BAB II

LANDASAN TEORI

2.1. Sistem Pakar

Sistem pakar merupakan sebuah sistem berbasis komputer yang menggunakan pengetahuan, fakta dan teknik penalaran yang dimiliki manusia sebagai pakar yang tersimpan di dalam komputer, dan digunakan untuk menyelesaikan masalah yang lazimnya memerlukan pakar tertentu. Sedangkan menurut pengertian lainnya. Sistem pakar (*expert system*) adalah sistem yang berusaha mengapdosi 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. Sistem pakar menanyakan fakta – fakta dan dapat membuat penalaran (*inferensi*) dan sampai pada suatu kesimpulan. Kemudian, sistem pakar memberikan penjelasan (memberikan kesimpulan dari hasil konsultasi yang telah dilakukan sebelumnya). (Dahria, 2013).

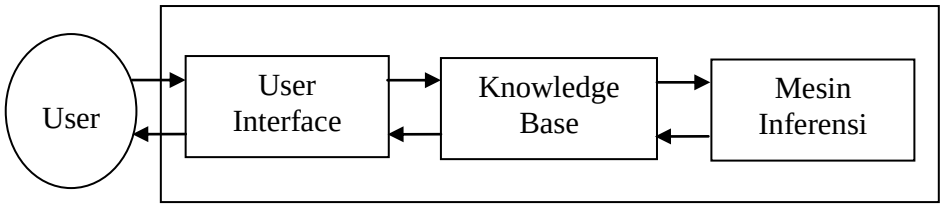
Dengan sistem pakar ini, orang awam pun dapat menyelesaikan masalahnya atau sekedar mencari suatu informasi berkualitas yang sebenarnya hanya dapat diperoleh dengan bantuan para ahli dibidangnya. Sistem pakar ini juga dapat membantu aktivitas para pakar sebagai asisten yang mempunyai pengetahuan yang dibutuhkan.

Pada dasarnya sistem pakar dirancang untuk mendukung aktivitas pemecahan masalah. Beberapa aktivitas pemecahan yang dimaksud seperti

pembuatan keputusan (*decision making*), pemanduan pengetahuan (*knowledge fusing*), pembuatan desain (*designing*), perencanaan (*planning*), prakiraan (*forescating*), pengaturan (*regulating*), pengendalian (*controlling*), diagnosa (*diagnosing*), perumusan (*prescribing*), penjelasan (*explaining*), pemberian nasihat (*advising*) dan pelatihan (*tutoring*).

2.2. Struktur Sistem Pakar

Sistem Pakar terdiri dari 3 komponen utama, yaitu : basis pengetahuan, motor *inferensi* dan *user inferensi.* Untuk lebih jelasnya, diagram blok umum *expert system* dapat dilihat pada gambar 2.1 dibawah ini :



Gambar 2.1. Diagram Blok Umum *Expert System*

Sistem pakar biasanya mengajukan pertanyaan-pertanyaan tertentu sampai dapat mengindentifikasi suatu objek yang sesuai dengan informasi yang diketahuinya. Ini merupakan bagian *software* spesialisasi tingkat tinggi yang berusaha menduplikasi fungsi suatu pakar dalam suatu bidang keahlian. Program ini bertindak sebagai penasehat atau konsultan dalam suatu lingkup keahlian tertentu, sebagai hasil pengetahuan yang telah dikumpulkan dari beberapa orang pakar.

Sistem pakar harus memiliki ciri-ciri sebagai berikut :

1. Memiliki fasilitas yang handal
2. Mudah dimodifikasi

3. Dapat digunakan dalam berbagai jenis komputer.
4. Memiliki komputer untuk belajar beradaptasi.

2.3. Keuntungan dan kelemahan sistem pakar

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

1. Membuat seorang awam dapat bekerja seperti layaknya seorang pakar.
2. Dapat bekerja dengan informasi yang tidak lengkap atau tidak pasti.
3. Meningkatkan *output* dan produktivitas.
4. Meningkatkan kualitas.
5. Menyediakan nasihat atau solusi yang konsisten dan dapat mengurangi tingkat kesalahan.
6. Membuat peralatan yang kompleks dan mudah dioperasionalkan karena sistem pakar dapat melatih pekerja yang tidak berpengalaman.
7. Sistem tidak dapat lelah atau bosan.
8. Merupakan panduan yang cerdas.
9. Memungkinkan pemindahan pengetahuan ke lokasi yang jauh serta memperluas jangkauan seorang pakar, dan dapat diperoleh atau dipakai dimana saja.

