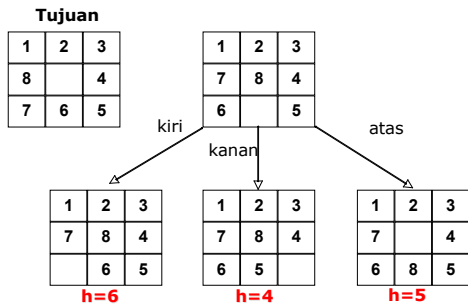


Materi: 4, 5, 6

### HEURISTIC SEARCH

- Alasan pemakaian: Pencarian buta tidak selalu dapat diterapkan dengan baik
  - Waktu akses lebih cepat
  - Tidak memerlukan space yang besar
  - Bisa menyelesaikan permasalahan yang lebih besar
  - Menggunakan fungsi heuristic yang menghitung biaya perkiraan (estimasi dari suatu simpul tertentu menuju ke simpul tujuan)
  - Aplikasi: Google, deep blue chess machine
- Contoh (pada masalah 8 puzzle)



- Langkah Awal hanya 3 operator yang bisa digunakan (Ubin kosong digeser ke kiri, ke kanan dan ke atas)
- Jika menggunakan pencarian buta, tidak perlu mengetahui operasi apa yang akan dikerjakan (sembarang)
- Pada pencarian heuristic perlu diberikan informasi khusus dalam domain tersebut:

Untuk jumlah ubin yang menempati posisi yang benar jumlah yang lebih tinggi adalah yang lebih diharapkan (lebih baik)

Untuk jumlah ubin yang menempati posisi yang salah jumlah yang lebih kecil adalah yang diharapkan (lebih baik). Menghitung total gerakan yang diperlukan untuk mencapai tujuan jumlah yang lebih kecil adalah yang diharapkan (lebih baik).

#### Jenis Pencarian Heuristik

- General and Test
- Hill Climbing
- Best First Search

### GENERATE AND TEST

- Teknik yang paling mudah dibandingkan Teknik search yang lain
- Relative lebih lama dalam mendapatkan Solusi
- Pencarian yang mengkombinasikan metode Depth-First Search dengan Backtracking (pelacakan balik/mundur) bergerak ke belakang menuju pada suatu keadaan awal
- prosedur Generate-and-Test, fungsi test hanya ditanggapi dengan Ya atau Tidak

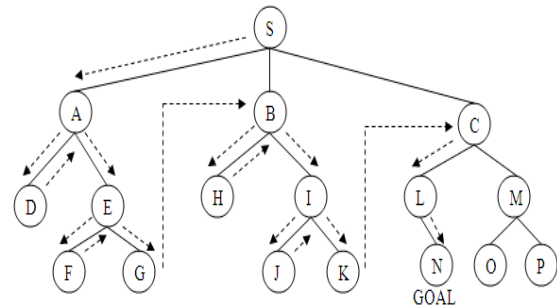
#### Algoritma Generate and Test

1. Bentuk solusi yang mungkin.  
Untuk beberapa masalah, ini berarti membentuk poin terpisah dari area permasalahan. Pada masalah lain, ini berarti membentuk jalur dari stata awal.
2. Lakukan test untuk melihat apakah poin yang ditemui adalah solusi dengan membandingkan poin yang dipilih atau poin terakhir dari jalur yang dipilih dengan kumpulan stata tujuan
3. Jika solusi sudah ditemukan, quit. Jika belum kembali ke langkah 1.

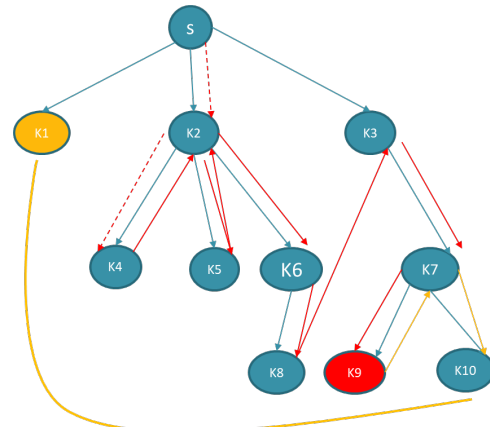
### 3. Selesai, jika solusi ditemukan

Kebaikan dan keburukan Generate and Test Jika penurunan solusi yang mungkin dilakukan secara sistematis, maka procedure diatas akan dapat menemukan solusi suatu saat, jika memang ada. Tapi sayangnya jika ruang permasalahan sangat luas maka saat ditemukannya solusi akan menjadi sangat lama.

Cara terbaik menerapkan generate-and-test yang sistematis adalah pada tree dari depth-first search dengan backtracking, yaitu kembali ke stata sebelumnya bila ditemui stata yg sudah pernah di test atau memodifikasi prosedurnya untuk menelusuri stata pada bentuk graph.



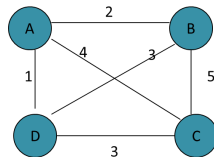
Solusi: S-A-D-A-E-F-E-G-B-H-B-I-J-I-K-C-L-N



### Cara Cepat Generate And Test:

- Jelajah dari fakta
  - o Apabila fakta dari tengah, telusuri ke dalam lalu lanjut ke node ke kanan (tidak ke kiri walaupun tujuan di kiri)
- Telusuri dari kiri ke kanan setelah node ke dalam
- Apabila ada cabang lakukan backtracking
- Apabila sudah node paling kanan, Kembali ke kiri

### Kasus TSP



Sebuah rute yang harus dilewati seorang sales dimana sales tersebut harus melewati setiap kota tepat sekali. Terdapat 4 kota, dengan jarak masing-masing kota AB=2, AC=4, AD=1, BC=5, BD=3, CD=3. Tujuannya adalah mencari jarak terpendek bagi sales untuk mengunjungi semua kota sekali. Penyelesaian menggunakan generate-test adalah dengan membangkitkan solusi-solusi yang mungkin ada sesuai permasalahan yang dihadapi oleh sales tersebut.

Kombinasi abjad sebagai solusi yang mungkin adalah  $n! = 4! = 24$ .

Tujuannya adalah mencari solusi dengan panjang terpendek.

