

# Sistem Pakar

---

MODUL KULIAH

Raissa Amanda Putri, M.Ti

PROGRAM STUDI ILMU KOMPUTER | FAKULTAS SAINS DAN TEKNOLOGI UINSU MEDAN

## DAFTAR ISI

DAFTAR ISI .....	1
Capaian Mata Kuliah.....	4
Materi Pembelajaran.....	4
BAB I .....	5
Kecerdasan buatan.....	5
A.    Defenisi Kecerdasan Buatan.....	5
B.    Kecerdasan Buatan Dengan Kecerdasan Alami .....	7
C.    Komputasi Kecerdasan Buatan dan Komputasi Konvensional .....	9
D.    Sejarah Kecerdasan Buatan .....	10
E.    Lingkup Kecerdasan Buatan Pada Aplikasi Komersial .....	11
BAB II .....	13
Konsep sistem pakar .....	13
A.    Konsep Dasar .....	13
B.    Keunggulan dan Keuntungan Menggunakan Sistem Pakar .....	15
C.    Konsep Umum Sistem Pakar .....	16
D.    Struktur Sistem Pakar .....	17
BAB III .....	24

	2
Jenis-jenis sistem pakar.....	24
BAB IV.....	26
Masalah dan pengetahuan dalam sistem pakar .....	26
A.    Sistem Perantaraan Maju (Forward Chaining Systems) .....	26
B.    Strategi penyelesaian konflik (conflict resolution strategy) .....	30
C.    Sistem Perantaraan Balik (Backward Chaining Systems).....	31
D.    Pemilihan Sistem Inferensi.....	34
E.    Ketidakpastian dalam Aturan .....	34
BAB V.....	36
Komponen – komponen sistem pakar.....	36
BAB VI.....	40
Mesin inferensi.....	40
A.    Strategi penyelesaian konflik (conflict resolution strategy) .....	40
B.    Sistem Perantaraan Maju (Forward Chaining Systems) .....	41
C.    Sistem Perantaraan Balik (Backward Chaining Systems).....	44
D.    Ketidakpastian dalam Aturan .....	47
BAB VII.....	48
Metode penanganan ketidakpastian dengan sistem pakar.....	48
A.    Ketidakpastian .....	48
B.    Teknik Probabilitas.....	50
C.    Faktor Kepastian.....	52

	3
D. Logika Fuzzy.....	55
BAB VIII.....	59
Pengembangan sistem pakar .....	59
A. Aplikasi Sederhana: Sistem Pakar Bengkel Mobil .....	59
B. Eliza.....	64
C. Parry .....	65
BAB IX.....	67
Contoh sistem pakar .....	67
A. Implementasi sistem pakar dalam bidang kesehatan .....	67
B. Contoh Skripsi Aplikasi Sistem Pakar .....	69
DAFTAR PUSTAKA.....	71

## **Capaian Mata Kuliah**

- (1) Mahasiswa mampu memahami model-model representasi pengetahuan,
- (2) Mahasiswa mampu memiliki kemampuan untuk menarik kesimpulan (inference) dari fakta yang digambarkan dalam model-model representasi,
- (3) Mahasiswa mampu menentukan pendekatan sistem cerdas yang sesuai dengan masalah yang dihadapi

## **Materi Pembelajaran**

- (1) Kecerdasan buatan
- (2) Konsep sistem pakar
- (3) Jenis-jenis sistem pakar
- (4) Masalah dan pengetahuan dalam sistem pakar
- (5) Komponen – komponen sistem pakar
- (6) Mesin inferensi
- (7) Metode penanganan ketidakpastian dengan sistem pakar
- (8) Pengembangan sistem pakar
- (9) Contoh sistem pakar

## **BAB I**

### **Kecerdasan buatan**

#### **A. Defenisi Kecerdasan Buatan**

Kecerdasan buatan atau artificial intelligence merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia.

Didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu entitas buatan. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia.

Artificial Intelligence (Inteligensi/Kecerdasan Buatan) merupakan salah satu bidang dari ilmu komputer yang membahas tentang kemungkinan komputer untuk dapat berlaku secara intelligen seperti halnya manusia.

Artificial Intelligence (AI) merupakan sub bidang pengetahuan computer yang khusus ditujukan untuk membuat software dan hardware yang sepenuhnya bisa menirukan beberapa fungsi otak manusia.

Lebih detilnya, pengertian kecerdasan buatan dapat dipandang dari berbagai sudut pandang, antara lain:

1. Sudut pandang kecerdasan.

Kecerdasan Buatan akan membuat mesin menjadi 'cerdas' (mampu berbuat seperti apa yang dilakukan oleh manusia).

## 2. Sudut pandang penelitian.

Kecerdasan Buatan adalah suatu studi bagaimana membuat agar komputer dapat melakukan sesuatu sebaik yang dikerjakan oleh manusia.

Domain yang sering dibahas oleh para peneliti meliputi:

### a. Mundane task

- Persepsi (vision & speech).
- Bahasa alami (understanding, generation & translation).
- Pemikiran yang bersifat commonsense.
- Robot control.

### b. Formal task

- Permainan/games.
- Matematika (geometri, logika, kalkulus integral, pembuktian).

### c. Expert task

- Analisis finansial.
- Analisis medikal.
- Analisis ilmu pengetahuan.
- Rekayasa (desain, pencarian kegagalan, perencanaan manufaktur).

## 3. Sudut pandang bisnis.

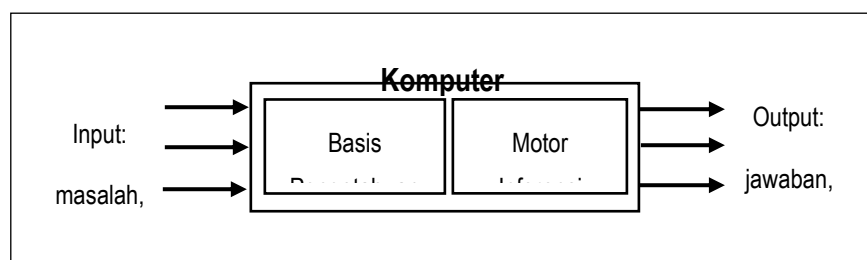
Kecerdasan buatan adalah kumpulan peralatan yang sangat powerful dan metodologis dalam menyelesaikan masalah-masalah bisnis.

## 4. Sudut pandang pemrograman.

Kecerdasan buatan meliputi studi tentang pemrograman simbolik, penyelesaian masalah (problem solving) dan pencarian (searching).

Untuk melakukan aplikasi kecerdasan buatan ada 2 bagian utama yang sangat dibutuhkan (Gambar 1.1), yaitu:

1. Basis Pengetahuan (Knowledge Base), berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
2. Motor Inferensi (Inference Engine), yaitu kemampuan menarik kesimpulan berdasarkan pengalaman.



Gambar 1.1 Penerapan Konsep Kecerdasan Buatan di Komputer.

## B. Kecerdasan Buatan Dengan Kecerdasan Alami

Kecerdasan Alami: kecerdasan yang dimiliki oleh manusia.

Keuntungan kecerdasan buatan secara komersil:

1. Kecerdasan buatan lebih bersifat permanen. Kecerdasan alami akan cepat mengalami perubahan. Hal ini dimungkinkan karena sifat manusia yang pelupa. Kecerdasan buatan tidak akan berubah sepanjang sistem komputer & program tidak mengubahnya.
2. Kecerdasan buatan lebih mudah diduplikasi & disebar. Mentransfer pengetahuan manusia dari satu orang ke orang lain membutuhkan proses yang sangat lama; dan juga suatu keahlian itu tidak akan pernah dapat diduplikasi



dengan lengkap. Oleh karena itu, jika pengetahuan terletak pada suatu sistem komputer, pengetahuan tersebut dapat disalin dari komputer tersebut dan dapat dipindahkan dengan mudah ke komputer yang lain.

3. Kecerdasan buatan lebih murah dibanding dengan kecerdasan alami. Menyediakan layanan komputer akan lebih mudah & lebih murah dibandingkan dengan harus mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.
4. Kecerdasan buatan bersifat konsisten. Hal ini disebabkan karena kecerdasan buatan adalah bagian dari teknologi komputer. Sedangkan kecerdasan alami akan senantiasa berubah-ubah.
5. Kecerdasan buatan dapat didokumentasi. Keputusan yang dibuat oleh komputer dapat didokumentasi dengan mudah dengan cara melacak setiap aktivitas dari sistem tersebut. Kecerdasan alami sangat sulit untuk direproduksi.
6. Kecerdasan buatan dapat mengerjakan pekerjaan lebih cepat dibanding dengan kecerdasan alami.
7. Kecerdasan buatan dapat mengerjakan pekerjaan lebih baik dibanding dengan kecerdasan alami.

Keuntungan kecerdasan alami:

1. Kreatif. Kemampuan untuk menambah ataupun memenuhi pengetahuan itu sangat melekat pada jiwa manusia. Pada kecerdasan buatan, untuk menambah pengetahuan harus dilakukan melalui sistem yang dibangun.
2. Kecerdasan alami memungkinkan orang untuk menggunakan pengalaman secara langsung. Sedangkan pada kecerdasan buatan harus bekerja dengan input-input simbolik.
3. Pemikiran manusia dapat digunakan secara luas, sedangkan kecerdasan buatan sangat terbatas.

### C. Komputasi Kecerdasan Buatan dan Komputasi Konvensional

Komputasi konvensional: komputer hanya diperuntukkan sebagai alat hitung.

Perbedaan komputasi kecerdasan buatan dengan komputasi konvensional terlihat pada Tabel 1.1.

Tabel 1.1 Kecerdasan Buatan Vs. Pemrograman Konvensional

<b>Dimensi</b>	<b>Kecerdasan Buatan</b>	<b>Pemrograman Konvensional</b>
<b>Pemrosesan</b>	Mengandung konsep-konsep simbolik	Algoritmik
<b>Sifat Input</b>	Bisa tidak lengkap	Harus lengkap

<b>Pencarian</b>	Kebanyakan bersifat heuristic	Biasanya didasarkan pada algoritma
<b>Keterangan</b>	Disediakan	Biasanya tidak disediakan
<b>Fokus</b>	Pengetahuan	Data & informasi
<b>Struktur</b>	Kontrol dipisahkan dari pengetahuan	Kontrol terintegrasi dengan informasi (data)
<b>Sifat output</b>	Kuantitatif	Kualitatif
<b>Pemeliharaan &amp; update</b>	Relatif mudah	Sulit
<b>Kemampuan menalar</b>	Ya	Tidak

#### D. Sejarah Kecerdasan Buatan

Uji Turing (Turing Test) oleh Alan Turing tahun 1950-an : Turing beranggapan bahwa, jika mesin dapat membuat seseorang percaya bahwa dirinya mampu berkomunikasi dengan orang lain, maka dapat dikatakan bahwa mesin tersebut cerdas (seperti layaknya manusia).

Seorang profesor dari Massachusetts Institute of Technology yang bernama John McCarthy pada tahun 1956 pada Dartmouth Conference yang dihadiri oleh para peneliti AI. Pada konferensi tersebut juga didefinisikan tujuan utama dari kecerdasan buatan, yaitu: mengetahui dan memodelkan proses-proses berfikir manusia dan mendesain mesin agar dapat menirukan kelakuan manusia tersebut.

Beberapa program AI yang mulai dibuat pada tahun 1956-1966, antara lain:

1. Logic Theorist, diperkenalkan pada Dartmouth Conference, program ini dapat membuktikan teorema-teorema matematika.
2. Sad Sam, diprogram oleh Robert K. Lindsay (1960). Program ini dapat mengetahui kalimat-kalimat sederhana yang ditulis dalam bahasa Inggris dan mampu memberikan jawaban dari fakta-fakta yang didengar dalam sebuah percakapan.
3. ELIZA, diprogram oleh Joseph Weizenbaum (1967). Program ini mampu melakukan terapi terhadap pasien dengan memberikan beberapa pertanyaan.

