

BAB II

LANDASAN TEORI

2.1 Sistem Pakar

“Sistem pakar adalah suatu program komputer yang didesain untuk memodelkan atau mengemulsi kemampuan seorang pakar dalam memecahkan suatu masalah” (Durkin, 1994).

Sistem pakar dimaksudkan untuk menyelesaikan masalah-masalah nyata dimana biasanya akan membutuhkan seorang pakar (seperti dokter, teknisi, dan lainnya). Sistem pakar menghasilkan hasil yang konsisten dan mempunyai kecepatan yang konstan daripada kepakaran seseorang. Hal ini disebabkan oleh beberapa hal yang bersifat manusiawi dimana dapat mempengaruhi pengambilan keputusan seorang pakar seperti lingkungan kerja, kondisi kesehatan dan masalah pribadi.

Menurut Durkin (1994) ada beberapa keunggulan dan kelemahan sistem pakar dibandingkan dengan seorang pakar, hal ini dapat dilihat pada tabel 2.1 dibawah ini.

Tabel 2.1 Perbandingan antara seorang pakar dan sistem pakar

Faktor	Pakar	Sistem Pakar
Ketersediaan waktu	Hari kerja	Setiap saat
Geografis	Lokal atau tertentu	Di mana saja
Keamanan	Tidak tergantikan	Dapat diganti
Dapat Habis (<i>Perishable</i>)	Ya	Tidak
Performansi	Variabel	Konsisten
Kecepatan	Variabel	Konsisten
Biaya	Tinggi	Terjangkau

Dari perbandingan di atas, maka dapat diketahui alasan mengapa perlu membangun sebuah sistem pakar, yaitu :

1. Untuk menggantikan seorang pakar
 - a. Memungkinkan untuk mendapatkan kepakaran setelah jam kerja atau dilokasi lain.
 - b. Otomatisasi pekerjaan rutin yang membutuhkan pakar.
 - c. Pakar mahal.
 - d. Kepakaran yang diperlukan pada tempat-tempat yang berbahaya (*hostile environment*).
2. Untuk membantu seorang pakar
 - a. Membantu seorang pakar untuk mengerjakan tugas-tugas rutinnya, sehingga pakar tersebut dapat meningkatkan produktivitasnya.
 - b. Membantu seorang pakar untuk menyelesaikan tugas-tugas sulit, sehingga pakar tersebut dapat lebih efektif dalam menangani masalah-masalah yang kompleks.
 - c. Menyediakan informasi yang sulit diingat kembali oleh seorang pakar.

Menurut Ivan Braiko (1990), sistem pakar yang baik harus mempunyai fungsi-fungsi sebagai berikut :

1. Penyelesaian masalah, kemampuan untuk menyelesaikan masalah dengan tepat pada ruang lingkup yang spesifik dengan informasi yang terbatas dan diperlukan (tidak harus lengkap).
2. Berinteraksi dengan *user*, sistem pakar harus dapat berinteraksi dengan *user* dan menjelaskan suatu sistem beserta kesimpulan-kesimpulan selama dan setelah proses penyelesaian masalah.

Hingga sekarang, sistem pakar telah banyak diaplikasikan untuk menyelesaikan masalah dalam bidang yang luas seperti pertanian, bisnis, ilmu kimia, komunikasi, sistem komputer, pendidikan, elektronik, teknik, obat-obatan dan lain sebagainya.

Berdasarkan dari pemecahan masalah (*problem solving tasks*), sistem pakar dapat dikategorikan menjadi beberapa tipe, yaitu :

a. Kontrol (*Control*)

Digunakan untuk mengontrol dan mengarahkan kerja sebuah sistem tertentu.

Misalnya : pengontrol mesin-mesin manufaktur atau *treatment* untuk seorang pasien di rumah sakit.

b. Desain (*Design*)

Digunakan untuk mengkonformasi beberapa obyek dengan batasan-batasan masalah tertentu. Misalnya : mendesain sebuah sistem komputer dengan batasan-batasan yang didefinisikan oleh *user*, seperti : batasan harga, kecepatan, kemampuan multimedia, dan lainnya.

c. Diagnosa (*Diagnosis*)

Digunakan untuk mendeteksi kerusakan di dalam sebuah sistem, dengan cara mengobservasi semua informasi yang didapat dari sistem tersebut. Misalnya : diagnosis penyakit seorang pasien berdasarkan atas gejala yang timbul.

d. Instruksi (*Instruction*)

Digunakan untuk mendeteksi dan memperbaiki kekurangan perilaku sistem dalam memahami bidang informasi tertentu. Misalnya : petunjuk belajar seorang siswa terhadap topik tertentu yang dipelajari.

e. Interpretasi (*Interpretation*)

Digunakan untuk menghasilkan sebuah pemahaman terhadap suatu situasi tertentu berdasarkan atas informasi yang dihasilkan dari sebuah situasi yang spesifik. Biasanya informasi-informasi tersebut berasal dari sensor, *instrument*, hasil tes, dan lainnya. Misalnya : *monitor* sensor mesin, sistem pencitraan, atau *speech analysis result*.

f. Pengamatan (*Monitoring*)

