

## Pertemuan 3

# Sistem Pakar (Expert System)

### A. Struktur Sistem Pakar

#### 1. Pengantar Sistem Pakar

Sistem Pakar (*Expert System*) adalah sistem berbasis komputer yang meniru cara berpikir dan pengambilan keputusan seorang pakar manusia dalam suatu bidang tertentu. Tujuan utamanya adalah membantu non-pakar untuk memecahkan masalah kompleks menggunakan pengetahuan dan aturan yang dikumpulkan dari pakar.

Contoh

Sistem pakar medis untuk diagnosa penyakit, sistem hukum untuk analisis peraturan, atau sistem pertanian untuk rekomendasi pupuk dan irigasi.

#### 2. Konsep Dasar Sistem Pakar

Sistem pakar bekerja berdasarkan dua komponen utama:

1. Pengetahuan (Knowledge) — informasi dan pengalaman dari pakar.
2. Penalaran (Reasoning) — mekanisme berpikir logis untuk menghasilkan kesimpulan.

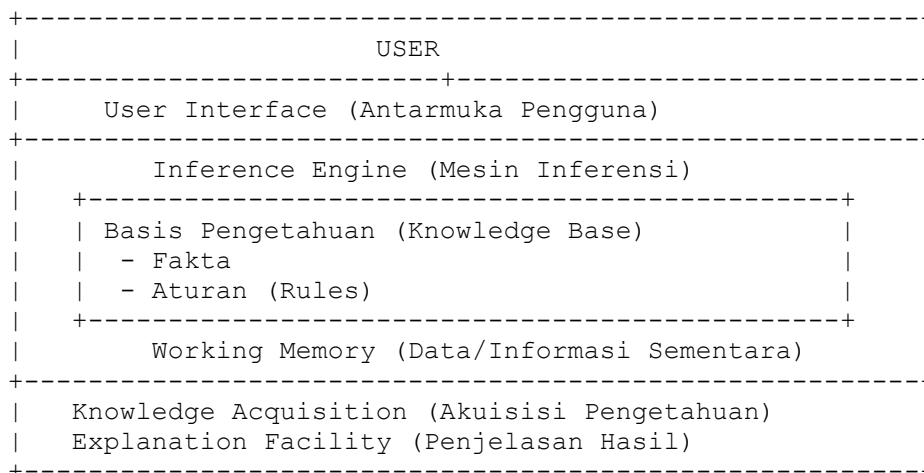
Prinsip utama:

“Jika sistem memiliki pengetahuan dari seorang pakar dan dapat menalar seperti pakar, maka sistem dapat berperilaku seperti pakar.”

#### 3. Struktur Umum Sistem Pakar

Secara umum, struktur sistem pakar terdiri atas tiga komponen utama dan dua komponen pendukung.





## 4. Komponen Utama Sistem Pakar

### A. Basis Pengetahuan (Knowledge Base)

Merupakan inti dari sistem pakar, berisi kumpulan:

- Fakta (Facts): kondisi, data, atau objek yang diketahui.
- Aturan (Rules): pengetahuan pakar dalam bentuk logika *IF-THEN*.

Fungsi:

- Menyimpan pengetahuan dari pakar ke dalam format yang bisa diproses komputer.
- Menjadi dasar pengambilan keputusan oleh mesin inferensi.

Contoh

Rule 1: IF gejala = demam AND batuk THEN penyakit = ISPA

Rule 2: IF penyakit = ISPA AND sesak THEN tindakan = Konsultasi Dokter

Jenis Pengetahuan yang Disimpan

Jenis	Deskripsi
Deklaratif	Menyatakan fakta, data, dan relasi (contoh: suhu pasien = 39°C).
Prosedural	Menyatakan cara atau prosedur (contoh: jika suhu > 38°C, beri obat penurun panas).



## B. Mesin Inferensi (*Inference Engine*)

Komponen inti yang berfungsi sebagai “otak” sistem pakar — melakukan proses penalaran otomatis (*reasoning*) berdasarkan aturan dan fakta yang ada dalam basis pengetahuan.

Fungsi Mesin Inferensi:

1. Mencocokkan fakta dengan aturan yang relevan.
2. Menarik kesimpulan baru atau memberikan rekomendasi.
3. Mengatur urutan penerapan aturan jika ada beberapa aturan yang sesuai.

