BAB II

LANDASAN TEORI

2.1 Penyakit Degeneratif

Menurut Suiraoka (2012), penyakit degeneratif adalah penurunan fungsi sel sebelum waktunya. Penyakit degeneratif adalah istilah medis untuk menjelaskan suatu penyakit yang muncul akibat proses kemunduran fungsi sel tubuh yaitu dari keadaan normal menjadi lebih buruk. Sedikitnya ada 50 jenis penyakit yang termasuk penyakit degeneratif diantaranya adalah diabetes melitus, stroke, jantung koroner, kardiovaskular, obesitas, dislipidemia, hipertensi, penyakit jantung, asam urat dan sebagainya. Menurut Khasanah (2011), penyakit degeneratif memiliki hubungan yang sangat kuat dengan bertambahnya umur seseorang, tetapi penyebab utama yang mempercepat munculnya penyakit degeneratif adalah perubahan gaya hidup. Sehingga sekarang ini, penyakit degeneratif dapat terjadi pada orang yang usianya jauh lebih muda. Perubahan gaya hidup ini terkait dengan perubahan pola makan dan berkurang nya aktifitas fisik. Hal ini dapat dilihat secara jelas antara lain dengan munculnya tempat-tempat makan *junk food* di hampir seluruh sudut kota. *Junk food* adalah makanan tidak sehat karena memiliki nilai nutrisi rendah. Junk food hampir tidak mengandung protein, vitamin serta serat yang sangat dibutuhkan tubuh. Selain itu, kurangnya aktifitas karena tuntutan pekerjaan juga menyebabkan penyakit ini lebih cepat terjadi. Menurut *World Health Organization* (WHO), hingga akhir tahun 2005 saja penyakit degeneratif telah menyebabkan kematian hampir 17 juta

orang di seluruh dunia. Jumlah ini menempatkan penyakit degeneratif menjadi penyakit pembunuh manusia terbesar.

2.1.1 Jantung

Jantung merupakan organ utama yang sangat penting bagi manusia, karena jantung diperlukan untuk memompa darah ke seluruh tubuh sehingga tubuh mendapatkan oksigen dan sari makanan yang diperlukan untuk metabolisme tubuh. Jantung perlu dijaga agar dapat menjalankan fungsinya dengan baik. Menurut Suiraoka (2012), penyakit jantung adalah kelas penyakit yang melibatkan pembuluh jantung atau darah (arteri dan vena). Sering kita jumpai banyak korban meninggal seketika akibat serangan jantung. Bahkan penyakit ini merupakan empat penyakit tidak menular yang paling banyak menyebabkan kematian selain kanker, diabetes, dan paru kronis.

Menurut Aisyah (2013), beberapa macam penyakit jantung :

1. Akut Miokard Infark (AMI)

   Jenis penyakit jantung inilah yang paling banyak membunuh dari sekian banyak penderita sakit jantung. Proses penyakitnya berjalan dengan sangat cepat dan juga membutuhkan pertolongan dengan segera. Penyakit akut miokard infark ini secara mudahnya adalah kematian dari otot jantung karena adanya penyumbatan pada pembuluh darah koroner. Sedangkan fungsi pembuluh darah koroner adalah memberikan nutrisi ke otot jantung untuk bekerja memompa darah ke seluruh tubuh. Bila fungsinya terhambat, terganggu bahkan sampai tidak berfungsi maka akan berakibat fatal. Karena adanya penyumbatan ini, maka pembuluh darah koroner yang berperan besar dalam menyuplai darah dan oksigen akan mengalami kerusakan bahkan

sampai kematian mendadak. Diagnosa dan pengobatan yang tepat dapat menyelamatkan penderita dari kematian.

2. Gagal Jantung Kongestif / CHF

   Penyakit gagal jantung kongestif ini adalah satu dari jenis macam penyakit jantung. Yang dimaksud dengan gagal jantung kongestif adalah ketidakmampuan jantung untuk memompa darah secara efektif ke seluruh tubuh. Padahal fungsi jantung salah satu diantaranya yaitu fungsi memompa darah ke seluruh tubuh. Dengan adanya kegagalan ini maka tubuh tidak tersuplai darah dengan baik. dan akan terjadi apa yang dinamakan dengan gagal jantung.

3. Aterosklerosis

   Penyakit jantung jenis ini kelainannya adalah penebalan dinding arteri sebelah dalam karena endapan plak (berupa lemak dan kolesterol) sehingga menghambat serta menyumbat pasokan darah ke sel-sel otot. Aterosklerosis dapat terjadi di seluruh bagian tubuh. Bila terjadi pada dinding arteri jantung, maka disebut penyakit jantung koroner atau penyakit jantung iskemik.

4. Penyakit Jantung Rematik

   Penyakit jantung rematik ini biasanya menyerang pada anak-anak dan juga bisa merupakan penyebab penyakit katup jantung. Penyakit jantung rematik adalah penyakit jantung yang terfokus pada kerusakan pada katup jantung karena demam rematik, yang disebabkan oleh bakteri streptokokus.

5. Penyakit Katup Jantung

   Jenis penyakit jantung ini sesuai dengan namanya maka penyakit ini menyerang pada katup jantung, sedangkan katup jantung itu mempunyai

fungsi dalam mengendalikan aliran darah dalam ruang-ruang jantung. Kelainan katup jantung yang dapat mengganggu aliran tersebut, antara lain karena pengecilan (stenosis), kebocoran (regurgitasi), atau tidak menutup sempurna (prolaps). Kelainan katup dapat terjadi sebagai bawaan lahir maupun karena infeksi. Demikian yang dimaksud dengan Penyakit katup jantung.