Digunakan untuk memonitor dan membandingkan cara kerja sebuah sistem dengan sistem yang diharapkan. Misalnya : kontrol instalasi nuklir.

g. Perencanaan (*Planning*)

Digunakan untuk melakukan aksi guna mencapai tujuan yang telah ditentukan dengan mengikuti batasan-batasan tertentu. Misalnya : merencanakan beberapa tugas yang berbeda untuk sebuah robot dalam industri manufaktur.

h. Prediksi (*Prediction*)

Digunakan untuk meramalkan aksi yang perlu dilakukan dimasa depan berdasarkan atas informasi yang ada dan model-model yang telah diberikan. Misalnya : prediksi kerusakan pada hasil panen yang diserang hama atau prediksi cuaca.

i. Preskripsi (*Prescription*)

Digunakan untuk menghasilkan rekomendasi solusi akan sebuah malfungsi yang terjadi pada sebuah sistem. Misalnya : *Blue Box* yang merekomendasikan terapi untuk pasien yang menderita depresi.

j. Seleksi (*Selection*)

Digunakan untuk mengidentifikasi pilihan terbaik dari suatu daftar kemungkinan atas dasar informasi yang didapat pada suatu kondisi tertentu.

k. Simulasi (*Simulation*)

Digunakan untuk memodelkan sebuah proses atau sistem pada bermacam-macam kondisi. Misalnya : pembelajaran bagi para operator sebuah sistem sebelum diterjunkan langsung pada kenyataan.

Adapun langkah-langkah dalam merancang suatu sistem pakar yaitu

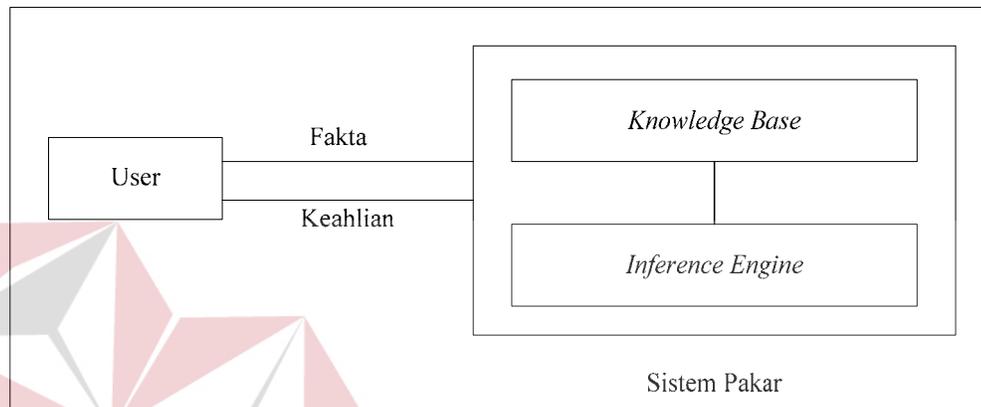
sebagai berikut :

1. Menentukan batasan-batasan dari suatu sistem pakar yang akan dirancang.
2. Memilih jenis keputusan yang akan diambil.
3. Meng-*extract* pengetahuan dari pakar, caranya yaitu dengan membuat *Dependency Diagram*.
4. Mempresentasikan pengetahuan dalam sistem pakar (membuat *rules*), salah satu teknik yang dapat digunakan yaitu mempresentasikan ke dalam bentuk *IF-THEN rules*.
5. Membuat *inference engine* dengan menggunakan metode yang sesuai.
6. Merancang *user interface*.

2.1.1 Konsep dasar sistem pakar

Konsep dasar sistem pakar dapat di lihat pada Gambar 2.1. Pengguna menyampaikan fakta atau informasi untuk sistem pakar dan kemudian menerima saran dari pakar atau jawaban ahlinya. Ada dua kapabilitas penting dari seorang pakar (*expert*), yang dicoba untuk dimodelkan pada sistem pakar yaitu

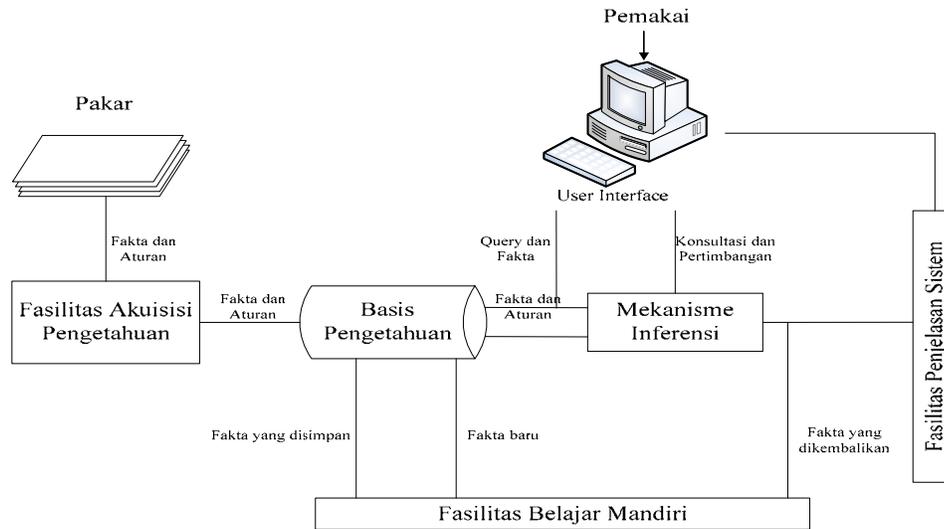
pengetahuan (*knowledge*) dan konsep berpikir (*reasoning*) dari sang pakar. Untuk menghasilkan kedua hal tersebut, sistem pakar harus terdiri dari dua komponen utama, yaitu *knowledge base* yang berisi *knowledge* dan *inference engine* yang menggambarkan kesimpulan. Kesimpulan tersebut merupakan respons dari sistem pakar atas permintaan pengguna.