Proses Kerja Mesin Inferensi:

1. Membaca fakta yang ada (*working memory*).
2. Memeriksa aturan mana yang cocok.
3. Menjalankan aturan tersebut (aplikasi IF-THEN).
4. Menambahkan hasil ke basis fakta.
5. Mengulangi hingga tidak ada aturan lagi yang dapat diterapkan.

Dua Metode Penalaran Umum:

Metode	Arah Penalaran	Deskripsi
Forward Chaining	Dari fakta ke kesimpulan	Cocok untuk diagnosa ( <i>modus ponens</i> ).
Backward Chaining	Dari kesimpulan ke fakta	Cocok untuk pembuktian hipotesis ( <i>modus tollens</i> ).

### Contoh Forward Chaining

Rule: IF suhu > 38°C THEN kondisi = demam

Fakta: suhu pasien = 39°C

→ Kesimpulan: kondisi = demam

### Contoh Backward Chaining:

Tujuan: Apakah pasien ISPA?

Cari aturan: IF batuk AND demam THEN ISPA

Periksa fakta: batuk = True, demam = True → ISPA = True



### C. Antarmuka Pengguna (*User Interface*)

Komponen yang berfungsi sebagai penghubung antara pengguna dan sistem pakar. Digunakan untuk:

- Memasukkan data atau pertanyaan.
- Menampilkan hasil inferensi dan penjelasan.

Fungsi Antarmuka Pengguna:

1. Input Data: menerima fakta/gejala dari pengguna.
2. Output Informasi: menampilkan hasil analisis, rekomendasi, atau diagnosa.
3. Interaksi Dialog: memungkinkan komunikasi dua arah antara pengguna dan sistem.
4. Visualisasi Penjelasan: menjelaskan dasar keputusan sistem (mengapa sistem memberikan hasil tertentu).

Contoh Antarmuka:

Tampilan	Deskripsi
Form Input	Pengguna memasukkan gejala atau data.
Tombol "Diagnosa"	Memicu mesin inferensi untuk memproses.
Output	Menampilkan hasil: "Pasien kemungkinan ISPA."
Penjelasan	"Karena gejala batuk dan demam ditemukan."

Jenis Antarmuka:

- Text-based Interface: input menggunakan teks (CLI).
- Menu-based Interface: pilihan berbentuk menu interaktif.
- Graphical Interface (GUI): tampilan visual dengan form, tombol, dan grafik.
- Web-based Interface: sistem pakar yang diakses melalui browser (PHP, Python, HTML).

### 5. Komponen Pendukung Sistem Pakar

Komponen Pendukung	Fungsi
Knowledge Acquisition Module	Proses akuisisi pengetahuan dari pakar manusia ke sistem.
Explanation Facility	Menjelaskan alasan di balik kesimpulan sistem.
Working Memory / BlackBoard	Tempat penyimpanan fakta sementara selama proses inferensi.

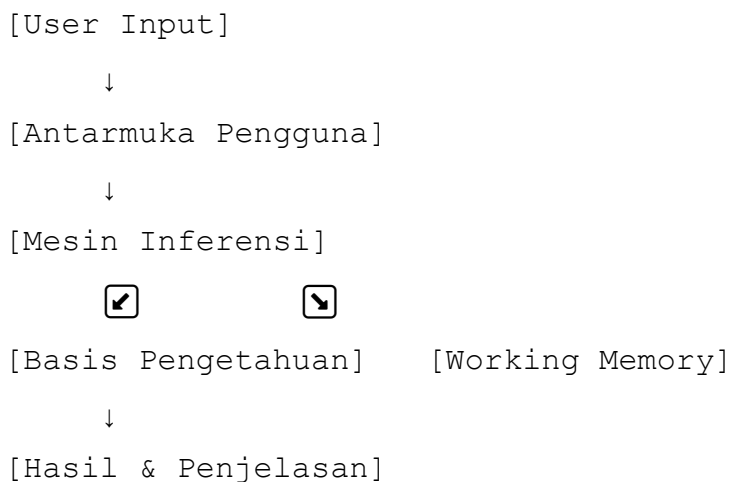


## 6. Alur Kerja Sistem Pakar

Tahapan Proses:

1. Input Data / Gejala → dimasukkan oleh pengguna.
2. Mesin Inferensi membaca fakta dan mencocokkannya dengan aturan.
3. Penerapan Aturan → menghasilkan kesimpulan baru.
4. Hasil dan Penjelasan ditampilkan kepada pengguna.

