

BAB V

REPRESENTASI PENGETAHUAN

A. Pengenalan Representasi Pengetahuan

Dalam penyelesaian masalah tentu membutuhkan pengetahuan pengetahuan yang cukup. Selain itu sistem harus bisa untuk menalar. Representasi pengetahuan dan kemampuan untuk melakukan penalaran dalam bidang kecerdasan buatan merupakan hal yang penting. Meskipun suatu sistem memiliki banyak pengetahua, namun jika tidak memiliki kemampuan untuk menalar akan percuma saja. Sebaliknya jika sistem memiliki kemampuan yang handal untuk menalar tapi basis pengetahuan yang dimilikinya tidak cukup, maka solusi yang diperolehnya menjadi tidak maksimal.

Banyak cara untuk menyatakan representasi pengetahuan dan setiap cara mempunyai kelebihan dan kelemahan masing-masing.

Adapun cara untuk representasi pengetahuan adalah:

- Kalkulus Predikat
- Logika
- Pohon
- Frame
- Script.
- Semantic Network
-

Dalam representasi pengetahuan ada dua hal yang harus diperhatikan, yaitu:

1. Fakta

Objek yang akan direpresentasikan. Ini menerangkan keadaan tentang benda yang ada dalam domain masalah. Fakta dapat berupa keterangan/kalimat dalam bahasa alami, logika atau benda.

2. Formula

Bentuk representasi yang dapat dimanipulasi dalam proses pemecahan soal. Bentuk ini harus dapat melukiskan hubungan antara komponen-komponen domain masalah.

B. Logika

Logika ini adalah bentuk representasi pengetahuan yang paling tua. Pada dasarnya proses logika adalah proses membentuk kesimpulan atau menarik inferensi berdasarkan fakta yang ada (gambar 5.1). Input dari proses logika berupa premis atau fakta yang diakui kebenarannya sehingga dengan melakukan penalaran pada proses logika dapat dibentuk suatu inferensi atau kesimpulan yang benar pula.



Ada dua penalaran untuk mendapatkan konklusi:

1. Penalaran Deduksi.

Penalaran dimulai dari prinsip umum untuk mendapatkan konklusi yang lebih khusus.

2. Penalaran Induktif

Penalaran dimulai dari fakta-fakta khusus untuk mendapatkan kesimpulan umum.

Pada representasi logika ini ada dua metode yaitu:

1. Logika Proposisi
2. Kalkulus Predikat

1. Logika preposisi

Proposisi adalah pernyataan yang dapat bernilai benar (B) atau salah (S). Simbol-simbol seperti P dan Q menunjukkan proposisi. Dua atau lebih proposisi dapat digabungkan dengan menggunakan operator logika yaitu:

- Konjungsi: \wedge (AND)
- Disjungsi: \vee (OR)
- Negasi : \neg (NOT)
- Implikasi : \rightarrow (IF Then)
- Ekuivalensi : \Leftrightarrow

Untuk nilai kebenaran dari semua operator logika dapat dilihat di table kebenaran (*truth table*) berikut:

P	Q	P AND Q	P OR Q	$P \rightarrow Q$	$P \Leftrightarrow Q$
B	B	B	B	B	B
B	S	S	B	S	S
S	B	S	B	B	S
S	S	S	S	B	B

2. Pembuktian Teorema (RESOLUSI)

Untuk melakukan inferensi pada logika proposisi dapat dilakukan dengan menggunakan resolusi. Resolusi adalah aturan untuk melakukan inferensi secara efisien dalam suatu bentuk khusus yang dikenal dengan *Conjunctive Normal Form* (CNF). Bentuk CNF ini mempunyai ciri-ciri sebagai berikut:

- Setiap kalimat merupakan disjungsi literal
- Semua kalimat terkonjungsi secara implicit.

Kalimat yang ditulis dalam logika proposisi dapat dikonversi ke bentuk CNF. Kita perlu menghapus operator-operator selain OR tanpa harus mengubah arti dari kalimat tersebut.