No pencarian	Lintasan	Panjang Lintasan	Lintasan yang dipilih	Panjang Lintasan
1	ABCD	10	ABCD	10
2	ABDC	8	ABDC	8
3	ACBD	12	ABDC	8
4	ACDB	10	ABDC	8
5	ADCB	9	ABDC	8
6	ADCB	9	ABDC	8
7	BACD	9	ABDC	8
8	BADC	6	BADC	6
9	BCAD	10	BADC	6
10	BCDA	9	BADC	6
11	BDAC	8	BADC	6
12	BDCA	10	BADC	6
13	CABD	9	BADC	6

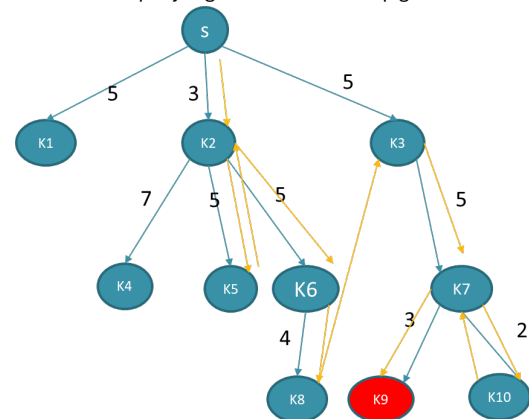
14	CADB	8	BADC	6
15	CBAD	8	BADC	6
16	CBDA	9	BADC	6
17	CDAB	6	BADC/CDAB	6
18	CDBA	8	BADC/CDAB	6
19	DABC	8	BADC/CDAB	6
20	DACB	10	BADC/CDAB	6
21	DBAC	9	BADC/CDAB	6
22	DBCA	12	BADC/CDAB	6
23	DCAB	9	BADC/CDAB	6
24	DCBA	10	BADC/CDAB	6

Dari tabel diatas, solusi pertama yang dibangkitkan adalah ABCD = 10, solusi kedua ABDC=8. Ternyata solusi kedua menghasilkan jarak yang lebih pendek sehingga dipilih lintasan ABDC=8. Lakukan untuk langkah selanjutnya. Pada tabel didapat solusi terpendek adalah BADC atau CDBA.

Kelemahan dari teknik ini perlu dibangkitkan semua kemungkinan yang ada sehingga apabila ditambahkan satu kota untuk permasalahan TSP ini diatas 5 kota. Maka akan diperlukan 120 kombinasi lintasan, kecuali diberikan kondisi tertentu misalnya kota awal bagi sales telah ditentukan.

### HILL CLIMBING

- Metode pelacakan mengkombinasikan pelacakan Generate and Test dengan Back Tracking
- Memilih node yang memiliki nilai terbaik (fungsi heuristic)
- pengembangan dari teknik Generate-and-Test, dengan penambahan adanya umpan balik dari prosedur test yang sudah digunakan untuk membantu memilih arah mana yang harus ditelusuri pada setiap area search.
- fungsi test ditambahkan dengan fungsi heuristic atau fungsi objectif yang memungkinkan perkiraan seberapa dekat simpul yang ditelusuri terhadap goal state.



Contoh : Hill Climbing untuk pencarian rute terpendek, maka yang dipilih adalah **nilai terkecil**

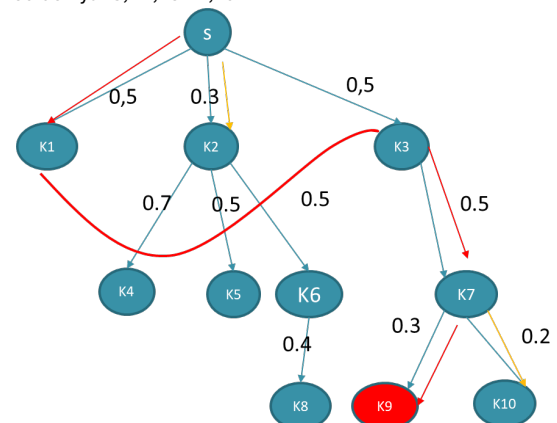
dengan fakta : S goal K9

solusinya: S,K2, K5,K2,K6,K8,K3,K7,K10,K7,K9

Contoh : Hill Climbing untuk pencarian penyakit berdasarkan nilai peluang, maka yang dipilih adalah **nilai terbesar**

dengan fakta : S goal K9

solusinya: S,K1,k3.k7,k9

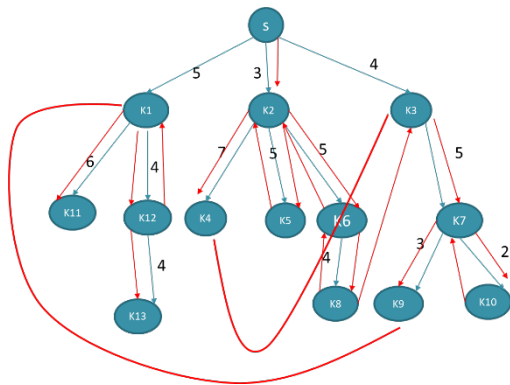


Contoh : Hill Climbing untuk pencarian rute terpendek, maka yang dipilih adalah **nilai terkecil**

dengan fakta : S goal tidak ditentukan , maka pelacakan berhenti pada node terakhir dari level terdalam dengan cara semua node dikunjungi dengan memprioritaskan nilai terbaik.

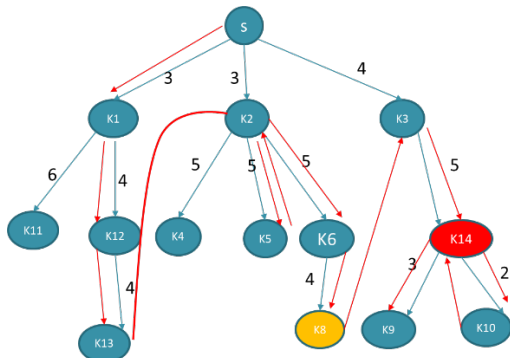
solusinya:

S, k2,k5,k2,k6,k8,k6,k2,k4,k3,k8,k10,k8,k9,k1,k12,k13,k12,k1,k11



Contoh : Hill Climbing untuk pencarian rute terpendek, maka yang dipilih adalah nilai terkecil dengan fakta : S goal tidak ditentukan , maka pelacakan berhenti pada node terakhir dari level terdalam dengan cara semua node dikunjungi dengan memprioritaskan nilai terbaik.

solusinya: S, k2, k5, k2, k6, k8, k6, k2, k4, k3, k8, k10, k8, k9, k1, k12, k13, k12, k1, k11

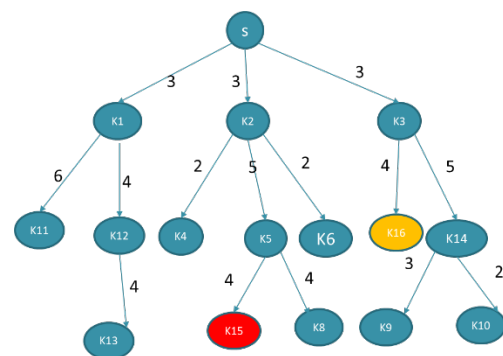


Contoh : Hill Climbing untuk pencarian rute terpendek, maka yang dipilih adalah nilai terkecil dengan fakta : faktanya S dengan goal di K8

solusinya: S, k1, k12, k13, k2, k5, k2, k6, k8

faktanya S dengan goal di K14

solusinya: s, k1, k12, k13, k2, k4, k2, k5, k2, k6, k8, k3, k14



#### Perbandingan Hill Climbing dan Generate And Test

Selesaikan kasus pelacakan berikut dengan Hill Climbing

Kasus 1: 1 jika Goal di K15 berapa solusinya..?

S, k1, k12, k13, k2, k4, k2, k6, k2, k5, k8, k5, k15, k3, k16

kasus 2: jika Goal di K6..berapa solusinya..?

S, k1, k12, k13, k2, k4, k2, k6, k2, k5, k15, k8, k5, k15

Selesaikan kasus pelacakan berikut dengan generate and test

Kasus 1: jika Goal di K15 berapa solusinya..?