### Diagram Alur:



## 7. Contoh Kasus Mini: Sistem Pakar Diagnosa Penyakit

Aturan (*Knowledge Base*)

Rule 1: IF batuk AND demam THEN penyakit = ISPA

Rule 2: IF ISPA AND pilek THEN tindakan = "Minum obat flu"

Rule 3: IF ISPA AND sesak THEN tindakan = "Konsultasi dokter"

Proses Kerja

1. Pengguna memasukkan gejala: batuk, demam, pilek.
2. Mesin inferensi mencocokkan Rule 1 → menghasilkan ISPA.
3. Rule 2 terpenuhi → tindakan = "Minum obat flu."
4. Sistem menampilkan hasil ke pengguna.



## Tampilan Output

Diagnosis: Pasien kemungkinan ISPA

Saran: Minum obat flu dan istirahat

Penjelasan: Karena gejala batuk, demam, dan pilek ditemukan.

## B. Jenis Penalaran

### 1. Pengantar Penalaran dalam Sistem Pakar

Salah satu kemampuan utama sistem pakar adalah melakukan penalaran otomatis (*automated reasoning*) untuk menarik kesimpulan dari aturan (*rules*) dan fakta (*facts*). Proses ini dilakukan oleh mesin inferensi (*inference engine*) dengan dua pendekatan utama:

1. Forward Chaining → *Penalaran maju (data-driven reasoning)*
2. Backward Chaining → *Penalaran mundur (goal-driven reasoning)*

### 2. Konsep Dasar Penalaran (*Reasoning*)

Komponen	Fungsi
Fakta (Facts)	Data awal yang diketahui atau dimasukkan oleh pengguna.
Aturan (Rules)	Pengetahuan pakar dalam bentuk IF-THEN.
Kesimpulan (Conclusion)	Hasil akhir dari proses penalaran.

Proses reasoning = mencocokkan fakta dengan aturan untuk menemukan kesimpulan logis.

### 3. Forward Chaining (Penalaran Maju)

Forward chaining adalah proses penalaran yang dimulai dari fakta awal menuju kesimpulan akhir. Sistem membaca semua fakta yang diketahui, lalu mencari aturan yang premisnya terpenuhi, dan menambahkan kesimpulan baru ke basis fakta.

- ◆ Disebut juga data-driven reasoning, karena dimulai dari data yang ada.



Langkah-langkah Forward Chaining:

1. Masukkan fakta awal ke sistem (*working memory*).
2. Mesin inferensi mencari aturan yang kondisi (IF) terpenuhi.
3. Jalankan aturan → tambahkan hasil (THEN) sebagai fakta baru.
4. Ulangi proses sampai tidak ada aturan lain yang bisa digunakan.

### 💡 Contoh Kasus: Diagnosa Penyakit

Basis Pengetahuan:

R1: IF batuk AND demam THEN penyakit = ISPA

R2: IF penyakit = ISPA AND pilek THEN tindakan = "Minum obat flu"

R3: IF penyakit = ISPA AND sesak THEN tindakan = "Konsultasi ke dokter"

Fakta Awal:

batuk = true

demam = true

pilek = true

Langkah Penalaran Maju:

1. R1 cocok → tambahkan fakta baru: penyakit = ISPA.
2. R2 cocok (karena ISPA dan pilek) → tambahkan fakta baru: tindakan = Minum obat flu.
3. Tidak ada aturan lain yang cocok.

### ➔ Kesimpulan:

Pasien kemungkinan ISPA → saran: minum obat flu.

Diagram Forward Chaining:

```
[Batuk] + [Demam]
  ↓
 [Rule 1] → ISPA
  ↓
 [Rule 2] + [Pilek]
  ↓
Tindakan: Minum obat flu
```



### Kelebihan Forward Chaining

- Cocok untuk diagnosis dan rekomendasi berbasis data awal.
- Sistem dapat menemukan semua kemungkinan kesimpulan dari fakta yang diketahui.
- Sangat intuitif bagi pengguna non-teknis.