## **E. Lingkup Kecerdasan Buatan Pada Aplikasi Komersial**

Lingkup utama dalam kecerdasan buatan adalah:

1. Sistem Pakar (Expert System). Disini komputer digunakan sebagai sarana untuk menyimpan pengetahuan para pakar. Dengan demikian komputer akan memiliki keahlian untuk menyelesaikan permasalahan dengan meniru keahlian yang dimiliki oleh pakar.

2. Pengolahan Bahasa Alami (Natural Language Processing). Dengan pengolahan bahasa alami ini diharapkan user dapat berkomunikasi dengan komputer dengan menggunakan bahasa sehari-hari.
3. Pengenalan Ucapan (Speech Recognition). Melalui pengenalan ucapan diharapkan manusia dapat berkomunikasi dengan komputer dengan menggunakan suara.
4. Robotika & Sistem Sensor (Robotics & Sensory Systems).
5. Computer Vision, mencoba untuk dapat menginterpretasikan gambar atau obyek-obyek tampak melalui komputer.
6. Intelligent Computer-aided Instruction. Komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.
7. Game Playing.

## BAB II

### Konsep sistem pakar

Menurut Turban yang namanya sering disebut-sebut sebagai ahli sistem pakar dunia di dalam bukunya, turban mengatakan bahwa:

“Sistem pakar (Expert System) adalah salah satu cabang dari AI (Artificial Intelligence) khusus untuk penyelesaian masalah tingkat manusia yang pakar. Sistem pakar adalah sistem yang menggunakan pengetahuan manusia yang terekam dalam komputer untuk memecahkan persoalan yang biasanya memerlukan keahlian manusia. Sedangkan AI itu sendiri atau yang disebut juga kecerdasan buatan adalah tingkah laku mesin yang jika dilakukan oleh manusia, akan disebut cerdas.”

#### A. Konsep Dasar

Konsep dasar sistem pakar menurut Turban mencakup beberapa persoalan mendasar, antara lain apa yang dimaksud dengan keahlian, siapa yang disebut pakar, bagaimana keahlian dapat ditransfer, dan bagaimana sistem bekerja.

Konsep dasar sistem pakar yaitu pengguna menyampaikan fakta atau informasi untuk sistem pakar dan kemudian menerima saran dari pakar atau jawaban ahlinya.

Bagian dalam sistem pakar terdiri dari 2 komponen utama yaitu knowledge base yang berisi knowledge dan mesin inferensi yang menggambarkan kesimpulan. Kesimpulan tersebut merupakan respon dari sistem pakar atas permintaan pengguna.

Pakar adalah orang yang memiliki pengetahuan, penilaian, pengalaman, dan metode khusus, serta kemampuan untuk menerapkan bakat ini dalam memberi nasihat dan memecahkan persoalan.

Keahlian adalah pengetahuan ekstensif yang spesifik terhadap tugas yang dimiliki pakar. Fitur-fitur yang harus dimiliki oleh sistem pakar adalah sebagai berikut:

1. Keahlian: pakar dibedakan dari tingkat keahlian mereka maka sistem pakar harus memiliki keahlian untuk memberi keputusan seperti seorang pakar.
2. Pertimbangan simbolik: pemikiran kecerdasan tiruan harus berdasarkan pada pertimbangan simbolik dari pada perhitungan matematika. Metode yang digunakan pada pertimbangan simbolik yaitu backward chaining atau forward chaining.
3. Deep knowledge (kedalaman pengetahuan): basis pengetahuan yang digunakan sistem pakar berasal dari seorang pakar, pengetahuan tersebut merupakan pengetahuan yang kompleks.

4. Self-knowledge: sistem pakar harus dapat menganalisis pertimbangannya sendiri dan menjelaskan kenapa bisa dicapai kesimpulan yang seperti itu

Fitur	Pakar Manusia	Sistem Pakar
Mortalitas	Ya	Tidak
Transfer pengetahuan	Sulit	Mudah
Dokumen Pengetahuan	Sulit	Mudah
Konsistensi Keputusan	Rendah	Tinggi
Unit Biaya Penggunaan	Tinggi	Rendah
Kreativitas	Tinggi	Rendah
Adaptabilitas	Tinggi	Rendah
Lingkup Pengetahuan	Luas	Sempit
Tipe Pengetahuan	Umum dan Teknis	Teknis
Isi Pengetahuan	Pengalaman	Simbol

Tabel 2.1. Perbedaan Pakar Manusia dan Sistem Pakar

## B. Keunggulan dan Keuntungan Menggunakan Sistem Pakar

Terdapat beberapa keunggulan sistem pakar, yang kita dapatkan jika kita menggunakannya, yaitu:

1. Menghimpun data dalam jumlah yang sangat besar.
2. Menyampaikan data tersebut untuk jangka waktu yang panjang dalam suatu bentuk tertentu.
3. Mengerjakan perhitungan secara cepat dan tepat dan tanpa jemu mencari kembali data yang tersimpan dengan kecepatan tinggi.

Sedangkan Keuntungan bila menggunakan sistem pakar diantaranya adalah:



1. Menjadikan pengetahuan dan nasihat lebih mudah didapat.
2. Meningkatkan output dan produktivitas.
3. Meningkatkan penyelesaian masalah, menerusi paduan pakar, penerangan, sistem pakar khas.
4. Meningkatkan reliabilitas.
5. Memberikan respon (jawaban) yang cepat.
6. Merupakan panduan yang intelligence (cerdas).
7. Dapat bekerja dalam informasi yang kurang lengkap dan mengandung ketidakpastian.
8. Intelligence database (basis data cerdas), bahwa sistem pakar dapat digunakan untuk mengakses basis data dengan cara cerdas.

### **C. Konsep Umum Sistem Pakar**

Kepakaran (expertise) adalah pengetahuan yang ekstensif dan spesifik yang diperoleh melalui rangkaian pelatihan, membaca, dan pengalaman. Pengetahuan membuat pakar dapat mengambil keputusan secara lebih baik dan lebih cepat daripada non-pakar dalam memecahkan problem yang kompleks. Kepakaran mempunyai sifat berjenjang, pakar top memiliki pengetahuan lebih banyak daripada pakar junior.

Tujuan Sistem Pakar adalah untuk mentransfer kepakaran dari seorang pakar ke komputer, kemudian ke orang lain (yang bukan pakar). Proses ini membutuhkan 4 aktivitas, yaitu tambahan pengetahuan (dari para ahli atau sumber lainnya), representasi pengetahuan (ke komputer), inferensi pengetahuan dan pengalihan pengetahuan ke pengguna.

Menurut Turban terdapat tiga orang yang terlibat dalam lingkungan sistem pakar, yaitu :

#### 1. Pakar

Pakar adalah orang yang memiliki pengetahuan khusus, pendapat, pengalaman dan metode, serta kemampuan untuk mengaplikasikan keahliannya tersebut guna menyelesaikan masalah.

#### 2. Knowledge engineer (Perekayasa Sistem)

Knowledge engineer adalah orang yang membantu pakar dalam menyusun area permasalahan dengan menginterpretasikan dan mengintegrasikan jawaban-jawaban pakar atas pertanyaan yang diajukan, menggambarkan analogi, mengajukan counter example dan menerangkan kesulitan-kesulitan konseptual.

#### 3. Pemakai

Sistem pakar memiliki beberapa pemakai, yaitu: pemakai bukan pakar, pelajar, pembangun sistem pakar yang ingin meningkatkan dan menambah basis pengetahuan, dan pakar.

### **D. Struktur Sistem Pakar**

Sistem Pakar dapat ditampilkan dengan dua lingkungan, yaitu: lingkungan pengembangan dan lingkungan konsultasi (runtime) (Lihat pada Gambar 1 di bawah).

Lingkungan pengembangan digunakan oleh ES builder untuk membangun komponen dan memasukkan pengetahuan ke dalam basis pengetahuan. Lingkungan konsultasi digunakan oleh nonpakar untuk memperoleh pengetahuan dan nasihat pakar. Lingkungan ini dapat dipisahkan setelah sistem lengkap.

Tiga komponen utama yang tampak secara virtual di setiap sistem pakar adalah:

1. Basis pengetahuan,
2. Mesin inferensi, dan
3. Antarmuka pengguna.

Sistem pakar yang berinteraksi dengan pengguna dapat pula berisi komponen tambahan berikut :

1. Subsistem akuisisi pengetahuan
2. Blackboard (tempat kerja)
3. Subsistem penjelas (justifier)
4. Subsistem perbaikan-pengetahuan

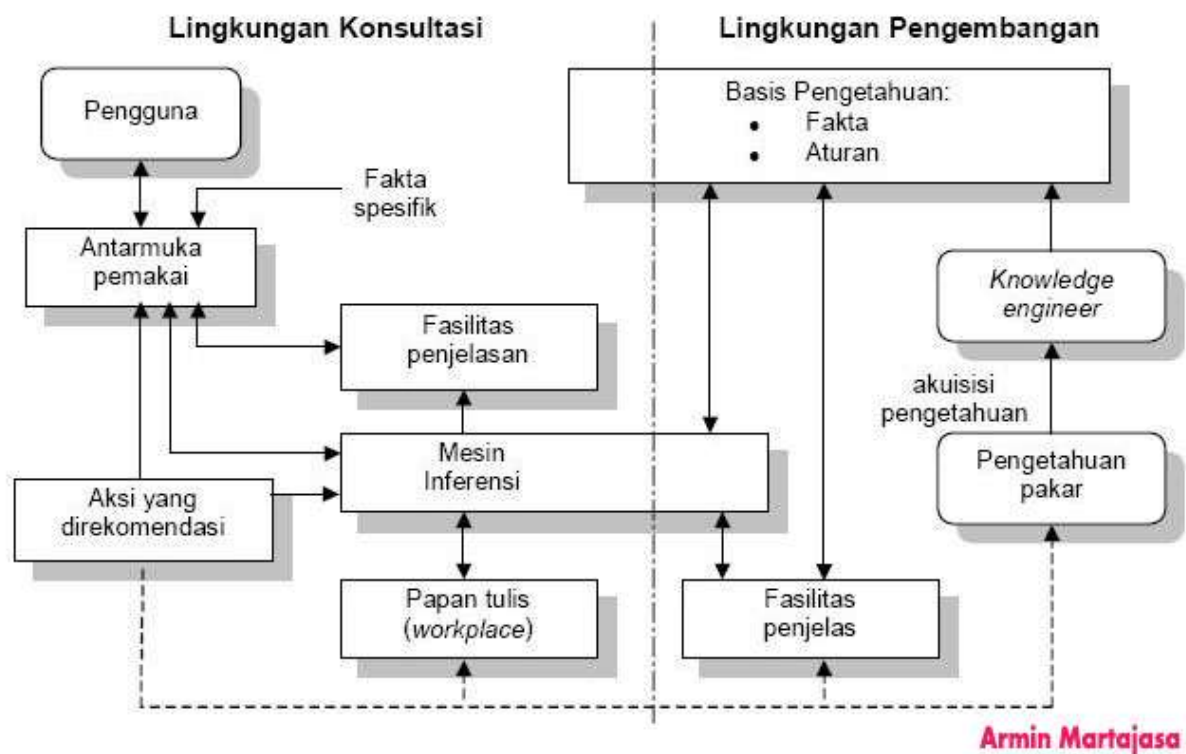
Kebanyakan sistem pakar saat ini tidak berisi komponen perbaikan pengetahuan.

Deskripsi singkat tiap komponen akan dijelaskan pada Gambar 1 di bawah.

1. Subsistem Akuisisi Pengetahuan

Akuisisi Pengetahuan adalah akumulasi, transfer, dan transformasi keahlian pemecah masalah dari pakar atau sumber pengetahuan terdokumentasi ke program komputer, untuk membangun atau memperluas basis pengetahuan. Sumber pengetahuan potensial antara lain pakar manusia, buku teks, dokumen multimedia, database (public dan privat), laporan riset khusus, dan informasi yang terdapat dalam web. Mendapatkan pengetahuan dari pakar adalah tugas

kompleks yang sering menimbulkan kemacetan dalam konstruksi ES. Dalam sistem pakar, seseorang memerlukan knowledge engineer atau pakar elitis pengetahuan untuk berinteraksi dengan satu atau lebih pakar manusia dalam membangun basis pengetahuan. Biasanya knowledge engineer membantu pakar menyusun area persoalan dengan menginterpretasikan dan mengintegrasikan jawaban manusia, menyusun analogi, mengajukan contoh perbandingan, dan menjelaskan kesulitan konseptual.



