

BAB III

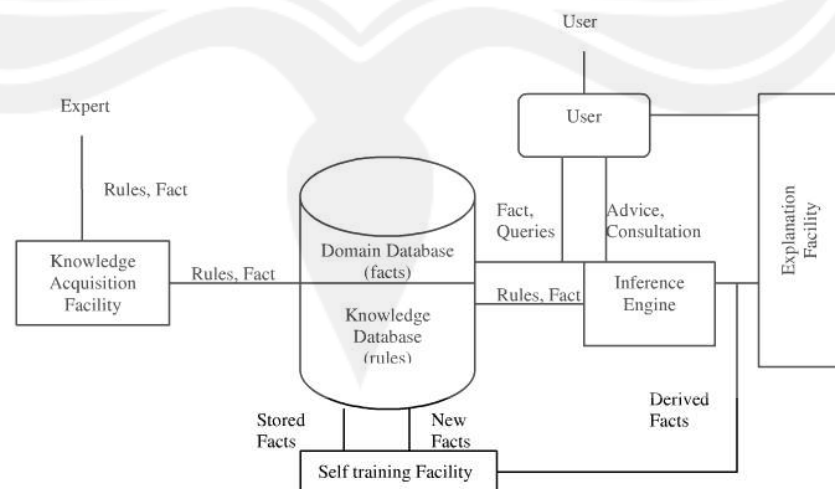
LANDASAN TEORI

3.1. Landasan Teori

3.1.1. Konsep Dasar Sistem Pakar

Sistem pakar adalah program komputer cerdas yang menggunakan pengetahuan dan prosedur-prosedur inferensi untuk menyelesaikan masalah-masalah yang membutuhkan seorang pakar dalam menemukan solusinya. Oleh sebab itu sistem pakar merupakan sebuah sistem yang mengemulasikan kemampuan membuat keputusan dari seorang pakar (Giarratano, 2005).

Arsitektur dalam sistem pakar memiliki variasi bentuk yang berbeda-beda, yang didasari oleh komponen-komponen yang berbeda juga. Tetapi pada umumnya komponen dasar yang pasti ada dalam sistem pakar antara lain, antarmuka pengguna basis pengetahuan, akusisi pengetahuan dan mekanisme inferensi (*inference engine*). Mekanisme inferensi merupakan pusat dari sistem yang mengatur jalannya sistem pakar tersebut. *Knowledge base* merupakan tempat penyimpanan pengetahuan dari seorang pakar. Antarmuka pengguna merupakan sarana berkomunikasi antara pengguna dengan sistem. Akusisi pengetahuan digunakan untuk menyimpan pengetahuan dari seorang pakar kedalam basis pengetahuan.



Gambar 3.1 Arsitektur Sistem Pakar

(James Martin & Steve Osman, 1988, halaman 30)

Penjelasan setiap komponen yang terdapat pada gambar diatas sebagai berikut :

1. Akusisi Pengetahuan (*Knowledge Acquisition Facility*)

Dahulu pengaturan basis pengetahuan merupakan hal yang sulit dan menyita banyak waktu. Setiap fakta, aturan dan hubungan harus dimasukkan ke dalam basis pengetahuan. Namun perangkat lunak yang ada sekarang memperbolehkan pakar untuk membuat dan mengubah basis pengetahuan melalui *Knowledge Acquisition Facility (KAF)*. *Knowledge Acquisition Facility (KAF)* ini bertujuan untuk menyediakan suatu dialog antara sistem pakar dengan seorang pakar untuk memperoleh pengetahuan, fakta dan aturan yang kemudian menemukannya dalam *knowledge base*, sehingga membuat *knowledge base* menjadi lebih mudah dalam pengaturan dan pemeliharaan.

2. Basis Pengetahuan (*Knowledge Base*)

Komponen ini merupakan bagian yang terdapat dalam proses inferensi yang bertujuan untuk menyimpan informasi dan aturan-aturan penyelesaian suatu pokok bahasan masalah beserta atributnya. *Knowledge Base* mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah yang dapat berasal dari pakar, jurnal, majalah, dan sumber pengetahuan lain. Basis pengetahuan disusun atas dua elemen dasar yaitu fakta dan aturan. Fakta merupakan informasi tentang obyek dalam area permasalahan tertentu. Sedangkan aturan merupakan informasi tentang cara memperoleh fakta baru dari fakta yang telah diketahui.