### Kelemahan

- Membutuhkan banyak aturan jika domain luas.
- Tidak efisien bila fakta awal sangat sedikit.
- Kadang menghasilkan kesimpulan yang terlalu umum.

## 4. Backward Chaining (Penalaran Mundur)

Backward chaining adalah proses penalaran yang dimulai dari suatu tujuan (*goal*) kemudian mencari fakta yang mendukung kesimpulan tersebut. Disebut juga goal-driven reasoning, karena sistem memulai dari hipotesis atau dugaan, lalu membuktikannya. Langkah-langkah Backward Chaining:

1. Tentukan tujuan (*goal*) yang ingin dibuktikan.
2. Cari aturan yang memiliki kesimpulan (THEN) sama dengan goal tersebut.
3. Uji apakah premis (IF) aturan tersebut benar berdasarkan fakta.
4. Jika premis belum diketahui, jadikan premis baru sebagai sub-goal.
5. Ulangi hingga goal terbukti atau tidak bisa dibuktikan.

### Contoh Kasus: Pembuktian Diagnosis ISPA

#### Basis Pengetahuan:

R1: IF batuk AND demam THEN penyakit = ISPA

R2: IF ISPA AND sesak THEN tindakan = "Konsultasi ke dokter"

R3: IF ISPA AND pilek THEN tindakan = "Minum obat flu"

#### Goal:

Apakah pasien perlu konsultasi ke dokter?

#### Langkah Penalaran Mundur:

1. Cari aturan dengan THEN tindakan = Konsultasi ke dokter → (R2).

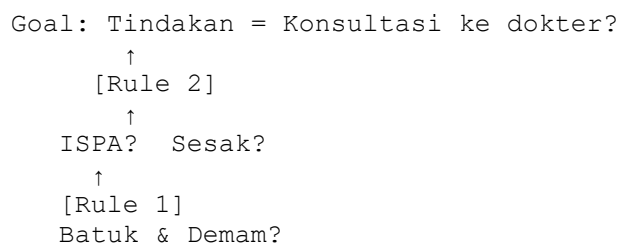
---

Kecerdasan Buatan (Artificial Intelligence)



2. Premis R2: pasien ISPA dan sesak.
3. Periksa fakta:
  - Sesak? ✗ Tidak ada fakta.
  - ISPA? Belum diketahui.
4. Untuk membuktikan ISPA, sistem mencari aturan R1:  
 IF batuk AND demam THEN ISPA.  
 Fakta: batuk , demam  → ISPA
5. Namun karena sesak ✗, maka R2 tidak bisa dijalankan.  
**➡ Kesimpulan: Pasien tidak perlu konsultasi ke dokter (karena tidak sesak).**

### Diagram Backward Chaining



### Kelebihan Backward Chaining

- Cocok untuk pembuktian hipotesis atau pencarian tujuan tertentu.
- Lebih efisien karena sistem hanya menelusuri aturan yang relevan dengan goal.
- Umumnya digunakan dalam sistem konsultasi atau diagnosa medis.

### Kelemahan

- Tidak cocok jika banyak kemungkinan tujuan.
- Sulit digunakan jika fakta awal tidak lengkap.
- Membutuhkan strategi kontrol (aturan mana yang diperiksa dulu).

## 5. Perbandingan Forward vs Backward Chaining

Aspek	Forward Chaining	Backward Chaining
Arah Penalaran	Dari fakta → kesimpulan	Dari kesimpulan → fakta



Sifat	Data-driven	Goal-driven
Cocok Untuk	Diagnosis, prediksi, klasifikasi	Pembuktian hipotesis, konsultasi
Contoh Proses	“Dari gejala menuju penyakit”	“Apakah pasien menderita penyakit X?”
Metode Logika	Modus Ponens	Modus Tollens
Kelebihan	Menemukan semua kemungkinan	Lebih fokus dan efisien
Kelemahan	Butuh banyak data awal	Butuh tujuan yang jelas

## 6. Contoh Penerapan di Dunia Nyata

Bidang	Jenis Penalaran	Contoh Sistem
Medis	Forward	Diagnosa gejala → penyakit (MYCIN)
Hukum	Backward	Pembuktian pelanggaran → cek fakta pendukung
Pertanian	Forward	Gejala tanaman → rekomendasi pupuk
Pendidikan	Backward	Apakah siswa siap UAS? → cek indikator kesiapan
Keuangan	Forward	Data nasabah → hasil kelayakan kredit