2.1.2 Diabetes Melitus

Menurut Khasanah (2011), diabetes melitus atau kencing manis adalah suatu kumpulan gejala yang timbul pada seseorang yang disebabkan oleh adanya peningkatan kadar gula (glukosa darah). Glukosa sangat penting bagi kesehatan karena merupakan sumber energi utama bagi otot dan jaringan. Jika seseorang memiliki diabetes tipe apapun, itu berarti memiliki terlalu banyak glukosa. Terlalu banyak glukosa dalam darah akan menyebabkan masalah yang serius.

Jenis-jenis penyakit diabetes :

1. Diabetes Melitus Tipe I

   Disebut juga *Insulin Dependent Diabetes Melitus* (IDDM), dimana penderita mengalami gangguan pada produksi hormon insulin oleh suatu bagian dari limpa.

2. Diabetes Melitus Tipe II

   Disebut juga *Non-Insulin Dependent Diabetes Melitus* (NIDDM), dimana penderita tidak kekurangan insulin, tetapi ada resistensi dari sel otot maupun sel jaringan lemak untuk dimasuki gula darah.

3. Diabetes Gestational

   Diabetes yang terjadi pada saat kehamilan. Sekitar 4% wanita hamil menderita tipe ini. Jika dilihat dari penyebab terjadinya, penyakit ini lebih mengarah ke dalam golongan diabetes tipe II.

2.1.3 Hipertensi

Menurut Khasanah (2011), hipertensi atau darah tinggi adalah gejala peningkatan tekanan darah yang mengakibatkan suplai oksigen dan nutrisi yang dibawa oleh darah terhambat sampai ke jaringan tubuh yang membutuhkan. Hipertensi berdasarkan penyebabnya dibagi menjadi 2 jenis :

1. Hipertensi primer atau esensial adalah hipertensi yang tidak atau belum diketahui penyebabnya (terdapat pada kurang lebih 90% dari seluruh hipertensi).
2. Hipertensi sekunder adalah hipertensi yang disebabkan atau sebagai akibat dari adanya penyakit lain.

Faktor gizi sangat berhubungan dengan terjadinya hipertensi melalui beberapa mekanisme. *Aterosklerosis* merupakan penyebab utama terjadinya hipertensi yang berhubungan dengan diet seseorang, walaupun faktor usia juga berperan, karena pada usia lanjut (usila) pembuluh darah cenderung menjadi kaku dan elastisitasnya berkurang. Dan sekitar 5% kasus hipertensi telah diketahui penyebabnya seperti penyakit ginjal, penyakit pembuluh darah dan penyakit endokrin.

2.1.4 Osteoartritis

Menurut J. D'Adamo (2007:4), osteoartritis adalah suatu penyakit sendi degeneratif, kebanyakan mempengaruhi tulang rawan. Tulang rawan yang sehat memungkinkan tulang meluncur di atas satu sama lain dan menyerap energi dari kejutan gerak fisik. Pada osteoartritis, lapisan permukaan tulang rawan rusak dan aus. Hal ini menyebabkan tulang bergesekan, menyebabkan rasa nyeri, pembengkakan, dan hilangnya gerakan sendi. Dengan berjalannya waktu, sendi dapat kehilangan bentuk normalnya.

Nyeri dan kekakuan dari sendi-sendi dapat terjadi setelah periode-periode yang panjang dari ketidakaktifan, contohnya, duduk dalam teater. Pada osteoartritis yang parah, kehilangan bantal *cartilage* yang komplit menyebabkan gesekan antara tulang-tulang, menyebabkan nyeri pada saat istirahat atau nyeri dengan gerakan yang terbatas.

2.1.5 Dislipidemia

Menurut Pramono (2009), dislipidemia adalah kelainan metabolisme lipid yang ditandai dengan peningkatan maupun penurunan fraksi lipid dalam plasma. Di mana peningkatan abnormal itu terjadi pada kadar kolesterol total, kolesterol LDL, trigliserida (TG), sedangkan kolesterol HDL yang berperan penting untuk kesehatan jantung mengalami penurunan.

Secara umum dislipidemia ada dua klasifikasi, yaitu primer dan sekunder. Dislipidemia primer merupakan kelainan kolesterol pada kolesterol total, kolesterol LDL, trigliserida, dan kadar kolsterol HDL. Bisa dikatakan, seseorang yang mengalami dislipidemia pasti memiliki kadar kolesterol yang abnormal, yang memicu penumpukan plak pada pembuluh darah tertentu,

sehingga aliran darah tak bisa mengalir. Dan pada kondisi inilah, seseorang umumnya akan mengalami serangan jantung.

Sedangkan untuk dislipidemia sekunder, serangan jantung umumnya dipicu karena penyakit kronis seperti diabetes, dan efek buruk merokok, alkohol serta obesitas. Kondisi itu bisa terjadi karena semua penyakit dan kebiasaan buruk itu memicu resistensi insulin dalam tubuh, sehingga peredaran pembuluh darah terganggu dan memicu gangguan pada penyakit jantung dan pembuluh darah.