Untuk mengubah suatu kalimat ke dalam bentuk CNF, dilakukan langkah berikut:

- Hilangkan implikasi dan ekuivalensi
 - o $x \rightarrow y$ menjadi $\neg x \vee y$
 - o $x \Leftrightarrow y$ menjadi $(\neg x \vee y) \wedge (\neg y \vee x)$
- Kurangi lingkup semua negasi menjadi satu negasi saja
 - o $\neg(\neg x)$ menjadi x
 - o $\neg(x \vee y)$ menjadi $(\neg x \wedge \neg y)$
 - o $\neg(x \wedge y)$ menjadi $(\neg x \vee \neg y)$

- Gunakan aturan asosiatif dan distributive untuk mengkonversi menjadi conjunctive of disjunctive
 - o Asosiatif: $(A \vee B) \vee C = A \vee (B \vee C)$
 - o Distributif: $(A \wedge B) \vee C = (A \vee C) \wedge (B \vee C)$
- Buat satu kalimat terpisah untuk tiap-tiap konjungsi

Pada logika proposisi, prosedur untuk membuktikan proposisi P dengan beberapa aksioma F yang telah diketahui, dengan menggunakan resolusi dapat dilakukan melalui algoritma sebagai berikut:

1. Konversikan semua proposes F ke bentuk CNF
2. Negasikan P, dan konversikan hasil negasi ke bentuk klausa. Tambahkan ke himpunan klausa yang telah ada pada langkah 1.
3. Kerjakan hingga terjadi “kontradiksi” atau proses tidak mengalami kemajuan:
 - a. Seleksi 2 klausa sebagai klausa parent
 - b. Bandingkan (*resolve*) secara bersama-sama. Klausa hasil resolve tersebut dinamakan *resolvent*. Jika ada pasangan literal L dan $\neg L$, eliminir dari *resolvent*.
 - c. *resolvent*. Berupa klausa kosong, maka ditemukan kontradiksi. Jika tidak, tambahkan ke himpunan klausa yang telah ada..

Contoh:

Diketahui basis pengetahuan (fakta-fakta yang bernilai benar) sebagai berikut:

1. P
2. $(P \wedge Q) \rightarrow R$
3. $(S \vee T) \rightarrow Q$
4. T

Buktikan kebenaran dari R !!

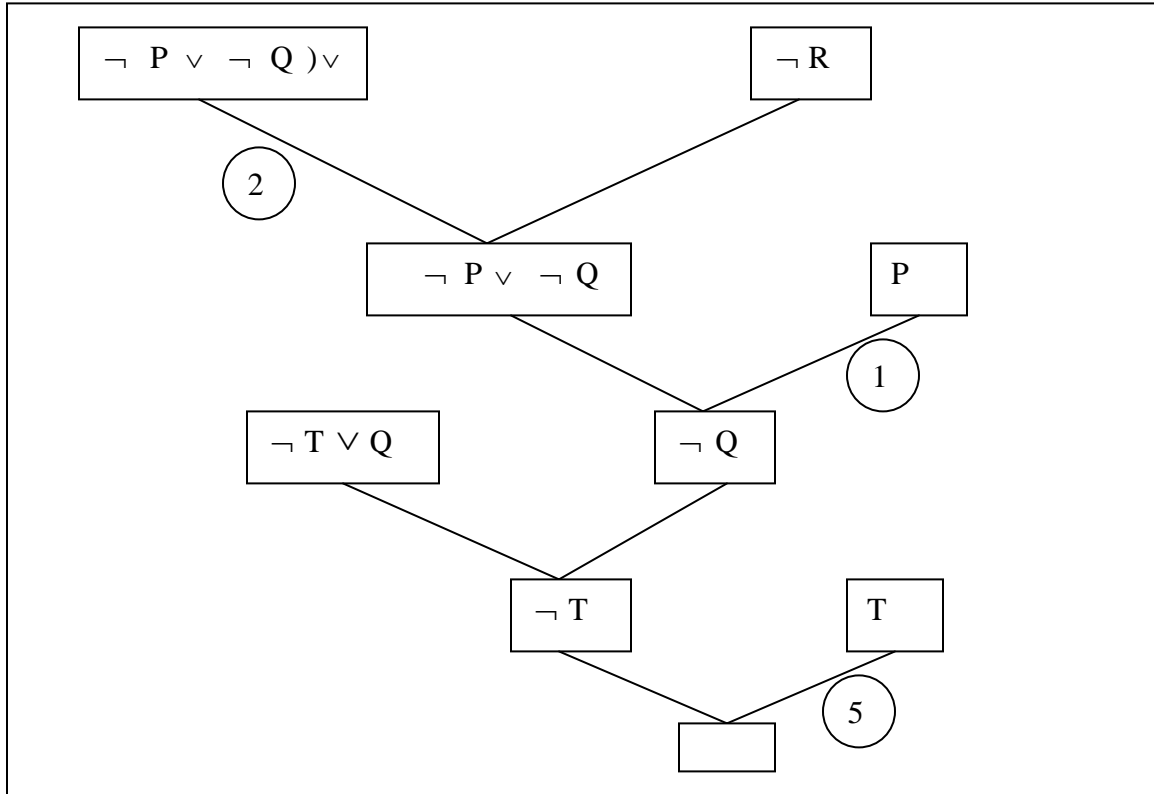
Apabila akan dibuktikan kebenaran R dengan menggunakan resolusi maka kita harus ubah keempat fakta di atas menjadi bentuk CNF. Konversi ke CNF dapat dilakukan sebagai berikut:

Kalimat	Langkah-Langkah	CNF
1. P	Sudah merupakan bentuk CNF	P
2. $(P \wedge Q) \rightarrow R$	<ul style="list-style-type: none"> - Menghilangkan implikasi $\neg (P \wedge Q) \vee R$ - Mengurangi lingkup negasi $(\neg P \vee \neg Q) \vee R$ - Gunakan asosiatif $\neg P \vee \neg Q \vee R$ 	$\neg P \vee \neg Q \vee R$
3. $(S \vee T) \rightarrow Q$	<ul style="list-style-type: none"> - Menghilangkan implikasi $\neg (S \vee T) \vee Q$ - Mengurangi lingkup negasi $(\neg S \wedge \neg T) \vee Q$ - Gunakan distributif $(\neg S \vee Q) \wedge (\neg T \vee Q)$ 	$\neg S \vee Q$ $\neg T \vee Q$
4. T	Sudah dalam bentuk CNF	T

Kemudian tambahkan kontradiksi pada tujuannya, R menjadi $\neg R$ sehingga fakta-fakta (dalam bentuk CNF) dapat disusun menjadi:

1. P
2. $\neg P \vee \neg Q) \vee R$
3. $\neg S \vee Q$
4. $\neg T \vee Q$
5. T
6. $\neg R$

Dengan demikian resolusi dapat dilakukan untuk membuktikan R sebagaimana terlihat pada gambar 5.1



Gambar 5.1 resolusi pada logika preposisi

Contoh apabila diterapkan dalam kalimat:

- P : Agung anak yang cerdas
- Q : Agung rajin belajar
- R : Andi akan menjadi juara kelas
- S : Agung makannya banyak
- T : Agung istirahatnya cukup

Kalimat yang terbentuk:

- Agung anak yang cerdas
- Jika Agung anak yang cerdas dan Agung rajin belajar, maka Agung akan menjadi juara kelas.

- Jika Agung makannya banyak atau Agung istirahatnya cukup, maka Agung rajin belajar
- Agung istirahatnya cukup

Setelah dilakukan konversi ke CNF, didapat:

- Fakta ke-2: Agung tidak cerdas atau Agung tidak rajin belajar atau Agung akan menjadi juara kelas
- Fakta ke-3: Agung tidak makan banyak atau Agung rajin belajar.
- Fakta ke4: agung tidak cukup istirahat atau Agung rajin belajar.

