

# **SISTEM PAKAR**

**SEMESTER II**

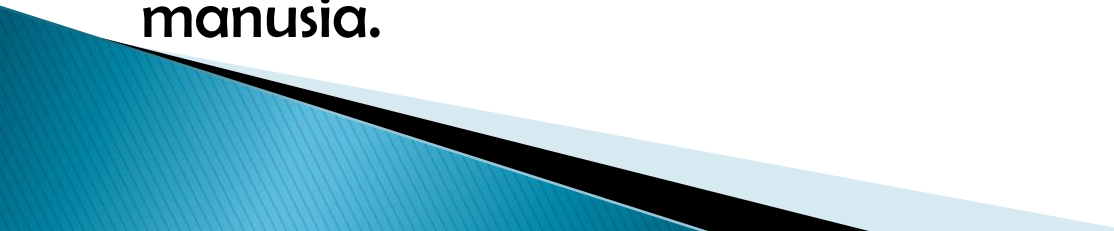
**PERTEMUAN KE – 11 & 12**

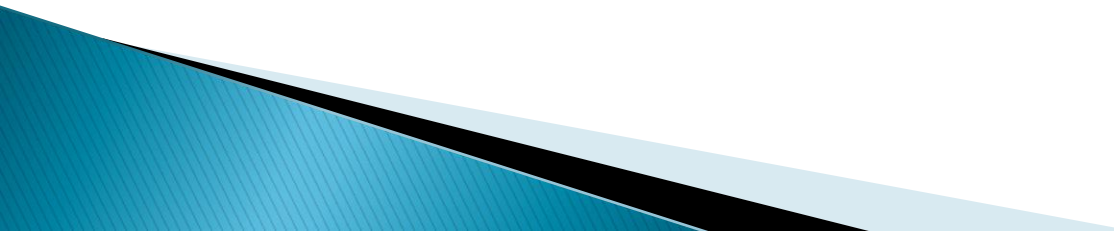
**DOSEN : ELISAWATI,M.KOM**

# Sistem Pakar


- ▶ Istilah sistem pakar berasal dari istilah knowledge-based expert system. Istilah ini muncul karena untuk memecahkan masalah, sistem pakar menggunakan pengetahuan seorang pakar yang di masukkan ke dalam komputer.

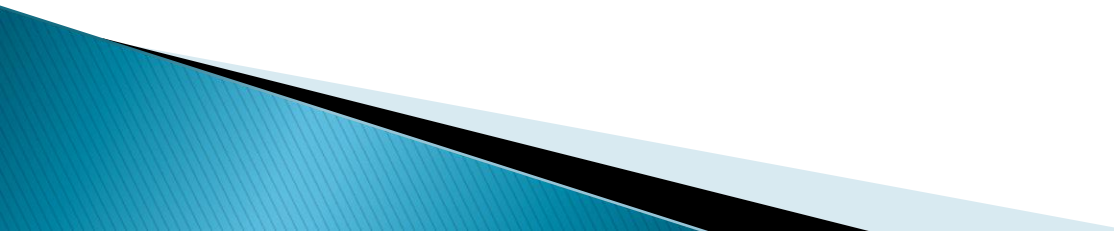
Berikut adalah beberapa pengertian sistem pakar.

- ▶ Turban (2001, p402) sistem pakar adalah sebuah sistem yang menggunakan pengetahuan manusia di mana pengetahuan tersebut dimasukkan ke dalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan kepakaran atau keahlian manusia.
- 

- ▶ Jackson (1999, p3) Sistem pakar adalah program komputer yang merepresentasikan dan melakukan penalaran dengan pengetahuan beberapa pakar untuk memecahkan masalah atau memeberikan saran.
  - ▶ Luger dan Stubblefield (1993,p308) sistem pakar adalah program yang berbasiskan pengetahuan yang menyediakan solusi “kualitas pakar” kepada masalah-masalah dalam bidang (domain) yang spesifik.
- 

# Manfaat Sistem Pakar

- ▶ Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
  - ▶ Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
  - ▶ Meningkatkan kualitas, dengan memberi nasihat yang konsisten dan mengurangi kesalahan
  - ▶ Mampu menangkap pengetahuan dan kepakaran seseorang
  - ▶ Dapat beroperasi di lingkungan yang berbahaya
  - ▶ Memudahkan akses pengetahuan seorang pakar
  - ▶ Andal. Sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit.
- 

- ▶ Meningkatkan kapabilitas sistem komputer
  - ▶ Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti.
  - ▶ Bisa di gunakan sebagai media pelengkap dalam pelatihan.
  - ▶ Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.
- 

# Kekurangan Sistem Pakar


- ▶ Biaya sangat mahal untuk membuat dan memeliharanya
- ▶ Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar
- ▶ Sistem pakar tidak 100% bernilai benar

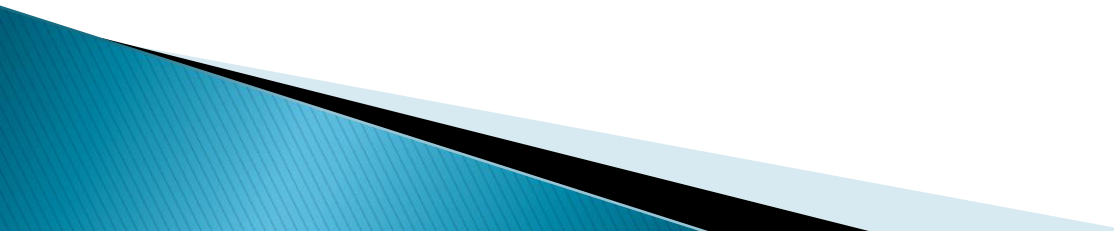
## Ciri-ciri sistem pakar

- ▶ Terbatas pada domain keahlian tertentu
- ▶ Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti
- ▶ Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami
- ▶ Bekerja berdasarkan kaidah/rule/tertentu
- ▶ Mudah dimodifikasi

- ▶ Basis pengetahuan dan mekanisme inferensi terpisah
- ▶ Keluarannya bersifat anjuran
- ▶ Sistem dapat mengaktifkan kaidah secara searah yang sesuai, di tuntun oleh dialog dengan pengguna.

## Area permasalahan aplikasi sistem pakar

- ▶ Interpretasi : menghasilkan deskripsi situasi berdasarkan data-data masukan
  - ▶ Prediksi : memperkirakan akibat yang mungkin terjadi dari situasi yang ada
  - ▶ Diagnosis : menyimpulkan suatu keadaan berdasarkan gejala-gejala yang di berikan
  - ▶ Desain : melakukan perancangan berdasarkan kendala-kendala yang di berikan.
- 

- ▶ **Planning** : merencanakan tindakan-tindakan yang akan dilakukan
  - ▶ **Monitoring** : membandingkan hasil pengamatan dengan proses perencanaan
  - ▶ **Debugging** : menentukan penyelesaian dari suatu kesalahan sistem
  - ▶ **Reparasi** : melaksanakan perbaikan
  - ▶ **Instruction** : melakukan instruksi untuk diagnosis, debugging dan perbaikan kinerja
  - ▶ **Kontrol** : melakukan kontrol terhadap hasil interpretasi, diagnosis, debugging, monitoring dan perbaikan tingkah laku sistem
- 



# Konsep dasar sistem pakar

- ▶ **Kepakaran (Expertise)**

Kepakaran merupakan suatu pengetahuan yang di peroleh dari pelatihan, membaca, dan pengalaman.

