau pilihan
5		Display, output yang ditampilkan dilayar terminal
6		Connector, suatu prosedur akan masuk atau keluar melalui simbol ini dalam lembar yang sama
7		Off Page Connector, merupakan symbol masuk atau keluarnya suatu prosedur pada kertas lembar lain
8		Arus? Flow, simbol ini digunakan untuk menggambarkan arus proses dari suatu kegiatan kegiatan lain
9		Hard Disk Storage, input/output yang menggunakan hardisk
10		Predefied Process, untuk menyatakan sekumpulan langkah proses yang ditulis sebagai prosedur
11		Stored Data, Input/ output yang menggunakan disket

(Sumber : Budi Sudtejo Dharma Oetomo, S.Kom, 2006, Hal: 127 - 128)