S, k1, k12, k13, k2, k4, k2, k6, k2, k5, k15

kasus 2: dengan metoda Breadth first search dengan goal di K16

S, k1, k2, k3, k11, k12, k4, k5, k6, k16

#### Cara cepat Hill Climbing:

- Rute terpendek: terkecil
- Nilai peluang: terbesar
- Jelajahi dari fakta
  - o Telusuri dari berat terbesar/terkecil
  - o Apabila ada berat sama, telusuri dari kiri ke kanan setelah node ke dalam
- Apabila ada cabang lakukan backtracking

#### Algoritma simple hill climbing

1. Evaluasi initial state. Jika ini goal state maka return dan keluar. Jika bukan maka lanjutkan dengan initial state sebagai current state.
2. Ulangi langkah berikut sampai menemukan solusi atau sampai tidak ada lagi operator yang dapat digunakan pada current state
  - a. Pilih operator yang belum digunakan pada current state dan gunakan untuk menghasilkan/menjujutkan state baru
  - b. Evaluasi state baru.
    - i. Jika ini goal state maka return dan keluar
    - ii. Jika bukan goal state tetapi lebih baik dari current state maka jadikan state baru sebagai current state
    - iii. Jika tidak lebih baik dari current state lanjutkan perulangan

#### BEST-FIRST SEARCH

- kombinasi dari metode Depth-First Search dan metode Breadth-First Search dengan mengambil kelebihan dari metode tersebut.
- menelusuri satu jalur saja pada satu saat, tapi dapat berpindah ketika jalur lain terlihat lebih menjanjikan dari jalur yang sedang ditelusuri.
- mendapatkan jalur yang menjanjikan adalah dengan memberikan skala prioritas pada setiap state saat dihasilkan dengan fungsi heuristic

#### Algoritma Best-First Search

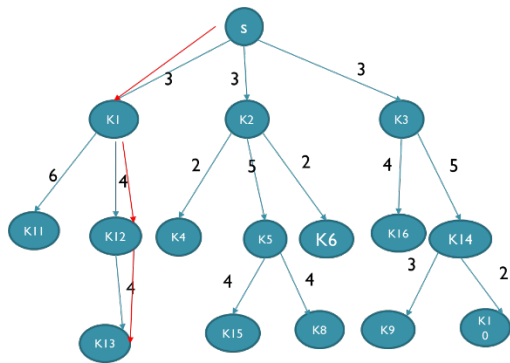
1. Mulai dengan OPEN hanya berisi initial state
2. Sampai goal ditemukan atau tidak ada lagi simpul yang tersisa dalam OPEN, lakukan :
  - a. Pilih simpul terbaik dalam OPEN
  - b. Telusuri successor-nya
  - c. Untuk tiap successor, lakukan :
    - i. Jika belum pernah ditelusuri sebelumnya, evaluasi simpul ini, tambahkan kedalam OPEN dan catat parentnya.
    - ii. Jika sudah pernah ditelusuri, ganti parent nya jika jalur baru lebih baik dari sebelumnya.

Contoh:

Contoh Metode Best First Search, jika tidak ditentukan goalnya

solusinya:

- S, K1
- k1, k2, k3
- k1, k12, k4, k6, k16, k5, k14, k11
- k12, k13, k10, k9, k8, k15



### REPRESENTASI PENGETAHUAN

- Metode yang digunakan untuk mengodekan pengetahuan dalam sebuah system
- Menangkap sifat-sifat penting problema
- Membuat informasi itu terdapat diakses oleh prosedur pemecahan problema

Pengetahuan diklasifikasi menjadi 3

- Prosedural (Bagaimana melakukan sesuatu [mendidihkan air dalam mangkok, memasak mie instan, menjalankan mobil])
- Deklaratif (Mengetahui sesuatu itu benar atau salah [Fakultas teknologi industri mempunyai 5 program studi, Dekan FTI UAD berjenis kelamin Perempuan])
- Tak terucap (tidak dapat diungkapkan dengan Bahasa [menggerakkan tangan, memejamkan mata])

Dua bagian dasar sistem AI (menurut Turban):

- 1) Basis Pengetahuan (Fakta tentang objek-objek dalam domain yang dipilih, Hubungan di antara domain domain tersebut)
- 2) Inference Engine (sekumpulan prosedur, menguji basis pengetahuan dalam menjawab suatu pertanyaan, menyelesaikan masalah, atau membuat keputusan)

### Karakteristik

Dapat diprogram dengan bahasa komputer dan disimpan dalam memori

Fakta dan pengetahuan lain yang terkandung di dalamnya dapat digunakan untuk penalaran

Dapat direpresentasikan dalam bentuk yang sederhana atau kompleks tergantung dari masalahnya (Schnupp, 1989)

### Model representasi pengetahuan

- 1) Logika (logic)
- 2) List & Tree
- 3) Jaringan Semantik (Semantic nets)
- 4) Bingkai (frame)
- 5) Tabel Keputusan (decision table)
- 6) Pohon keputusan (decision tree)
- 7) Kaidah/sistem produksi (production rule)
- 8) Naskah (script)

### LOGIKA

Suatu pengkajian ilmiah tentang serangkaian penalaran, sistem kaidah, dan prosedur yang membantu penalaran.

Jenis Penalaran

#### A. Penalaran Deduktif

Bergerak dari penalaran umum menuju ke konklusi khusus atau premis dan inferensi terdiri dari:

- 1) Premis Mayor
- 2) Premis Minor
- 3) Konklusi

Contoh :

Premis mayor : Jika hujan turun saya tidak akan kuliah

Premis minor : Pagi ini hujan turun

Konklusi : Oleh karena itu pagi ini saya tidak akan kuliah

#### B. Penalaran Induktif

Bergerak dari masalah khusus ke masalah umum

Menggunakan sejumlah fakta atau premis yang mantap untuk menarik kesimpulan umum

Contoh :

Premis 1 : Aljabar adalah pelajaran yang sulit

Premis 2 : Geometri adalah pelajaran yang sulit

Premis 3 : Kalkulus adalah pelajaran yang sulit

Konklusi : Matematika adalah pelajaran yang sulit

Konklusi tidak selalu mutlak, dapat berubah jika ditemukan fakta baru.