- ▶ **Pakar (Expert)**

Pakar adalah seseorang yang mempunyai pengetahuan, pengalaman dan metode khusus, serta mampu menerapkannya untuk memecahkan masalah atau memberi nasihat.

- ▶ **Pemindahan Kepakaran (Transferring Expertise)**

Tujuan dari sistem pakar adalah memindahkan kepakaran dari seorang pakar ke dalam komputer, kemudian di transfer kepada orang lain yang bukan pakar.

- ▶ **Inferensi (Inferencing)**

Inferensi adalah sebuah prosedur yang mempunyai kemampuan dalam melakukan penalaran.

- ▶ **Aturan-Aturan (Rule)**

Kebanyakan software sistem pakar komersial adalah sistem yang berbasis rule (rule-based systems), yaitu pengetahuan disimpan terutama dalam bentuk rule, sebagai prosedur-prosedure pemecahan masalah.

- ▶ **Kemampuan Menjelaskan (Explanation Capability)**

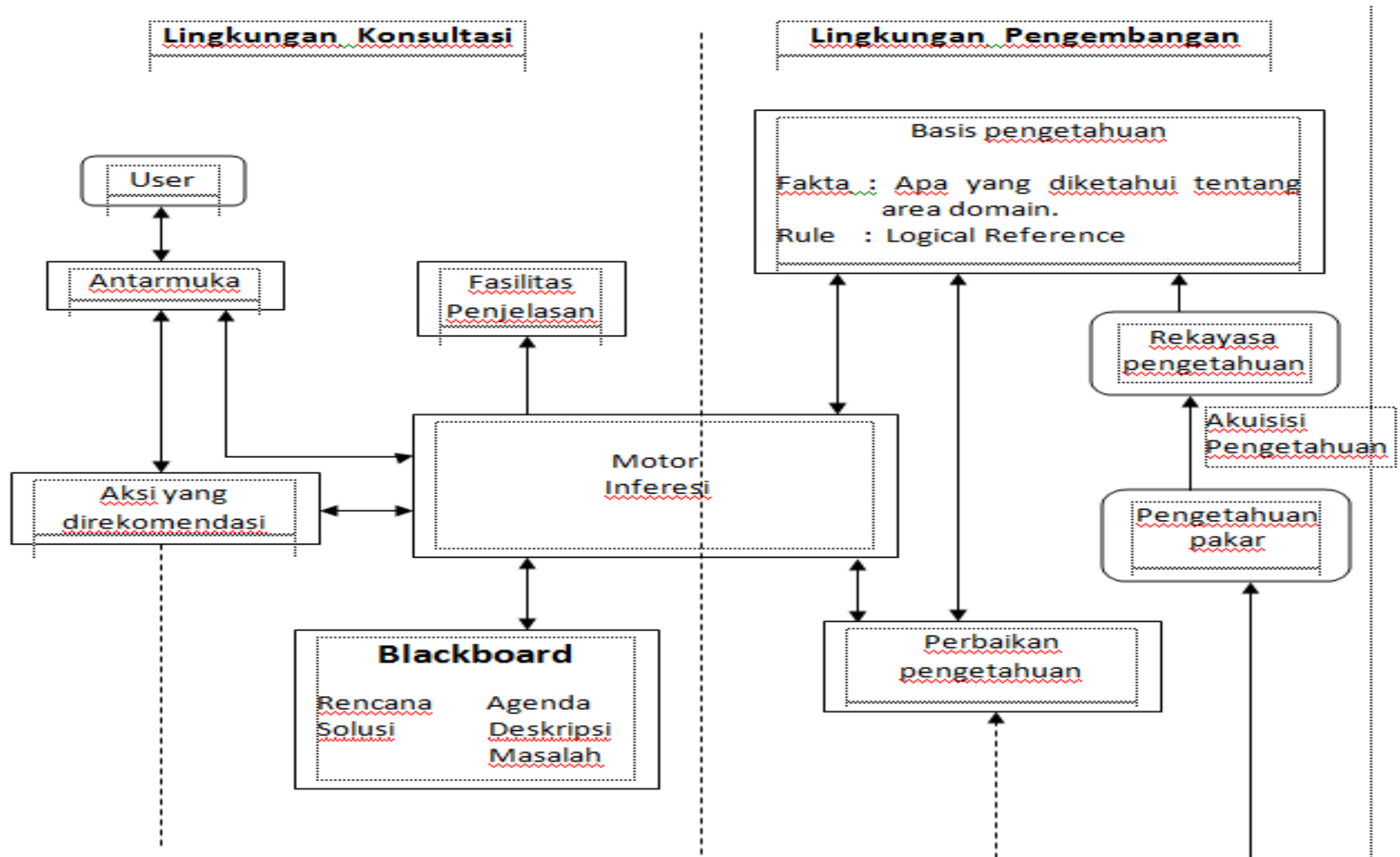
Fasilitas lain dari system pakar adalah kemampuannya untuk menjelaskan saran atau rekomendasi yang diberikannya.

# Perbandingan sistem konvensional dengan sistem pakar

Sistem Konvensional	Sistem Pakar
Informasi dan pemrosesannya biasanya digabungkan dalam satu program	Basis pengetahuan dipisahkan secara jelas dengan mekanisme inferensi
Program tidak membuat kesalahan (Yang membuat kesalahan : Pemrograman atau pengguna)	Program dapat berbuat kesalahan
Biasanya tidak menjelaskan mengapa data masukan diperlukan atau bagaimana output dihasilkan	Penjelasan merupakan bagian terpenting dari semua sistem pakar
Perubahan program sangat menyulitkan	Perubahan dalam aturan-aturan mudah untuk dilakukan
Sistem hanya bisa beroperasi setelah lengkap atau selesai	Sistem dapat beroperasi hanya dengan aturan-aturan yang sedikit (sebagai prototipe awal)

Sistem Konvensional	Sistem Pakar
Eksekusi dilakukan langkah demi langkah (algoritmik)	Eksekusi dilakukan dengan menggunakan heuristik dan logika pada seluruh basis pengetahuan
Perlu informasi lengkap agar bisa beroperasi	Dapat beroperasi dengan informasi yang tidak lengkap atau mengandung ketidakpastian
Manipulasi efektif dari basis data yang besar	Manipulasi efektif dari basis pengetahuan yang besar
Menggunakan data	Menggunakan pengetahuan
Tujuan utama : Efisiensi	Tujuan utama efektivitas
Mudah berurusan dengan data kuantitatif	Mudah berurusan dengan data kualitatif
Menangkap, menambah dan mendistribusikan akses ke data numerik atau informasi	Menangkap, menambah dan mendistribusikan akses ke pertimbangan dan pengetahuan.

# Struktur Sistem Pakar



# Rule Sebagai Teknik Representasi Pengetahuan

Setiap rule terdiri dari dua bagian, yaitu bagian **IF** disebut evidence (fakta-fakta) dan bagian **THEN** di sebut Hipotesis atau kesimpulan.

Syntax Rule adalah :

**IF E THEN H**

Secara umum, rule mempunyai evidence lebih dari satu yang di hubungkan oleh kata penghubung **AND** atau **OR** atau kombinasi keduanya. Tetapi sebaiknya biasakan menghindari penggunaan **AND** dan **OR** secara sekaligus dalam satu rule.