Gambar 2.1. Struktur Sistem Pakar Turban

## 2. Basis Pengetahuan

Basis pengetahuan berisi pengetahuan relevan yang diperlukan untuk memahami, merumuskan, dan memecahkan persoalan. Basis tersebut mencakup dua elemen dasar:

- a. Fakta, misalnya situasi persoalan dan teori area persoalan.

b. Heuristik atau aturan khusus yang mengarahkan penggunaan pengetahuan untuk memecahkan persoalan khusus dalam domain tertentu. (Selain itu, mesin inferensi dapat menyertakan pemecahan persoalan untuk tujuan umum dan aturan pengambilan keputusan). Heuristik menyatakan pengetahuan penilaian informal dalam area aplikasi. Pengetahuan, tidak hanya fakta, adalah bahan mentah primer dalam sistem pakar.

### 3. Mesin Inferensi

Otak ES adalah mesin inferensi, yang dikenal juga sebagai struktur kontrol atau penerjemah aturan (dalam ES berbasis-aturan). Komponen ini sebenarnya adalah program komputer yang menyediakan metodologi untuk mempertimbangkan informasi dalam basis pengetahuan dan blackboard, dan merumuskan kesimpulan. Komponen ini menyediakan arahan bagaimana menggunakan pengetahuan sistem, yakni dengan mengembangkan agenda yang mengatur dan mengontrol langkah yang diambil untuk memecahkan persoalan kapan pun konsultasi berlangsung.

### 4. Antarmuka Pengguna

Sistem pakar berisi prosesor bahasa untuk komunikasi berorientasi-persoalan yang mudah antara pengguna dan komputer. Komunikasi ini paling baik dilakukan dalam bahasa alami. Dikarenakan batasan teknologi, maka kebanyakan sistem yang ada menggunakan pendekatan pertanyaan dan jawaban untuk berinteraksi dengan pengguna.

## 5. Blackboard (Tempat Kerja)

Blackboard adalah area kerja memori yang disimpan sebagai database untuk deskripsi persoalan terbaru yang ditetapkan oleh data input; digunakan juga untuk perekam hipotesis dan keputusan sementara. Tiga tipe keputusan dapat direkam dalam blackboard: rencana (bagaimana mengatasi persoalan), agenda (tindakan potensial sebelum eksekusi), dan solusi (hipotesis kandidat dan arah tindakan alternative yang telah dihasilkan sistem sampai dengan saat ini).

## 6. Subsistem Penjelas (Justifier)

Kemampuan untuk melacak tanggung jawab suatu kesimpulan terhadap sumbernya adalah penting untuk transfer keahlian dan dalam pemecahan masalah. Subsistem penjelas (disebut juga justifier) dapat melacak tanggung jawab tersebut dan menjelaskan perilaku ES dengan menjawab pertanyaan berikut sebagai interaktif :

- a. Mengapa suatu pertanyaan ditanyakan oleh sistem pakar?
- b. Bagaimana suatu kesimpulan dicapai?
- c. Mengapa suatu alternative ditolak?
- d. Apa rencana untuk mencapai solusi? Misalnya, apa yang tetap tersisa sebelum diagnosis akhir ditetapkan?

Dalam ES sederhana, penjelasan menunjukkan aturan yang digunakan untuk memperoleh rekomendasi tertentu.

## 7. Sistem Perbaikan Pengetahuan

Pakar manusia memiliki sistem-perbaikan-pengetahuan, yakni mereka dapat menganalisis pengetahuannya sendiri dan kegunaannya, belajar darinya, dan meningkatkannya untuk konsultasi mendatang. Serupa pula, evaluasi tersebut diperlukan dalam pembelajaran komputer sehingga program dapat menganalisis alasan keberhasilan atau kegagalannya. Hal ini dapat mengarah kepada peningkatan sehingga menghasilkan basis pengetahuan yang lebih akurat dan pertimbangan yang lebih efektif. Komponen tersebut tidak tersedia dalam sistem pakar komersial pada saat ini, tetapi sedang dikembangkan dalam ES eksperimental pada beberapa universitas dan lembaga riset.

Contoh dialog sistem pakar dengan penggunanya adalah sebagai berikut:

Sistem : Halo, saya pakar buah, silahkan jawab pertanyaan saya, ya atau tidak.

Sistem : Apakah warnanya hijau?

User : Tidak

Sistem : Apakah warnanya ungu kemerahan?

User : Ya

Sistem : Apakah batangnya merambat?

User : Ya

Sistem : Apakah bentuknya bulat kecil?

User : Ya

Sistem : Buah yang anda maksud adalah ANGGUR

Agar tanya jawab ini bisa berlangsung dengan baik maka sistem pakar ini harus memiliki basis-pengetahuan sebagai berikut:

Apel :

- warna kuning atau merah
- batangnya adalah pohon
- bentuknya bulat sedang

Anggur :

- warna ungu kemerahan
- batangnya merambat
- bentuknya bulat kecil

Jeruk :

- warna oranye
- batangnya adalah pohon
- bentuknya bulat sedang

Ketika Sistem melakukan tanya jawab dengan User maka jawaban Ya akan dijadikan basis untuk memeriksa knowledge-base, misalnya pada contoh diatas data masukan adalah (ungu kemerahan, merambat, bulat kecil) dan data ini cocok dengan karakteristik dari Anggur, maka jawaban dari sistem adalah: ANGGUR.



## **BAB III**

### **Jenis-jenis sistem pakar**

Menurut Staugaard (1987) suatu sistem pakar disusun oleh tiga modul utama yaitu :

1. Modul Penerimaan Pengetahuan (Knowledge Acquisition Mode) Sistem berada pada modul ini, pada saat ia menerima pengetahuan dari pakar. Proses mengumpulkan pengetahuan-pengetahuan yang akan digunakan untuk pengembangan sistem, dilakukan dengan bantuan knowledge engineer. Peran knowledge engineer adalah sebagai penghubung antara suatu sistem pakar dengan pakarnya.

2. Modul Konsultasi (Consultation Mode)

Pada saat sistem berada pada posisi memberikan jawaban atas permasalahan yang diajukan oleh user, sistem pakar berada dalam modul konsultasi. Pada modul ini, user berinteraksi dengan sistem dengan menjawab pertanyaan-pertanyaan yang diajukan oleh sistem.

3. Modul Penjelasan (Explanation Mode)

Modul ini menjelaskan proses pengambilan keputusan oleh system (bagaimana suatu keputusan dapat diperoleh).

Ada 4 bentuk sistem pakar, yaitu :

1. Berdiri sendiri. Sistem pakar jenis ini merupakan software yang berdiri-sendiri tidak tergantung dengan software yang lainnya.
2. Tergabung. Sistem pakar jenis ini merupakan bagian program yang terkandung didalam suatu algoritma (konvensional), atau merupakan program dimana didalamnya memanggil algoritma subrutin lain (konvensional).
3. Menghubungkan ke software lain . Bentuk ini biasanya merupakan sistem pakar yang menghubungkan ke suatu paket program tertentu, misalnya DBMS.
4. Sistem Mengabdikan. Sistem pakar merupakan bagian dari komputer khusus yang dihubungkan dengan suatu fungsi tertentu. Misalnya sistem pakar yang digunakan untuk membantu menganalisis data radar.

#### Jenis-Jenis Sistem Pakar

1. Interpretasi: Menghasilkan deskripsi situasi berdasarkan data sensor.
2. Prediksi: Memperkirakan akibat yang mungkin dari situasi yang diberikan.
3. Diagnosis: Menyimpulkan kesalahan sistem berdasarkan gejala (symptoms).
4. Disain: Menyusun objek-objek berdasarkan kendala.
5. Planning: Merencanakan tindakan
6. Monitoring: Membandingkan hasil pengamatan dengan proses perencanaan.
7. Debugging: Menentukan penyelesaian dari kesalahan sistem.
8. Reparasi: Melaksanakan rencana perbaikan.
9. Instruction: Diagnosis, debugging, dan reparasi kelakuan pelajar.
10. Control: Diagnosis, debugging, dan reparasi kelakuan sistem

## BAB IV

### Masalah dan pengetahuan dalam sistem pakar

#### A. Sistem Perantaraan Maju (Forward Chaining Systems)

Pada sistem perantaraan maju, fakta-fakta dalam sistem disimpan dalam memori kerja dan secara kontinyu diperbarui. Aturan dalam sistem merepresentasikan aksiaksi yang harus diambil apabila terdapat suatu kondisi khusus pada item-item dalam memori kerja, sering disebut aturan kondisi-aksi. Kondisi biasanya berupa pola yang cocok dengan item yang ada di dalam memori kerja, sementara aksi biasanya berupa penambahan atau penghapusan item dalam memori kerja.

Aktivitas sistem dilakukan berdasarkan siklus mengenal-beraksi (recognise-act). Mula-mula, sistem mencari semua aturan yang kondisinya terdapat di memori kerja, kemudian memilih salah satunya dan menjalankan aksi yang bersesuaian dengan aturan tersebut. Pemilihan aturan yang akan dijalankan (fire) berdasarkan strategi tetap yang disebut strategi penyelesain konflik. Aksi tersebut menghasilkan memori kerja baru, dan siklus diulangi lagi sampai tidak ada aturan yang dapat dipicu (fire), atau goal (tujuan) yang dikehendaki sudah terpenuhi.

Sebagai contoh, lihat pada sekumpulan aturan sederhana berikut (Di sini kita memakai kata yang diawali huruf kapital untuk menyatakan suatu variabel. Pada sistem lain, mungkin dipakai cara lain, misalnya menggunakan awalan ? atau ^):

1. JIKA (mengajar X)

DAN (mengoreksi\_tugas X)

MAKA TAMBAH (terlalu\_banyak\_bekerja X)

2. JIKA (bulan maret)

MAKA TAMBAH (mengajar dosen)

3. JIKA (bulan maret)

MAKA TAMBAH (mengoreksi\_tugas dosen)

4. JIKA (terlalu\_banyak\_bekerja X)

ATAU (kurang\_tidur X)

MAKA TAMBAH (mood\_kurang\_baik X)

5. JIKA (mood\_kurang\_baik X)

MAKA HAPUS (bahagia X)

6. JIKA (mengajar X)

MAKA HAPUS (meneliti X)

Kita asumsikan, pada awalnya kita mempunyai memori kerja yang berisi fakta berikut:

(bulan maret)

(bahagia dosen)

(meneliti dosen)

Sistem Pakar mula-mula akan memeriksa semua aturan yang ada untuk mengenali aturan manakah yang dapat memicu aksi, dalam hal ini aturan 2 dan 3.

Sistem kemudian memilih salah satu di antara kedua aturan tersebut dengan strategi penyelesaian konflik. Katakanlah aturan 2 yang terpilih, maka fakta (mengajar dosen) akan ditambahkan ke dalam memori kerja. Keadaan memori kerja sekarang menjadi:

(mengajar dosen)

(bulan maret)

(bahagia dosen)

(meneliti dosen)

Sekarang siklus dimulai lagi, dan kali ini aturan 3 dan 6 yang kondisinya terpenuhi. Katakanlah aturan 3 yang terpilih dan terpicu, maka fakta (mengoreksi\_tugas dosen) akan ditambahkan ke dalam memori kerja. Lantas pada siklus ketiga, aturan 1 terpicu, sehingga variabel X akan berisi (bound to) dosen, dan fakta (terlalu\_banyak\_bekerja dosen) ditambahkan, sehingga isi memori kerja menjadi:

(terlalu\_banyak\_bekerja dosen)

(mengoreksi\_tugas dosen)

(mengajar dosen)

(bulan maret)

(bahagia dosen)

(meneliti dosen)

Aturan 4 dan 6 dapat diterapkan. Misalkan aturan 4 yang terpicu, sehingga fakta (mood\_kurang\_baik dosen) ditambahkan. Pada siklus berikutnya, aturan 5 terpilih dan dipicu, sehingga fakta (bahagia dosen) dihapus dari memori kerja. Kemudian aturan 6 akan terpicu dan fakta (meneliti dosen) dihapus pula dari memori kerja menjadi:

(mood\_kurang\_baik dosen)

(terlalu\_banyak\_bekerja dosen)

(mengoreksi\_tugas dosen)

(mengajar dosen)

(bulan maret)

Urutan aturan yang dipicu bisa jadi sangat vital, terutama di mana aturan-aturan yang ada dapat mengakibatkan terhapusnya item dari memori kerja. Tinjau kasus berikut: andaikan terdapat tambahan aturan pada kumpulan aturan di atas, yaitu:

7. JIKA (bahagia X)

MAKA TAMBAH (memberi\_nilai\_bagus X)

Jika aturan 7 ini terpicu sebelum (bahagia dosen) dihapus dari memori, maka Sistem Pakar akan berkesimpulan bahwa saya akan memberi nilai bagus( ) 1 . Namun jika aturan 5 terpicu dahulu, maka aturan 7 tidak akan dijalankan (artinya saya tidak akan memberi nilai bagus).