Gambar 2.1 Konsep Dasar Sistem Pakar

2.1.2 Komponen utama sistem pakar

Menurut Andi (2003) suatu sistem disebut sistem pakar jika mempunyai ciri dan karakteristik tertentu. Hal ini juga harus didukung oleh komponen-komponen sistem pakar yang mampu menggambarkan tentang ciri dan karakteristik tersebut. Komponen utama pada sistem pakar dapat digambarkan pada gambar 2.2.



Gambar 2.2 Struktur Bagan Sistem Pakar

Dari gambar 2.2 komponen sistem pakar, sebenarnya dapat disimpulkan bahwa ada tiga unsur penting dari pengembangan sistem pakar, yaitu adanya pakar, pemakai dan sistem.

Pakar adalah orang yang mempunyai penjelasan khusus akan suatu masalah. Dalam sistem, pengalaman tersebut disimpan sebagai basis aturan. Sedangkan pemakai adalah orang yang ingin berkonsultasi dengan pakar lewat sistem. Sistem sendiri menyediakan berbagai fasilitas untuk menghubungkan pakar dengan pemakai.

2.1.3 Keuntungan dan kelemahan sistem pakar

Keuntungan pemakaian sistem pakar menurut Kusri (2006) adalah :

1. Membuat seorang awam dapat bekerja seperti layaknya seorang pakar
2. Dapat bekerja dengan informasi yang tidak lengkap atau tidak pasti

3. Meningkatkan output yang produktivitas. Sistem pakar dapat bekerja lebih cepat dari manusia. Keuntungan ini berarti mengurangi pekerja yang dibutuhkan, dan akhirnya dapat mereduksi biaya
4. Meningkatkan kualitas
5. Sistem pakar menyediakan nasehat yang konsisten dan dapat mengurangi tingkat kesalahan
6. Membuat peralatan yang kompleks lebih mudah dioperasikan karena sistem pakar dapat melatih pekerja yang tidak berpengalaman
7. Handal (*reliability*)
8. Sistem pakar tidak dapat lelah atau bosan. Juga konsisten dalam memberikan jawaban dan selalu memberikan perhatian penuh.
9. Memiliki kemampuan untuk memecahkan masalah yang kompleks
10. Memungkinkan pemindahan pengetahuan ke lokasi yang jauh serta memperluas jangkauan seorang pakar, dapat diperoleh dan dapat dipakai dimana saja. Merupakan arsip yang terpercaya dari sebuah keahlian sehingga *user* seolah-olah berkonsultasi langsung dengan sang pakar meskipun sang pakar telah pensiun.

Kelemahan sistem pakar menurut Kusri (2006) adalah :

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. Hal ini dapat dilihat pada tabel 2.2.

Tabel 2.2 Perbandingan perangkat lunak konvensional dengan perangkat lunak Sistem Pakar

Perangkat Lunak Konvensional	Perangkat Lunak Sistem Pakar
Fokus pada solusi	Fokus pada permasalahan
Pengembangan dapat dilakukan secara individu	Pengembangan dilakukan oleh tim kerja
Pengembangan secara sekuensial	Pengembangan secara iteratif

2.1.4 Knowledge base

“*Knowledge base* adalah bagian dari sistem pakar yang berisi sumber pengetahuan” (Durkin, 1994).

Knowlede base berisi pengetahuan spesifik yang disediakan oleh seorang pakar untuk memecahkan suatu masalah tertentu, dimana di dalamnya terdapat fakta, *rules*, konsep, dan hubungan antara *rules* maupun antara fakta dari suatu masalah tersebut.

Untuk mengkodekan pengetahuan dari seorang pakar ke dalam *knowledge base* dapat digunakan beberapa teknik. Salah satu teknik yang paling umum yang digunakan untuk menggambarkan pengetahuan dalam sistem pakar yaitu *rules*.

Pada sistem pakar setiap kasus akan diproses berdasarkan fakta yang ada, baru kemudian dapat diambil suatu kesimpulan. Apabila terjadi suatu kasus yang belum pernah ada, maka sistem pakar tidak mampu mengambil kesimpulan dari kasus tersebut. Untuk itu diperlukan penambahan fakta baru untuk *knowledge base*. Jadi, suatu *knowledge base* dikatakan baik apabila memiliki *rules* yang dapat digunakan dalam setiap kemungkinan kasus dalam ruang lingkup tertentu.