Selain memiliki keuntungan, sistem pakar juga memiliki beberapa kelemahan di dalam penerapannya, antara lain :

1. Daya kerja dan produktivitas manusia menjadi berkurang karena semuanya dilakukan secara otomatis oleh sistem.

2. Pengembangan perangkat lunak sistem pakar lebih sulit dibandingkan dengan perangkat lunak konvensional.
3. Biaya pembuatan mahal, karena seorang pakar membutuhkan pembuat aplikasi untuk membuat sistem pakar yang diinginkan.

2.4. *Certainty Factor*

Dalam menghadapi suatu masalah, sering ditemukan jawaban yang memiliki kepastian penuh. Ketidakpastian ini biasa berupa probabilitas atau kebolehjadian yang bergantung pada hasil suatu kejadian. Hasil yang tidak pasti disebabkan oleh dua faktor, yaitu aturan yang tidak pasti dan jawaban yang tidak pasti. Hal ini sangat mudah dilihat pada sistem diagnosis penyakit, dimana pakar tidak dapat mendefinisikan hubungan antara gejala dengan penyebabnya secara pasti. Pada akhirnya ditemukan banyak kemungkinan penyakit.

Metode statistik standar didasari pada asumsi bahwa ketidakpastian adalah probabilitas dari sebuah kejadian/fakta adalah benar atau salah. Dalam teori kepastian (*Certainty Factor*), sama halnya dengan *fuzzy logic*, ketidakpastian direpresentasikan dengan derajat kepercayaan.

Faktor Kepastian (*Certainty Faktor*) merupakan sebuah teknik algoritma untuk mengukur seberapa besar nilai tingkat keyakinan hipotesis terhadap suatu hasil kesimpulan pengetahuan yang diberikan oleh pakar pada aplikasi sistem pakar. CFs mengekspresikan kepercayaan dalam hipotesis berdasarkan penilaian seorang pakar. (Sari, 2013)

Certainty Factor (CFs) menunjukan ukuran tingkat kepastian/ketidakpastian terhadap suatu fakta / Rule yang berlaku dengan pendekatan dari metode *Certainty Factor* sebagai berikut :

Notasi faktor kepastian (Yastita, 2012) :

CFs[h,e] = MB[h,e] – MD[h,e]……(1)

Keterangan :

CFs[h,e] = *Certainty Factor* (Faktor Kepastian) dalam Hipotesa H dipengaruhi oleh fakta E

MB[h,e] = Ukuran kepercayaan atau tingkat keyakinan terhadap hipotesis h, dipengarui oleh *evidence* (e) antara 0 dan 1

MD[h,e] = Ukuran ketidakpercayaan atau tingkat hipotesis terhadap h, dipengarui oleh *evidence* (e) antara 0 dan 1

Untuk menentukan nilai *Certainty Factor* dengan beberapa evidence, dibutuhkan nilai MB dan MD. Kedua nilai tersebut ditentukan oleh seorang pakar dengan menggunakan :

MB(h, e1^e2) = MB(h,e1)+MB(h,e2)*(1-MB[h,e1])…..(2)

MD(h,e1^e2) = MD(h,e1)+MD(h,e2)*(1-MD[h,e1])…...(3)

Keterangan :

CFs [h,e] = *Certainty Factor* (Faktor Kepastian) dalam Hipotesa H yang dipengaruhi oleh fakta E

MB [h,e] = *Measure of Belief* (Kepastian),merupakan ukuran kenaikan dari kepercayaan hipotesa H dipengaruhi oleh fakta E.