## B. Strategi penyelesaian konflik (conflict resolution strategy)

Strategi penyelesaian konflik dilakukan untuk memilih aturan yang akan diterapkan apabila terdapat lebih dari 1 aturan yang cocok dengan fakta yang terdapat dalam memori kerja. Di antaranya adalah:

1. No duplication. Jangan memicu sebuah aturan dua kali menggunakan fakta/data yang sama, agar tidak ada fakta yang ditambahkan ke memori kerja lebih dari sekali.
2. Recency. Pilih aturan yang menggunakan fakta yang paling baru dalam memori kerja. Hal ini akan membuat sistem dapat melakukan penalaran dengan mengikuti rantai tunggal ketimbang selalu menarik kesimpulan baru menggunakan fakta lama.
3. Specificity. Picu aturan dengan fakta prakondisi yang lebih spesifik (khusus) sebelum aturan yang menggunakan prakondisi lebih umum. Contohnya: jika kita mempunyai aturan “JIKA (burung X) MAKA TAMBAH (dapat\_terbang X)” dan “JIKA (burung X) DAN (penguin X) MAKA TAMBAH (dapat\_berenang X)” serta fakta bahwa tweety adalah seekor penguin, maka lebih baik memicu aturan kedua dan menarik kesimpulan bahwa tweety dapat berenang.
4. Operation priority. Pilih aturan dengan prioritas yang lebih tinggi. Misalnya ada fakta (bertemu kambing), (ternak kambing), (bertemu macan), dan (binatang\_buas macan), serta dua aturan: “JIKA (bertemu X) DAN (ternak X) MAKA TAMBAH (memberi\_makan X)” dan “JIKA (bertemu X) DAN (binatang\_buas X) MAKA TAMBAH (melarikan\_diri)”, maka kita akan memilih aturan kedua karena lebih tinggi prioritasnya.

### C. Sistem Perantaraan Balik (Backward Chaining Systems)

Sejauh ini kita telah melihat bagaimana sistem berbasis aturan dapat digunakan untuk menarik kesimpulan baru dari data yang ada, menambah kesimpulan ini ke dalam memori kerja. Pendekatan ini berguna ketika kita mengetahui semua fakta awalnya, namun tidak dapat menebak konklusi apa yang bisa diambil. Jika kita tahu kesimpulan apa yang seharusnya, atau mempunyai beberapa hipotesis yang spesifik, maka perantaraan maju di atas menjadi tidak efisien. Sebagai contoh, jika kita ingin mengetahui apakah saya dalam keadaan mempunyai mood yang baik sekarang, kemungkinan kita akan berulang kali memicu aturan-aturan dan memperbarui memori kerja untuk mengambil kesimpulan apa yang terjadi pada bulan Maret, atau apa yang terjadi jika saya mengajar, yang sebenarnya perlu terlalu kita ambil pusing. Dalam hal ini yang diperlukan adalah bagaimana dapat menarik kesimpulan yang relevan dengan tujuan atau goal.

Hal ini dapat dikerjakan dengan perantaraan balik dari pernyataan goal (atau hipotesis yang menarik bagi kita). Jika diberikan sebuah goal yang hendak dibuktikan, maka mula-mula sistem akan memeriksa apakah goal tersebut cocok dengan fakta-fakta awal yang dimiliki. Jika ya, maka goal terbukti atau terpenuhi. Jika tidak, maka sistem akan mencari aturan-aturan yang konklusinya (aksinya) cocok dengan goal. Salah satu aturan tersebut akan dipilih, dan sistem kemudian akan mencoba membuktikan fakta-fakta prakondisi aturan tersebut menggunakan prosedur yang sama, yaitu dengan menset prakondisi tersebut sebagai goal baru yang harus dibuktikan.

Perhatikan bahwa pada perantaraan balik, sistem tidak perlu memperbarui memori kerja, namun perlu untuk mencatat goal-goal apa saja yang dibuktikan untuk



membuktikan goal utama (hipotesis). Secara prinsip, kita dapat menggunakan aturan-aturan yang sama untuk perantaraan maju dan balik. Namun, dalam prakteknya, harus sedikit dimodifikasi. Pada perantaraan balik, bagian MAKA dalam aturan biasanya tidak diekspresikan sebagai suatu aksi untuk dijalankan (misalnya TAMBAH atau HAPUS), tetapi suatu keadaan yang bernilai benar jika premisnya (bagian JIKA) bernilai benar. Jadi aturan-aturan di atas diubah menjadi:

1. JIKA (mengajar X)  
     DAN (mengoreksi\_tugas X)  
     MAKA (terlalu\_banyak\_bekerja X)
2. JIKA (bulan maret)  
     MAKA (mengajar dosen)
3. JIKA (bulan maret)  
     MAKA (mengoreksi\_tugas dosen)
4. JIKA (terlalu\_banyak\_bekerja X)  
     ATAU (kurang\_tidur X)  
     MAKA (mood\_kurang\_baik X)
5. JIKA (mood\_kurang\_baik X)  
     MAKA TIDAK BENAR (bahagia X)

**dengan fakta awal:**

(bulan maret)

(meneliti dosen)

Misalkan kita hendak membuktikan apakah mood saya sedang kurang baik. Mulamula kita periksa apakah goal cocok dengan fakta awal. Ternyata tidak ada fakta awal yang menyatakan demikian, sehingga langkah kedua yaitu mencari aturan mana yang mempunyai konklusi (mood\_kurang\_baik dosen). Dalam hal ini aturan yang cocok adalah aturan 4 dengan variabel X diisi dengan (bound to) dosen. Dengan demikian kita harus membuktikan bahwa prakondisi aturan ini, (terlalu\_banyak\_bekerja dosen) atau (kurang\_tidur dosen), salah satunya adalah benar (karena memakai ATAU). Lalu diperiksa aturan mana yang dapat membuktikan bahwa adalah (terlalu\_banyak\_bekerja dosen) benar, ternyata aturan 1, sehingga prakondisinya, (mengajar X) dan (mengoreksi\_tugas X), duaduanya adalah benar (karena memakai DAN). Ternyata menurut aturan 2 dan 3, keduanya bernilai bernilai benar jika (bulan maret) adalah benar.

Karena ini sesuai dengan fakta awal, maka keduanya bernilai benar. Karena semua goal sudah terpenuhi maka goal utama (hipotesis) bahwa mood saya sedang kurang baik adalah benar (terpenuhi).

Untuk mencatat goal-goal yang harus dipenuhi/dibuktikan, dapat digunakan stack (tumpukan). Setiap kali ada aturan yang konklusinya cocok dengan goal yang sedang dibuktikan, maka fakta-fakta prakondisi dari aturan tersebut ditaruh (push) ke dalam stack sebagai goal baru. Dan setiap kali goal pada tumpukan teratas terpenuhi atau dapat dibuktikan, maka goal tersebut diambil (pop) dari tumpukan. Demikian seterusnya sampai tidak ada goal lagi di dalam stack, atau dengan kata lain goal utama (yang terdapat pada tumpukan terbawah) sudah terpenuhi.

#### D. Pemilihan Sistem Inferensi

Secara umum kita dapat memakai panduan berikut untuk menentukan apakah kita hendak memilih perantaraan maju atau balik untuk Sistem Pakar yang kita bangun. Panduan tersebut tercantum dalam Tabel berikut ini.

Tabel 3.1. Panduan untuk memilih sistem inferensi

Perantaraan Maju	Perantaraan Balik
<ul style="list-style-type: none"> <li>• Ada banyak hal yang hendak dibuktikan</li> <li>• Hanya sedikit fakta awal yang dipunyai</li> <li>• Ada banyak aturan berbeda yang dapat memberikan kesimpulan yang sama</li> </ul>	<ul style="list-style-type: none"> <li>• Hanya akan membuktikan fakta (hipotesis) tunggal</li> <li>• Terdapat banyak fakta awal</li> <li>• Jika terdapat banyak aturan yang memenuhi syarat untuk dipicu (fire) pada suatu siklus</li> </ul>

#### E. Ketidakpastian dalam Aturan

Sejauh ini kita menggunakan nilai kebenaran tegas dalam fakta dan aturan yang dipakai, misalnya: jika terlalu banyak bekerja maka pasti mood kurang baik. Pada kenyataannya, seringkali kita tidak bisa membuat aturan yang absolut untuk mengambil kesimpulan secara pasti, misalnya: jika terlalu banyak bekerja maka kemungkinan besar mood kurang baik. Untuk itu, seringkali aturan yang dipakai

memiliki nilai kepastian (certainty value). Contohnya: jika terlalu banyak bekerja maka pasti mood kurang baik (kepastian 0,75)

## BAB V

### Komponen – komponen sistem pakar

Diantara komponen-komponen dalam Gambar 1 di atas, basis pengetahuan dan mesin inferensi adalah modul paling kritis agar sistem pakar dapat berfungsi dengan baik. Pengetahuan harus direpresentasikan dan diatur secara tepat dalam basis pengetahuan. Mesin inferensi kemudian dapat menggunakan pengetahuan tersebut untuk menarik kesimpulan baru dari fakta dan aturan yang ada. Dalam bagian ini, struktur berbasis pengetahuan dan mesin inferensi pada sistem berbasis-aturan.

#### 1. Representasi dan Organisasi Pengetahuan

Pengetahuan pakar harus direpresentasikan dalam format yang dapat dipahami komputer dan diatur dengan tepat dalam basis pengetahuan sistem pakar. Terdapat beberapa cara yang berbeda untuk merepresentasikan pengetahuan manusia, antara lain aturan produksi, jaringan semantik, dan pernyataan logika.

Dalam sistem berbasis aturan, pengetahuan dalam basis pengetahuan direpresentasikan dalam aturan JIKA MAKA yang menggabungkan kondisi dan kesimpulan untuk menangani situasi tertentu.

Bagian JIKA mengindikasikan kondisi aturan tersebut diaktifkan, dan bagian MAKA menunjukkan tindakan atau kesimpulan jika semua kondisi JIKA dipenuhi.

Keuntungan menggunakan aturan produksi adalah aturan tersebut mudah dipahami dan aturan baru dapat ditambahkan dengan mudah ke dalam basis pengetahuan tanpa memengaruhi aturan yang telah ada. Ketidakpastian yang

dihubungkan dengan tiap aturan dapat ditambahkan untuk meningkatkan keakuratannya.

Tugas utama pengembangan sistem pakar adalah memperoleh pengetahuan dari manusia dan mengubahnya menjadi aturan produksi yang dapat ditangani mesin inferensi. Mesin inferensi memilih aturan yang dapat diterapkan dari basis pengetahuan, mengintegrasikannya, dan mempertimbangkannya untuk mendapatkan kesimpulan.