3. Mekanisme Inferensi (*Inference Engine*)

Mesin inferensi merupakan perangkat lunak yang melakukan penalaran dengan menggunakan pengetahuan dari pakar untuk menganalisa data dan menghasilkan suatu kesimpulan atau hasil akhir yang direpresentasikan melalui *user interface* kepada pengguna. Komponen ini memiliki tugas utama untuk mengevaluasi kondisi (*condition*) dan memeriksa semua kondisi dalam sebuah aturan (*rule*) telah terpenuhi. Mekanisme inferensi yang terdapat dalam penelitian ini menggunakan algoritma *forward chaining* dalam penyusunan aturannya,

dimana dari gejala-gejala yang diberikan oleh user dan kemudian mendapatkan hasil atau kesimpulan berupa penyakit yang diderita ikan laut tersebut.

4. Fasilitas Penjelas (*Explanation Facility*)

Komponen tambahan ini akan meningkatkan kemampuan dari sistem pakar yaitu untuk menggambarkan penalaran sistem kepada pengguna (*user*). Dimana komponen ini berfungsi untuk menjelaskan kepada pengguna tentang bagaimana kesimpulan diambil oleh sistem pakar.

5. Antarmuka Pengguna (*User Interface*)

Komponen dalam sistem pakar yang digunakan untuk berkomunikasi antara sistem dengan pengguna (*user*). Antarmuka menerima informasi yang diberikan oleh pengguna lalu mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menerima informasi dari sistem lalu menyajikannya ke dalam bentuk yang dapat dimengerti oleh pengguna.

6. Memori Kerja (*Working Memory*)

Komponen ini menyimpan sementara setiap fakta jawaban dari pengguna yang kemudian digunakan oleh mesin inferensi untuk membandingkan fakta tersebut dengan aturan yang ada dalam basis aturan dengan menggunakan metode *forward chaining* yang nantinya menghasilkan jawaban berupa penyakit yang dicari oleh *user*.

7. Sistem Pembelajaran (*Self Training Facility*)

Komponen dalam sistem pakar yang berguna untuk pembelajaran sistem secara mandiri, sehingga dapat mengolah aturan (*rule*) dan fakta (*fact*) dengan benar. Komponen ini merupakan fasilitas untuk mengatur posisi dari pengetahuan yang didapat dari pakar ke dalam basis pengetahuan (*knowledge base*), dimana pengetahuan tersebut secara mandiri dikonversikan oleh sistem kedalam bahasa yang dapat dibaca oleh komputer. Metode *Ripple Down Rules (RDR)* merupakan bagian dari *self training facility* dimana metode ini akan melakukan pembelajaran secara mandiri terhadap suatu fakta yang diberikan oleh pengguna, yang mengubahnya menjadi bentuk aturan-aturan agar dapat dilakukan proses inferensi berdasarkan aturan tersebut.

3.1.2. *Ripple Down Rules (RDR)*

Ripple Down Rules (RDR) pada awalnya adalah teknik akuisi pengetahuan secara umum dimana area aplikasi yang dimaksud di sini adalah kemampuan untuk menambahkan komentar klinis atau interpretasi laporan laboratorium untuk membantu dokter dalam memberi rujukan. Artinya, dokter yang memerintahkan tes darah patologi kimia, menerima tidak hanya hasil laboratorium tetapi saran dari ahli patologi terhadap interpretasi hasil pengujian, pengujian lebih lanjut yang mungkin diperlukan dan sebagainya. Banyak laporan patologi berisi semacam komentar sederhana dan seadanya; tujuan dari penggunaan *Knowledge Base System (KBS)* adalah untuk memberikan komentar jauh lebih rinci yang diberikan oleh ahli patologi di klinik pengelolaan pasien tertentu.

Keuntungan dari area ini untuk sistem pakar atau teknologi *Artificial Intelligence (AI)* lainnya adalah bahwa tidak ada permintaan atau harapan terhadap dokter untuk menerima laporan. Dokter tidak harus berinteraksi dengan sistem, atau mengubah cara operasionalnya. Dokter bisa memilih untuk mengagendakan tes diagnostik untuk pasien, melihat laporan, termasuk memberikan komentar interpretatif.