Berdasarkan dari *knowledge base* yang dimilikinya, sistem pakar dapat digolongkan menjadi 5 tingkatan, yaitu :

1. Tingkat 1
 - a. Menggunakan *internal knowledge base*.
 - b. Dapat melakukan penambahan data dan membuat laporan data yang ada pada *knowledge base*.
 - c. Menggunakan metode *Backward Chaining* atau metode *Forward Chaining* untuk menyelesaikan suatu kasus.
 - d. Mampu menampilkan hasil kerja dari *inference engine* melalui *user interface*.
2. Tingkat 2
 - a. Menggunakan *external knowledge base*.
 - b. Mampu memilih metode *Backward Chaining* atau *Forward Chaining* yang akan digunakan untuk menyelesaikan suatu kasus.
 - c. Mampu melakukan penghitungan matematis.
3. Tingkat 3
 - a. Mampu melakukan ekspor atau impor suatu *knowledge base*.
 - b. Mampu melakukan penghitungan matematis secara dinamis.
4. Tingkat 4
 - a. Dapat dioperasikan pada berbagai *operating sistem*.
5. Tingkat 5
 - a. Mampu merubah atau menambah isi *knowledge base* secara otomatis dan mempelajari suatu pola baru dari beberapa kasus yang pernah dialaminya.

2.1.5 Working memory

“*Working memory* adalah bagian dari sistem pakar yang berisi fakta-fakta dari suatu masalah yang ditemukan selama proses” (Durkin, 1994).

Working memory berisi semua informasi mengenai suatu masalah baik yang itu disediakan oleh *user* maupun dari kesimpulan yang diperoleh sistem. Banyak aplikasi sistem pakar dapat menggunakan informasi yang terdapat dalam penyimpanan eksternal seperti *database*, *spreadsheets*, atau *sensors*.

2.1.6 Inference engine

“*Inference engine* adalah *processor* dalam suatu sistem pakar dimana mencocokkan fakta-fakta yang terdapat dalam *working memory* dengan domain pengetahuan yang terdapat dalam *knowledge base*, untuk menggambarkan kesimpulan mengenai suatu masalah” (Durkin, 1994).

Inference engine akan mencari *rules* dimana premis dari *rules* tersebut cocok dengan informasi yang terdapat dalam *working memory*. Setelah menemukan kecocokan, *Inference engine* akan menambahkan kesimpulan *rules* tersebut ke dalam *working memory* dan melanjutkan mencari *rules* untuk kecocokan yang baru.

2.1.7 IF-THEN rules

Rule dimana secara logika menghubungkan informasi di bagian *IF* dengan informasi di bagian *THEN*.

IF : Sebagai penanda awal kondisi pada sebuah *rule*.

THEN : Sebagai penanda awal kesimpulan pada sebuah *rule*.

ELSE : Sebagai penanda awal alternatif kesimpulan pada sebuah *rule*, bersifat *optional*.

Operator yang dapat digunakan pada *IF-THEN rule* adalah :

AND : Semua kondisi yang dihubungkan oleh operator *AND* harus bernilai benar, agar kondisi keseluruhan *rule* tersebut bernilai benar. Bila ada satu kondisi yang bernilai salah, keseluruhan *rule* tersebut bernilai salah.

OR : Semua kondisi yang dihubungkan oleh operator *OR* harus bernilai salah, agar kondisi keseluruhan *rule* tersebut bernilai salah. Bila ada satu kondisi atau lebih yang bernilai benar, keseluruhan *rule* tersebut bernilai benar.

Contoh :

Diagnosa masalah pada mobil

RULE 1

IF The car will not start

THEN The problem may be in the electrical sistem

RULE 2

IF The problem may be in the electrical sistem

AND The battery voltage is below 10 volts

THEN The fault is a bad battery

2.1.8 User interface

User Interface merupakan interaksi antara seorang pakar dengan *user* yang disajikan dalam *natural language style*, dimana *user* memasukkan *input* dan akan menerima *output*. *User interface* memberikan fasilitas komunikasi antara pemakai dan sistem, memberikan berbagai fasilitas informasi dan berbagai keterangan

yang bertujuan untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan solusi. Pada umumnya, *user interface* juga berfungsi untuk menginputkan pengetahuan baru ke dalam basis pengetahuan sistem pakar, menampilkan fasilitas penjelasan sistem dan memberikan tuntunan penggunaan sistem secara menyeluruh langkah demi langkah sehingga *user* mengerti apa yang harus dilakukan terhadap sistem. Karena dimaksudkan untuk *user*, maka *interface* harus dirancang sebisa mungkin memberi kemudahan kepada *user* dalam mengoperasikan sistem dan se jelas mungkin sehingga tidak terjadi salah interpretasi. Jadi, *interface* haruslah *user friendly*.

2.1.9 Verifikasi

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 yang 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 kecocokan antara sistem dengan apa yang sistem kerjakan dan juga untuk memastikan bahwa sistem itu terbebas dari error.

Berikut adalah yang harus di cek dalam suatu basis pengetahuan :

1. *Redundant rules*

Dikatakan *redundant rule* jika dua *rule* atau lebih mempunyai premis dan *conclusion* yang sama.