3. Kalkulus Predikat

Kalkulus Predikat adalah satu formula yang terdiri dari predikat, variable, konstanta, atau fungsi.. Kalkulus predikat digunakan untuk merepresentasikan hal-hal yang tidak dapat direpresentasikan dengan menggunakan logika proposisi. Pada logika predikat (kalkulus predikat) kita dapat merepresentasikan fakta-fakta sebagai pernyataan yang disebut dengan *wff* (*well formed formula*).

Contoh :

1. WARNA (RUMAH, MERAH):

Predikat ini menggambarkan warna rumah merah, dimana WARNA adalah predikat, RUMAH dan MERAH adalah sautu konstanta.

2. WARNA (x,MERAH) :

x adalah variable yang menyatakan sembarang benda yang berwarna merah.

3. WARNA (x,y) :

Predikat ini menyatakan suatu sifat warna antara variable x dan y.

Biasanya predikat dan konstanta ditulis dengan huruf besar, sedangkan variabel atau fungsi ditulis dalam huruf kecil. Formula-formula ini mempunyai beberapa kaidah yang menghubungkan satu dengan yang lain

Kalkulus predikat bersifat rangkaian (conective) seperti :

1. Konjungtif (\wedge)

Fakta : Amin tinggal dirumah yang berwarna kuning

Formula : TINGGAL (AMIN,RUMAH) \wedge WARNA (RUMAH, KUNING)

2. Disjungtif (\vee)

Fakta : Amin bisa main biola dan piano

Formula : MAIN (AMIN,PIANO) \vee MAIN (AMIN,BIOLA)

3. Negasi (-)

Fakta : Amin tidak bisa main biola

Formula : - MAIN(AMIN,BIOLA).

4. Implikasi (\rightarrow)

Formula sebelah kiri dari \rightarrow adalh premis (*antecedent*) dan sebelah kanan adalah konklusi (*consequence*).

Fakta : Amin mempunyai mobil biru.

Fakta tersebut mengandung arti : bila Amin mempunyai mobil maka mobil itu berwarna biru.

Formula : PUNYA (AMIN, MOBIL_A) \rightarrow WARNA (MOBIL_A, BIRU)

5. Kuantifier (penghitung):

Kuantufier adalah satu symbol dalam satu formula yang membenarkan formula itu dalam satu domain. Misal :

Fakta : amin punya mobil

Formula : PUNYA (x,y)

Fakta tersebut dapat ditulis PUNYA (AMIN, MOBIL) dimana AMIN dan MOBIL adalah kuantifier dari variable x dan y. Kuantifier ini ada beberapa tipe yaitu:

- Kuantifier universal dimana semua constant membenarkan formula itu .

Fakta : Semua kucing mempunyai empat kaki.

Formula : $(\forall x) \text{BINATANG}(x, \text{KUCING}) \rightarrow \text{KAKI}(x, 4)$

- Kuantifier yang berlaku untuk suatu kadaan saja.

Fakta : Ada satu kucing yang berkaki tiga

Formula : $(\exists x) \text{BINATANG}(x, \text{KUCING}) \rightarrow \text{KAKI}(x, 3)$

Contoh Kasus :

Misalkan ada pernyataan-pernyataan sebagai berikut:

1. Karto adalah seorang laki-laki
2. Karto adalah orang jawa.
3. Karto lahir pada tahun 1840
4. Setiap laki-laki pasti mati.
5. Semua orang jawa mati pada saat Krakatau meletus tahun 1883.
6. Setiap orang pasti mati setelah hidup lebih dari 150 tahun.
7. Sekarang tahun 2003
8. Mati berarti tidak hidup.
9. Jika seseorang mati, maka beberapa waktu kemudian ia pasti dinyatakan mati.

Diubah menjadi predikat:

1. Laki2(Karto)
2. Jawa (Karto)
3. lahir (Karto, 1840)
4. $\forall x : \text{laki2}(x) \rightarrow \text{pastimati}(x)$.
5. $\text{meletus}(\text{Krakatau}, 1883) \wedge \forall x : [\text{Jawa}(x) \rightarrow \text{mati}(x, 1883)]$.