**IF (E1 AND E2 AND E3 ..... AND En) THEN H**

**IF (E1 OR E2 OR E3..... OR En) THEN H**

Satu evidence bisa juga mempunyai hipotesis lebih dari satu.

**IF E THEN (H1 AND H2 AND H3 ..... AND Hn)**

# Forward Chaining

Forward Chaining adalah teknik pencarian yang di mulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian **IF** dari rules **IF – THEN**.

- ▶ Bila ada fakta yang cocok dengan bagian IF, maka rule tersebut di eksekusi. Bila sebuah rule di eksekusi, maka sebuah fakta baru (bagian THEN) di tambahkan ke dalam databases. Setiap kali pencocokan, di mulai dari rule teratas. Setiap rule hanya boleh di eksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi rule yang bisa di eksekusi. Metode pencarian yang di gunakan adalah DFS, BFS.

# Contoh :

Misalkan diketahui sistem pakar menggunakan 5 buah rule berikut :

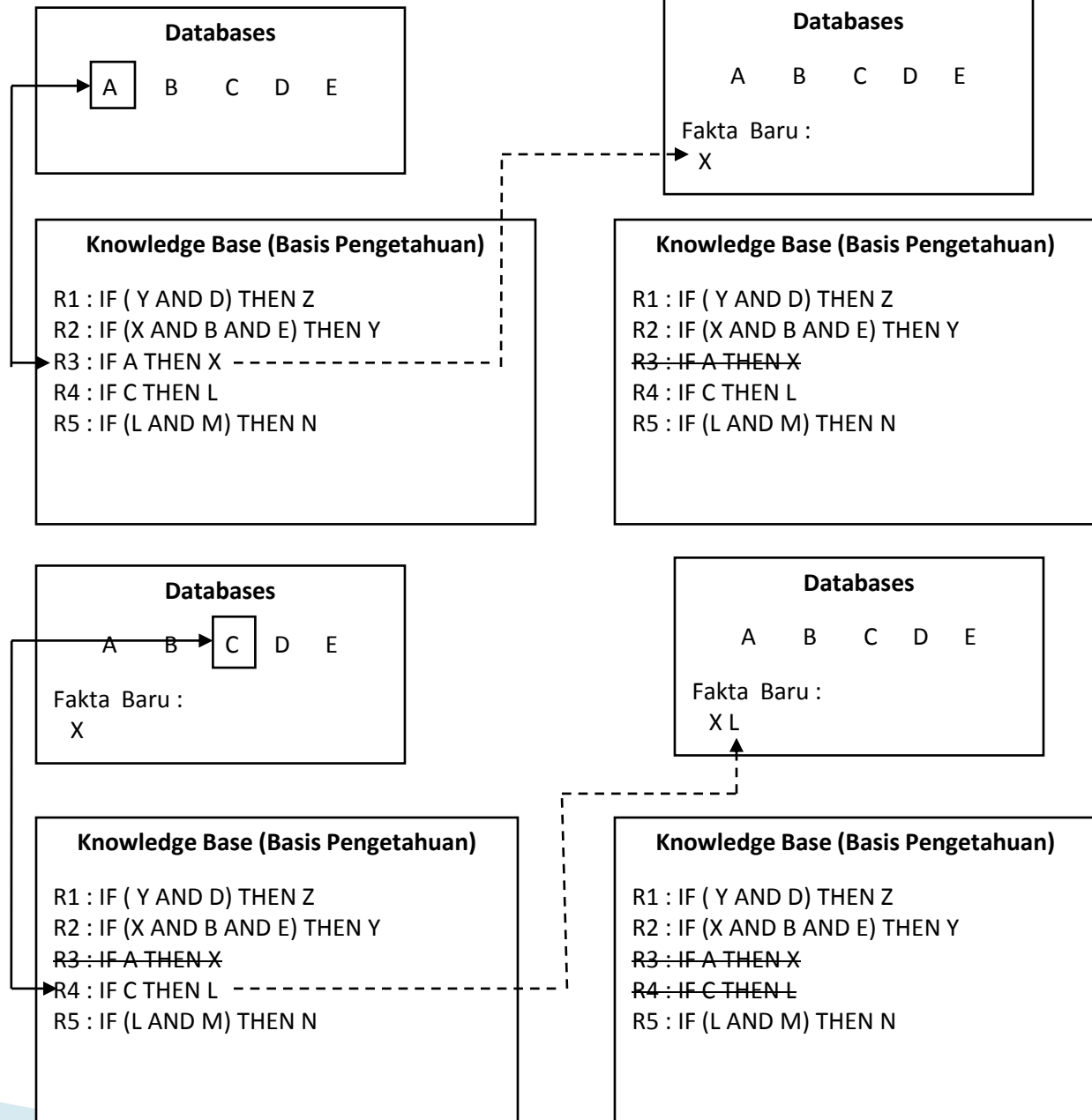
- ▶ [R1] : IF (Y AND D) THEN Z
- ▶ [R2] : IF (X AND B AND E) THEN Y
- ▶ [R3] : IF A THEN X
- ▶ [R4] : IF C THEN L
- ▶ [R5] : IF (L AND M) THEN N

Fakta-fakta : A, B, C, D dan E bernilai benar

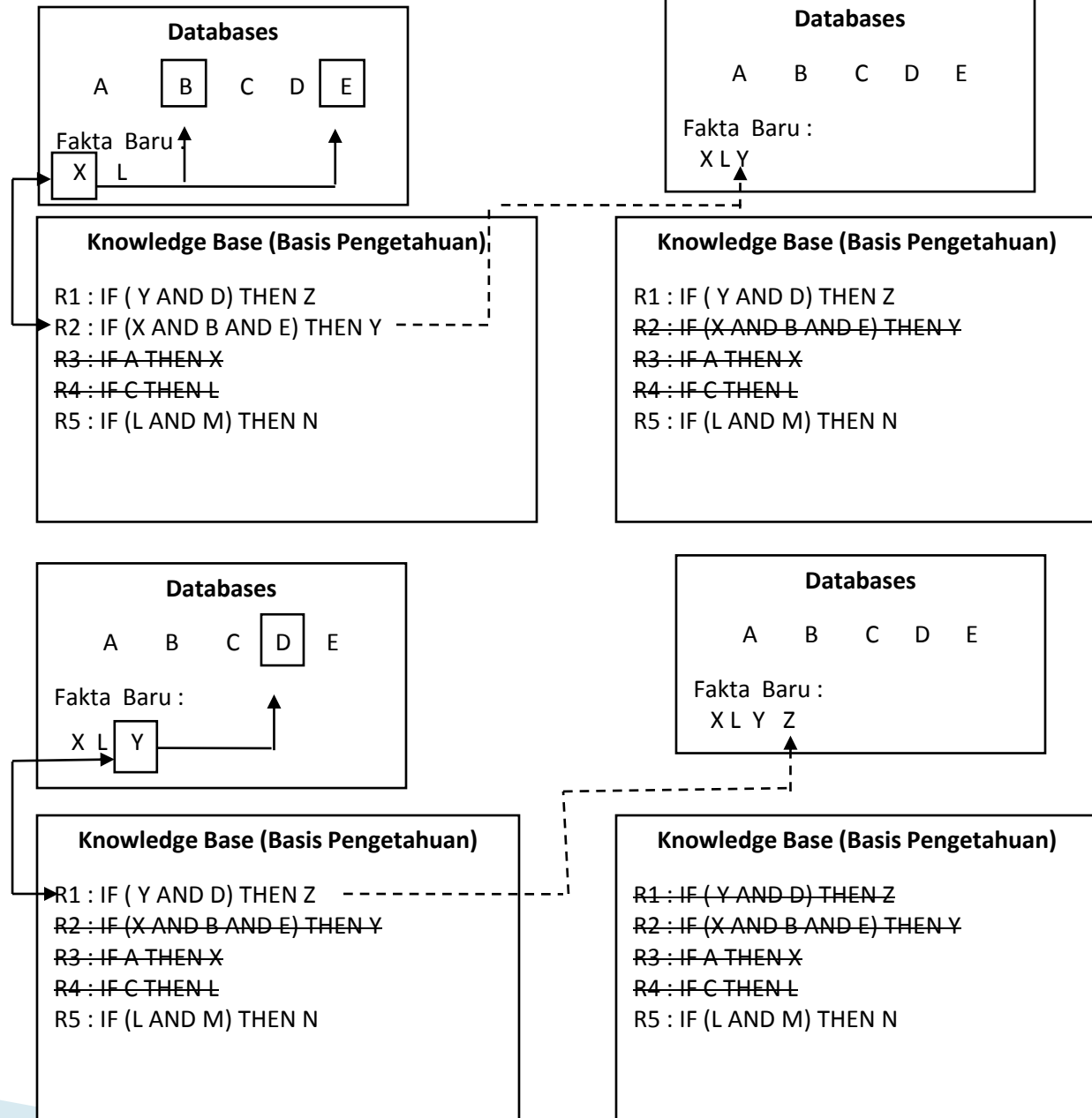
Goal : Menentukan apakah Z bernilai benar atau salah