Contoh :

Rule 1 : If Motor starter rusak And Temperatur mesin naik

Then Mesin tidak berfungsi.

Rule 2 : If Temperatur mesin naik And Motor starter rusak

Then Mesin tidak berfungsi

2. *Conflicting rules*

Conflicting rules terjadi ketika dua buah *rule* atau lebih mempunyai premis yang sama, tetapi mempunyai *conclusion* yang berlawanan.

Contoh :

Rule 1 : If Motor starter rusak And Temperatur mesin naik

Then Mesin tidak berfungsi

Rule 2 : If Temperatur mesin naik And Motor starter rusak

Then mesin berfungsi

3. *Subsumed rules*

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

Contoh :

Rule 1 : If Motor rusak And Temperatur mesin naik

And Pompa bahan bakar rusak

Then Mesin tidak berfungsi

Rule 2 : If Temperatur mesin naik And Motor starter rusak
Then Mesin tidak berfungsi

4. *Circular rules*

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

Contoh :

Rule 1 : If Lala dan Lili kakak beradik

Then Lala dan Lili mempunyai orang tua yang sama

Rule 2 : If Lala dan Lili mempunyai orang tua yang sama

Then Lala dan Lili kakak beradik

5. *Unecessary IF condition*

Unecessary if terjadi ketika dua rule atau lebih mempunyai *conclusion* yang sama, tetapi salah satu dari rule tersebut mempunyai premis yang tidak perlu dikondisikan dalam rule karena tidak mempunyai pengaruh apapun.

Rule 1 : If Mesin tidak dapat di-starter And Motor starter tidak bekerja And Klakson tidak berbunyi

Then Baterai perlu di setrum

Rule 2 : If Mesin tidak dapat di-starter And Motor starter tidak berbunyi And Klakson berbunyi

Then Baterai perlu disetrum

6. *Dead-end rules*

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

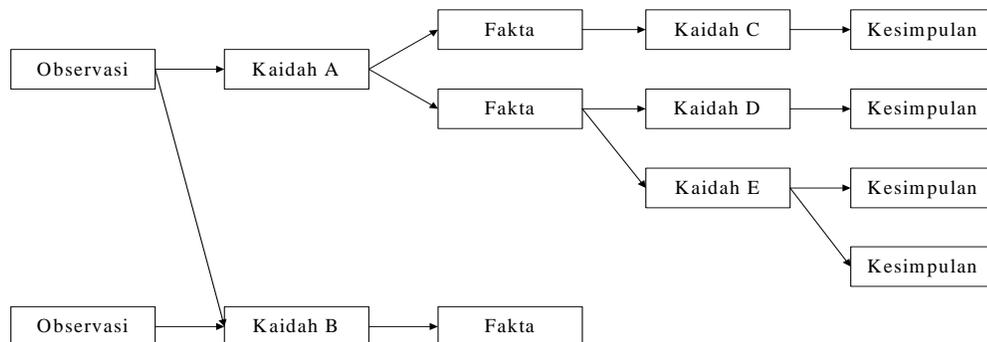
Contoh :

Rule 1 : If Mesin tidak dapat di-start And Motor starter tidak
Bekerja And Klakson tidak berbunyi
Then Nonton pesawat.

2.2 Metode Forward Chaining

“*Forward Chaining* adalah strategi *inference* yang bermula dari sejumlah fakta yang diketahui, dengan menggunakan *rules* yang premisnya cocok dengan fakta yang diketahui tersebut untuk memperoleh fakta baru dan melanjutkan proses hingga *goal* dicapai atau hingga sudah tidak ada *rules* lagi yang premisnya cocok dengan fakta yang diketahui maupun fakta yang diperoleh” (Durkin, 1994).

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, maka kaidah distimulasi (Agung Saputra, 2007). Untuk lebih jelasnya dapat kita lihat pada gambar 2.3



Gambar 2.3 Metode *Forward Chaining*

Pada gambar 2.3 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. motor inferensi mulai melakukan pelacakan, mencocokkan kaidah A dalam pangkalan pengetahuan terhadap informasi yang ada didalam 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 1, demikian seterusnya.

Metode *Forward Chaining* dimulai dari kiri ke kanan, yaitu dimulai dari premis kepada kesimpulan akhir, disebut sebagai *data driven* (yaitu, pencarian yang dikendalikan oleh data yang diberikan). Metode ini lebih baik digunakan apabila memiliki sedikit premis dan banyak kesimpulan.

Menurut Durkin (1994) adapun kelebihan dan kelemahan dari metode *Forward Chaining* yaitu :

a. Kelebihan:

1. Kelebihan utama dari *Forward Chaining* yaitu metode ini akan bekerja dengan baik ketika problem bermula dari mengumpulkan atau menyatukan informasi lalu kemudian mencari kesimpulan apa yang dapat diambil dari informasi tersebut.
2. Metode ini mampu menyediakan banyak sekali informasi dari hanya sejumlah kecil data.

3. Merupakan pendekatan paling sempurna untuk beberapa tipe dari *problem solving tasks*, yaitu *planning*, *monitoring*, *control*, dan *interpretation*.