Dapat dipecah menjadi:

5a. meletus (Krakatau, 1883)

5b. $\forall x: [\text{Jawa}(x) \rightarrow \text{mati}(x, 1883)]$.

6. $\forall x: \forall t1: \forall t2: \text{pastimati}(x) \wedge \text{lahir}(x, t1) \wedge \text{lahir}(x, t2) \wedge \text{lb}(t2-t1, 150) \rightarrow \text{mati}(x, t2)$

7. sekarang = 2003

8. $\forall x: \forall t: [\text{mati}(x, t) \rightarrow \neg \text{hidup}(x, t) \wedge [\neg \text{hidup}(x, t) \rightarrow \text{mati}(x, t)]]$

9. $\forall x: \forall t1: \forall t2: \text{mati}(x, t1) \wedge \text{lb}(t2-t1) \rightarrow \text{mati}(x, t2)$

Andaikan akan menjawab pertanyaan: “ Apakah Karto masih hidup sekarang??”

Maka dengan menggunakan fakta atau aturan yang ada, kita dapat membuktikan bahwa:

$\neg \text{hidup}(\text{Karto}, \text{sekarang})$

Nilai NIL pada akhir pembuktian menunjukkan bahwa pembuktian sukses.

$\neg \text{hidup}(\text{Karto}, \text{sekarang})$

↑ (8, substitusi)

$\text{mati}(\text{karto}, \text{sekarang})$

↑ (9, substitusi)

$\text{mati}(\text{KArjo}, t1) \wedge \text{lb}(\text{sekarang}, t1)$

↑ (5b, substitusi)

$\text{Jawa}(\text{Karto}) \wedge \text{lb}(\text{sekarang}, 1883)$

↑ (2)

$\text{lb}(\text{sekarang}, 1883)$

↑ (7, substitusi)

$\text{lb}(2003, 1883)$

↑ (menghitung lb)

NIL

Cara lain untuk membuktikan bahwa Kato sudah mati adalah:

\neg hidup (Karto, sekarang)

↑ (8, substitusi)

mati (karto, sekarang)

↑ (6, substitusi)

pastimati (Karto) \wedge lahir(Karto, t1) \wedge lb (sekarang-t1, 150)

↑ (1)

lahir (Karto, t1) \wedge lb (sekarang-1840,150)

↑ (3, substitusi)

lb (sekarang-1840, 150)

↑ (7, substitusi)

lb (2003-1840,150)

↑ (menghitung minus)

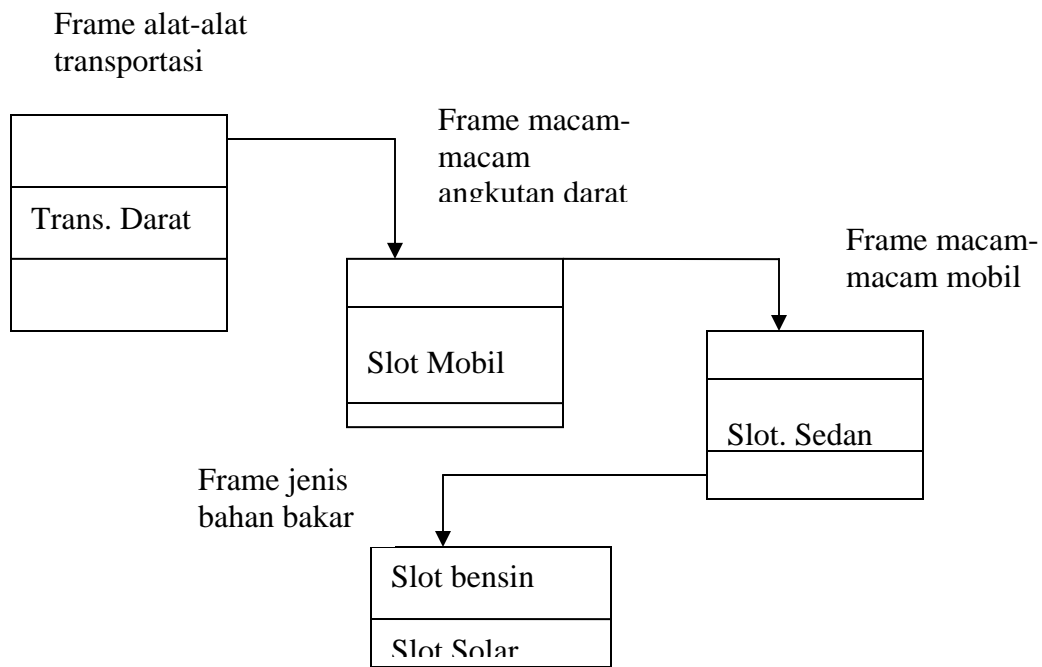
lb (163,150)