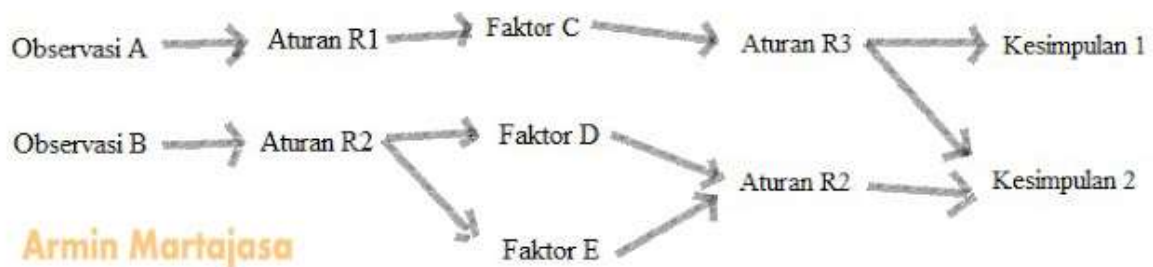
## 2. Mesin Inferensi

Dalam keputusan kompleks, pengetahuan pakar sering tidak dapat direpresentasikan dalam aturan tunggal. Sebaliknya, aturan dapat digabungkan secara dinamis untuk mencakup berbagai kondisi. Proses penggabungan banyak aturan berdasarkan data yang tersedia, disebut inferensi. Komponen yang melakukan inferensi dalam sistem pakar disebut mesin inferensi. Ada dua pendekatan populer untuk menarik kesimpulan adalah sebagai berikut:

### a. Forward Chaining

Forward chaining adalah mencari bagian JIKA terlebih dahulu. Setelah semua kondisi JIKA dipenuhi, aturan dipilih untuk mendapatkan kesimpulan. Jika kesimpulan diambil dari keadaan pertama, bukan dari yang terakhir, maka ia akan digunakan sebagai fakta untuk disesuaikan dengan kondisi JIKA aturan

yang lain untuk mendapatkan kesimpulan yang lebih baik. Proses ini berlanjut hingga dicapai kesimpulan terbaik. Yang digambarkan pada Gambar 2.2.

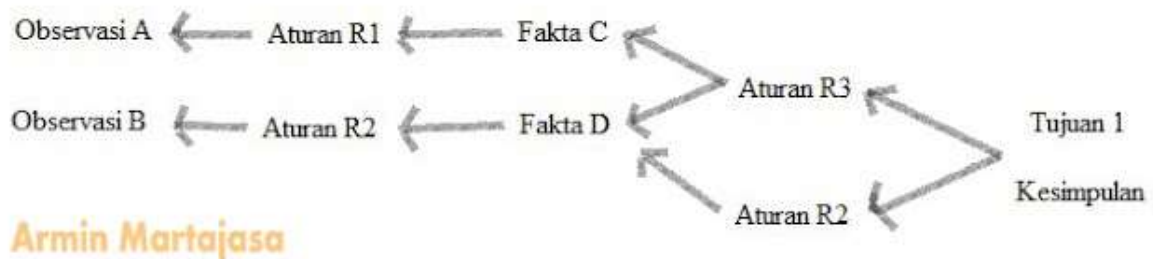


Gambar 2.2. Forward Chaining

#### b. Backward Chaining

Backward Chaining adalah kebalikan dari Forward Chaining. Pendekatan ini mulai dari kesimpulan dan hipotesis bahwa kesimpulan adalah benar. Mesin inferensi kemudian mengidentifikasi kondisi JIKA yang diperlukan untuk membuat kesimpulan benar dan mencari fakta untuk menguji apakah kondisi JIKA adalah benar. Jika semua kondisi JIKA adalah benar, maka aturan dipilih dari kesimpulan yang dicapai. Jika beberapa kondisi salah, maka aturan dibuang dan aturan berikutnya digunakan sebagai hipotesis kedua. Jika tidak ada fakta yang membuktikan bahwa semua kondisi JIKA adalah benar atau salah, maka mesin inferensi terus mencari aturan yang kesimpulannya sesuai dengan kondisi JIKA yang tidak diputuskan untuk bergerak satu langkah ke depan memeriksa kondisi tersebut. Serupa pula, proses chaining ini berlanjut hingga suatu set aturan didapat untuk mencapai kesimpulan atau untuk

membuktikan tidak dapat mencapai kesimpulan. Yang digambarkan pada Gambar 2.3.



Armin Martajasa

Gambar 2.3. Backward Chaining

Kedua metode inferensi tersebut dipengaruhi oleh tiga macam penelusuran, yaitu:

- Depth-first search, melakukan penelusuran kaidah secara mendalam dari simpul akar bergerak menurun ke tingkat dalam yang berurutan.
- Breadth-first search, bergerak dari simpul akar, simpul yang ada pada setiap tingkat diuji sebelum pindah ke tingkat selanjutnya.
- Best-first search, bekerja berdasarkan kombinasi kedua metode sebelumnya.



## BAB VI

### Mesin inferensi

#### A. Strategi penyelesaian konflik (conflict resolution strategy)

Strategi penyelesaian konflik dilakukan untuk memilih aturan yang akan diterapkan apabila terdapat lebih dari 1 aturan yang cocok dengan fakta yang terdapat dalam memori kerja. Di antaranya adalah :

1. No duplication. Jangan memicu sebuah aturan dua kali menggunakan fakta/data yang sama, agar tidak ada fakta yang ditambahkan ke memori kerja lebih dari sekali.
2. Recency. Pilih aturan yang menggunakan fakta yang paling baru dalam memori kerja. Hal ini akan membuat sistem dapat melakukan penalaran dengan mengikuti rantai tunggal ketimbang selalu menarik kesimpulan baru menggunakan fakta lama.
3. Specificity. Picu aturan dengan fakta prakondisi yang lebih spesifik (khusus) sebelum aturan yang menggunakan prakondisi lebih umum. Contohnya: jika kita mempunyai aturan “JIKA (burung X) MAKA TAMBAH (dapat\_terbang X)” dan “JIKA (burung X) DAN (penguin X) MAKA TAMBAH (dapat\_berenang X)” serta fakta bahwa tweety adalah seekor penguin, maka lebih baik memicu aturan kedua dan menarik kesimpulan bahwa tweety dapat berenang.
4. Operation priority. Pilih aturan dengan prioritas yang lebih tinggi. Misalnya ada fakta (bertemu kambing), (ternak kambing), (bertemu macan), dan (binatang\_buas macan), serta dua aturan: “JIKA (bertemu X) DAN (ternak X) MAKA TAMBAH (memberi\_makan X)” dan “JIKA (bertemu X) DAN

(binatang\_buas X) MAKA TAMBAH (melarikan\_diri)", maka kita akan memilih aturan kedua karena lebih tinggi prioritasnya.

## **B. Sistem Perantaraan Maju (Forward Chaining Systems)**

Pada sistem perantaraan maju, fakta-fakta dalam dalam sistem disimpan dalam memori kerja dan secara kontinyu diperbarui. Aturan dalam sistem merepresentasikan aksi-aksi yang harus diambil apabila terdapat suatu kondisi khusus pada item-item dalam memori kerja, sering disebut aturan kondisi-aksi. Kondisi biasanya berupa pola yang cocok dengan item yang ada di dalam memori kerja, sementara aksi biasanya berupa penambahan atau penghapusan item dalam memori kerja.

Aktivitas sistem dilakukan berdasarkan siklus mengenal-beraksi (recognise-act). Mula-mula, sistem mencari semua aturan yang kondisinya terdapat di memori kerja, kemudian memilih salah satunya dan menjalankan aksi yang bersesuaian dengan aturan tersebut. Pemilihan aturan yang akan dijalankan (fire) berdasarkan strategi tetap yang disebut strategi penyelesain konflik. Aksi tersebut menghasilkan memori kerja baru, dan siklus diulangi lagi sampai tidak ada aturan yang dapat dipicu (fire), atau goal (tujuan) yang dikehendaki sudah terpenuhi.

Sebagai contoh, lihat pada sekumpulan aturan sederhana berikut (Di sini kita memakai kata yang diawali huruf kapital untuk menyatakan suatu variabel. Pada sistem lain, mungkin dipakai cara lain, misalnya menggunakan awalan ? atau ^):

1. JIKA (mengajar X) DAN (mengoreksi\_tugas X) MAKA TAMBAH (terlalu\_banyak\_bekerja X)
2. JIKA (bulan maret) MAKA TAMBAH (mengajar kuncoro)
3. JIKA (bulan maret) MAKA TAMBAH (mengoreksi\_tugas kuncoro)
4. JIKA (terlalu\_banyak\_bekerja X) ATAU (kurang\_tidur X) MAKA TAMBAH (mood\_kurang\_baik X)
5. JIKA (mood\_kurang\_baik X) MAKA HAPUS (bahagia X)
6. JIKA (mengajar X) MAKA HAPUS (meneliti X)

Kita asumsikan, pada awalnya kita mempunyai memori kerja yang berisi fakta berikut:

(bulan maret)

(bahagia kuncoro)

(meneliti kuncoro)

Sistem Pakar mula-mula akan memeriksa semua aturan yang ada untuk mengenali aturan manakah yang dapat memicu aksi, dalam hal ini aturan 2 dan 3. Sistem kemudian memilih salah satu di antara kedua aturan tersebut dengan strategi penyelesaian konflik. Katakanlah aturan 2 yang terpilih, maka fakta (mengajar kuncoro) akan ditambahkan ke dalam memori kerja. Keadaan memori kerja sekarang menjadi:

(mengajar kuncoro)

(bulan maret)

(bahagia kuncoro)

(meneliti kuncoro)

Sekarang siklus dimulai lagi, dan kali ini aturan 3 dan 6 yang kondisinya terpenuhi. Katakanlah aturan 3 yang terpilih dan terpicu, maka fakta (mengoreksi\_tugas kuncoro) akan ditambahkan ke dalam memori kerja. Lantas pada siklus ketiga, aturan 1 terpicu, sehingga variabel X akan berisi (bound to) kuncoro, dan fakta (terlalu\_banyak\_bekerja kuncoro) ditambahkan, sehingga isi memori kerja menjadi:

(terlalu\_banyak\_bekerja kuncoro)

(mengoreksi\_tugas kuncoro)

(mengajar kuncoro)

(bulan maret)

(bahagia kuncoro)

(meneliti kuncoro)

Aturan 4 dan 6 dapat diterapkan. Misalkan aturan 4 yang terpicu, sehingga fakta (mood\_kurang\_baik kuncoro) ditambahkan. Pada siklus berikutnya, aturan 5 terpilih dan dipicu, sehingga fakta (bahagia kuncoro) dihapus dari memori kerja. Kemudian aturan 6 akan terpicu dan fakta (meneliti kuncoro) dihapus pula dari memori kerja menjadi:

(mood\_kurang\_baik kuncoro)

(terlalu\_banyak\_bekerja kuncoro)

(mengoreksi\_tugas kuncoro)

(mengajar kuncoro)

(bulan maret)

Urutan aturan yang dipicu bisa jadi sangat vital, terutama di mana aturan-aturan yang ada dapat mengakibatkan terhapusnya item dari memori kerja. Tinjau kasus berikut: andaikan terdapat tambahan aturan pada kumpulan aturan di atas, yaitu:

7. JIKA (bahagia X) MAKA TAMBAH (memberi\_nilai\_bagus X)

Jika aturan 7 ini terpicu sebelum (bahagia kuncoro) dihapus dari memori, maka Sistem Pakar akan berkesimpulan bahwa saya akan memberi nilai bagus. Namun jika aturan 5 terpicu dahulu, maka aturan 7 tidak akan dijalankan (artinya saya tidak akan memberi nilai bagus).

### **C. Sistem Perantaraan Balik (Backward Chaining Systems)**

Sejauh ini kita telah melihat bagaimana sistem berbasis aturan dapat digunakan untuk menarik kesimpulan baru dari data yang ada, menambah kesimpulan ini ke dalam memori kerja. Pendekatan ini berguna ketika kita mengetahui semua fakta awalnya, namun tidak dapat menebak konklusi apa yang bisa diambil. Jika kita tahu kesimpulan apa yang seharusnya, atau mempunyai beberapa hipotesis yang spesifik, maka perantaraan maju di atas menjadi tidak efisien. Sebagai contoh, jika kita ingin mengetahui apakah saya dalam keadaan mempunyai mood yang baik sekarang, kemungkinan kita akan berulang kali memicu aturan-aturan dan memperbarui memori kerja untuk mengambil kesimpulan apa yang terjadi pada bulan Maret, atau apa yang terjadi jika saya mengajar, yang sebenarnya tidak perlu terlalu kita ambil pusing. Dalam hal ini yang diperlukan adalah bagaimana dapat menarik kesimpulan yang relevan dengan tujuan atau goal.