Tentu saja, kualitas komentar akan menjadi penting terutama apakah dokter memperhatikan komentar tersebut, tetapi kualitas komentar murni tergantung pada tingkat keahlian dari sistem, bukan pada isu-isu integrasi ke dalam alur kerja klinis. Menurut laporan dari Buchanan 1986 mengenai sistem pakar yang rutin digunakan untuk mencatat bahwa tiga dari empat sistem medis pertama rutin digunakan untuk menyediakan interpretasi klinis dari hasil tes diagnostik.

Ripple Down Rules (RDR) awalnya dikembangkan untuk menangani masalah pemeliharaan salah satu sistem pakar medis pertama. Mereka pertama kali diuji dalam bidang kedokteran di sistem *PEIRS*. Namun, dalam studi ini ada domain seorang ahli tunggal yang sangat erat terlibat dalam pengembangan dan penggunaan sistem, sehingga selalu menjadi pertanyaan apakah teknik ini akan berguna di tangan yang berbeda. Telah dilakukan berbagai evaluasi untuk jenis masalah yang berbeda, tapi ini semua dilakukan dalam konteks penelitian.

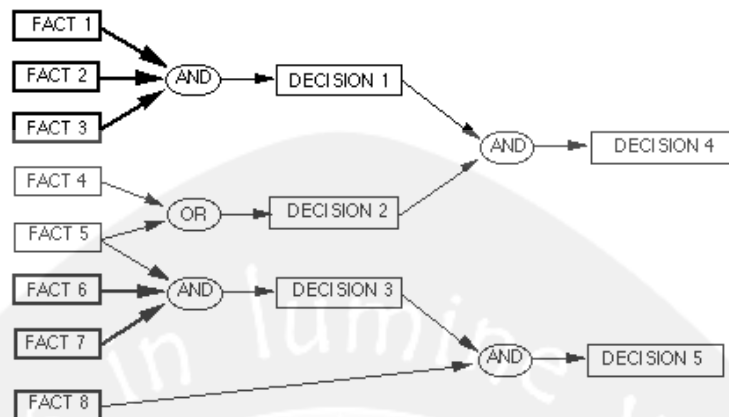
Pada perkembangannya *Ripple Down Rules (RDR)* merupakan strategi dalam mengembangkan sistem secara bertahap dimana sistem tersebut sudah digunakan. Saat sistem tidak memberikan respon yang benar maka perubahan perlu dilakukan tanpa mempengaruhi kompetensi sistem. Perubahan harus dapat dilakukan dengan mudah dan cepat serta kesulitan dalam melakukan perubahan tidak boleh meningkat meskipun sistem berkembang.

Berbagai sistem *Ripple Down Rules (RDR)* komersial telah dikembangkan untuk berbagai macam aplikasi. Riset telah membuktikan penggunaan *Ripple Down Rules (RDR)* di berbagai aplikasi seperti : pengklasifikasian masalah, konfigurasi dan pencocokan parameter, pengolahan teks, pengolahan citra, pencarian heuristik dan pencocokan algoritma genetika.

3.1.3. Penalaran Maju (*Forward Chaining*)

Metode yang digunakan dalam penelitian ini adalah penalaran maju (*forward chaining*). Penalaran maju (*forward chaining*) adalah metode inferensi yang memulai dengan menyusun fakta yang diketahui dan fakta-fakta baru dengan menggunakan aturan yang memiliki premis sesuai dengan fakta yang diketahui, dan proses ini berlanjut sampai sebuah kesimpulan tercapai atau sampai tidak ada aturan yang sesuai dengan fakta yang diketahui (Durkin 1994).

Aturan-aturan yang ada dalam metode ini diuji satu demi satu dalam suatu urutan tertentu. Saat tiap aturan diuji, sistem pakar akan mengevaluasi apakah kondisinya benar atau salah. Jika kondisi benar, maka aturan itu disimpan kemudian aturan berikutnya diuji. Namun sebaliknya jika kondisi salah, aturan tidak disimpan. Proses ini akan berulang sampai seluruh aturan yang ada telah teruji dengan berbagai kondisi.