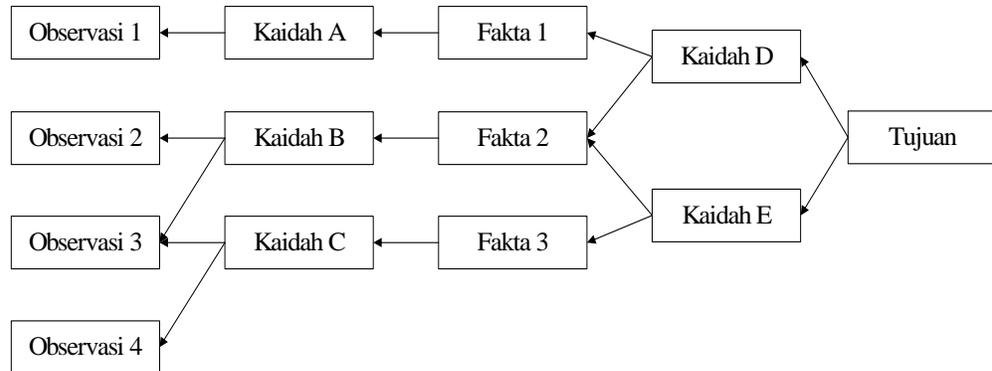
b. Kelemahan:

1. Kelemahan utama dari metode ini adalah tidak dapat mengenali dimana beberapa fakta lebih penting dari fakta lainnya.
2. Sistem dapat menanyakan pertanyaan yang tidak berhubungan, meskipun jawaban dari pertanyaan tersebut penting. Hal ini dapat membingungkan *user* dalam menjawab pertanyaan pada subyek yang tidak berhubungan.

2.3 Metode Backward Chaining

Metode *Backward Chaining* merupakan kebalikan dari metode Forward Chaining atau disebut sebagai penalaran mundur yaitu suatu metode yang digunakan dalam mekanisme inferensi untuk melakukan pelacakan atau penalaran dari sekumpulan hipotesa menuju fakta-fakta yang mendukung kesimpulan tersebut. Jadi interpreter kaidah mulai menguji kaidah sebelah kanan yaitu then. Mesin inferensi akan melacak bukti-bukti yang mendukung hipotesa awal. Jika ternyata sesuai, maka basis data akan mencatat kondisi if yang benar-benar sesuai digunakan untuk menghasilkan hipotesa yang baru dan keadaan tujuan, yang kemudian direkam dalam basis data. Keadaan diatas terus berlangsung sampai hipotesa terbukti kebenarannya (Agung Saputra, 2007).

Untuk lebih jelasnya dapat kita lihat alur dari metode backward chaining pada gambar 2.4.



Gambar 2.4 Metode *Backward Chaining*

Dalam melakukan penelusuran pada *Backward Chaining* berawal dari goal atau pada gambar 2.4 disebut sebagai tujuan, kemudian mencari informasi untuk memenuhi tujuan tersebut. Pertama-tama mulai dengan memberitahu sistem, bahwa kita ingin membuktikan keadaan tujuan. Motor inferensi melihat pangkalan data yaitu fakta untuk dicocokkan dengan pangkalan kaidah.

Terdapat tiga faktor yang mempengaruhi pada pemilihan metode forward chaining dan backward chaining yaitu :

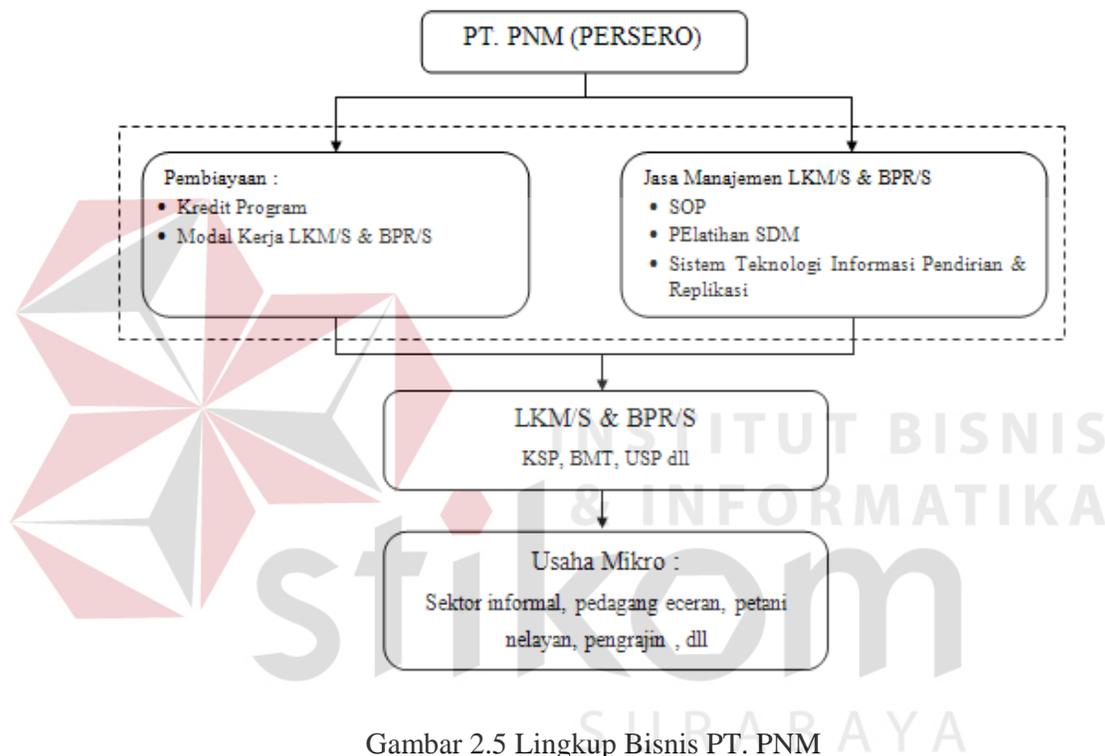
1. Manakah yang lebih banyak, kaidah awal atau kaidah akhir yang dimungkinkan? Kita akan bergerak dari himpunan keadaan yang lebih kecil menuju ke himpunan yang lebih besar (sehingga akan lebih mudah untuk mendapatkan yang kita inginkan).
2. Kearah manakah faktor percabangan menjadi lebih besar (yakni jumlah simpul teratas yang dapat dicapai secara langsung dari sebuah simpul tunggal)? Kita akan bergerak kearah yang memiliki faktor percabangan lebih kecil (<http://iel.ipb.ac.id/agrinetmedia/modul/aplikasi/servercabai/pakar.htm>).