## 7. Ilustrasi Perbandingan (Diagram Umum)

Forward Chaining (Data-driven)

Fakta → [Aturan 1] → Fakta Baru → [Aturan 2] → Kesimpulan

Backward Chaining (Goal-driven)

Goal → [Aturan yang relevan] → Premis → [Cari Fakta Pendukung] → Bukti atau Tolak

### C. Studi Kasus Sistem Pakar Medis

#### Kasus: Sistem Pakar Diagnosa Penyakit Pernapasan

Seorang pakar medis ingin membuat sistem yang dapat membantu pasien mengenali kemungkinan penyakit pernapasan awal, seperti ISPA (Infeksi Saluran Pernapasan Atas), Asma, dan Bronkitis, berdasarkan gejala yang dialami.



## Tahap 1 — Basis Pengetahuan (Knowledge Base)

Pengetahuan pakar dikumpulkan dalam bentuk **aturan IF–THEN**, seperti berikut:

### ■ Aturan Diagnosa:

R1: IF batuk AND demam THEN penyakit = ISPA

R2: IF sesak AND batuk THEN penyakit = Asma

R3: IF batuk AND dahak THEN penyakit = Bronkitis

### ■ Aturan Rekomendasi:

R4: IF penyakit = ISPA THEN tindakan = "Minum obat flu & istirahat"

R5: IF penyakit = Asma THEN tindakan = "Gunakan inhaler & hindari debu"

R6: IF penyakit = Bronkitis THEN tindakan = "Konsultasi ke dokter paru"

## Tahap 2 — Basis Fakta (Working Memory)

Pengguna (pasien) mengisi data gejala yang dialami melalui antarmuka pengguna (*User Interface*):

Gejala	Nilai
Batuk	<input checked="" type="checkbox"/> Ya
Demam	<input checked="" type="checkbox"/> Ya
Sesak	<input checked="" type="checkbox"/> Tidak
Dahak	<input checked="" type="checkbox"/> Tidak

## Tahap 3 — Mesin Inferensi (*Inference Engine*)

### Langkah Penalaran (*Forward Chaining*)

1. Fakta awal dari pengguna:

batuk = True, demam = True

2. Periksa aturan:

R1 cocok: IF batuk AND demam → penyakit = ISPA

Tambahkan fakta baru: penyakit = ISPA.



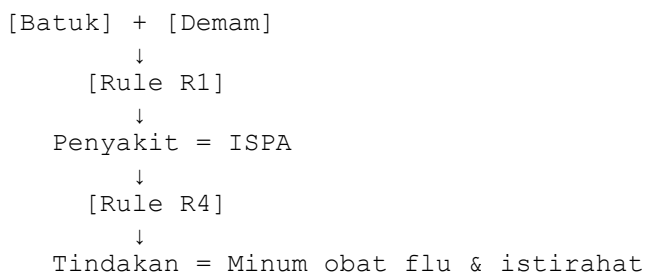
3. Fakta baru (penyakit = ISPA) digunakan untuk memeriksa aturan berikutnya:  
R4 cocok: IF penyakit = ISPA → tindakan = “Minum obat flu & istirahat.”
4. Tidak ada aturan lain yang bisa dijalankan.

**➔ Kesimpulan akhir sistem:**

Kemungkinan penyakit: ISPA

Rekomendasi tindakan: Minum obat flu & istirahat

**Diagram Inferensi (Forward Chaining)**



**Tahap 4 — Antarmuka Pengguna (User Interface)**

Antarmuka pengguna berperan untuk menghubungkan sistem dengan pasien atau pengguna non-teknis.