Hal ini dapat dikerjakan dengan perantaraan balik dari pernyataan goal (atau hipotesis yang menarik bagi kita). Jika diberikan sebuah goal yang hendak dibuktikan, maka mula-mula sistem akan memeriksa apakah goal tersebut cocok dengan fakta-fakta awal yang dimiliki. Jika ya, maka goal terbukti atau terpenuhi. Jika tidak, maka sistem akan mencari aturan-aturan yang konklusinya (aksinya) cocok dengan goal. Salah satu aturan tersebut akan dipilih, dan sistem kemudian akan mencoba membuktikan fakta-fakta prakondisi aturan tersebut menggunakan prosedur yang sama, yaitu dengan menset prakondisi tersebut sebagai goal baru yang harus dibuktikan.

Perhatikan bahwa pada perantaraan balik, sistem tidak perlu memperbarui memori kerja, namun perlu untuk mencatat goal-goal apa saja yang dibuktikan untuk membuktikan goal utama (hipotesis).

Secara prinsip, kita dapat menggunakan aturan-aturan yang sama untuk perantaraan maju dan balik. Namun, dalam prakteknya, harus sedikit dimodifikasi. Pada perantaraan balik, bagian MAKA dalam aturan biasanya tidak diekspresikan sebagai suatu aksi untuk dijalankan (misalnya TAMBAH atau HAPUS), tetapi suatu keadaan yang bernilai benar jika premisnya (bagian JIKA) bernilai benar. Jadi aturan-aturan di atas diubah menjadi:

1. JIKA (mengajar X) DAN (mengoreksi\_tugas X) MAKA (terlalu\_banyak\_bekerja X)
2. JIKA (bulan maret) MAKA (mengajar kuncoro)
3. JIKA (bulan maret) MAKA (mengoreksi\_tugas kuncoro)
4. JIKA (terlalu\_banyak\_bekerja X) ATAU (kurang\_tidur X) MAKA (mood\_kurang\_baik X)

5. JIKA (mood\_kurang\_baik X) MAKA TIDAK BENAR (bahagia X)

dengan fakta awal:

(bulan maret)

(meneliti kuncoro)

Misalkan kita hendak membuktikan apakah mood sedang kurang baik. Mula-mula kita periksa apakah goal cocok dengan fakta awal. Ternyata tidak ada fakta awal yang menyatakan demikian, sehingga langkah kedua yaitu mencari aturan mana yang mempunyai konklusi (mood\_kurang\_baik kuncoro). Dalam hal ini aturan yang cocok adalah aturan 4 dengan variabel X diisi dengan (bound to) kuncoro. Dengan demikian kita harus membuktikan bahwa prakondisi aturan ini, (terlalu\_banyak\_bekerja kuncoro) atau (kurang\_tidur kuncoro), salah satunya adalah benar (karena memakai ATAU). Lalu diperiksa aturan mana yang dapat membuktikan bahwa adalah (terlalu\_banyak\_bekerja kuncoro) benar, ternyata aturan 1, sehingga prakondisinya, (mengajar X) dan (mengoreksi\_tugas X), keduanya adalah benar (karena memakai DAN). Ternyata menurut aturan 2 dan 3, keduanya bernilai benar jika (bulan maret) adalah benar. Karena ini sesuai dengan fakta awal, maka keduanya bernilai benar. Karena semua goal sudah terpenuhi maka goal utama (hipotesis) bahwa mood saya sedang kurang baik adalah benar (terpenuhi).

Untuk mencatat goal-goal yang harus dipenuhi/dibuktikan, dapat digunakan stack (tumpukan). Setiap kali ada aturan yang konklusinya cocok dengan goal yang sedang dibuktikan, maka fakta-fakta prakondisi dari aturan tersebut ditaruh (push) ke dalam stack sebagai goal baru. Dan setiap kali goal pada tumpukan teratas terpenuhi atau dapat dibuktikan, maka goal tersebut diambil (pop) dari tumpukan.

Demikian seterusnya sampai tidak ada goal lagi di dalam stack, atau dengan kata lain goal utama (yang terdapat pada tumpukan terbawah) sudah terpenuhi.

#### **D. Ketidakpastian dalam Aturan**

Sejauh ini kita menggunakan nilai kebenaran tegas dalam fakta dan aturan yang dipakai, misalnya: jika terlalu banyak bekerja maka pasti mood kurang baik. Pada kenyataannya, seringkali kita tidak bisa membuat aturan yang absolut untuk mengambil kesimpulan secara pasti, misalnya: jika terlalu banyak bekerja maka kemungkinan besar mood kurang baik. Untuk itu, seringkali aturan yang dipakai memiliki nilai kepastian (certainty value). Contohnya: jika terlalu banyak bekerja maka pasti mood kurang baik (kepastian 0,75). Isi



## BAB VII

### Metode penanganan ketidakpastian dengan sistem pakar

#### A. Ketidakpastian

Jika sistem kecerdasan buatan yang dikembangkan memiliki pengetahuan yang lengkap tentang permasalahan yang akan ditanganinya, maka sistem tersebut dapat dengan mudah memberikan solusi dengan menggunakan pendekatan logika. Akan tetapi, system hampir tidak pernah dapat mengakses seluruh fakta yang ada dalam lingkungan permasalahan yang akan ditanganinya, sehingga sistem harus bekerja dalam ketidakpastian dan kesamaran.

Untuk itu, sistem harus menggunakan teknik-teknik khusus yang dapat menangani ketidakpastian dan kesamaran dalam menyelesaikan permasalahan yang ditanganinya.

Untuk mengawali pembahasan ini, diberikan sebuah kasus mengenai diagnosa media.

Potongan pengetahuan berikut yang direpresentasikan dalam aturan produksi menggambarkan gejala-gejala dari suatu penyakit.

Rule 1:

IF Has\_fever (Patient) AND

Has\_rash (Patient) AND

Has\_high\_body\_ache(Patient)

THEN Bears-Typhoid (Patient)

Berdasarkan aturan diatas, terlihat bahwa jika seorang pasien mengalami ketiga gejala yang disebutkan dalam aturan, maka sistem akan mendiagnosa pasien tersebut menderita tifus (typhoid). Aturan ini akan diterapkan pada seluruh pasien yang mengalami ketiga gejala tersebut. Jika sistem ini diterapkan dalam kasus nyata, apakah gejala-gejala yang disebutkan

telah mewakili seluruh gejala dari penyakit tifus? Bagaimana jika ada penyakit lain yang memiliki gejala yang sama dengan ketiga gejala tersebut? Bagaimana jika derajat gejala yang

dialami seorang pasien dengan pasien yang lainnya berbeda? Pertanyaan-pertanyaan ini tidak

dapat ditangani oleh aturan tersebut karena jawaban dari pasien sebagai pengguna tidak hanya “ya” atau “tidak”, sehingga muncul ketidakpastian dan kesamaran pengetahuan dalam

permasalahan ini.

Ada tiga teknik yang dapat digunakan untuk menangani ketidakpastian dan kesamaran

pengetahuan, yaitu:

1. Teknik Probabilitas, yang dikembangkan dengan memanfaatkan teorema Bayes yang menyajikan hubungan sebab akibat yang terjadi diantara evidence-evidence yang ada. Pendekatan alternatif lainnya yang dapat digunakan adalah teori Dempster-Shafer.

2. Faktor Kepastian, merupakan teknik penalaran tertua, yang digunakan pada system MYCIN. Teknik ini bersifat semi probabilitas, karena tidak sepenuhnya menggunakan notasi probabilitas.
3. Logika Fuzzy, merupakan teknik baru yang diperkenalkan oleh Zadeh. Setiap variable dalam teknik ini memiliki rentang nilai tertentu, yang akan digunakan untuk menghitung nilai fungsi keanggotaannya.

## B. Teknik Probabilitas

### 1. Teorema Bayes

Thomas Bayes menemukan pendekatan penalaran statistik yang jauh lebih maju dibandingkan dengan pola pikir matematis tradisional pada saat itu. Fokus matematika pada saat itu adalah pada tingkah laku sampel dari populasi yang diketahui. Akan tetapi, Bayes mengemukakan ide untuk menentukan properti dari populasi berdasarkan sampel tersebut.

Dalam “An Essay Towards the Solving a Problem in the Doctrines of Chance”, dia menyajikan tentang “Proposition 9”, yang akhirnya dikenal dengan “Teorema Bayes”. Selanjutnya, teorema ini menjadi dasar dalam pengambilan keputusan modern.

Teorema Bayes diperoleh dari aturan produksi konjungsi dengan notasi

$$P(Y | X) = \frac{P(X | Y) \cdot P(Y)}{P(X)}$$

sebagai berikut:

Atau dapat juga ditulis sebagai berikut:

$$P(Y | X, e) = \frac{P(X | Y, e) \cdot P(Y | e)}{P(X | e)}$$

## 2. Teori Dempster-Shaffer

Secara umum teori Dempster-Shafer dapat ditulis dalam suatu interval: [Belief, Plausability] Belief (Bel) adalah ukuran kekuatan evidence dalam mendukung suatu himpunan proposisi.

Jika bernilai 0 maka mengindikasikan bahwa tidak ada evidence, dan jika bernilai 1 maka

menunjukkan adanya kepastian. Plausability (Pl) dinotasikan sebagai:

$$Pl(s) = 1 - Bel(\neg s)$$

Plausability juga bernilai antara 0 dan 1. Jika kita yakin terhadap  $\neg s$ , maka dapat dikatakan

bahwa  $Bel(\neg s) = 1$  dan  $Pl(s) = 0$ . Pada teori ini dikenal juga adanya frame of discernment yang dinotasikan dengan  $q$ , merupakan semesta pembicaraan dari sekumpulan hipotesis. Tidak semua evidence secara langsung mendukung tiap-tiap elemen. Untuk itu diperlukan adanya probabilitas fungsi densitas ( $m$ ). Nilai  $m$  tidak hanya mendefinisikan elemen-elemen saja, namun juga semua subset-nya, sehingga jika  $q$  berisi  $n$  elemen, maka  $m$  mendefinisikan  $2^n$  elemen. Jumlah semua  $m$  dalam subset  $q$  sama dengan 1.

$$M3(Z) = \frac{\sum_{X \cap Y = Z} m1(X) * m2(Y)}{1 - \sum_{X \cap Y = \emptyset} m1(X) * m2(Y)}$$

Andaikan tidak ada informasi apapun untuk memilih keempat hipotesis

tersebut, maka  $m\{q\} = 1,0$ . Andaikan diketahui  $X$  adalah subset dari  $q$ , dengan  $m_1$  sebagai fungsi densitasnya, dan  $Y$  juga merupakan subset dari  $q$  dengan  $m_2$  sebagai fungsi densitasnya, maka dapat dibentuk fungsi kombinasi  $m_1$  dan  $m_2$  sebagai  $m_3$ , yaitu:

### C. Faktor Kepastian

Faktor kepastian (Certainty Factor-CF) diperkenalkan oleh Shortliffe Buchanan dalam pembuatan MYCIN. Faktor kepastian merupakan nilai parameter klinis yang diberikan MYCIN

untuk menunjukkan besarnya kepercayaan.

Ada dua macam faktor kepastian yang dapat digunakan, yaitu faktor kepastian yang diisikan oleh pakar bersama aturan dan faktor kepastian yang diberikan oleh pengguna. Faktor kepastian yang diisikan oleh pakar menggambarkan kepercayaan pakar terhadap hubungan antara antecedent dan consequent pada aturan kaidah produksi. Faktor kepastian dari pengguna menunjukkan besarnya kepercayaan terhadap keberadaan masing-masing elemen

dalam antecedent.