Akankah program diminta untuk menjelaskan proses penalaran yang digunakan kepada pemakai? Jika ya, maka pemrosesan ke arah yang bersesuaian

”jarak” yang lebih dekat seperti yang diharapkan oleh pemakai akan menjadi penting.

2.4 Lingkup Bisnis PT. PNM

Lingkup bisnis pada PT. PNM terdapat pada gambar 2.5



Gambar 2.5 Lingkup Bisnis PT. PNM

2.4.1 Visi perusahaan

Menjadi lembaga pembiayaan terkemuka dalam meningkatkan nilai tambah secara berkelanjutan bagi Usaha Mikro, Kecil, Menengah dan Koperasi (UMKMK) dengan berlandaskan prinsip-prinsip *Good Corporate Governance (GCG)*.

2.4.2 Misi perusahaan

1. Meningkatkan kelayakan usaha dan kemampuan wirausaha UMKMK
2. Meningkatkan akses pembiayaan UMKMK kepada lembaga keuangan baik bank maupun non bank dalam rangka perluasan lapangan kerja dan peningkatan kesejahteraan masyarakat
3. Meningkatkan kretifitas dan produktifitas karyawan untuk mencapai kinerja terbaik dalam mengembangkan UMKMK.

2.4.3 Tujuan perusahaan

Turut melaksanakan dan menunjang kebijakan dan program Pemerintah di bidang ekonomi dan pembangunan nasional pada umumnya, khususnya di bidang pemberdayaan dan pengembangan koperasi, usaha mikro kecil dan menengah dengan menerapkan prinsip-prinsip perseroan terbatas.

2.4.4 Strategi perusahaan

Strategi perusahaan pada PT. PNM terdiri dari dua yaitu :

a. Strategi Pembiayaan

1. Meningkatkan fokus pada pembiayaan usaha mikro melalui Lembaga Keuangan Mikro/Syariah
2. Advancement skema kredit program untuk Kredit Usaha Kecil & Mikro (KUKM), bekerjasama dengan bank pelaksana dan Bank Indonesia untuk pola kemitraan, serta UMKMK di sentra usaha.

b. Strategi Jasa Manajemen LKM

1. Fokus dan penyederhanaan model bisnis dan mengikuti praktek umum konsultasi *Survey-Training-Consultancy Monitoring* (STCM)
2. Merupakan sumber dana jasa manajemen dari dana PKBL (Program Kemitraan dan Bina Lingkungan) BUMN.

2.4.5 Sasaran strategi perusahaan

Sasaran strategi perusahaan pada PT. PNM adalah :

a. Perspektif Keuangan

1. Memaksimalkan profit dan mengelola biaya lebih efisien
2. Meningkatkan *shareholder value* yang berkesinambungan.

b. Perspektif Stakeholder

1. Memperluas jangkauan serta akses pembiayaan dan jasa
2. Manajemen dalam rangka memberdayakan UMKMK & LKMS.

c. Perspektif Proses Bisnis

1. Meningkatkan penghimpunan dan pengelolaan dana
2. Memperluas serta memperkuat jaringan dan pelayanan
3. Menerapkan Sistem Teknologi & informasi yang tepat guna
4. Meningkatkan efektifitas penerapan Sistem Manajemen.

d. Perspektif SDM dan Budaya

1. Meningkatkan kualitas dan produktivitas SDM
2. Meningkatkan kepuasan SDM dan pengembangan budaya kerja.

2.4.6 Produk pembiayaan

Sebagai salah satu Koordinator pelaksanaan 12 Skim Kredit Program eks Kredit Likuiditas Bank Indonesia (KLBI), PT. PNM berusaha menjangkau seluruh wilayah Indonesia dalam penyaluran pembiayaan kepada UMKMK melalui bank pelaksana. Selain penyaluran kredit, PT. PNM juga memberikan bantuan pembiayaan kepada UMKMK melalui Lembaga Keuangan baik Konvensional maupun Syariah.