2.1.6 Hiperurisemia

Menurut Antoro (2012), hiperurisemia adalah istilah kedokteran yang mangacu pada kondisi kadar asam urat dalam darah melebihi normal, yaitu lebih dari 7,0 mg/dl. Hiperurisemia dapat terjadi akibat meningkatnya produksi ataupun menurunnya pembuangan asam urat, atau kombinasi dari keduanya. Hiperurisemia mempunyai gejala khas peradangan sendi yang mendadak, disebabkan oleh reaksi jaringan sendi terhadap pembentukan kristal asam urat yang bentuknya menyerupai jarum. Hiperurisemia dapat menyebabkan peningkatan kadar asam urat dalam darah tetapi tidak selalu disertai gejala-gejala penyakit asam urat. Pada penyakit hiperurisemia, kadang-kadang dapat terjadi pembentukan kristal asam urat dalam ginjal, kristal ini akan larut dalam urin yang bersifat alkalis (basa). Senyawa asam urat dihasilkan oleh tubuh dalam metabolisme purin dan dikeluarkan keluar melalui metabolisme ginjal. Penyakit hiperurisemia ada 2 jenis, yaitu :

1. Primer

   Disebabkan oleh produksi asam urat yang berlebihan.

2. Sekunder

   Disebabkan oleh obat / racun yang mengakibatkan produksi asam urat naik dan menyebabkan serangan akut / mendadak (obat golongan salisilat, diuretik).

2.2 Kedudukan Sistem Pakar Dalam Kecerdasan Buatan

Menurut Subakti (2006), sistem pakar adalah salah satu bagian dari bidang kecerdasan buatan. Sistem dirancang untuk meniru perilaku seorang ahli yang bisa menyelesaikan permasalahan-permasalahan yang cukup kompleks. Kunci sukses dari sistem pakar adalah bagaimana sistem tersebut dapat memproses basis pengetahuan yang ada. Apabila basis pengetahuan tersebut dikombinasikan dengan teknik inferensi yang ada dalam kecerdasan buatan, tidak menutup kemungkinan menghasilkan pemecahan yang lebih baik dari seorang pakar pada satu area masalah yang spesifik dan biasanya lebih sempit.

Proses dari sistem pakar bisa dinilai sederhana. Kepakaran dipindahkan dari seorang pakar ke komputer yang disimpan dalam basis data dan bila user meminta saran spesifik yang dibutuhkan, komputer mencari, mengolah, dan menampilkan kesimpulan yang spesifik.

2.3 Sistem Pakar

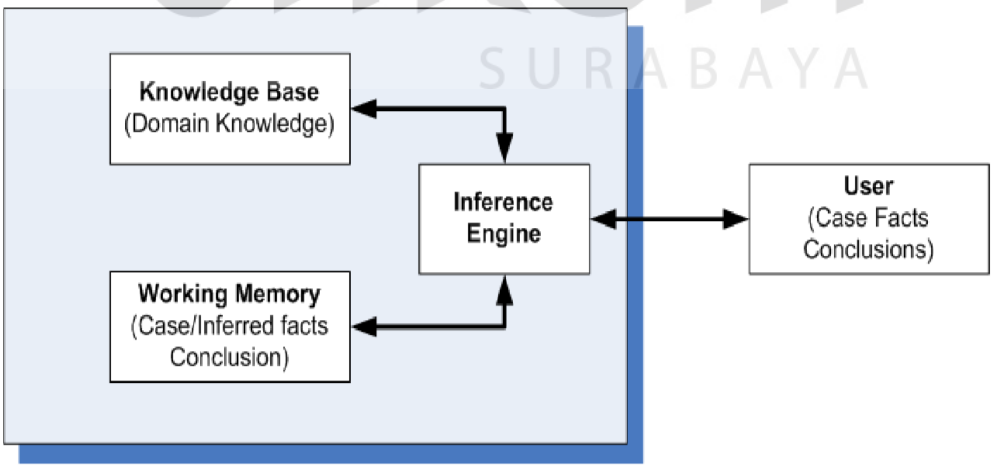
Menurut Kusrini (2006:11), sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar

dalam bidang tersebut. Sistem pakar bekerja berdasarkan pengetahuan yang dimasukkan oleh seorang atau beberapa orang pakar dalam rangka mengumpulkan informasi hingga sistem pakar dapat menemukan jawabannya.

Di dalam menyelesaikan suatu masalah, sistem pakar mengajukan berbagai pertanyaan kepada pemakai dalam rangka pengumpulan informasi hingga sistem pakar tersebut dapat memberikan suatu penyelesaian yang dianggap tepat atau sesuai bagi seseorang, yang harus memiliki serangkaian alternatif terbaik dari alternatif yang ada. Berdasarkan kriteria yang diberikan, sistem pakar bisa menentukan pilihan yang tepat. Tujuan utama sistem ini adalah untuk memindahkan secara efektif ilmu pengetahuan kepada mereka yang bukan pakar.

2.4 Komponen Sistem Pakar

Menurut Irawan (2007), sistem pakar mempunyai 3 komponen utama, yaitu k*nowledge base*, *working memory*, dan i*nference engine*. Model sistem pakar dapat dijelaskan melalui diagram gambar 2.1 di bawah ini.