Faktor kepastian didefinisikan sebagai berikut.

$$CF(H,E) = MB(H,E) - MD(H,E)$$

dimana:

$CF(H,E)$  faktor kepastian dari hipotesis  $H$  yang dipengaruhi oleh premis (evidence)  $E$ .

$MB(H,E)$  ukuran kepercayaan terhadap hipotesis H yang dipengaruhi oleh premis (evidence) E

$MD(H,E)$  ukuran ketidakpercayaan terhadap hipotesis H yang dipengaruhi oleh premis (evidence) E.

Pengguna akan memberikan faktor kepastian terhadap setiap premis yang ada dalam aturan. Untuk menentukan faktor kepastian dari suatu aturan yang didalamnya terdapat beberapa premis dengan faktor kepastiannya masing-masing, maka perlu dilakukan perhitungan CF paralel.

CF paralel merupakan CF yang diperoleh dari beberapa premis pada sebuah aturan.

Besarnya CF paralel dipengaruhi oleh CF pengguna untuk masing-masing premis dan operator dari premis. Rumus untuk masing-masing operator diberikan sebagai berikut.

$$CF(x \text{ and } y) = \text{Min}(CF(x), CF(y))$$

$$CF(y \text{ or } x) = \text{Max}(CF(x), CF(y))$$

$$CF(\text{not } x) = - CF(x)$$

Suatu aturan akan memiliki nilai faktor kepastian dari seorang pakar, sementara aturan tersebut juga memiliki faktor kepastian yang diperoleh dari premis-premis yang ada didalamnya (CF paralel), sehingga perlu dilakukan perhitungan nilai faktor kepastian untuk suatu aturan berdasarkan CF paralel dan CF yang diberikan oleh pengguna, yang disebut dengan CF sekuensial.

CF sekuensial diperoleh dari hasil perhitungan CF paralel dari semua premis dalam satu aturan dengan CF aturan yang diberikan oleh pakar. Rumus untuk menghitung CF sekuensial adalah sebagai berikut.

$$CF(x,y) = CF(x) * CF(y)$$

dimana:

CF(x,y) CF sekuensial

CF(x) CF paralel

CF(y) CF pakar

Suatu hipotesis pada kenyataannya bisa dihasilkan dari beberapa aturan yang berbeda, dimana setiap aturan memiliki faktor kepastian masing-masing, sehingga perlu dilakukan perhitungan CF gabungan dari seluruh aturan yang ada untuk suatu hipotesis.

CF gabungan merupakan CF akhir dari sebuah calon kesimpulan. CF ini dipengaruhi oleh semua CF sekuensial dari aturan yang menghasilkan kesimpulan tersebut. Rumus untuk melakukan perhitungan CF gabungan adalah sebagai berikut.

Jika  $CF(x) > 0$  dan  $CF(y) > 0$ , maka:

$$CF(x,y) = CF(x) + CF(y) - (CF(x)*CF(y))$$

Jika salah satu,  $CF(x)$  atau  $CF(y) < 0$ , maka:

$$CF(x,y) = \frac{CF(x) + CF(y)}{(1 - (\text{Min}(|CF(x)|, |CF(y)|)))}$$

Jika  $CF(x) < 0$  dan  $CF(y) < 0$ , maka:

$$CF(x,y) = CF(x) + (CF(y) * (1 + CF(x)))$$

#### D. Logika Fuzzy

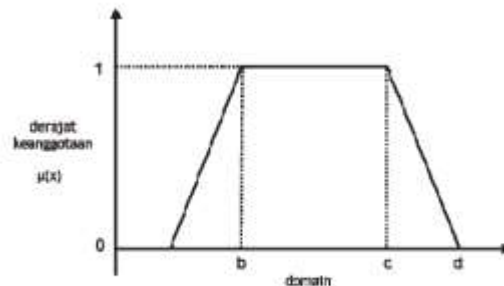
Logika fuzzy pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh pada tahun 1965. Dasar logika fuzzy adalah teori himpunan fuzzy. Pada teori himpunan fuzzy, peranan derajat keanggotaan/nilai keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangatlah penting.

Logika fuzzy digunakan sebagai suatu cara untuk memetakan permasalahan dari input menuju ke output yang diharapkan. Pada himpunan fuzzy, nilai keanggotaan terletak pada rentang 0 sampai 1. Himpunan fuzzy merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel fuzzy. Variabel fuzzy merupakan variabel yang hendak dibahas dalam suatu sistem fuzzy, misalnya umur, temperature, dan lain-lain.

Fungsi keanggotaan (membership function) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya yang memiliki interval antara 0 sampai 1. Ada beberapa fungsi keanggotaan yang dapat digunakan dalam logika fuzzy, tetapi fungsi yang paling sering digunakan dalam pembangunan system pakar adalah representasi kurva trapesium.



Kurva trapezium memiliki bentuk dasar seperti kurva segitiga, tetapi memiliki beberapa titik yang mempunyai nilai keanggotaan 1 seperti yang terlihat pada gambar 1.



Gambar 1 Representasi Kurva Trapezium

Fungsi keanggotaannya adalah:

$$\mu(x) = \begin{cases} 0; & x \leq a \text{ atau } x \geq d \\ (x-a)/(b-a); & a \leq x \leq b \\ 1; & b \leq x \leq c \\ (d-x)/(d-c) & x \geq d \end{cases}$$

Suatu himpunan fuzzy dapat dikombinasikan dengan himpunan fuzzy lainnya dengan menggunakan operator. Ada tiga operator dasar yang diciptakan oleh Zadeh, yaitu:

#### 1. Operator AND

Hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan

$$\mu_{AB} = \min(\mu_A(x), \mu_B(y))$$

## 2. Operator OR

Hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan

terbesar antar elemen pada himpunan-himpunan yang bersangkutan

$$\mu_{AB} = \max(\mu_A(x), \mu_B(y))$$

## 3. Operator NOT

Hasil operasi dengan operator NOT diperoleh dengan mengurangi nilai keanggotaan

elemen pada himpunan yang bersangkutan dengan 1

$$\mu_{A'} = 1 - \mu_A(x)$$

Tiap-tiap aturan pada basis pengetahuan fuzzy akan berhubungan dengan suatu relasi

fuzzy. Bentuk umum dari aturan yang digunakan dalam fungsi implikasi adalah:

$$\text{IF } x \text{ is } A \text{ THEN } y \text{ is } B$$

dengan  $x$  dan  $y$  adalah skalar, sedangkan  $A$  dan  $B$  adalah himpunan fuzzy.

Secara umum, ada dua fungsi implikasi yang dapat digunakan, yaitu:

- a. Min (minimum), akan memotong output himpunan fuzzy
- b. Dot (product), akan menskala output himpunan fuzzy

Metode penalaran fuzzy ada tiga, yaitu metode Tsukamoto, metode Mamdani dan metode Sugeno. Pada metode Tsukamoto, setiap konsekuen pada aturan yang berbentuk IF-THEN harus direpresentasikan dengan suatu himpunan fuzzy dengan fungsi keanggotaan monoton.

Sebagai hasilnya, output hasil inferensi dari tiap-tiap aturan diberikan secara tegas berdasarkan  $\alpha$ -predikat. Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot.

Metode Mamdani sering dikenal dengan metode Max-Min, diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Untuk mendapatkan output diperlukan empat tahapan, yaitu pembentukan himpunan fuzzy, aplikasi fungsi implikasi, komposisi dan defuzzifikasi.

Metode Sugeno memiliki penalaran yang hampir sama dengan penalaran Mamdani, tetapi output yang dihasilkan tidak berupa himpunan fuzzy, melainkan berupa konstanta atau persamaan linear. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985, terdiri dari dua jenis, yaitu metode Fuzzy Sugeno Orde-Nol dan metode Fuzzy Sugeno Orde-Satu.

## BAB VIII

### Pengembangan sistem pakar

#### A. Aplikasi Sederhana: Sistem Pakar Bengkel Mobil

Ini adalah contoh Sistem Pakar sederhana, yang bertujuan untuk mencari apa yang salah sehingga mesin mobil pelanggan yang tidak mau hidup, dengan memberikan gejala-gejala yang teramati. Anggap Sistem Pakar kita memiliki aturan-aturan berikut:

1. JIKA mesin\_mendapatkan\_bensin  
DAN starter\_dapat\_dihidupkan  
MAKA ada\_masalah\_dengan\_pengapian
2. JIKA TIDAK BENAR starter\_dapat\_dihidupkan  
DAN TIDAK BENAR lampu\_menyala  
MAKA ada\_masalah\_dengan\_aki
3. JIKA TIDAK BENAR starter\_dapat\_dihidupkan  
DAN lampu\_menyala  
MAKA ada\_masalah\_dengan\_starter
4. JIKA ada\_bensin\_dalam\_tangki\_bahan\_bakar  
MAKA mesin\_mendapatkan\_bensin

Terdapat 3 masalah yang mungkin, yaitu: ada\_masalah\_dengan\_pengapian, ada\_masalah\_dengan\_aki dan ada\_masalah\_dengan\_starter. Dengan sistem

terarah-tujuan (goal-driven), kita hendak membuktikan keberadaan setiap masalah tadi.

Pertama, Sistem Pakar berusaha untuk membuktikan kebenaran `ada_masalah_dengan_pengapian`. Di sini, aturan 1 dapat digunakan, sehingga Sistem Pakar akan menset goal baru untuk membuktikan apakah `mesin_mendapatkan_bensin` serta `starter_dapat_dihidupkan`. Untuk membuktikannya, aturan 4 dapat digunakan, dengan goal baru untuk membuktikan `mesin_mendapatkan_bensin`. Karena tidak ada aturan lain yang dapat digunakan menyimpulkannya, sedangkan sistem belum memperoleh solusinya, maka Sistem Pakar kemudian bertanya kepada pelanggan: "Apakah ada bensin dalam tangki bahan bakar?". Sekarang, katakanlah jawaban klien adalah "Ya", jawaban ini kemudian dicatat, sehingga klien tidak akan ditanyai lagi dengan pertanyaan yang sama. Nah, karena sistem sekarang sudah dapat membuktikan bahwa mesin mendapatkan bensin, maka sistem sekarang berusaha mengetahui apakah `starter_dapat_dihidupkan`. Karena sistem belum tahu mengenai hal ini, sementara tidak ada aturan lagi yang dapat menyimpulkannya, maka Sistem Pakar bertanya lagi ke klien: "Apakah starter dapat dihidupkan?". Misalkan jawabannya adalah "Tidak", maka tidak ada lagi aturan yang dapat membuktikan `ada_masalah_dengan_pengapian`, sehingga Sistem Pakar berkesimpulan bahwa hal ini bukanlah solusi dari problem yang ada, dan kemudian melihat hipotesis berikutnya: `ada_masalah_dengan_aki`. Sudah diketahui (dibuktikan) bahwa mesin tidak dapat distarter, sehingga yang harus dibuktikan adalah bahwa lampu tidak menyala. Sistem Pakar kemudian bertanya: "Apakah lampu menyala?". Misalkan jawabannya adalah "Tidak", maka sudah terbukti bahwa ada masalah dengan aki.

Sistem ini mungkin berhenti sampai di sini, tetapi biasanya ada kemungkinan terdapat lebih dari satu solusi (misalnya terdapat lebih dari satu kerusakan), atau ada kemungkinan terdapat solusi lain yang lebih tepat, sehingga biasanya semua hipotesis diperiksa kebenarannya. Sistem Pakar ini kemudian mencoba membuktikan bahwa ada masalah dengan starter, namun dari fakta yang sudah diperoleh, yaitu lampu tidak menyala, maka pembuktiannya menjadi gagal. Dengan demikian solusi yang diberikan oleh Sistem Pakar adalah ada masalah dengan aki.

Secara lengkap, interaksi antara Sistem Pakar dengan klien mungkin seperti ini:

System Pakar : Apakah ada bensin dalam tangki bahan bakar?

Klien : Ya

System Pakar : Apakah starter dapat dihidupkan?

Klien : Tidak

System Pakar : Apakah lampu menyala?