Jenis produk pembiayaan yang terdapat pada PT. PNM cabang Surabaya untuk BPR/S dan LKM/S adalah :

a. PMK (Pembiayaan Modal Kerja) Madani

1. Definisi

Produk ini diajukan kepada LKM/S atau BPR/S yang sudah beroperasi minimal 2 tahun yang membutuhkan tambahan modal kerja (Komersial/Syariah) untuk pengembangan usahanya.

2. Target

LKM/S dan BPR/S seluruh Indonesia.

3. Jangka Waktu Pembayaran

Maksimal 3 tahun (36 bulan).

4. Batasan Pembiayaan

Minimal Rp.100.000.000,-(seratus juta rupiah).

b. SOL (Sub Ordinate Loan) Madani

1. Definisi

Produk pembiayaan kepada BPR/S dalam bentuk pinjaman sub ordinasi (yang masuk kedalam bagian dari modal pelengkap). Produk ini

memberikan fasilitas kepada BPR/S untuk memperbaiki rasio kecukupan modal secara bertahap, serta menyediakan alternatif sumber dana jangka panjang untuk perkuatan portofolio pembiayaan.

2. Target

BPR/S diseluruh Indonesia, baik berbadan hokum PT, PD, maupun koperasi, yang memenuhi persyaratan dan ketentuan produk.

3. Jangka Waktu Pembiayaan

Maksimal 5 (lima) tahun, termasuk masa tenggang 1 (satu) tahun.

4. Batasan Pembiayaan

Plafond pembiayaan SOL MADANI BPR/S besarnya sampai dengan 50% dari modal inti.

c. SUP (Sarana Usaha Produktif) Madani BPR/S

1. Definisi

Produk pembiayaan modal kerja berbunga rendah yang disalurkan kepada pelaku usaha mikro, kecil dan koperasi untuk usaha produktif melalui BPR/S dengan pola executing.

2. Target

BPR/S diseluruh Indonesia, baik berbadan hukum PT, PD, maupun koperasi.

3. Jangka Waktu Pembiayaan

Maksimal 3 tahun (36 bulan).

4. Batasan Pembiayaan

Sesuai dengan hasil evaluasi kebutuhan BPR/S.

d. SUP (Sarana Usaha Produktif) Madani LKM/S

1. Definisi

Produk pembiayaan modal kerja bunga rendah yang disalurkan kepada pelaku usaha mikro, kecil dan koperasi untuk usaha produktif melalui Lembaga Keuangan Mikro/Syariah (LKM/S) Non Bank dengan pola *executing* (“SUP = Sarana Usaha Produktif Madani” melalui Koperasi Simpan Pinjam, Unit Simpan Pinjam Koperasi dan BaitulMaal wat Tamwil, serta Lembaga Keuangan Mikro lainnya).

2. Target

LKM/S Non Bank di seluruh Indonesia, antara lain KSP, USP, BMT, dan LKM/S bukan bank lainnya yang telah berbadan hukum.

3. Jangka Waktu Pembayaran

Maksimal 3 tahun (36 bulan).

4. Batasan Pembiayaan

Sesuai dengan hasil evaluasi kebutuhan LKM/S Non Bank.

e. PMK (Pembiayaan Modal Kerja) Perumahan BRP/BPRS

1. Definisi

Produk pembiayaan khusus masalah perumahan (perbaikan, pembangunan dan yang berkenaan dengan kualitas pemukiman dan lingkungan dengan pola penyaluran *Executing* melalui Lembaga Keuangan Lokal yang berupa Lembaga Keuangan Mikro khususnya BPR/S).

2. Target

BPR/S yang ada di 32 kota/kota kabupaten yang menjadi target NUSPP.

3. Jangka Waktu Pembayaran

Maksimal 5 tahun (60 bulan).

4. Batasan Pembiayaan

Sesuai dengan hasil evaluasi kebutuhan BPR/S Maksimum Rp. 2 Milyar dan debitur (*end user*) maksimum Rp. 10 juta.

f. PMK (Pembiayaan Modal Kerja) Perumahan LKM/S Non Bank

1. Definisi

Produk pembiayaan khusus masalah perumahan (Perbaikan, pembangunan dan yang berkenaan dengan peningkatan kualitas pemukiman dan lingkungan dengan pola penyaluran *Executing* melalui Lembaga Keuangan

Lokal yang berupa Lembaga Keuangan Mikro khususnya LKM/S Non Bank).

2. Target

LKM/S Non Bank yaitu Koperasi Simpan Pinjam, Unit Simpan Pinjam Koperasi, BMT, Serta Lembaga Keuangan Mikro yang ada di 32 kota/kota kabupaten yang menjadi target NUSSP.

3. Jangka Waktu Pembayaran

Maksimal 5 tahun (60 bulan).

4. Batasan Pembiayaan

Sesuai dengan hasil evaluasi kebutuhan BPR/S Maksimum Rp. 2 Milyar dan debitur (*end user*) maksimum Rp. 10 juta.