MD[h,e] = *Measure of Disbelief* (Ketidakpastian),merupakan ukuran kenaikan dari ketidakkepercayaan hipotesa H dipengaruhi oleh fakta E.

E = *Evidence* (peristiwa atau fakta)

H = Hipotesa (dugaan).

Bentuk dasar rumus *Certainty Factor* sebuah aturan JIKA E MAKA H adalah seperti yang ditunjukan oleh persamaan 4 berikut :

$$CF(H,e) = CF(E,e) * CF(H,E) \quad \text{.....(4)}$$

Dimana :

CF (H,e) : *Certainty Factor* hipotesis yang dipengaruhi oleh *evidence* e.

CF(E,e) : *Certainty Factor evidence* E yang dipengaruhi oleh *evidence* e.

CF(H,E) : *Certainty Factor* hipotesis yang dipengaruhi oleh *evidence* e.

Jika semua *evidence* pada *antecedent* diketahui dengan pasti maka persamaannya akan menjadi :

$$CF(H,e) = CF(H,E) \quad \text{.....(5)}$$

Dalam aplikasinya, CF(H,E) merupakan nilai kepastian yang diberikan oleh pakar terhadap sebuah aturan, sedangkan CF(E,e) merupakan nilai kepercayaan yang diberikan oleh pengguna terhadap gejala yang dialami.

Dalam CF dikenalkan konsep *Measures of Belief* (MB) atau ukuran kepercayaan dan *Measures of Disbelief* (MD) atau ukuran ketidakpercayaan. Dalam memberikan ukuran MB, MD, dan CF, tim MYCIN mempunyai parameter untuk menujukkan ukuran kepercayaan. Berikut tabel 2.1 aturan nilai-nilai kepercayaan dan tabel 2.2 nilai interpretasi untuk MB dan MD yang diberikan oleh MYCIN.

Tabel 2.1 Aturan nilai-nilai kepercayaan

Kepercayaan	CF
Tidak Pasti	-1,0 sampai -0,79
Hampir Tidak Pasti	-0,8 sampai -0,59
Kemungkinan Tidak	-0,6 sampai -0,39
Mungkin Tidak	-0,4 sampai -0,19
Tidak Tahu	-0,2 sampai 0,2
Mungkin	0,4 sampai 0,59
Kemungkinan Besar	0,6 sampai 0,79
Hampir Pasti	0,8 sampai 0,89
Pasti	0,9 sampai 1,0

Tabel 2.2 Nilai interpretasi untuk MB dan MD

Kepercayaan	MB / MD
Tidak tahu	0 – 0,29
Mungkin	0,3 – 0,49
Kemungkinan Besar	0,5 – 0,69
Hampir Pasti	0,7 – 0,89
Pasti	0,9 – 1,0

2.5. PHP

PHP atau *Hypertext Preprocessor*, merupakan bahasa script yang digunakan untuk membuat halaman WEB yang dinamis. PHP bersifat open source product. Pengguna dapat merubah source code dan mendistribusikannya secara bebas serta diedarkan secara gratis. PHP bersifat server side scripting yang dapat ditambahkan ke dalam HTML,

sehingga suatu halaman WEB tidak lagi bersifat statis, namun bersifat dinamis.

PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (*Form Interpreter*), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari WEB. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI, kependekan dari Personal Home Page/Form Interpreter. Dengan perilisan kode sumber ini menjadi open source, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP.

Pada tahun 1997, sebuah perusahaan bernama zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada juni 1998, perusahaan tersebut merilis interpreter terbaru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP daru dan rilis tersebut dikenal dengan PHP 4.0 adalah versi PHP yang banyak dipakai pada awal abad ke-21.Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi WEB kompleks tetapi memiliki kecepatan dan stabilitas yang tinggi.

Pada juni 2004, Zend merilis PHP 5.0.dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga mendukung penuh model pemrograman berorientasi objek (PBO), integrasi XML, mendukung semua ekstensi terbaru MySQL. (Riyanto, 2014).