Gambar 2.1 Diagram Komponen Sistem Pakar (Sumber: Irawan, 2007:7)

A. *Knowledge base* merupakan berfungsi untuk menyimpan data atau pengetahuan yang memuat fakta-fakta, yang diperlukan untuk membuat suatu keputusan. *Knowledge base* terdiri dari dua bagian yaitu:

1. Fakta

   Fakta adalah suatu kenyataan atau kebenaran yang diketahui. Fakta menyatakan hubungan (relasi) antara dua objek atau lebih. Fakta dapat pula menunjukkan sifat.

2. Aturan

   Dalam menerangkan masalah digunakan aturan untuk menentukan hal apa yang harus dilakukan dalam situasi tertentu dan aturan tersebut terdiri dari 2 bagian yaitu *IF* dan *THEN*. *IF* merupakan kondisi yang mungkin benar atau mungkin tidak benar, sedangkan *THEN* adalah tindakan yang dilakukan jika kondisi benar.

B. *Working memory* berfungsi untuk menyimpan fakta-fakta yang ditemukan selama proses konsultasi. Selama proses konsultasi, *user* memasukkan fakta yang dibutuhkan, kemudian sistem mencari padanan tentang fakta dengan informasi yang ada di *knowledge base* untuk menghasilkan fakta baru. Sistem akan memasukkan fakta baru ke dalam *working memory*, sehingga *working memory* menyimpan fakta-fakta yang ditemukan baik dari *user* maupun hasil kesimpulan sistem.

C. *Inference engine* adalah bagian dari sistem pakar yang melakukan penalaran dengan menggunakan isi *knowledge base* berdasarkan urutan tertentu. *Inference engine* berfungsi untuk mencari padanan antara fakta yang ada dalam *working memory* dengan fakta yang ada di dalam *knowledge base*,

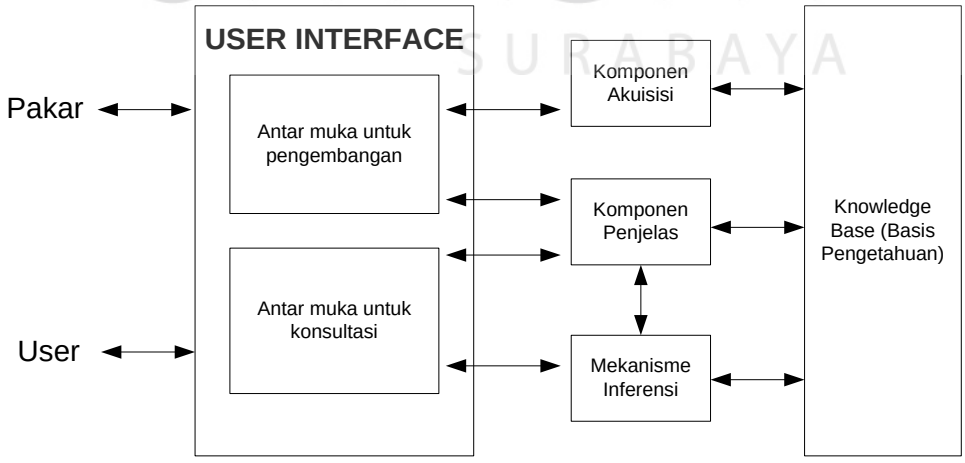
yang selanjutnya *inference engine* akan menarik kesimpulan dari problem yang diajukan kepada sistem. Ada dua metode utama yang digunakan *inference engine* untuk melakukan penelusuran yaitu penalaran maju (*Forward Chaining*) dan penalaran mundur (*Backward Chaining*).

2.5 Sistem Berbasis Aturan

Menurut Subakti (2006), sistem berbasis aturan merupakan suatu sistem yang pengetahuannya direprensentasikan sebagai serangkaian *rule-rule* (*production rules*). Dengan kata lain bahwa sistem berbasis aturan adalah suatu perangkat lunak yang menyajikan keahlian pakar dalam bentuk aturan-aturan pada suatu domain tertentu untuk menyelesaikan suatu permasalahan.

2.6 Komponen Sistem Berbasis Aturan

Untuk membangun suatu sistem berbasis aturan diperlukan beberapa komponen, secara umum dapat dilihat pada gambar 2.2 di bawah ini.



Gambar 2.2 Diagram Komponen Sistem (Sumber: Ulysses, 2012)

A. *Knowledge base* (basis pengetahuan) adalah bagian dari sebuah sistem pakar yang mengandung/menyimpan pengetahuan (*domain knowledge*). *Knowledge base* yang dikandung oleh sebuah sistem pakar berbeda antara satu dengan yang lain tergantung pada bidang kepakaran dari sistem yang dibangun (Irawan, 2007).

B. Mekanisme inferensi berfungsi untuk mensimulasikan strategi penyelesaian masalah dari seorang pakar. Sebuah konklusi akan dicapai dengan menjalankan suatu aturan tertentu pada fakta yang ada.

C. Komponen penjelas berfungsi menjelaskan strategi penyelesaian masalah bagi *user* yang meliputi:

   1. Pertanyaan apa yang akan diajukan pada pemakai dan jika diperlukan mengapa mengajukan pertanyaan tersebut.
   2. Alasan bagaimana sistem tersebut memperoleh hasil demikian.
   3. Karakteristik apa yang dimiliki tiap-tiap objek.

D. *User interface* yaitu bagian program yang berhubungan langsung dengan pemakai, baik selama konsultasi maupun untuk pengembangan sistem. Oleh karena itu sistem haruslah menggunakan bahasa dan sistem pengoperasian yang mudah dimengerti.