**Contoh Tampilan Dialog:**

```

=====
SISTEM PAKAR DIAGNOSA PENYAKIT ISPA
=====
Masukkan kondisi Anda:

Batuk? (Y/T): Y
Demam? (Y/T): Y
Sesak napas? (Y/T): T
Dahak? (Y/T): T

Proses inferensi...
→ Berdasarkan gejala Anda:
  Kemungkinan penyakit : ISPA
  Rekomendasi tindakan : Minum obat flu & istirahat
-----
Penjelasan:
  Karena Anda mengalami batuk dan demam,
  sistem menyimpulkan penyakit = ISPA.
=====

```



## Tahap 5 — Fasilitas Penjelasan (Explanation Facility)

Fasilitas penjelasan memberikan **transparansi logika sistem**, menjawab pertanyaan seperti:

“Mengapa sistem menyimpulkan saya terkena ISPA?”

### ■ Contoh Penjelasan Otomatis:

Sistem menggunakan aturan:

```
IF batuk AND demam THEN penyakit = ISPA
```

Kondisi Anda memenuhi kedua gejala tersebut, maka sistem menyimpulkan Anda kemungkinan ISPA.

Fasilitas ini penting agar pengguna mempercayai hasil inferensi dan memahami dasar keputusan sistem.

## Tahap 6 — Simulasi Implementasi (Pseudocode Python)

```
# Basis Pengetahuan
def diagnosa(gejala):
    penyakit = None
    tindakan = None

    if gejala['batuk'] and gejala['demam']:
        penyakit = 'ISPA'
    elif gejala['batuk'] and gejala['sesak']:
        penyakit = 'Asma'
    elif gejala['batuk'] and gejala['dahak']:
        penyakit = 'Bronkitis'

    if penyakit == 'ISPA':
        tindakan = 'Minum obat flu & istirahat'
    elif penyakit == 'Asma':
        tindakan = 'Gunakan inhaler & hindari debu'
    elif penyakit == 'Bronkitis':
        tindakan = 'Konsultasi ke dokter paru'

    return penyakit, tindakan

# Fakta input
gejala = {'batuk': True, 'demam': True, 'sesak': False, 'dahak':
False}

penyakit, tindakan = diagnosa(gejala)

print("Kemungkinan Penyakit :", penyakit)
print("Rekomendasi Tindakan :", tindakan)
```



## Output:

Kemungkinan Penyakit : ISPA

Rekomendasi Tindakan : Minum obat flu & istirahat

## Soal

### A. Pemahaman Konsep Dasar

1. Jelaskan dengan kata-kata Anda sendiri apa yang dimaksud dengan Sistem Pakar (*Expert System*) dan sebutkan dua karakteristik utama yang membedakannya dari program komputer biasa.
2. Sebutkan dan jelaskan tiga komponen utama dalam struktur sistem pakar serta peran masing-masing dalam proses penalaran.
3. Mengapa basis pengetahuan (*knowledge base*) disebut sebagai “jantung” dari sistem pakar? Berikan contoh bentuk aturan yang biasanya disimpan di dalamnya.
4. Apa fungsi dari antarmuka pengguna (*user interface*) dalam sistem pakar? Jelaskan mengapa peran antarmuka sangat penting dalam penerapan sistem pakar di masyarakat.
5. Jelaskan perbedaan antara pengetahuan deklaratif dan pengetahuan prosedural yang tersimpan dalam basis pengetahuan sistem pakar.

### B. Analisis Penalaran

6. Jelaskan perbedaan mendasar antara Forward Chaining dan Backward Chaining dalam sistem pakar, baik dari sisi arah penalaran maupun tujuan penggunaannya.
7. Dalam konteks sistem pakar medis, metode penalaran mana yang lebih tepat digunakan untuk *diagnosa gejala menjadi penyakit*? Jelaskan alasan logisnya.
8. Diberikan aturan berikut:  
R1: IF hujan THEN tanah\_basah  
R2: IF tanah\_basah THEN rumput\_tumbuh  
R3: IF rumput\_tumbuh THEN kambing\_senang



Jika diketahui fakta: “*hujan*”, jelaskan langkah-langkah inferensi Forward Chaining untuk mencapai kesimpulan akhir.

9. Pada sistem pakar hukum, pengguna ingin mengetahui apakah seseorang “melanggar peraturan lalu lintas.”

Jelaskan bagaimana Backward Chaining bekerja untuk membuktikan kesimpulan tersebut dengan memeriksa fakta-fakta pendukung.

10. Mengapa dalam beberapa kasus sistem pakar modern digunakan kombinasi kedua metode penalaran (hybrid reasoning)? Jelaskan keuntungannya.