# Iterasi



# Iterasi ke



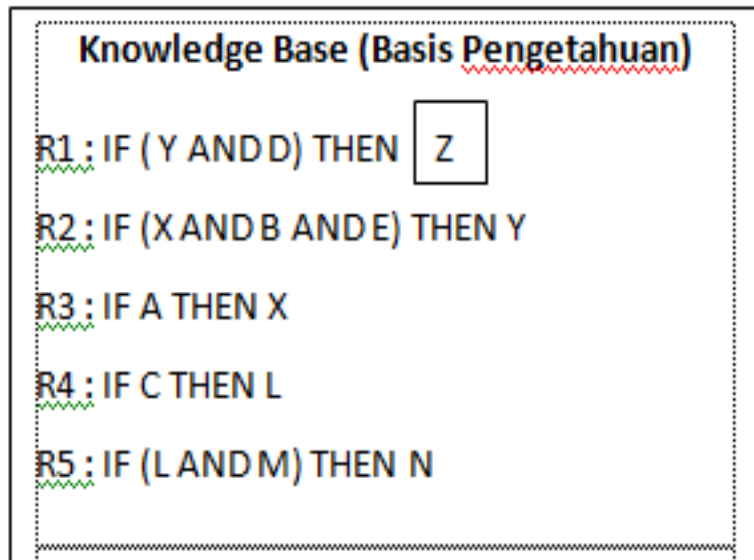
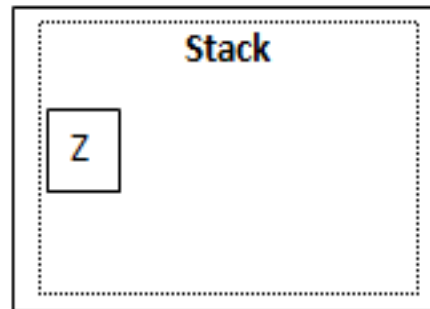
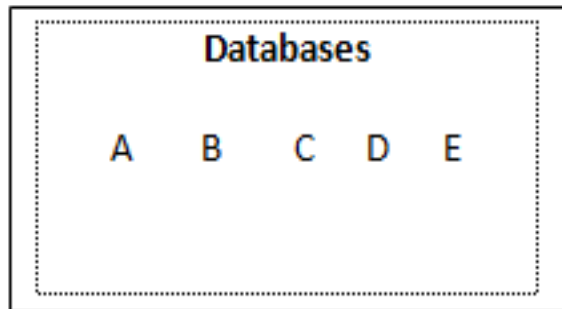
- ▶ Sampai di sini proses dihentikan karena sudah tidak ada lagi rule yang bisa di eksekusi. Hasil pencarian adalah Z bernilai benar. (lihat database di bagian fakta baru).

# Backward Chaining

Backward chaining adalah metode inferensi yang bekerja mundur ke arah kondisi awal. Proses diawali dari Goal (yang berada di bagian **THEN** dari rule **IF – THEN**), kemudian pencarian di mulai dijalankan untuk mencocokkan apakah fakta-fakta yang ada cocok dengan premis-premis di bagian **IF**.

Jika cocok, rule dieksekusi, kemudian hipotesis di bagian **THEN** di tempatkan di basis data sebagai fakta baru. Jika tidak cocok, simpan premis di bagian **IF** ke dalam stack sebagai subGoal. Proses berakhir jika Goal ditemukan atau tidak ada rule yang bisa membuktikan kebenaran dari subGoal atau Goal

## Iterasi ke- 1

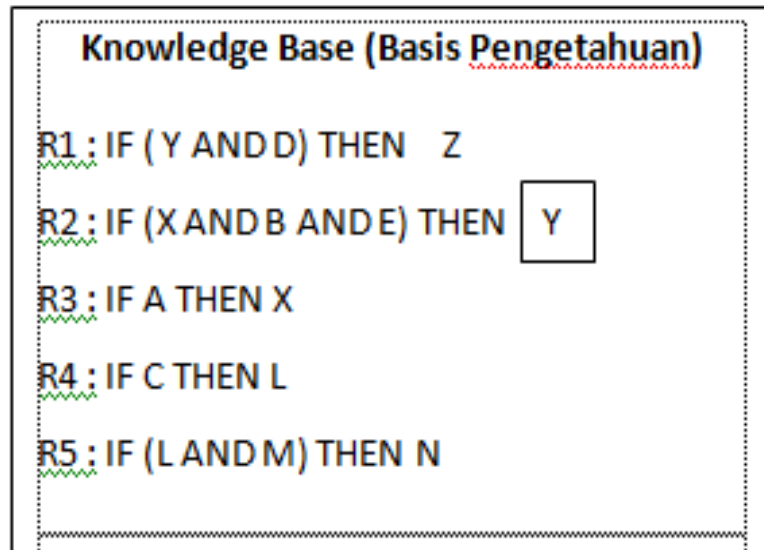
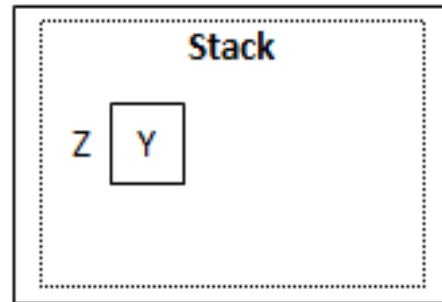
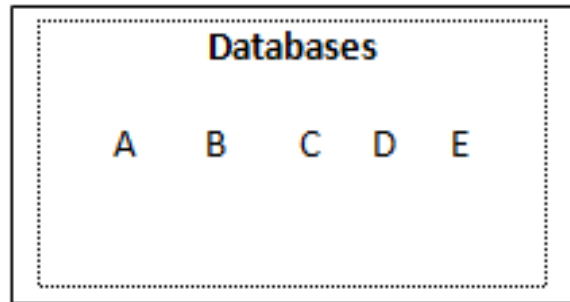