E. Komponen akuisisi berfungsi untuk menyusun dan mengimplementasikan pengetahuan dalam basis pengetahuan. Komponen ini memiliki beberapa karakteristik antara lain:

1. Pengetahuan yang terdiri dari pengetahuan dan fakta harus mudah untuk dimasukkan.
2. Metode penyajian informasi dalam basis pengetahuan harus mudah dimengerti.
3. Sangat baik jika memiliki sistem pengecekan atas format yang salah.

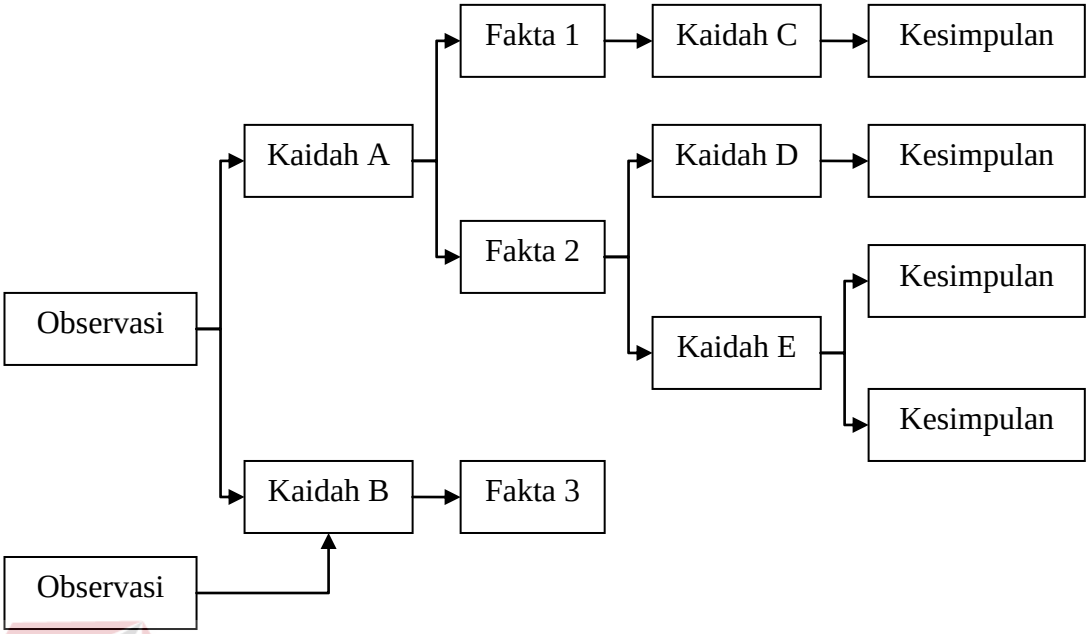
2.7 Inference Engine

Menurut Andi (2003), *inference engine* adalah bagian dari sistem pakar yang melakukan penalaran dengan menggunakan isi daftar *rule* berdasarkan urutan dan pola tertentu. Selama proses konsultasi antara *user* dengan sistem, *inference engine* menguji *rule* satu demi satu sampai kondisi *rule* itu benar.

Ada 2 metode utama yang telah dibuat bagi *inference engine* untuk menguji aturan yaitu penalaran maju (*Forward Chaining*) dan penalaran mundur *(Backward Chaining)*.

1. Forward Chaining

   Menurut Wijaya (2006), metode *forward chaining* adalah suatu metode dari mesin inferensi untuk memulai penalaran atau pelacakan suatu data dari fakta-fakta yang ada menuju suatu kesimpulan. Dalam metode ini, kaidah interpreter mencocokkan fakta atau *statement* dalam pangkalan data dengan situasi yang dinyatakan dalam bagian sebelah kiri atau kaidah *IF*. Apabila fakta yang ada dalam pangkalan data itu sudah sesuai dengan kaidah *IF*, maka kaidah distimulasi. Tahapan metode *forward chaining* dapat dilihat pada gambar 2.3.



Gambar 2.3 Metode Forward Chaining (Sumber: Fitriawanti, 2009:14)