Contoh

Premis 4 : AI adalah pelajaran yang sulit

Komputer harus dapat menggunakan proses penalaran deduktif dan induktif ke dalam bentuk yang sesuai dengan manipulasi komputer, yaitu logika simbolik atau matematika Disebut Logika Komputasional:

#### 1. Logika Proporsional

Suatu statemen atau pernyataan yang menyatakan benar (TRUE) atau salah (FALSE)

Operator	Simbol
AND	$\wedge, \cdot$
OR	$\vee, +$
NOT	$\sim, \neg$
IMPLIES	$\rightarrow$
Bi-implikasi	$\leftrightarrow, \leftrightarrow$

A	B	$\sim A$	$\sim B$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
T	T	F	F	T	T	T	T
T	F	F	T	F	T	F	F
F	T	T	F	F	T	T	F
F	F	T	T	F	F	T	T

### Resolusi

suatu aturan untuk melakukan inferensi yang dapat berjalan secara efisien dalam suatu bentuk khusus yaitu conjunctive normal form (CNF)

untuk melakukan inferensi pada logika proposi

Langkah langkah mengubah suatu kalimat ke bentuk CNF

Langkah-langkah untuk mengubah suatu kalimat (konversi)

ke bentuk CNF :

Hilangkan implikasi dan ekuivalensi

$x \rightarrow y$  menjadi  $\neg x \vee y$

$x \leftrightarrow y$  menjadi  $(\neg x \vee y) \wedge (\neg y \vee x)$

Kurangi lingkup semua negasi menjadi satu negasi saja

$\neg(\neg x)$  menjadi  $x$

$\neg(x \vee y)$  menjadi  $(\neg x \wedge \neg y)$

$\neg(x \wedge y)$  menjadi  $(\neg x \vee \neg y)$

Gunakan aturan asosiatif dan distributif untuk

mengkonversi menjadi conjunction of disjunction

Asosiatif :  $(A \vee B) \vee C$  menjadi  $A \vee (B \vee C)$

Distributif :  $(A \wedge B) \vee C$  menjadi  $(A \vee C) \wedge (B \vee C)$

Buat satu kalimat terpisah untuk tiap-tiap konjungsi

Contoh:

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

- P
- $(P \wedge Q) \rightarrow R$
- $(S \vee T) \rightarrow Q$
- T

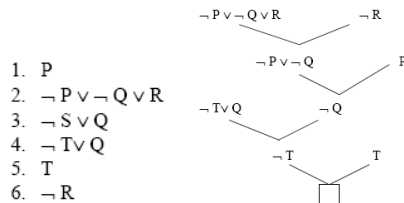
Tentukan kebenaran R

Jawab:

Ubah ke dalam bentuk CNF

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

Tambahkan Kontradiksi pada tujuannya, sehingga fakta-fakta dalam CNF menjadi :



Contoh bila diterapkan dalam kalimat :

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

Kalimat yang terbentuk :

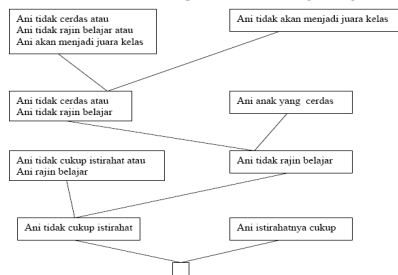
- Ani anak yang cerdas
- Jika ani anak yang cerdas dan ani rajin belajar, maka ani akan menjadi juara kelas
- Jika ani makannya banyak atau ani istirahatnya cukup, maka ani rajin belajar
- Ani istirahatnya cukup

Setelah dilakukan konversi ke bentuk CNF, didapat :

Fakta ke-2 : Ani tidak cerdas atau ani tidak rajin belajar atau ani akan menjadi juara kelas

Fakta ke-3 : Ani tidak makan banyak atau ani rajin belajar

Fakta ke-4 : Ani tidak cukup istirahat atau ani rajin belajar



## 2. Logika Predikat

Suatu logika yang lebih canggih yang seluruhnya menggunakan konsep dan kaidah proporsional yang sama. Disebut juga kalkulus predikat, yang memberi tambahan kemampuan untuk merepresentasikan pengetahuan dengan sangat cermat dan rinci.

Memungkinkan memecah statemen ke dalam bagian komponen yang disebut obyek, karakteristik obyek atau beberapa keterangan obyek.

Suatu proposisi atau premis dibagi menjadi dua bagian

1) Argumen (atau obyek)

Individu atau obyek yang membuat keterangan

2) Predikat (keterangan)

Keterangan yang membuat argumen dan predikat

Dalam suatu kalimat, predikat dapat berupa kata kerja atau bagian kata kerja

**PREDIKAT** (individu[obyek]1, individu[obyek]2)

- Misalnya proposisi:  
Mobil berada dalam garasi  
Di dalam (mobil, garasi)
- Dinyatakan menjadi  
Di dalam = produk (keterangan)  
Mobil = Argumen (obyek)

Garasi = Argumen (obyek)

Contoh lain

Proposisi : Rojali suka Juleha

Kalkulus Predikat : SUKA (Rojali, Juleha)

Proposisi : Pintu Terbuka

Kalkulus Predikat : BUKA (pintu)

Proposisi : Sensor cahaya aktif

Kalkulus Predikat : AKTIF (sensor cahaya)

Pengetahuan diekspresikan dalam kalkulus predikat yang dapat dimanipulasi agar dapat diinferensi/dinalar.

Pangkalan pengetahuan dibentuk dengan menggunakan variabel sebagai simbol-simbol untuk merancang obyek misalnya

x = Rojali

y = Juleha

Maka proposisinya menjadi Suka(x,y)

Predikat kalkulus membolehkan penggunaan simbol untuk mewakili fungsi-fungsi

Misalnya

ayah(Juleha) = Jojon

ibu(Rojali) = Dorce

Fungsi dapat digunakan bersamaan dengan predikat

Misalnya predikat berikut menjelaskan bahwa Jojon dan Dorce adalah berteman

teman(ayah(Juleha),ibu(Rojali))=teman(Jojon,Dorce)

Adalah simbol yang mengijinkan untuk menyatakan suatu rangkaian atau cakrawala variabel dalam suatu ekspresi logika

Dua pengukuran kuantitas, yaitu:

- Kuantitas universal ( $\forall$ ) = Untuk semua
- Kuantitas eksistensial ( $\exists$ ) = Ada / terdapat

Contoh:

Semua sapi berkaki empat

$(\forall x)[\text{Sapi}(x), \text{berkaki empat}(x)]$

Beberapa sapi berwarna putih

$(\exists x)[\text{Sapi}(x), \text{berwarna putih}(x)]$

Contoh

1. mahasiswa(Andi)

Andi adalah seorang mahasiswa

2. elektro(Andi)

Andi masuk jurusan elektro

3.  $\forall x : \text{elektro}(x) \rightarrow \text{teknik}(x)$

Setiap mahasiswa elektro pasti mahasiswa teknik

4. sulit(Kalkulus)

Kalkulus adalah matakuliah yang sulit

5.  $\forall x : \text{teknik}(x) \rightarrow \text{suka}(x, \text{kalkulus}) \vee \text{benci}(x, \text{kalkulus})$

Setiap mahasiswa teknik pasti akan suka kalkulus atau akan membencinya

6.  $\forall x : \exists y : \text{suka}(x, y)$

Setiap mahasiswa pasti akan suka terhadap suatu matakuliah

7.  $\forall x : \forall y : \text{mahasiswa}(x) \wedge \text{sulit}(y) \wedge \neg \text{hadir}(x, y) \rightarrow \neg \text{suka}(x, y)$