Goal : Z (sebagai isi awal dari stack)

D : ada di database

Y : Tidak ada di databases, simpan di stack

## Iterasi ke 2



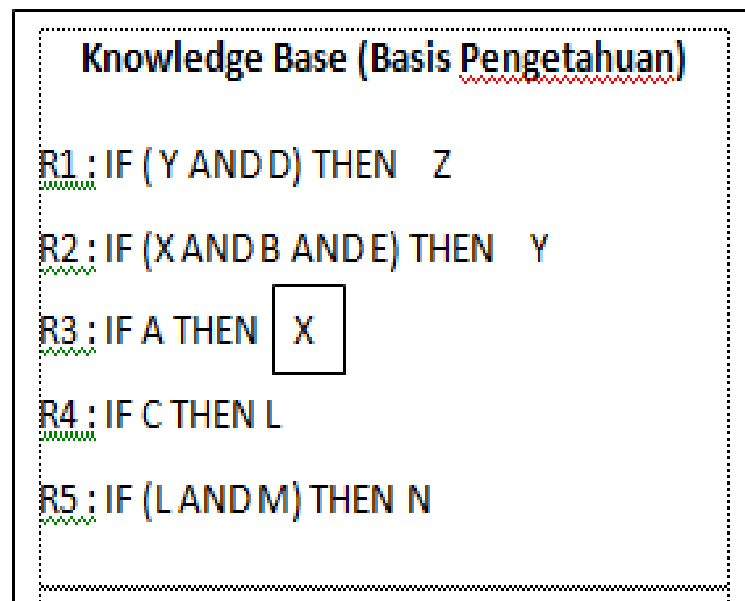
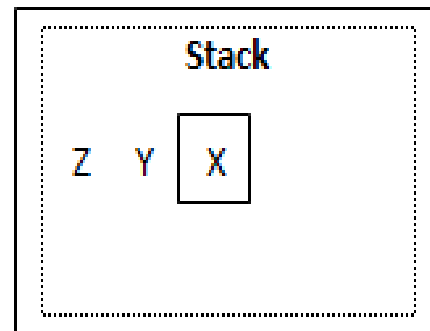
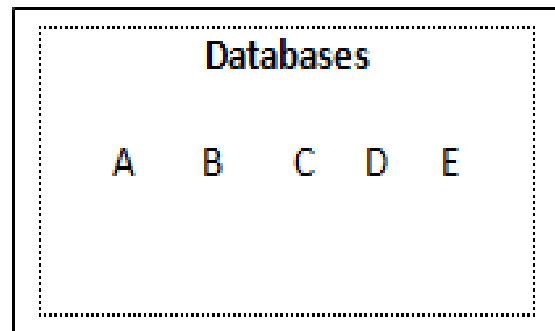
subGoal : Y

B : ada di database

E : ada di database

X : Tidak ada di databases, simpan di stack

### Iterasi - 3

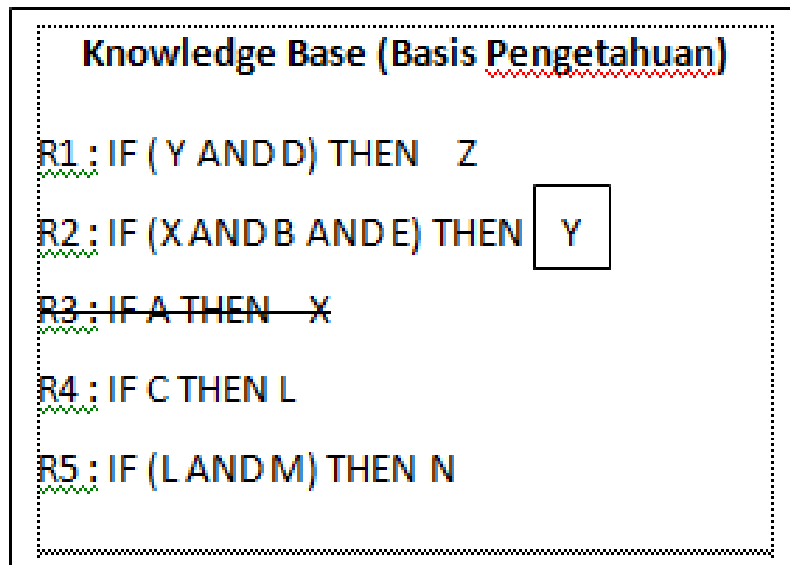
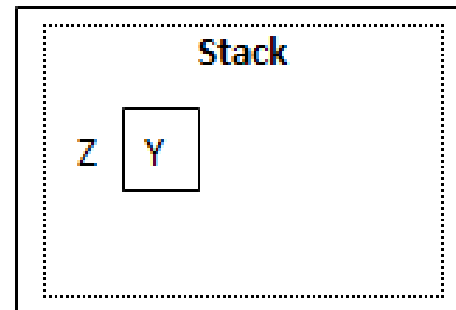
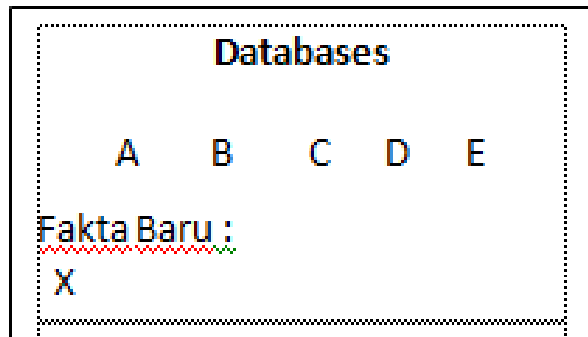


subGoal : X

A : ada di database

X : hapus dari stack dan masukkan ke  
fakta baru

## Iterasi ke -4



subGoal : Y

X : ada di database

B : ada di database

E : ada di database

Y : hapus dari stack dan masukkan ke  
fakta baru



## Iterasi ke -5

### Databases

A B C D E

Fakta Baru :