Gambar 3.2 Forward Chaining
(Bratko ed. 4, bab 15, hal 343-)

Diketahui fakta dan aturan yang diberikan oleh pakar yaitu P dan $P \rightarrow Q$, lalu mesin inferensi memasukan Q sebagai kesimpulan dan disimpan dalam basis data. Lalu pakar mempelajari suatu fakta dan mendapat kesimpulan bahwa fakta P adalah salah dan dimasukan kedalam basis data. Hal ini mengakibatkan basis data menghasilkan suatu kontradiksi, sehingga sistem menghapus P dalam basis data agar sistem tetap stabil. Namun jika P dihapus maka kesimpulan Q tidak dapat terbukti kebenarannya sehingga harus dihapus juga dari basis data. Perbaikan fakta dan aturan ini adalah salah satu fungsi dari metode *Ripple Down Rules (RDR)* (Padhy, 2005). Metode *Ripple Down Rules (RDR)* adalah metode dimana seorang pakar dapat melakukan pengembangan dan perbaikan pada sistem, agar sistem tersebut dapat selalu memperoleh akuisisi pengetahuan yang tepat dan dapat diperbarui (Carrico,2006).

Jika ditemukan identifikasi suatu kasus tidak benar, maka pakar hanya perlu memasukkan fakta yang benar untuk diletakkan pada bagian fakta yang salah tadi, lalu sistem akan membentuk aturan yang baru berdasarkan fakta yang diberikan oleh pakar dan juga aturan tersebut akan ditandai sebagai *updating rule*. *Updating rule* adalah penambahan suatu *aturan* pengecualian tanpa megubah susunan dan kualitas dari *rule- rule* yang telah ada sebelumnya. Jika suatu *rule* telah mengalami *updating rule*, maka *user* akan melihat tampilan terbaru dari kasus tersebut.

Contoh pengimplementasian metode *Ripple Down Rules (RDR)* pada kasus ini :

Aturan awal yang terbentuk dalam sistem

IF perilaku berenang ikan yang tidak beraturan

AND beberapa ikan tenggelam ke dasar bak

AND kemudian mengapung lagi di permukaan

AND pembengkakan gelembung renang

AND letargik

AND warna tubuh terlihat lebih gelap

AND hilang nafsu makan

Then

Penyakit yang diderita *Nervous Necrosis Virus (NNV)*

Lalu pakar ingin memperbaiki gejala yang ada dalam sistem dan menggantinya menjadi gejala baru, seperti dalam kasus ini pakar ingin mengubah gejala letargik menjadi gerakan lemah.

Berikut ini pengimplementasian perubahan gejala yang dimasukan oleh pakar :

1. Sistem akan menunjukkan semua yang berhubungan dengan penyakit yang akan diubah :

Kode Penyakit : P025

Nama penyakit : *Nervous Necrosis Virus (NNV)*

Kode gejala : G001

Nama gejala : perilaku berenang ikan yang tidak beraturan

Kode Gejala : G002

Nama Gejala : beberapa ikan tenggelam ke dasar bak

Kode gejala : G003

Gejala : kemudian mengapung lagi di permukaan

Kode gejala : ...

Gejala : ...

2. Setelah itu sistem menanyakan gejala lama apa yang akan diubah di dalam sistem ini :

Kode penyakit : P025

Penyakit : *Nervous Necrosis Virus (NNV)*

Kode Gejala : G005

Gejala lama yang akan diubah : letargik

3. Lalu sistem akan meminta aturan baru yang akan ditambahkan ke dalam sistem pakar :

Gejala Baru : gerakan lemah

4. Setelah itu sistem akan melakukan pelacakan basis aturan yang berhubungan dengan *Nervous Necrosis Virus (NNV)* dan letargik dan mengubah gejala lama tersebut menjadi gerakan lemah. Setelah itu sistem menyusun bentuk aturan yang baru sebagai berikut :

IF perilaku berenang ikan yang tidak beraturan

AND beberapa ikan tenggelam ke dasar bak

AND kemudian mengapung lagi di permukaan

AND pembengkakan gelembung renang

AND gerakan lemah

AND warna tubuh terlihat lebih gelap

AND hilang nafsu makan

Then

Penyakit yang diderita *Nervous Necrosis Virus (NNV)*