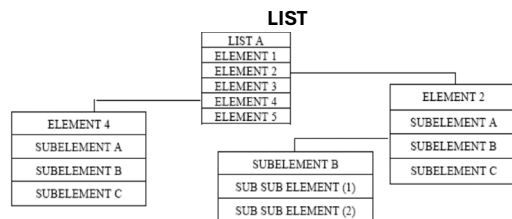
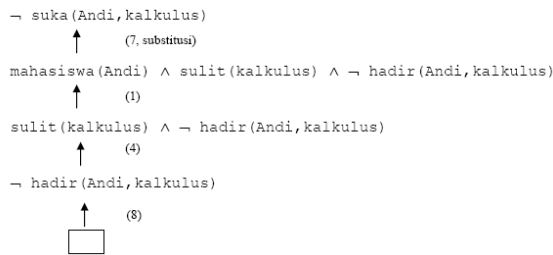
Mahasiswa yang tidak pernah hadir pada kuliah matakuliah sulit, maka mereka pasti tidak suka terhadap matakuliah tersebut

8.  $\neg \text{hadir}(\text{andi}, \text{kalkulus})$

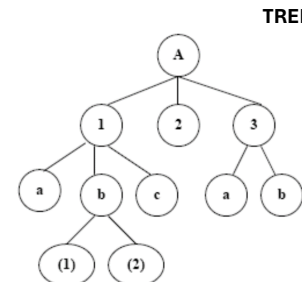
Andi tidak pernah hadir kuliah matakuliah kalkulus

“Apakah andi suka matakuliah kalkulus ?”

Gunakan penalaran backward :



- Struktur sederhana untuk representasi pengetahuan
- Daftar dari rangkaian materi yang terkait
- List digunakan untuk objek yang dikelompokkan, dikategorikan atau digabungkan



- Merupakan struktur grafik hirarki
- Struktur sederhana untuk representasi pengetahuan



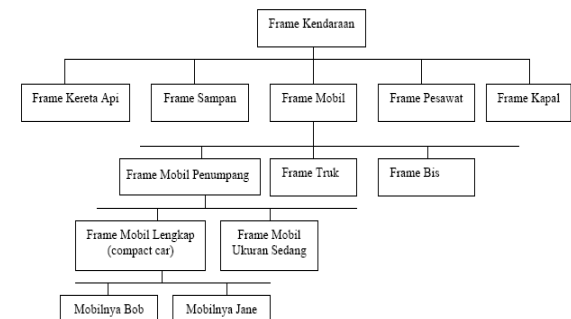
- Struktur sederhana untuk representasi pengetahuan grafis yang menunjukkan hubungan antar berbagai objek

- FRAME**
- kumpulan pengetahuan tentang suatu objek tertentu, peristiwa, lokasi, situasi, berdasarkan pengalaman
  - memiliki slot yang menggambarkan rincian (atribut) dan karakteristik objek.
  - Hierarki Frame: susunan hirarki dari frame memungkinkan pewarisan frame

Contoh Frame:

<b>Frame Mobil</b>
Class : Transportasi
Nama pabrik : Audi
Negara : Jerman
Model : 5000 Turbo
Tipe : Sedan
Bobot : 3300 lb
Ukuran dasar roda : 105.8 inci
Jumlah pintu : 4 (default)
Transmisi : 3-speed otomatis
Jumlah roda : 4 (default)
Mesin : (referensi kerangka mesin)
• Tipe : in-line, overhead cam
• Jumlah silinder : 5
Akselerasi
0-60 : 40,4 detik
¼ mil : 17,1 detik, 85 mph
Jarak gas : rata-rata 22 mpg
<b>Frame Mesin</b>
Kaliber silinder : 3.19 inci
Tak silinder : 3,4 inci
Rasio kompresi : 7,8 : 1
Sistem bahan bakar : injeksi dengan pertukaran turbo
Tenaga : 140 HP
Torsi : 160/ft/LB

Contoh Hierarki Frame:



Parent Frame	Child Frame
Nama : Compact Car	Nama : Mobilnya Jane
Slot	Facets
Pemilik	Cek daftar registrasi
Warna	Daftar per manufaktur
No silinder	No silinder
Range	4 atau 6
Jika dibutuhkan	Tanya pemilik
Buatan	Senna manufaktur
Daftar range	Tanya pemilik
Jika dibutuhkan	Gunakan hubungan frame
Model (tahun)	Model (tahun)
Range	1950 - 2001
Jika dibutuhkan	Tanya pemilik

### TABEL KEPUTUSAN (DECISION TABLE)

Tabel keputusan dalam format tabel

Tabel dibagi 2 bagian pertama untuk atribut & bagian kedua untuk nilai & kesimpulan

Atribut	Bulat	Bulat	Bulat	Bulat	Lonjong	Lonjong	Lonjong	Lonjong
Bentuk	Bulat	Bulat	Bulat	Bulat	Lonjong	Lonjong	Lonjong	Lonjong
Aroma	Asam	Asam	Manis	Manis	Manis	Manis	Asam	Manis
Warna	Kuning	Kuning	Merah	Merah	Kuning	Kuning	Oranye	Hijau
Rasa	Asam	Manis	Manis	Manis	Manis	Manis	Asam	Manis
Kulit	Kasar	Kasar	Halus	Halus	Halus	Halus	Halus	Halus
Kesimpulan								
Anggur	X							
Jeruk		X						
Apel			X	X				
Pisang					X			
Pir						X		X

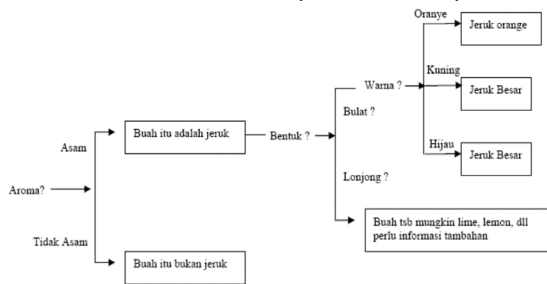
Contoh:

Buah dan Atribut	bentuk		Aroma dan rasa		warna			kulit	
	lonjong	Bulat	asam	manis	orens	kuning	hijau	halus	kasar
Anggur	X	X	X	X	-	X	X	X	-
Pir	X	-	-	X	-	X	X	X	-
Jeruk	-	X	X	X	X	X	X	X	X
Apel	-	X	-	X	-	-	X	X	-
Pisang	X	-	-	X	-	X	X	X	-

JIKA FAKTANYA ASAM DAN LONJONG..?

APA HASIL SOLUSINYA...? ANGGUR

## POHON KEPUTUSAN (DECISION TREE)

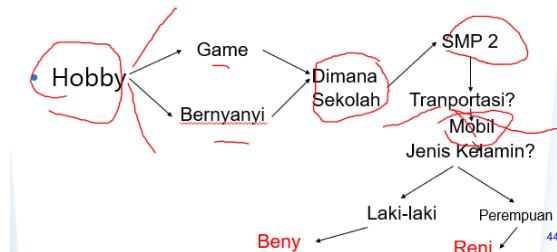


Diberikan Fakta sebagai berikut:

Nama	Hoby	Sekolah	Jenis Kelamin	Transportasi ke sekolah
Joko	bersepeda	SMP 1	Laki-laki	Bus
Sinta	Membaca Komik	SMP 1	Perempuan	Sepeda
Beny	Game	SMP 2	Laki-laki	Mobil
Reny	Bernyanyi	SMP 2	Perempuan	Mobil

Siapakah Dia..? Buatlah Pohon pelacakan!

Fakta : bersekolah di SMP 2 dengan Transportasi Mobil  
Apakah Hobby anak tersebut?