Klien : Tidak

System Pakar : Saya berkesimpulan bahwa ada masalah dengan aki

### **Fasilitas Penjelasan**

Kebanyakan Sistem Pakar mempunyai fasilitas penjelasan yang mengizinkan klien untuk bertanya mengapa dia diberi suatu pertanyaan atau bagaimana dapat mencapai suatu kesimpulan. Pertanyaan dari klien tersebut dijawab dengan mengacu pada goal, aturan yang dipakai, dan data-data yang ada. Contoh

pemakaian fasilitas penjelas dalam Sistem Pakar di atas dapat dilihat pada percakapan berikut:

System Pakar : Apakah ada bensin dalam tangki bahan bakar?

Klien : Ya

System Pakar : Apakah starter dapat dihidupkan?

Klien : Tidak

System Pakar : Apakah lampu menyala?

Klien : Mengapa?

System Pakar : Saya mencoba membuktikan bahwa ada masalah dengan aki.

Sudah ditunjukkan adalah tidak benar starter dapat dihidupkan.

Sehingga jika tidak benar bahwa lampu menyala, maka ada masalah dengan aki.

Apakah lampu menyala?

Klien : Tidak

System Pakar : Saya berkesimpulan bahwa ada masalah dengan aki

Klien : Bagaimana bisa?

System Pakar : Hal ini sesuai dengan aturan no 2:

JIKA                      TIDAK                      BENAR

starter\_dapat\_dihidupkan

DAN TIDAK BENAR lampu\_menyala

MAKA ada\_masalah\_dengan\_aki

bahwa tidak benar starter\_dapat\_dihidupkan  
sudah dijawab,

bahwa tidak benar lampu\_menyala sudah  
dijawab,

sehingga dapat disimpulkan bahwa  
ada\_masalah\_dengan\_aki.

Fasilitas penjelasan berhubungan dengan pencatatan aturan apa saja yang digunakan untuk mengambil kesimpulan, dan menggunakan catatan tersebut untuk menyusun penjelasan. Pembangkitan penjelasan sederhana seperti contoh di atas bukanlah persoalan yang sulit, namun pemahaman klien terhadap penjelasan tersebut tidaklah sama, sehingga seringkali klien tetap tidak mengerti maksudnya. Hal ini karena biasanya penjelasan hanya meliputi pengetahuan yang dangkal atau hanya permukaan saja sebagaimana tercantum dalam aturan yang dipunyai, bukannya pengetahuan yang mendalam yang mendasari aturan tersebut. Dalam contoh di atas, Sistem Pakar hanya mengatakan bahwa hal ini sesuai dengan aturan no 2, tetapi tidak menjelaskan secara lebih detil mengenai aturan 2 tersebut.

Mungkin klien perlu diberi pengertian bahwa lampu dan starter dua-duanya menggunakan aki untuk menyalakannya. Hal penting lain adalah ketika klien gagal mengerti atau menerima penjelasan yang diberikan, biasanya Sistem Pakar tidak



dapat memberi penjelasan dengan cara lain (tidak begitu halnya dengan seorang pakar yang mampu menggunakan kalimat yang lain untuk mencoba menjelaskannya).

## **B. Eliza**

Eliza adalah salah satu Sistem Pakar yang paling awal dikembangkan. Ini adalah program komputer terapis yang dibuat oleh Joseph Weizenbaum di MIT. Pengguna berkomunikasi dengannya sebagaimana sedang berkonsultasi dengan seorang terapis.

Berikut ini adalah contoh percakapan antara Eliza dengan seorang klien:

Eliza: Hello. I am ELIZA. How can I help you?

Klien: My name is Balza.

Eliza: I have told you before, I do not care about names.

Klien: I don't have any money. That's my problem.

Eliza: Why do you say your problem?

Klien: Because you asked me.

Eliza: Do any other reasons not come to mind?

Klien: No.

Eliza: You are being a bit negative.

Klien: Why?

Eliza: Please go on.

Klien: OK. I am bankrupt.

Eliza: How long have you been bankrupt?

Klien: A year.

Eliza: Please go on.

Klien: So, what should I do?

Eliza: Does that question interest you?

Klien: Yes, how can I make big money?

Eliza: You are sure?

Klien: Of course I'm sure.

...

### **C. Parry**

Parry adalah Sistem Pakar yang juga paling awal dikembangkan di Stanford University oleh seorang psikiater, Kenneth Colby, yang mensimulasikan seorang paranoid.

Berikut ini contoh respon Parry terhadap pertanyaan seorang psikiater yang menginterviewnya.

Psikiater : Why are you in the hospital?

Parry : I shouldn't be here.

Psikiater : Who brought you here?

Parry : The police.

Psikiater : What trouble did you have with the police?

Parry : Cops don't do their job.

Psikiater : What do they do instead?

Parry : Cops arrest the wrong people.

Psikiater : They arrest the right ones sometimes

Parry : The law should get those Italian crooks.

Psikiater : Are you Italian?

Parry : My background is British but I was born in this country.

Psikiater : Do you know any Italian crooks?

Parry : I try to avoid the underworld

## BAB IX

### Contoh sistem pakar

#### A. Implementasi sistem pakar dalam bidang kesehatan

Bidang kedokteran sangat erat hubungannya dengan kesehatan. Penerapan sistem pakar pada bidang ini akan sangat membantu dalam kelangsungan hidup seseorang. Beberapa alat kedokteran saat ini sudah memanfaatkan sistem pakar. Salah satunya dapat dijadikan sebagai penentu keputusan dan ada juga yang bekerja untuk menyembuhkan suatu penyakit mulai yang sederhana hingga yang kronis.

Contoh alat kedokteran yang menerapkan sistem pakar di dalamnya antara lain USG (ultrasonografi). Alat ini bekerja berdasarkan pantulan gelombang suara ultrasonik. Banyak digunakan untuk mendeteksi janin dalam kandungan. Alat ini bekerja dengan menerima input berupa suara yang kemudian diolah menjadi sebuah informasi berupa visual. Alat terapi kanker yang menghasilkan keputusan berupa bentuk terapi yang otomatis dilakukan oleh alat ini. Sangat membantu memang bila tidak terjadi kesalahan. Tetapi karena kesalahan dalam pengambilan keputusan maka menimbulkan korban jiwa. Hal ini yang tidak diinginkan dari penerapan sistem pakar pada dunia kesehatan.

Maka alat-alat yang dilengkapi sistem pakar pada bidang ini hanya bersifat membantu menghasilkan keputusan bukan secara otomatis melakukan tindakan. Bagaimana pun keputusan final tetap berada pada tangan ahlinya. Dan sistem pakar terancang adalah manusia. Sistem pakar yang diterapkan semata-mata hanya sebagai pendukung keputusan. Bila mana dimungkinkan untuk kerja

otomatis, itu juga hanya mengerjakan input yang merupakan keputusan dari ahli di bidangnya (dokter/spesialis).

Adapun pengembangan sebuah sistem pakar dapat dilakukan dengan dua cara yaitu:

1. Membangun sendiri semua komponen, sedangkan
2. Memakai semua komponen yang sudah ada (dalam rangka pengembangan), kecuali isi basis pengetahuan. Penggunaan cara kedua disebut sebagai membangun sistem pakar dengan shell.

Dalam contoh sebuah jurnal, telah menyimpulkan setelah mempelajari, membahas dan menganalisis pengetahuan sistem pakar untuk diagnosa penyakit gigi dan mulut pada manusia, maka diambil beberapa kesimpulan sebagai berikut :

1. Sistem pakar diagnosa penyakit gigi dan mulut sangat diperlukan dan dapat dijadikan sebagai asisten yang cukup cerdas untuk membantu pekerjaan staf/pegawai di Puskesmas serta dapat meningkatkan pemahaman pasien dalam mengetahui dengan benar gejala penyakit gigi dan mulut.
2. Dengan menggunakan forward chaining sebagai metode inferensinya, aplikasi sistem pakar ini memudahkan user dalam melakukan proses konsultasi, dimana hasil diagnosis berupa nama penyakit serta solusi pengobatannya yang sesuai dengan data gejala yang di inputkan oleh user.
3. Hasil pengujian dengan menggunakan metode Black box bahwa pada aplikasi sistem pakar bisa berjalan dengan baik dan bisa menghasilkan output yang diharapkan serta bisa dipakai pada tahap konsultasi.

## B. Contoh Skripsi Aplikasi Sistem Pakar

Berikut merupakan beberapa contoh skripsi untuk aplikasi maupun analisis dari penerapan sistem pakar, yang mungkin bisa membantu teman-teman yang sedang akan mengambil tugas akhir, penelitian perusahaan/pribadi, atau sekedar untuk menambah pengetahuan:

1. Sistem Pakar Berbasis Web Identifikasi Penyakit Ayam (Metode Forward Chaining)
2. Aplikasi Sistem Pakar Untuk Membantu Deteksi Dini Penyakit Immunologi (Studi Kasus Lupus Erythematosus) (Metode Inferensi Forward Chaining)
3. Sistem Pakar Diagnosa Awal Penyakit Kulit Sapi Berbasis Web Dengan Menggunakan Metode Certainty Factor
4. Pembangunan Sistem Pakar Pada Perangkat Mobile Dengan WML Dan PHP Untuk Menemukan Penyebab Kerusakan Mesin Isuzu Panther (Metode Inferensi Forward Dan Backward Chaining)
5. Sistem Pakar Identifikasi Bentuk Keris Jawa Dengan Metode CF (Certainty Factor)
6. Sistem Pakar Diagnosa Penyakit Paru Pada Anak Berbasis WEB (Metode Inferensi Forward Chaining Dan Backward Chaining)
7. Sistem Pakar Identifikasi Hama Dan Penyakit Tanaman Apel Berbasis WEB (Metode Forward Chaining)
8. Sistem Pakar Untuk Mengidentifikasi Gizi Buruk Pada Anak Berbasis Web (Metode Inferensi Forward Chaining)
9. Sistem Pakar Konsultasi Siswa Bermasalah (Metode Depth First Search)
10. Sistem Pakar Untuk Mendiagnosa Penyakit Kulit Wajah Berbasis Mobile

11. Rancang Bangun Sistem Pakar Menentukan Penyakit Tanaman Kakao Menggunakan Metode Forward Chaining Berbasis WAP
12. Sistem Pakar Diagnosa Penyakit Infeksi Virus Pada Anak Menggunakan Metode Forward Chaining Dan Certainty Factor
13. Sistem Pakar Diagnosis Kerusakan Pada Televisi Berwarna (Metode Inferensi Forward Chaining Dan Backward Chaining)
14. Aplikasi Sistem Pakar Diagnosis Pada Sistem Computer (Metode Inferensi Forward Chaining Dan Backward Chaining)
15. Sistem pakar untuk mendiagnosis penyakit alopesia pada manusia (Metode Inferensi Forward Chaining)
16. Aplikasi Sistem Pakar Untuk Simulasi Diagnosa Hama Dan Penyakit Tanaman Bawang Merah Dan Cabai Menggunakan Forward Chaining Dan Pendekatan Berbasis Aturan
17. Sistem pakar pendiagnosa penyakit berbasis WEB
18. Aplikasi sistem pakar berbasis web untuk diagnose penyakit gigi danmulut WEB (Metode Forward Chaining)
19. Perancangan Sistem Pakar Untuk Diagnosa Penyakit Anak (Metode Inferensi Forward Chaining)
20. Alat Bantu Ajar Penerapan Metodeforward Chaining Dan Backward Chaining Pada Sistem Pakar

## DAFTAR PUSTAKA

- (1) Giarratano, J. and Riley, G., 1994, "Expert Systems Principles and Programming", Publishing Company, Boston
- (2) Giarratano, J. and Riley, G., 2005, "Expert Systems: Principles and Programming", 4th edition, Thomson Course Technology, Boston
- (3) Jogiyanto, 2005, Pengembangan Sistem Pakar Menggunakan Visual Basic, Andi, Yogyakarta
- (4) Kusumadewi, S., 2003, Artificial Intelligence: Teknik dan Aplikasinya, Graha Ilmu, Yogyakarta
- (5) Kusrini, 2008, Aplikasi Sistem Pakar, Andi, Yogyakarta
- (6) Turban, E., 1995, Decision Support and Expert Systems, Management Support System, Prentice Hall International Inc., New York