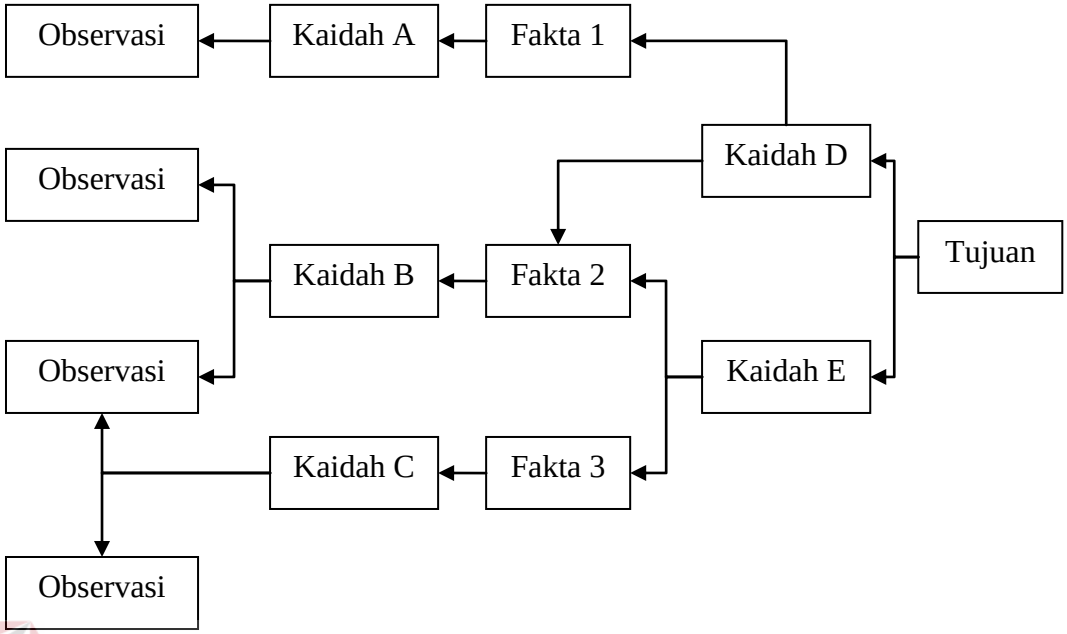
Pada gambar di atas menunjukkan pangkalan kaidah yang terdiri dari 5 buah yaitu kaidah A, kaidah B, kaidah C, kaidah D dan kaidah E. Sedangkan pangkalan data terdiri dari pengawalan fakta yang sudah diketahui, yaitu fakta 1, fakta 2 dan fakta 3.

Melalui observasi 1 mulai melacak pangkalan kaidah untuk mencari premis dengan menguji semua kaidah secara berurutan. Pada observasi 1 pertama-tama melacak kaidah A dan kaidah B. *Inference engine* mulai melakukan pelacakan, mencocokkan kaidah A dalam pangkalan pengetahuan terhadap informasi yang ada di dalam pangkalan data, yaitu fakta 1 dan fakta 2. Jika pelacakan pada kaidah A tidak ada yang cocok dengan fakta 1, maka terus bergerak menuju kaidah C yang kemudian menghasilkan kesimpulan, demikian seterusnya.

2. Backward Chaining

Menurut Irawan (2007), metode *backward chaining* dilakukan dengan cara memilih beberapa kesimpulan yang mungkin dan mencoba membuktikan kesimpulan tersebut dari bukti-bukti yang ada. Metode *backward chaining* merupakan kebalikan dari metode *forward chaining* dan sering disebut penalaran mundur. Pada metode ini pelacakan atau penalaran dari dilakukan dari sekumpulan hipotesa menuju fakta-fakta yang mendukung kesimpulan tersebut. Jadi interpreter kaidah mulai menguji kaidah sebelah kanan yaitu *THEN*.

Inference engine akan melacak bukti-bukti yang mendukung hipotesa awal. Jika ternyata sesuai, maka basis data akan mencatat kondisi terhadap status sistem yang berlaku. Semua sisi kaidah *IF* yang benar-benar sesuai digunakan untuk menghasilkan hipotesa yang baru dan keadaan tujuan, yang kemudian direkam dalam basis data. Keadaan di atas terus berlangsung sampai hipotesa terbukti kebenarannya. Alur dari metode *backward chaining* dapat dilihat pada gambar 2.4 di bawah ini.



Gambar 2.4 Metode Backward Chaining (Sumber: Fitriawanti, 2009:16)

Dalam melakukan penelusuran pada *backward chaining* berawal dari *goal* atau pada gambar disebut sebagai tujuan, kemudian mencari informasi untuk memenuhi tujuan tersebut. Pertama-tama mulai dengan memberitahu sistem bahwa kita ingin membuktikan keadaan tujuan. *Inference engine* melihat pangkalan data yaitu fakta untuk dicocokkan dengan pangkalan kaidah.

2.8 Verifikasi

Menurut Gonzales (2000), verifikasi adalah proses untuk memastikan bahwa sistem cerdas sesuai dengan spesifikasi serta basis pengetahuannya konsisten dan terbebas dari kesalahan. Suatu kualitas dari basis pengetahuan dapat dilihat dari ukuran, kompleksitas dan sifat kritikal dari aplikasi-aplikasi yang ada. Semuanya itu dapat diwujudkan dari proses-proses verifikasi. Elemen ini sangat penting bagi suatu sistem berbasis pengetahuan. Verifikasi adalah membangun sistem yang benar. Verifikasi itu sendiri terdiri dari 2 proses yaitu:

1. memeriksa pelaksanaan suatu sistem secara spesifik.
2. memeriksa konsistensi dan kelengkapan dari basis pengetahuan.

Verifikasi dijalankan ketika ada penambahan atau perubahan pada *rule*, karena *rule* tersebut sudah ada pada sistem. Sedangkan tujuan verifikasi adalah untuk memastikan adanya kecocokkan antara sistem dengan apa yang sistem kerjakan dan juga untuk memastikan bahwa sistem itu terbebas dari *error*.

Berikut ini adalah yang harus dicek dalam suatu basis pengetahuan :

1. *Redundant rules*

   Dikatakan *redundant rules* jika 2 *rule* atau lebih mempunyai *premise* dan *conclusion* yang sama.

   Contoh :

   Rule 1 : *If the humidity is high and the temperature is hot*
   Then there will be thunderstorms

   Rule 2 : *If the temperature is hot and the humidity is high*
   Then there will be thunderstorms

2. *Conflicting rules*

   Conflicting rules terjadi ketika 2 *rule* atau lebih mempunyai *premise* yang sama tetapi *conclusion* yang berbeda.