## NASKAH (SCRIPT)

### Script

sama dengan frame, bedanya menggambarkan urutan peristiwa

### Elemen script

Kondisi input : kondisi yang harus dipenuhi

Track : variasi yang mungkin terjadi

Prop : berisi objek-objek pendukung

Role : peran yang dimainkan

Scene : adegan yang dimainkan

Hasil : kondisi yang ada setelah urutan peristiwa dalam script terjadi.

Contoh Script: Pelaksanaan Praktikum

Jalur (track)

Pelaksanaan Praktikum Kecerdasan Buatan

Role (peran)

mahasiswa, laboran, asisten

Prop (pendukung)

lembar soal pre test/pos test, lembar jawab, presensi, form nilai, dll

Kondisi input

mahasiswa terdaftar untuk melaksanakan

praktikum

Contoh Naskah

Adegan (scene) -1 : Persiapan praktikum

Laboran menerima pendaftaran

Laboran mempersiapkan sarana praktikum

Laboran menyiapkan lembar presensi

Adegan-2 : Mahasiswa masuk Laboratorium

Asisten mempersilahkan mahasiswa masuk

Asisten membagikan soal pre test

Asisten mengoreksi lembar jawab pre test

Asisten memberi nilai pretest

Adegan – 3 : Mahasiswa mengerjakan tugas praktikum

Mahasiswa membaca modul praktikum

Mahasiswa membaca intruksi tugas praktikum/pos test

Mahasiswa mengerjakan tugas praktikum/pos test

Mahasiswa mengumpulkan tugas pos test

Adegan – 4 : Mahasiswa telah selesai praktikum

Asisten mempersilahkan mahasiswa keluar ruangan

Asisten menerima tugas post test

Mahasiswa keluar dari laboratorium

Adegan – 5 : Asisten merekap nilai

Asisten mengoreksi tugas post test

Asisten menginputkan nilai ke dalam form nilai

Asisten meninggalkan ruangan lab.

Hasil :

Mahasiswa merasa senang dan lega

Mahasiswa merasa kecewa

Mahasiswa pusing

Mahasiswa memaki – maki

Mahasiswa sangat bersyukur

## SISTEM/ATURAN PRODUKSI (PRODUCTION RULES)

Penalaran merupakan kemampuan komputer untuk melakukan pelacakan

2 metode penalaran yang menggunakan aturan

- Forward Reasoning (penalaran maju)
  - o Pelacakan dimulai dari keadaan awal (informasi atau fakta yang ada) dan kemudian dicoba untuk mencocokkan dengan tujuan yang diharapkan
  - o Gunakan jika jumlah keadaan awal lebih kecil daripada tujuan & kejadian itu berupa fakta baru
- Backward Chaining
  - o Penalaran dimulai dari tujuan atau hipotesa, baru dicocokkan dengan keadaan awal atau fakta-fakta yang ada.
  - o Jika jumlah keadaan awal lebih banyak daripada tujuan
  - o Jika kejadian itu berupa query

Representasi pengetahuan yang berupa aturan (rule) :

Antecedent : mengekspresikan situasi / premis (berawalan if)

Konsekuensi : menyatakan tindakan tertentu jika premis benar (berawalan THEN)

Contoh :

IF lalu lintas pagi ini padat

THEN saya naik sepeda motor saja

Aturan dapat ditulis dalam beberapa bentuk :

1. IF premis THEN kesimpulan

**Jika** pendapatan tinggi **MAKA** pajak yang harus dibayar juga tinggi

2. Kesimpulan IF premis

Pajak yang harus dibayar tinggi **JIKA** pendapatan tinggi

3. Inclusion of ELSE

**IF** pendapatan tinggi **OR** pengeluaran tinggi, **THEN** pajak yang harus dibayar tinggi **ELSE** pajak yang harus dibayar rendah

4. Aturan yang lebih kompleks

**IF** rating kredit tinggi **AND** gaji lebih besar dari \$30,000 **OR** aset lebih dari \$75,000 **AND** sejarah pembayaran tidak miskin **THEN** pinjaman diatas \$ 10,000 disetujui dan daftar pinjaman masuk kategori "B"

IF G01 and G02 and G03 and G04 and G05 and G06 and G07 then P01 → [Rule 1]

IF G04 and G05 and G06 and G07 then P01 → [Rule 2]

IF G01 and G02 and G03 and G08 and G09 and G10 and G11 then P02 → [Rule 3]

IF G08 and G09 and G10 and G11 then P02 → [Rule 4]



Contoh:

R1 : IF suku bunga turun THEN harga obligasi naik

R2 : IF suku bunga naik THEN harga obligasi turun

R3 : IF suku bunga tidak berubah THEN harga obligasi tidak berubah

R4 : IF dolar naik THEN suku bunga turun

R5 : IF dolar turun THEN suku bunga naik

R6 : IF harga obligasi turun THEN beli obligasi

Apabila diketahui bahwa dolar turun, apa keputusan yang diambil, apakah akan membeli obligasi atau tidak

Forward Reasoning :  $R5 \rightarrow R2 \rightarrow R6$

Dari fakta dolar turun,

berdasarkan Rule 5, diperoleh konklusi suku bunga naik.

Dari Rule 2, suku bunga naik menyebabkan harga obligasi turun.

Dengan Rule 6, jika harga obligasi turun, maka kesimpulan yang diambil adalah membeli obligasi.

Backward Reasoning :

Dari solusi yaitu membeli obligasi

dengan menggunakan Rule 6 diperoleh anteseden harga obligasi turun

Dari Rule 2 dibuktikan harga obligasi turun bernilai benar jika suku bunga naik bernilai benar .

Dari Rule 5 suku bunga naik memang bernilai benar karena diketahui fakta dolar turun.

Contoh:

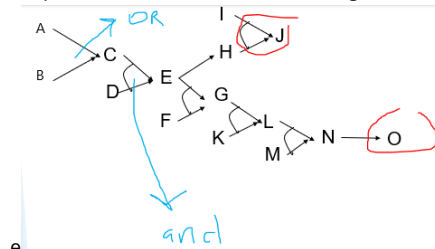
No	Aturan
R1	IF A Or B Then C
R2	IF C and D Then E
R3	IF E and F Then G
R4	IF E then H
R5	IF H and I Then J
R6	IF G and K Then L
R7	IF L and M Then N
R8	IF N Then O

Jika Fakta awal : A  
Solusi: ...?  
Jawab :  $R_1 \rightarrow R_2 \rightarrow R_4 \rightarrow R_5$

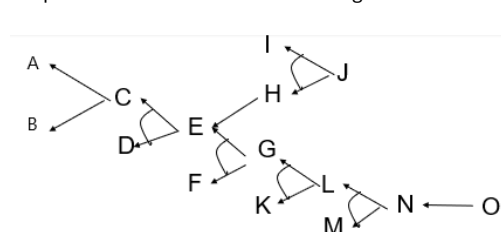
Graph FC

Graph BC

Graph Pelacakan Forward Reasoning



Graph Pelacakan Backward Reasoning



Keterkaitan basis aturan, table keputusan dan metode penalaran

penyakit	Rule	gejala								Solusi
		G1	G2	G3	G4	G5	G6	G7	G8	
P01	R1	✓		✓	✓	✓	✓			S01
P01	R2	✓				✓	✓			S02
P02	R3			✓	✓			✓	✓	S02
p03	R4		✓		✓					S04
P04	R5		✓			✓				S05

Diatas merupakan tabel keputusan yang merupakan bentuk representasi pengetahuan lainnya.