X Y

### Stack

Z

### Knowledge Base (Basis Pengetahuan)

R1 : IF (Y AND D) THEN Z

~~R2 : IF (X AND B AND E) THEN Y~~

~~R3 : IF A THEN X~~

~~R4 : IF C THEN L~~

~~R5 : IF (L AND M) THEN N~~

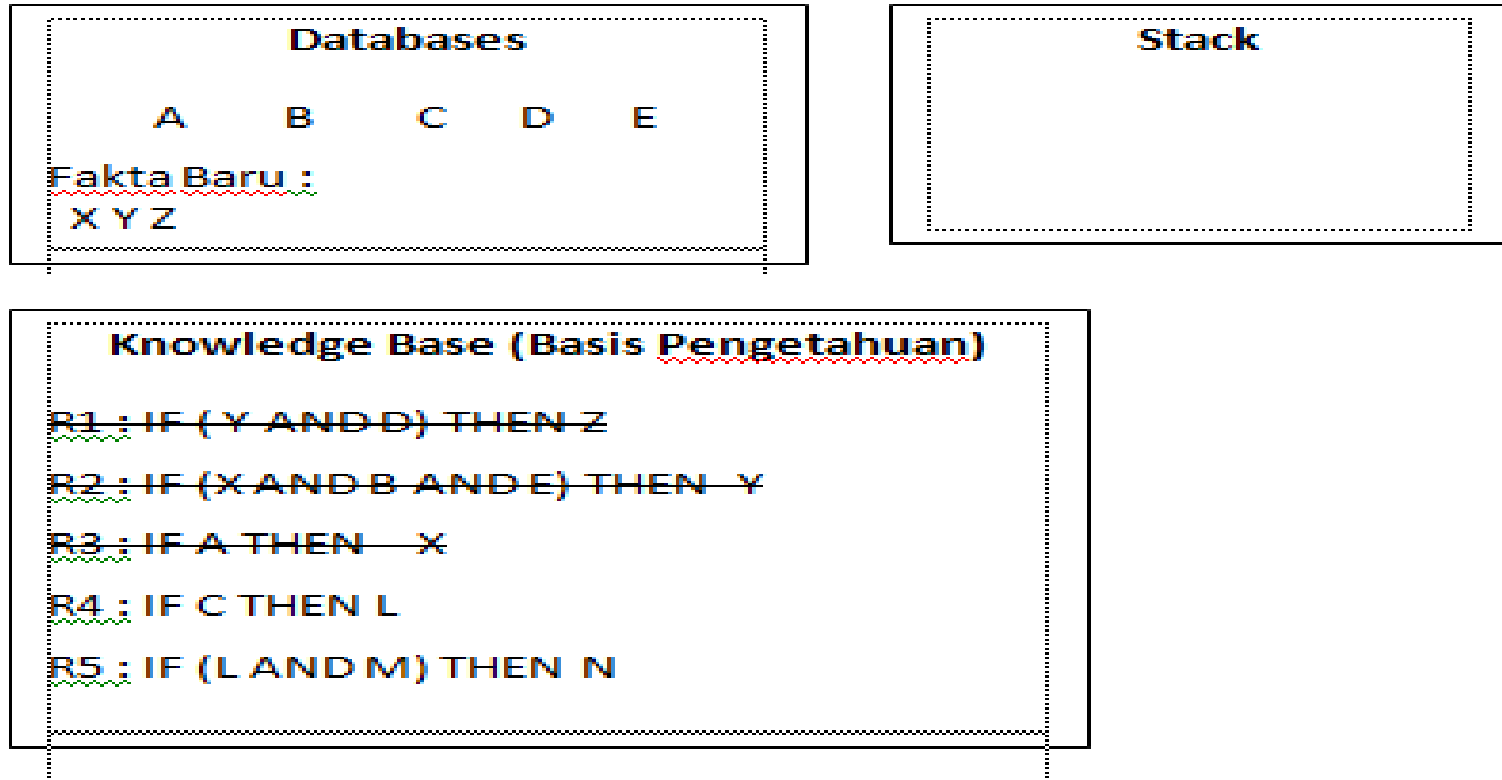
Goal : Z

Y: ada di database

D: ada di database

Z: hapus dari stack dan masukkan ke  
fakta baru

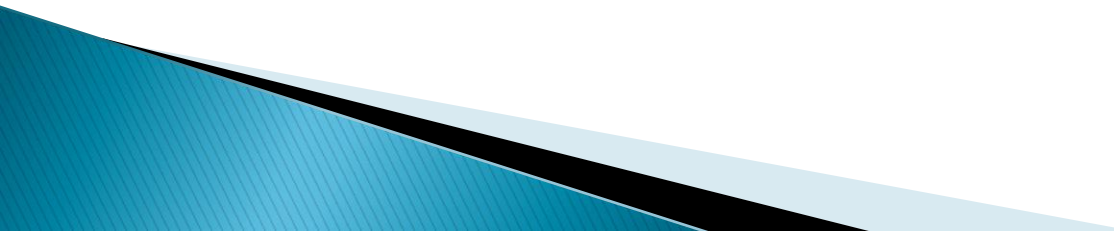
## Iterasi ke - 6



Karena Goal Z ditemukan di database, maka proses pencarian di hentikan. Disini terbukti bahwa Z bernilai benar.

# Soal :

Berikut adalah contoh sistem pakar yang menggunakan 10 rule berikut :

1. IF (bukan hari minggu) AND (Toyes sehat) THEN (Toyes kuliah)
  2. IF (Toyes kuliah) THEN (Sylvi kuliah)
  3. IF (Bukan hari minggu) AND (Toyes sakit) THEN (Toyes di rumah sakit)
  4. IF (Bukan hari minggu) THEN (Mahasiswa UDINUS kuliah)
  5. IF (Toyes di rumah sakit) AND (Mahasiswa UDINUS kuliah) THEN (Sylvi kuliah)
  6. IF (Mahasiswa UDINUS kuliah) AND (Toyes sakit) THEN (Sylvi sakit)
  7. IF (Toyes kuliah) AND (Sylvi tidak kuliah) THEN (Sylvi sakit)
  8. IF (Bukan hari minggu) AND (Sylvi sakit) THEN (kuliah tidak libur)
  9. IF (Mahasiswa UDINUS kuliah) THEN (kuliah tidak libur)
  10. IF (kuliah tidak libur) THEN (Toyes belajar di kampus UDINUS)
- 

# Soal :

Semula diberikan dua buah fakta berikut :

Hari senin (bukan hari minggu)

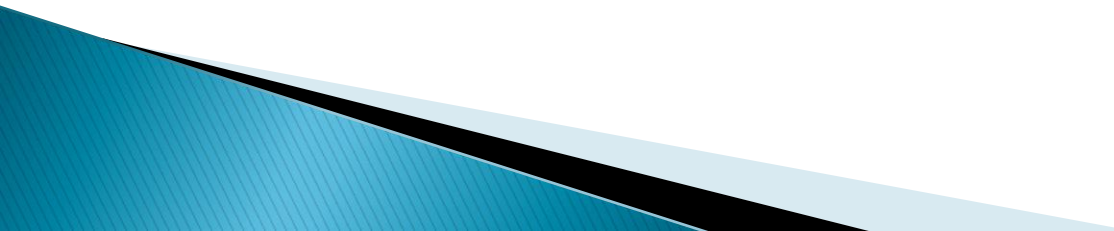
Toyes kuliah

Ini berarti kedua fakta tersebut tersimpan di dalam memori kerja. Ingin di buktikan apakah fakta “Toyos belajar di kampus UDINUS” dan “Sylvi tidak kuliah” bernilai benar.

Buktikan dengan menggunakan Forward Chaining dan Backward Chaining.

# Certainty Factor (Faktor Kepastian)

Teori Certainty Factor (CF) di usulkan oleh Shortliffe dan Buchanan pada tahun 1975 untuk mengakomodasi ketidakpastian pemikiran (inexact reasoning) seorang pakar. Seorang pakar, (misalnya dokter) sering kali menganalisa informasi yang ada dengan ungkapan seperti “mungkin”, “kemungkinan besar”, “hampir pasti”. Untuk mengakomodasi hal ini kita menggunakan certainty factor (CF) guna menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang di hadapi.