   Contoh :

   Rule 1 : *If the temperatur is hot and the humidity is high*
   Then there will be sunshine

   Rule 2 : *If the temperatur is hot and the humidity is high*
   Then there will not be sunshine

3. *Subsumed rules*

   Suatu keadaan dapat dikatakan *subsumed rules* jika *rule* tersebut mempunyai *constraint* yang lebih atau kurang tetapi mempunyai *conclusion* yang sama.

   Contoh :

   Rule 1 : *If the temperatur is hot and the humidity is high*

   Then there will be thunderstorms

   Rule 2 : *If the temperatur is hot*

   Then there will be thunderstorms

4. *Circular rules*

   Circular rules ialah suatu keadaan dimana terjadinya proses perulangan dari suatu *rule*. Ini dikarenakan suatu *premise* dari salah satu *rule* merupakan *conclusion* dari *rule* yang lain, atau kebalikannya.

   Contoh :

   Rule 1 : *If X and Y are brothers*

   Then X and Y have the same parents

   Rule 2 : *If X and Y have the same parents*

   Then X and Y are brothers

5. *Unnecessary IF condition*

   Unnecessary IF terjadi ketika 2 *rule* atau lebih mempunyai *conclusion* yang sama tetapi salah satu dari *rule* tersebut mempunyai *premise* yang tidak perlu dikondisikan dalam *rule* karena tidak mempunyai pengaruh apapun.

Contoh :

Rule 1 : *If the patient has pink spots and the patient has a fever*

Then the patient has measles

Rule 2 : *If the patient has pink spots and the patient does not have fever*

Then the patient has measles

6. *Dead-end rules*

Dead-end rules adalah suatu *rule* yang *conclusion*nya tidak diperlukan oleh *rule-rule* lainnya.

Contoh :

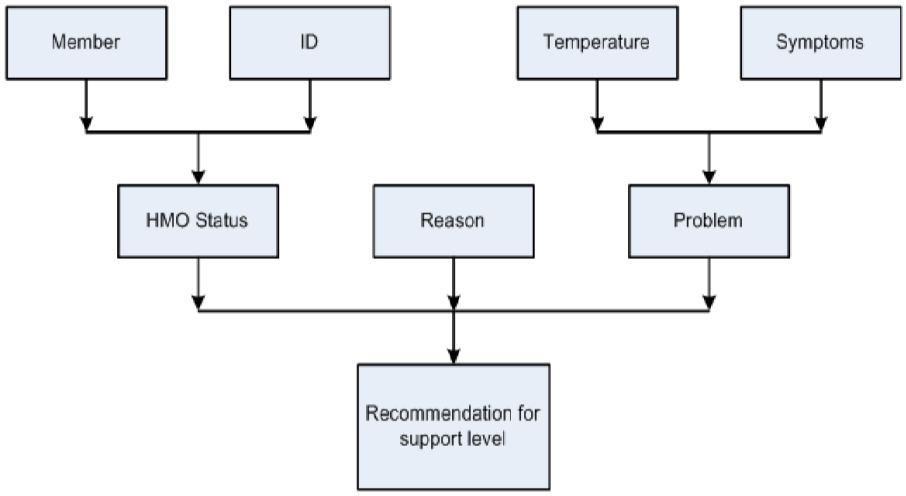
Rule 1 : *If the gauge reads empty*

Then the gas tank is empty

2.9 Diagram Blok

Langkah awal yang dilakukan dalam menerjemahkan suatu bidang ilmu ke dalam sistem berbasis aturan yaitu melalui diagram blok *(block diagram)*. Menurut Ogata (2002), diagram blok dari sebuah sistem merupakan gambaran dari proses yang dijalankan oleh tiap komponen dan aliran sinyal.

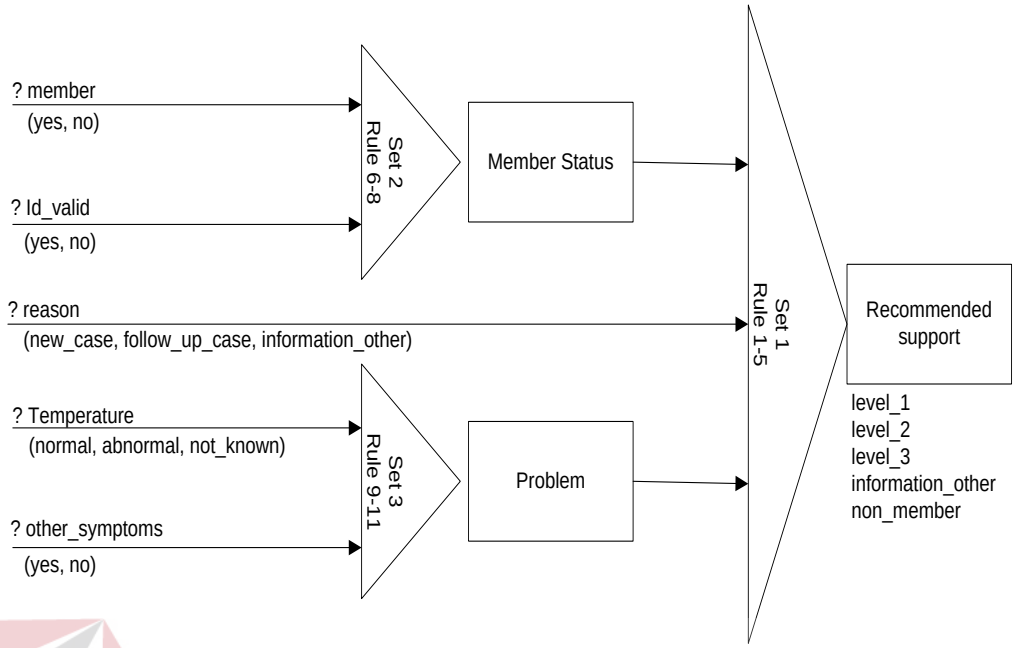
Dengan membuat diagram blok di dalam sistem berbasis aturan maka dapat diketahui urutan kerja sistem dalam mencari keputusan. Contoh diagram blok dapat dilihat pada gambar 2.5 di bawah ini.