Jika diberikan rp seperti tabel keputusan diatas, maka diubah kedalam bentuk rule production if...then, setelah itu buatlah graph pelacakan dengan forward chaining atau backward chaining

Rule 1: IF G1&G3&G4&G5&G6 Then P01 & S01

Rule 2: IF G1&G5&G6 Then P01 & S02

Rule 3: IF G3&G4&G6&G7 Then P02 & S02

Rule 4: IF G2&G4&G5&G6 Then P03 & S04

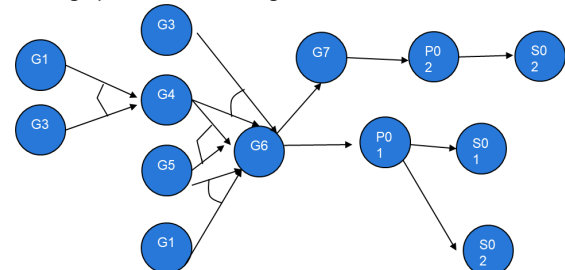
Rule 5: IF G2&G5&G6 Then P04 & S05

Untuk Rule 1: Rule 1: IF G1&G3&G4&G5&G6 Then P01 & S01

Rule 2: IF G1&G5&G6 Then P01 & S02

Rule 3: IF G3&G4&G6&G7 Then P02 & S02

Maka graph forward chaining:



## Representasi Pengetahuan

1. Jelaskan bagaimana peran representasi pengetahuan dalam membangun mesin inferensi pada suatu aplikasi kecerdasan buatan?

Representasi pengetahuan memainkan peran penting dalam membangun mesin inferensi dalam aplikasi kecerdasan buatan (AI). Representasi pengetahuan adalah cara di mana pengetahuan tentang dunia direpresentasikan dalam sistem kecerdasan buatan. Ini mencakup fakta-fakta tentang dunia, hubungan antara berbagai objek atau entitas, dan aturan-aturan yang mengatur bagaimana entitas-entitas ini berinteraksi satu sama lain.

Berikut adalah beberapa cara di mana representasi

pengetahuan membantu dalam pembuatan mesin inferensi:

- Representasi pengetahuan digunakan untuk membentuk basis pengetahuan, yang merupakan kumpulan fakta dan aturan yang digunakan oleh sistem AI untuk membuat inferensi. Basis pengetahuan ini digunakan oleh mesin inferensi untuk menarik kesimpulan atau membuat keputusan berdasarkan informasi yang disediakan.
- Representasi pengetahuan memfasilitasi pemrosesan informasi dalam mesin inferensi. Misalnya, dalam model representasi berbasis logika, pengetahuan direpresentasikan dalam bentuk pernyataan logika yang dapat dievaluasi sebagai benar atau salah. Mesin inferensi kemudian dapat menggunakan pernyataan ini untuk menarik kesimpulan logis.
- Representasi pengetahuan memungkinkan mesin inferensi untuk melakukan penalaran. Misalnya, dalam model representasi berbasis aturan, pengetahuan direpresentasikan dalam bentuk aturan "IF-THEN". Mesin inferensi dapat menggunakan aturan ini untuk melakukan penalaran berbasis aturan dan menarik kesimpulan.
- Dalam beberapa kasus, representasi pengetahuan juga dapat membantu mesin inferensi dalam



belajar dari pengalaman. Misalnya, dalam model representasi berbasis jaringan semantik, mesin inferensi dapat memperbaiki jaringan semantik berdasarkan pengalaman baru, sehingga memungkinkannya untuk belajar dari pengalaman tersebut.

Jadi representasi pengetahuan memberikan struktur dan konteks untuk informasi yang digunakan oleh mesin inferensi dalam sistem AI. Ini memungkinkan sistem AI untuk memahami dunia, membuat inferensi, dan belajar dari pengalaman.

2. Dalam membangun basis pengetahuan, diperlukan adanya basis aturan dan basis data. Jelaskan bagaimana kaitan kedua hal tersebut dalam membangun basis pengetahuan untuk suatu aplikasi kecerdasan buatan!

Dalam membangun basis pengetahuan untuk suatu aplikasi kecerdasan buatan, diperlukan adanya basis aturan (rule base) dan basis data (database). Keduanya saling terkait dan memiliki peran penting dalam proses penalaran dan pengambilan keputusan oleh sistem kecerdasan buatan.

Basis Data (Database):

Basis data berfungsi sebagai tempat penyimpanan fakta-fakta atau informasi tentang objek-objek dalam domain pengetahuan yang dipilih. Basis data berisi data-data yang relevan dengan domain masalah yang akan ditangani oleh aplikasi kecerdasan buatan. Misalnya, dalam sistem pakar untuk diagnosis penyakit, basis data dapat berisi informasi tentang gejala-gejala, penyakit-penyakit, dan data-data pasien.

Basis Aturan (Rule Base):

Basis aturan berisi kumpulan aturan atau kaidah-kaidah yang mewakili pengetahuan domain yang diperoleh dari pakar atau sumber pengetahuan lainnya. Aturan-aturan ini digunakan untuk melakukan penalaran dan mengambil kesimpulan berdasarkan informasi yang tersedia dalam basis data. Aturan-aturan ini biasanya diekspresikan dalam bentuk "IF-THEN" atau aturan produksi (production rules).

Kaitan antara basis data dan basis aturan dalam membangun basis pengetahuan adalah sebagai berikut:

Basis data menyediakan fakta-fakta yang diperlukan untuk proses penalaran oleh basis aturan.

Basis aturan menggunakan informasi dari basis data sebagai masukan (input) untuk menerapkan aturan-aturan dan melakukan penalaran.

Proses penalaran oleh basis aturan dapat menghasilkan kesimpulan atau rekomendasi yang berguna bagi pengguna aplikasi kecerdasan buatan.

Kesimpulan atau rekomendasi ini dapat digunakan untuk mengambil tindakan atau keputusan yang tepat dalam domain masalah yang ditangani.

Dengan demikian, basis data dan basis aturan saling melengkapi dalam membangun basis pengetahuan yang efektif untuk aplikasi kecerdasan buatan. Basis data menyediakan informasi faktual, sedangkan basis aturan mewakili pengetahuan domain dan kemampuan penalaran untuk menghasilkan kesimpulan atau rekomendasi yang berguna.