↑ (menghitung lb)



C. Bingkai (Frame)

Frame merupakan kumpulan pengetahuan tentang suatu objek tertentu, peristiwa, lokasi, situasi dll. Frame memiliki slot yang menggambarkan rincian (atribut) dan karakteristik objek. Frame biasanya digunakan untuk merepresentasikan pengetahuan yang berdasarkan karakteristik yang sudah dikenal yang merupakan pengalaman-pengalaman. Dengan menggunakan frame, sangat mudah untuk membuat inferensi tentang objek karena menyediakan basis pengetahuan yang berasal dari pengalaman.



Gambar 5.2 Frame

Gambar di atas menunjukkan frame alat-alat transportasi. Frame memiliki 3 slot, yaitu alat-alat transportasi di darat, udara, dan laut. Ada beberapa slot yang bernilai tetap, dan tidak tetap (procedural). Slot yang bernilai tetap misal jumlah roda pada sedan (4). Jenis slot lain yang bersifat procedural artinya slot yang memungkinkan penambahan informasi baru yang bisa ditambahkan pada aturan IF. Misal informasi tentang kecepatan perjalanan, pengisian tangki bahan bakar, atau pemakaian bahan bakar tiap km.

D. Naskah (Script)

Script adalah skema representasi pengetahuan yang sama dengan frame. Perbedaannya frame menggambarkan objek sedangkan script menggambarkan urutan peristiwa. Dalam menggambarkan urutan peristiwa, skrip menggunakan slot yang berisi informasi tentang orang, objek, dan tindakan yang terjadi dalam suatu peristiwa.

Elemen-elemen script meliputi:

1. Kondisi input, yaitu kondisi yang harus dipenuhi ebelum terjadi suatu peristiwa,
2. Track, yaitu variasi yang mungkin terjadi dalam suatu script
3. Prop, berisi objek-objek pendukung yang digunkana selama peristiwa terjadi.
4. Role, yaitu peran yang dimainkan oleh seseorang dalam peristiwa.
5. Scene, yaitu adegan yang dimainkan .
6. Hasil, yaitu kondisi yang ada setelah urutan peristiwa yang terjadi.
- 7.

Contoh:

Jalur (track) : Ujian tertulis mata kuliah Kecerdasan Buatan

Role (peran) : Mahasiswa, Pengawas.

Prop (pendukung) : lembar soal, lembar jawab, presensi, pena dll

Kondisi input : Mahasiswa terdaftar untuk mengikuti ujian

Adegan (scene) - 1 : Persiapan pengawas

- Pengawas menyiapkan lembar soal.
- Pengawas menyiapkan lembar jawab
- Pengawas menyiapkan lembara presensi

Adegan – 2 : mahasiswa masuk ruangan

- Pengawas mempersilahkan mahasiswa untuk masuk ruangan
- Pengawas membagikan lembar soal
- Pengawas membagikan lembar jawab
- Pngawas memimpin doa

Adegan – 3 : Mahasiswa mengerjakan soal ujian

- Mahasiswa menulis identitas di lembar jawab
- Mahasiswa menandatangani lembar jawab
- Mahasiswa mengerjakan soal
- Mahasiswa mengecek jawban