Ada dua cara dalam mendapatkan tingkat keyakinan (CF) dari sebuah rule, yaitu :

- ▶ Metode “Net Belief” yang di usulkan oleh E.H. Shortliffe dan B.G. Buchanan

$$CF(\text{rule}) = MB(H,E) - MD(H, E) \quad \dots\dots\dots (1)$$

$$MB(H,E) = \begin{cases} 1 & P(H) = 1 \\ \frac{\max[P(H|E), P(H)] - P(H)}{\max[1,0] - P(H)} & \dots\dots\dots (2) \\ & \text{Lainnya} \end{cases}$$

$$MD(H,E) = \begin{cases} 1 & P(H)=0 \\ \frac{\min[P(H|E), P(H)] - P(H)}{\min[1,0] - P(H)} & \dots\dots\dots (3) \\ & \text{Lainnya} \end{cases}$$

**Dimana :**

**$CF(Rule)$  = Faktor Kepastian**

**$MB(H,E)$  = Measure of belief (ukuran kepercayaan) terhadap hipotesis H, jika di berikan evidence E (antara 0 dan 1)**

**$MD(H,E)$  = measure of disbelief, (ukuran ketidakpercayaan) terhadap Hipotesis H, jika diberikan evidence E (antara 0 dan 1)**

**$P(H)$  = Probabilitas kebenaran hipotesis H**

**$P(H|E)$  = Probabilitas bahwa H benar karena fakta E**

Contoh soal :

Seandainya seorang pakar penyakit kelamin menyatakan bahwa probabilitas seseorang berpenyakit phimosi adalah 0,02. dari data lapangan menunjukkan bahwa dari 100 orang penderita penyakit phimosi, 40 orang memiliki gejala kulup berminyak. Dengan menganggap  $H$  = Phimosi dan  $E$  = Kulup berminyak, hitung faktor kepastian bahwa phimosi disebabkan oleh adanya kulup berminyak.

Jawab :

$$P(\text{Phimosi}) = 0,02$$

$$P(\text{Phimosi} \mid \text{Kulup Berminyak}) = 40/100 = 0,4$$

$$\begin{aligned} MB(H,E) &= \frac{\max[P(H \mid E), P(H)] - P(H)}{\max[1,0] - P(H)} \\ &= \frac{\max[0,4, 0,02] - 0,02}{1 - 0,02} = \frac{0,4 - 0,02}{1 - 0,02} = 0,39 \end{aligned}$$



$$\begin{aligned}
 MD(H,E) &= \frac{\min[P(H|E), P(H)] - P(H)}{\min[1,0] - P(H)} \\
 &= \frac{\min[0,4, 0,02] - 0,02}{0 - 0,02} = \frac{0,02 - 0,02}{- 0,02} = 0
 \end{aligned}$$

$$CF = 0,39 - 0 = 0,39$$

Rule : IF (Gejala = kulup berminyak) THEN Penyakit =  
Phimosi (CF = 0,39)

- ▶ Dengan cara mewawancarai seorang pakar

Nilai  $CF(Rule)$  di dapat dari interpretasi “term” dari pakar, yang di ubah menjadi nilai CF tertentu sesuai tabel berikut :

Uncertain Term	CF
Definitely not (pasti tidak)	-1.0
Almost Certainly not (hampir pasti tidak)	-0.8
Probably not (kemungkinan besar tidak)	-0.6
May be not (mungkin tidak)	-0.4
Unknown (tidak tahu)	-0.2 to 0.2
Maybe (mungkin)	0.4
Probably (Kemungkinan besar)	0.6
Almost certainly (hampir pasti)	0.8
Definitely (Pasti)	1.0

Contoh soal :

Pakar :

Jika batuk dan panas, maka “hampir dipastikan” (Almost certainly) penyakitnya adalah influenza.

Rule : IF (Batuk AND panas) THEN penyakit = influenza  
(CF=0.8)

# Perhitungan Cenrtainly Factor Gabungan

Secara umum, rule direpresentasikan dalam bentuk sebagai berikut :

**IF** E1 **AND** E2.....**AND** En **THEN** H (CF rule)

Atau

**IF** E1 **OR** E2.....**OR** En **THEN** H (CF rule)

Dimana :

E1 .... En : Fakta-fakta (evidence) yang ada

H : Hipotesis atau konklusi yang dihasilkan

CF Rule : Tingkat keyakinan terjadinya hipotesis H akibatnya adanya fakta-fakta E1 .... En

- ▶ Rule dengan evidence E Tunggal dan Hipotesis H Tunggal  
IF E THEN H (CF rule)  
 $CF(H,E) = CF(E) \times CF(rule) \dots\dots\dots(4)$

Catatan :

Secara praktik, nilai CF rule ditentukan oleh pakar, sedangkan nilai CF(E) ditentukan oleh pengguna saat berkonsultasi dengan sistem pakar.

Contoh soal :

IF hari ini terang (CF=0.4) THEN besok hujan (CF=0.6)

Jawab :

$CF(\text{besok hujan, hari ini terang}) = 0.4 \times 0.6 = 0.24$

Artinya : Jika hari ini terang, tingkat kepastian besok hujan adalah 0.24

► Rule dengan evidence E ganda dan Hipotesis H tunggal

IF E1 AND E2.....AND En THEN H (CF rule)

$$CF(H,E) = \min [CF(E1),CF(E2),.....,CF(En)] \times CF \text{ (rule) } .....(5)$$

IF E1 OR E2.....OR En THEN H (CF rule)

$$CF(H,E) = \max[CF(E1),CF(E2),.....,CF(En)] \times CF \text{ (rule) } .....(6)$$

Contoh soal :

IF demam (CF=0.4) AND batuk (CF=0.2) AND muntah (CF=0.7) THEN penyakit = TBC (CF=0.3)

Jawab :

$$CF(TBC \cap \text{demam} \cap \text{batuk} \cap \text{muntah}) = \min[0.4;0.2;0.7] \times 0.3 = 0.2 \times 0.3 = 0.06$$

Artinya : Jika gejala demam dan batuk dan muntah, maka tingkat kepastian terkena penyakit TBC adalah 0.06

- Kombinasi dua buah rule dengan evidence berbeda (E1 dan E2), tetapi hipotesis sama.

IF E1 THEN H

Rule 1      $CF(H,E1)=CF1=C(E1) \times CF(rule1)$

IF E2 THEN H

Rule 2      $CF(H,E2)=CF2=C(E2) \times CF(rule2)$

$$CF(CF1,CF2) = \begin{cases} CF1 + CF2 (1-CF1) & \text{Jika } CF1 > 0 \text{ dan } CF2 > 0 \\ \frac{CF1 + CF2}{1-\min[|CF1|, |CF2|]} & \text{Jika } CF1 > 0 \text{ atau } CF2 < 0 \\ CF1 + CF2 \times (1 + CF1) & \text{Jika } CF1 < 0 \text{ dan } CF2 < 0 \end{cases}$$

## Contoh Soal :

Rule 1 : IF batuk THEN penyakit = batuk rejan (CF = 0.8)

Rule 2 : IF demam THEN penyakit = batuk rejan (CF = 0.6)

Hitung CF gabungan jika :

- a.  $CF(\text{batuk}) = 1$  dan  $CF(\text{demam}) = 1$
- b.  $CF(\text{batuk}) = 1$  dan  $CF(\text{demam}) = -1$
- c.  $CF(\text{batuk}) = -1$  dan  $CF(\text{demam}) = -1$

Jawab :