2.6. MySQL

MySQL adalah salah satu perangkat lunak (*Software*) sistem manajemen database relasi (*Relation Database Management System*) yang bersifat "terbuka" (*open source*), artinya bebas untuk digunakan, diedarkan, maupun dikembangkan kembali oleh siapa saja tanpa harus khawatir dengan hak cipta, MySQL merupakan hasil buah pikiran dari Michael "monty" Widenius, David Axmark, dan Allan Larson dimulai tahun 1995. Mereka bertiga kemudian mendirikan perusahaan bernama MySQL AB Swedia.

Tujuan awal didirikannya program MySQL adalah untuk mengembangkan aplikasi WEB yang digunakan salah satu client MySQL AB. Pada saat itu MySQL AB adalah sebuah perusahaan konsultan database dan pengembangan software.

MySQL versi 1.0 dirilis pada Mei 1996 dan penggunaannya hanya terbatas dikalangan internal saja. Pada bulan Oktober 1996, MySQL versi 3.11.0 dirilis kemasyarakat luas dibawah lisensi "terbuka tapi terbatas". Dengan lisensi ini, maka siapapun boleh melihat program aslinya dan menggunakan server MySQL secara gratis untuk kegiatan–kegiatan non komersial. Tetapi, untuk kegiatan komersial, maka harus membayar lisensi tersebut.

Pada bulan juni 2000, MySQL AB mengumumkan bahwa mulai versi 3.23.19 diterapkan sebagai *General Public License* (GPL). Dengan lisensi ini, maka siapapun boleh melihat program aslinya dan menggunakan program executablenya secara open source atau gratis. (Nugroho, 2012).

2.7. Adobe Dreamweaver

Adobe Dreamweaver adalah sebuah editor HTML professional yang digunakan untuk mendesain secara visual dan mengelola situs WEB maupun halaman WEB. Adobe Dreamwever adalah salah satu produk dari vendor macromedia Inc.

Adobe Dreamweaver memiliki kemampuan untuk menyunting kode dengan lebih baik, serta mampu menggabungkan desain layout site dengan kode programing webnya. Kehebatan Dreamweaver ini menjadi Dreamweaver banyak digunakan oleh WEB Desainer maupun WEB Programmer guna mengembangkan situs WEB. Ruang kerja, fasilitas, dan kemampuan Dreamweaver mampu meningkatkan produktivitas dan efektifitas dalam desain maupun membangun situs WEB. (Andi, 2013).

2.8. Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram Konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke sistem atau output dari sistem. Dalam diagram konteks ini akan memberi gambaran tentang keseluruhan sistem. (Afyenni, 2013).

2.9. *Data Flow Diagram* (DFD)

Data Flow Diagram (DFD) disebut juga dengan Diagram Arus Data (DAD). DFD adalah: suatu model logika data atau proses yang dibuat untuk menggambarkan: darimana asal data, dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data

tersebut, dan interaksi antara data yang tersimpan, dan proses yang dikenakan pada data tersebut (S.Pressman, 2012).

2.10. *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem secara abstrak. ERD juga menggambarkan hubungan antara satu entitas yang memiliki sejumlah atribut dengan entitas yang lain dalam suatu sistem yang terintegrasi. *Entity Relationship Diagram* sering disebut dengan ERD dengan tujuan untuk menghubungkan antara suatu tabel dengan tabel yang lain yang masih berhubungan dari table yang dibuat. (Yakub, 2012).

ERD menggunakan sejumlah simbol untuk menggambarkan struktur dan hubungan antar data. Pada dasarnya ada tiga komponen yang digunakan, yaitu :

a. *Entity*

Entity merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari *entity* ini biasanya digambarkan dengan persegi panjang.

b. Atribut

Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasikan isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol *elips*.

c. Hubungan / Relasi

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi dapat digambarkan sebagai berikut:

Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu basis data yaitu:

1) Satu ke satu (*One to one*)

Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.

2) Satu ke banyak (*One to many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

3) Banyak ke banyak (*Many to many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.