Adegan – 4 : Mahasiswa telah selesai ujian

- Pengawas mempersilahkan mahasiswa keluar ruangan
- Mahasiswa mengumpulkan kembali lembar jawab
- Mahasiswa keluar ruangan

Adegan – 5 : Pengawas mengemasi lembar jawab

- Pengawas mengurutkan lembar jawab
- Pengawas mengecek lembar jawab dan mengurutkan
- Pengawas meninggalkan ruangan

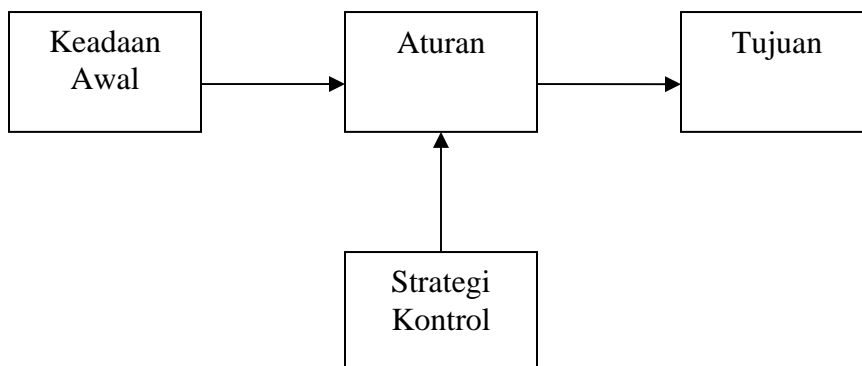
Hasil :

- Mahasiswa merasa lega dan senang sudah ujian
- Mahasiswa merasa kecewa
- Mahasiswa pusing
- Mahasiswa sangat bersyukur

E. Sistem Produksi (Production Rule)

Sistem Produksi secara umum terdiri dari komponen-komponen:

1. Ruang keadaan, yang berisi keadaan awal (Initial state), tujuan (Goal State) dan kumpulan aturan (production rule) yang digunakan untuk mencapai tujuan.
2. Strategi Kontrol, yang berguna untuk mengarahkan bagaimana proses pencarian akan berlangsung dan mengendalikan arah eksplorasi.



Gambar 5.3 sistem produksi

Sistem produksi ini sangat [opuler dan banyak digunakan. Reprsentasi pengetahuna dengan sistem produksi pada dasarnya berupa aplikasi aturan yang berupa:

1. Antecedent, yaitu bagian yang mengekspresikan situasi atau premis (pernyataan berawalan IF)
2. Konsekuen, yaitu bagian yang menyatakan suatu tindakan tertentu atau konklusi yang diterapkan jika situasi atau premis bernilai benar (berawalan THEN).

Apabila pengetahuan direpresentasikan dengan aturan, maka ada 2 penalaran yaitu:

1. Forward Reasoning (Penalaran Maju).

Pelacakan dimulai dari keadaan awal (informasi atau fakta yang ada) dan kemudian dicoba untuk mencocokkan **dengan tujuan diharapkan**

2. Backward reasoning (Penalaran Mundur).

Dimulai dari tujuan atau hipotesa, baru dicocokkan dengan keadaan awal atau fakta yang ada.

Ada beberapa factor yang mempengaruhi pemilihan *Forward Reasoning* atau *Backward reasoning* dalam memilih metode penalaran :

- Banyaknya keadaan awal dan tujuan. Jika jumlah keadaan awal lebih kecil dari pada tujuan, maka digunakan Forward Reasoning dan sebaliknya.
- Rata-rata jumlah node yang dapat diraih secara langsung dari suatu node. Lebih baik dipilih yang jumlah node tiap cabangnya lebih sedikit.
- Apakah program butuh menanyai user untuk melakukan justifikasi terhadap proses penalaran ? Jika iya, maka dipilih yang memudahkan user.
- Bentuk kejadian yang akan memicu penyelesain masalah. Jika kejadian itu berupa fakta baru, maka dipilih penalaran Forward Reasoning. tapi jika kejadian berupa queri, maka lebih baik digunakan Backward reasoning