- a.  $CF1 = C(\text{batuk}) \times CF(\text{Rule1}) = 1 \times 0.8 = 0.8$   
 $CF2 = C(\text{demam}) \times CF(\text{Rule2}) = 1 \times 0.6 = 0.6$   
 $CF = CF1 + CF2 (1 - CF1)$   
 $= 0.8 + 0.6 (1 - 0.8)$   
 $= 0.28$



b.  $CF1 = C(\text{batuk}) \times CF(\text{Rule1}) = 1 \times 0.8 = 0.8$   
 $CF2 = C(\text{demam}) \times CF(\text{Rule2}) = -1 \times 0.6 = -0.6$   
 $CF = \frac{CF1 + CF2}{1 - \min[|CF1|, |CF2|]} = \frac{0.8 - 0.6}{1 - \min(0.8; 0.6)} = \frac{0.2}{1 - 0.6} = 0.5$

c.  $CF1 = C(\text{batuk}) \times CF(\text{Rule1}) = -1 \times 0.8 = -0.8$   
 $CF2 = C(\text{demam}) \times CF(\text{Rule2}) = -1 \times 0.6 = -0.6$   
 $CF = CF1 + CF2 (1 + CF1)$   
 $= -0.8 - 0.6 (1 - 0.8)$   
 $= -0.28$

► **Contoh soal : Sistem pakar ramalan cuaca**

Seseorang berkonsultasi pada sistem pakar untuk mengetahui cuaca apakah besok akan datang hujan atau terang. Sistem pakar ramalan cuaca mempunyai basis pengetahuan berikut:

**Rule 1 : IF hari ini hujan THEN besok hujan (CF = 0.5)**

**Rule2 : IF NOT (hari ini hujan) THEN besok terang (CF = 0.5)**

**Rule3 : IF hari ini hujan AND hujan gerimis THEN besok terang (CF = 0.6)**

**Rule4 : IF hari ini hujan AND hujan gerimis AND suhu dingin THEN besok terang (CF = 0.7)**

**Rule5 : IF NOT (hari ini hujan) AND suhu hangat THEN besok hujan (CF = 0.65)**

**Rule6 : IF NOT (hari ini hujan) AND suhu hangat AND langit mendung THEN besok hujan (CF = 0.56)**

**Catatan : NOT (hari ini hujan) artinya sama dengan “hari ini terang”**

# Dialog antara sistem pakar dan pengguna

- ▶ Untuk mendapatkan fakta-fakta dari pengguna, berdasarkan Rule ke 1, pertama kali sistem pakar menanyakan kepada pengguna pertanyaan berikut :

*Sistem pakar : "Bagaimana cuaca hari ini (isi kepastian CF dengan angka  $[0,1]$  ?*

*Pengguna : "Hujan,  $CF = 1$ "*

$CF_1$  (Besok hujan, hari ini hujan) =  $1 \times 0.5 = 0.5$

Basis data (Fakta-fakta) :

hari ini hujan (Evidence  $CF = 1$ )

Besok Hujan (Hypothesis  $CF_1 = 0.5$ )

~~Rule 1 : IF hari ini hujan THEN besok hujan ( $CF = 0.5$ )~~

Berdasarkan fakta “hari ini hujan”, Rule ke 2 tidak di eksekusi.  
Berdasarkan Rule ke 3 sistem pakar menanyakan pada pengguna pertanyaan berikut :

*Sistem pakar : “Hujan deras atau gerimis (isi kepastian CF dengan angka [0,1] ?*

*Pengguna : “Gerimis, CF = 0.8”*

$$\begin{aligned} \text{CF3}(\text{Besok terang, hari ini hujan} \cap \text{gerimis}) &= \min [1; 0.8] \times 0.6 \\ &= 0.8 \times 0.6 \\ &= 0.48 \end{aligned}$$

Basis data (Fakta-fakta) :

hari ini hujan (Evidence CF =1)

Besok Hujan (Hypothesis CF1 = 0.5)

Hujan gerimis (Evidence CF =0.8)

besok terang (Hypothesis CF3 = 0.48)

Rule 3 : ~~IF hari ini hujan AND hujan gerimis THEN besok terang (CF = 0.6)~~

Berdasarkan Rule ke 4, sistem pakar menanyakan pada pengguna pertanyaan berikut :

*Sistem pakar : "Bagaimana suhu hari ini (isi kepastian CF dengan angka [0,1] ?*

*Pengguna : "Dingin, CF = 0.9"*

$$\begin{aligned}\text{CF4 (Besok terang, hujan} \cap \text{gerimis} \cap \text{dingin)} &= \min [1;0.8;0.9] \times 0.7 \\ &= 0.8 \times 0.7 \\ &= 0.56\end{aligned}$$

Basis data (Fakta-fakta) :

hari ini hujan (Evidence CF =1)

Besok Hujan (Hypothesis CF1 = 0.5)

Hujan gerimis (Evidence CF =0.8)

Besok terang (Hypothesis CF3 = 0.48)

Suhu dingin (Evidence CF = 0.9)

Besok Terang (Hypothesis CF4 = 0.56)

~~Rule 3 IF hari ini hujan AND hujan gerimis AND suhu dingin THEN  
besok terang (CF = 0.7)~~

Rule3 dan Rule4 mempunyai hipotesis yang sama, yaitu besok terang. Perhitungan CF gabungan kedua rule tersebut adalah :

$$\begin{aligned} CF &= CF3 + CF4 (1 - CF3) \\ &= 0.48 + 0.56 (1 - 0.48) \\ &= 0.77 \end{aligned}$$

Basis data (Fakta-fakta) :

hari ini hujan (Evidence CF =1)

Besok Hujan (Hypothesis CF1 = 0.5)

Hujan gerimis (Evidence CF =0.8)

Besok terang (Hypothesis CF3 = 0.77)

Suhu dingin (Evidence CF = 0.9)

Sampai di sini proses berhenti karena sudah tidak ada lagi rule yang bisa di bangkitkan. Ada dua hipotesis dalam database, yaitu besok terang (CF = 0.77) dan besok hujan (CF = 0.5)

**Kesimpulan** yang di berikan sistem pakar kepada pengguna adalah yang mempunyai CF terbesar yaitu besok hari akan terang dengan tingkat kepastian 77%

## Kasus : sistem pakar diagnosis penyakit pada anak

Rule 1 : IF demam AND batuk AND napas berbunyi THEN infeksi saluran napas (CF = 0.25)

Rule 2 : IF demam AND batuk AND napas sangat cepat THEN infeksi dada (CF = 0.45)

Rule 3 : IF demam AND batuk AND napas megap-megap THEN influenza (CF = 0.74)

### Dialog antara sistem pakar dan pengguna

Sistem pakar : “Apakah Anda mengalami demam (isi kepastian CF dengan angka [0,1] ?

Pengguna : “Ya, CF = 0.7”

Sistem pakar : “Apakah Anda mengalami batuk (isi kepastian CF dengan angka [0,1] ?

Pengguna : “Ya, CF = 0.3”

Sistem pakar : “Apakah napas anda berbunyi (isi kepastian CF dengan angka [0,1] ?

Pengguna : “Ya, CF = 0.6”

Sistem pakar : “Apakah anda bernapas sangat cepat (isi kepastian CF dengan angka [0,1]?)

Pengguna : “Ya, CF = 0.2”

Sistem pakar : “Apakah napas anda megap-megap (isi kepastian CF dengan angka [0,1]?)

Pengguna : “tidak”