## Representasi Pengetahuan (penalaran)

Bila diberikan suatu gejala yang mengarah penyakit Gastro Entero Reflux Disease, sebagai berikut

Kode Gejala	Gejala	Bobot
G01	Nyeri dada	0,6
G02	Nyeri pada tenggorokan	0,3
G03	Nyeri dada saat menelan	0,7
G04	Rasa asam/ pahit di mulut	0,7
G05	Kepala pusing	0,8
G06	Mata berkunang-kunang	0,8
G07	Keluar lender dari mulut	0,9
G08	Nafas tidak teratur	0,7

Tentukanlah:

3. Model representasi pengetahuan apa yang paling sesuai untuk membangun aplikasi kecerdasan buatan yang dapat mengidentifikasi penyakit tersebut? Jelaskan Alasannya

Model representasi pengetahuan yang paling sesuai untuk membangun aplikasi kecerdasan buatan yang dapat mengidentifikasi penyakit berdasarkan gejala adalah model Pohon Keputusan (Decision Tree) dan Kaidah/Sistem Produksi (Production Rule).

- Pohon Keputusan (Decision Tree): Model ini sangat sesuai untuk mendiagnosis penyakit berdasarkan gejala. Pohon keputusan memungkinkan sistem untuk membuat keputusan berdasarkan berbagai kondisi atau gejala. Setiap cabang dari pohon mewakili pilihan yang mungkin dari gejala dan setiap daun mewakili hasil (dalam hal ini, diagnosis penyakit). Hal ini memudahkan penalaran dan juga memudahkan penjelasan kepada pengguna. Bobot dari setiap gejala dapat digunakan sebagai faktor pembobot dalam proses pengambilan keputusan.
- Kaidah/Sistem Produksi (Production Rule): Model ini juga cocok untuk tujuan ini. Sistem produksi terdiri dari serangkaian aturan "IF-THEN" yang dapat digunakan untuk mewakili pengetahuan. Dalam konteks ini, setiap aturan bisa mewakili hubungan antara gejala dan penyakit. Misalnya, aturan mungkin berbunyi "IF nyeri dada AND rasa asam/pahit di mulut THEN Gastro Entero Reflux Disease". Bobot dari setiap gejala dapat digunakan sebagai faktor pembobot dalam proses pengambilan keputusan.

4. Bila seorang pasien menggunakan aplikasi kecerdasan buatan untuk berkonsultasi penyakit, dan dia menginputkan fakta G02 dan G03, buktikan apakah benar sakit pasien tersebut sakit Gastro Entero Reflux Disease. Buatlah representasi pengetahuan untuk membuktikannya dalam bentuk: Pohon Pelacakan dan Produksi If ... Then

R1: If G07 = True Then Total\_weight = Total\_weight + 0.9

R2: If G05 = True Then Total\_weight = Total\_weight + 0.8

R3: If G06 = True Then Total\_weight = Total\_weight + 0.8

R4: If G03 = True Then Total\_weight = Total\_weight + 0.7

R5: If G04 = True Then Total\_weight = Total\_weight + 0.7

R6: If G08 = True Then Total\_weight = Total\_weight + 0.7

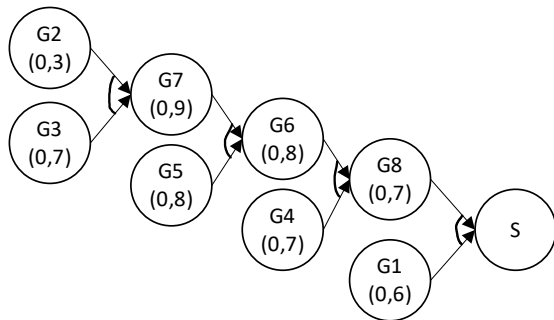
R7: If G01 = True Then Total\_weight = Total\_weight + 0.6

R8: If G02 = True Then Total\_weight = Total\_weight + 0.3

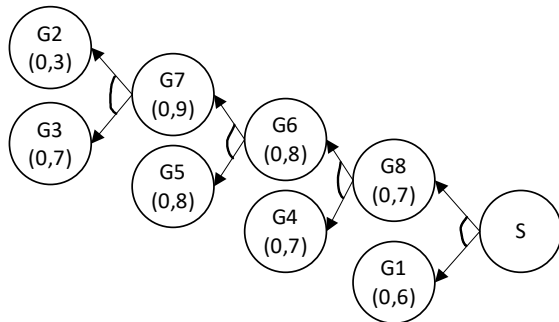
R9: If Total\_weight >= weight\_GERD Then Gerd = True

5. Buatlah graf penalarannya dengan menggunakan backward reasoning

#### Forward Chaining



#### Backward Chaining



Total\_bobot\_gejala = G02+G03

IF total\_bobot\_gejala >= bobot\_kemungkinan\_terjadi\_gejala

6. Buatlah graf penalarannya dengan menggunakan backward reasoning

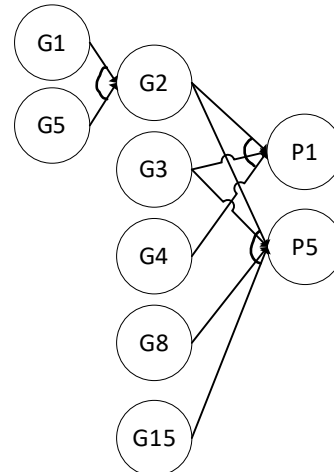
#### Analisis Kasus

Apabila diberi suatu tabel Keputusan sebagai berikut:

Gejala dan Penyakit	P01	P02	P03	P04	P05	P06
G001	✓	✓		✓	✓	✓
G002	✓				✓	
G003	✓		✓		✓	
G004	✓					
G005	✓				✓	
G006		✓				
G007		✓				
G008		✓	✓		✓	
G009			✓			
G010			✓			
G011						✓
G012				✓		
G013				✓		
G015					✓	
G016						✓
G017						✓

Maka berdasarkan tabel Keputusan tersebut buktikanlah Kesimpulan yang teridentifikasi adalah Penyakit P05 jika gejalanya adalah G001 dan G005 dengan menggambarkan pohon pelacakannya!

No	Aturan
R1	IF G001&G002&G003&G004&G005 THEN P01
R2	IF G001&G006&G007&G008 THEN P02
R3	IF G003&G008&G009&G010 THEN P03
R4	IF G001&G012&G013 THEN P04
R5	IF G001&G002&G003&G005&G008&G015 THEN P05
R6	IF G001&G011&G016&G017 THEN P06



Kesimpulan bahwa penyakitnya adalah P05 jika gejalanya G001 dan G005 **tidak sepenuhnya benar**. Meskipun kedua gejala tersebut memang muncul pada P05, penting untuk dicatat bahwa G001 dan G005 juga merupakan gejala dari P01. Tabel keputusan yang diberikan hanya menunjukkan ada atau tidaknya gejala pada suatu penyakit, tanpa memberikan informasi tentang seberapa sering gejala tersebut muncul pada setiap penyakit atau seberapa kuat asosiasinya dengan penyakit tertentu.

Dengan demikian, berdasarkan data yang tersedia, kita tidak dapat secara pasti menyimpulkan bahwa pasien menderita P05 hanya karena menunjukkan gejala G001 dan G005. Gejala-gejala tersebut bisa saja mengindikasikan P01 atau P05. Untuk memastikan diagnosis P05, diperlukan adanya gejala yang unik dan hanya muncul pada P05, bukan pada penyakit lain seperti P01. Tanpa adanya gejala pembeda tersebut, diagnosis P05 tidak memiliki cukup bukti.