Gambar 2.5 *Block Diagram* Health Maintenance Organization (HMO)
(Sumber: Irawan, 2007:56)

2.10 Diagram Ketergantungan

Setelah diketahui urutan kerja sistem dalam mencari keputusan dari diagram blok, langkah selanjutnya adalah membuat diagram ketergantungan *(dependency diagram)*. Menurut Viony (2012), *dependency diagram* menggambarkan hubungan pertanyaan, *rule*, nilai, dan faktor-faktor penentu yang sudah dibuat dalam diagram blok. Contoh *dependency diagram* dapat dilihat pada gambar 2.6.



Gambar 2.6 *Dependency diagram* HMO (Sumber: Irawan, 2007:57)

2.11 Decision Table

Dari data-data yang diolah dan dibuat diagram ketergantungan, langkah yang berikutnya adalah pembuatan *decision table*. Menurut Viony (2012), *decision table* merupakan tabel yang menunjukkan semua kombinasi inputan dan hasilnya. Sebagai contoh dari pembuatan *decision table* dapat dilihat tabel 2.1 di bawah ini.

Tabel 2.1 *Decision Table HMO Rule Set 1*

Rule	Member Status	Reason	Problem	Concluding Recommendation for support level
A 1	ok	new-case	serious	level-1
A 2	ok	new-case	non-serious	level-2
A 3	ok	follow-up-case	serious	level-1
A 4	ok	follow-up-case	non-serious	level-3
A 5	ok	information-other	serious	information-other
A 6	ok	information-other	non-serious	information-other
A 7	not-ok	new-case	serious	non-member
A 8	not-ok	new-case	non-serious	non-member
A 9	not-ok	follow-up-case	serious	non-member
A 10	not-ok	follow-up-case	non-serious	non-member
A 11	not-ok	information-other	serious	non-member
A 12	not-ok	information-other	non-serious	non-member

(Sumber: Irawan, 2007:57)

2.12 Reduced Decision Table

Setelah didapatkan nilai dari *decision table* akan direduksi untuk mendapatkan nilai dari kondisi terakhir. Menurut Lee (2011), *reduced decision table* adalah penyederhanaan dari *decision table* dengan menggunakan *rule-rule* yang hasilnya sama dan memiliki inputan yang tidak berpengaruh. Sebagai contoh dari mereduksi *decision table* dapat dilihat pada tabel 2.2.

Tabel 2.2 *Reduced Decision Table HMO Rule Set 1*

Rule	Member Status	Reason	Problem	Concluding Recommendation for support level
B 1	ok	new-case	serious	level-1
B 2	ok	new-case	non-serious	level-2
B 3	ok	follow-up-case	serious	level-1
B 4	ok	follow-up-case	non-serious	level-3
B 5	ok	information-other	–	information-other
B 6	not-ok	–	–	non-member

(Sumber: Irawan, 2007:58)

2.13 Penyajian aturan (rule) dengan treeview

Treeview adalah sebuah fasilitas yang disediakan bahasa pemrograman *Visual Basic* untuk penyusunan aturan-aturan. Menurut Tarigan (2010), *object treeview* adalah sebuah diagram pohon yang menggambarkan hubungan logis antara komponen visual dan non visual yang terletak pada *form*, *data module*, maupun *frame*. Dalam sebuah *treeview* ada beberapa fungsi dan prosedur yang bisa membantu menyusun aturan-aturan dan memanfaatkannya sebagai *inference engine* ketika sistem dijalankan.

Dengan *treeview* langkah-langkah untuk mengubah diagram ketergantungan menjadi *rule* tidak diperlukan karena diagram ketergantungan dapat langsung diaplikasikan dalam *treeview*. *Treeview* menyediakan fasilitas untuk menambah, menyisipkan ataupun memotong node-node yang ada.

2.14 Basis Data

Menurut Minartiningtyas (2013), basis data (*database*) adalah kumpulan informasi yang disusun berdasarkan cara tertentu dan merupakan suatu kesatuan yang utuh. Berdasarkan pengertian tersebut, data yang terhimpun dalam suatu database dapat menghasilkan informasi yang berguna.

Manfaat dari pembentukkan *database* adalah untuk mempermudah penciptaan struktur data. Selain itu suatu *database* dapat digunakan untuk sejumlah program aplikasi yang berlainan sehingga dapat meningkatkan produktifitas *progammer*. Kumpulan *file* yang saling berkaitan dengan program untuk pengelolaanya disebut sebagai *database*.