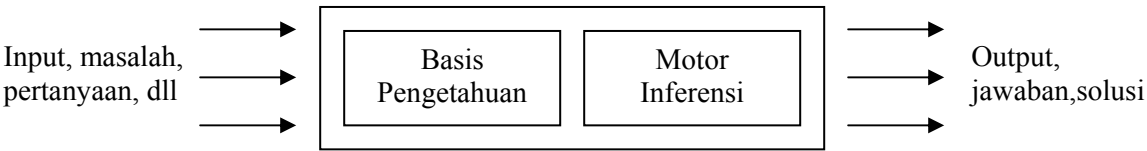


II. MASALAH DAN METODE PEMECAHAN MASALAH

Sistem yang menggunakan kecerdasan buatan akan memberikan output berupa solusi dari suatu masalah berdasarkan kumpulan pengetahuan yang ada.



Gambar sistem yang menggunakan kecerdasan buatan

Pada gambar, input yg diberikan pada sistem yg menggunakan kecerdasan buatan adalah berupa masalah. Sistem harus dilengkapi dengan sekumpulan pengetahuan yang ada pada basis pengetahuan. Sistem harus memiliki motor inferensi agar mampu mengambil kesimpulan berdasarkan fakta atau pengetahuan. Output yang diberikan berupa solusi masalah sebagai hasil dari inferensi.

Secara umum, untuk membangun suatu sistem yang mampu menyelesaikan masalah, perlu dipertimbangkan 4 hal :

- 1. Mendefinisikan masalah dengan tepat.
Pendefinisian ini mencakup spesifikasi yang tepat mengenai *keadaan awal* dan *solusi yang diharapkan*.
- 2. Menganalisis masalah tersebut serta mencari beberapa teknik penyelesaian masalah yang sesuai.
- 3. Merepresentasikan pengetahuan yang perlu untuk menyelesaikan masalah tersebut.
- 4. Memilih teknik penyelesaian masalah yang terbaik

MENDEFINISIKAN MASALAH SEBAGAI SUATU RUANG KEADAAN

Misalkan permasalahan yang dihadapi adalah permainan catur, maka harus ditentukan :

- 1. posisi awal pada papan catur
posisi awal setiap permainan catur selalu sama, yaitu semua bidak diletakkan di atas papan catur dalam 2 sisi, yaitu kubu putih dan kubu hitam.
- 2. aturan – aturan untuk melakukan gerakan
aturan – aturan ini sangat berguna untuk menentukan gerakan suatu bidak, yaitu melangkah dari satu keadaan ke keadaan lain. Misalkan untuk mempermudah menunjukkan posisi bidak, setiap kotak ditunjukkan dalam huruf (a,b,c,d,e,f,g,h) pada arah horisontal dan angka (1,2,3,4,5,6,7,8) pada arah vertikal. Suatu aturan untuk menggerakkan bidak dari posisi (e,2) ke (e,4) dapat ditunjukkan dengan aturan :
if bidak putih pada kotak(e,2),
and kotak(e,3) kosong,
and kotak(e,4) kosong
then gerakkan bidak dari (e,2) ke (e,4)
- 3. tujuan (goal)
tujuan yang ingin dicapai adalah posisi pada papan catur yang menunjukkan kemenangan seseorang terhadap lawannya. Kemenangan ini ditandai dengan posisi raja yang sudah tidak dapat bergerak lagi.

Contoh tersebut menunjukkan representasi masalah dalam Ruang Keadaan (State Space), yaitu suatu ruang yang berisi semua keadaan yang mungkin. Kita dapat memulai bermain catur dengan menempatkan diri pada keadaan awal, kemudian bergerak dari satu keadaan ke keadaan yang lain sesuai dengan aturan yang ada, dan mengakhiri permainan jika salah satu telah mencapai tujuan.

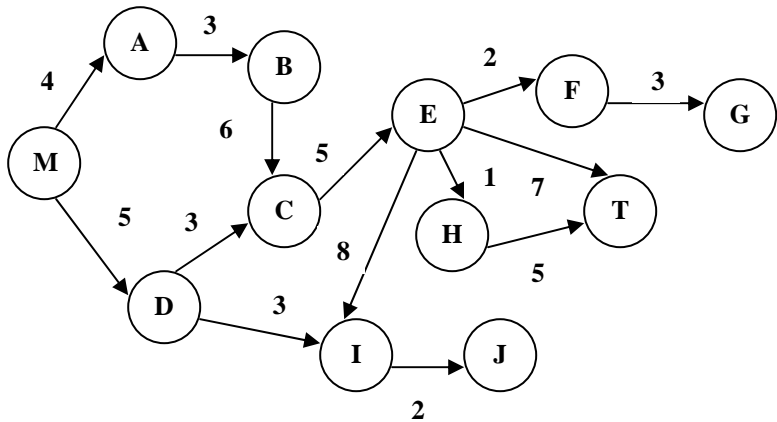
Jadi untuk *mendeskrripsikan masalah dengan baik* harus :

- 1. Mendefinisikan suatu ruang keadaan (state space)
- 2. Menetapkan satu atau lebih keadaan awal (initial state)
- 3. Menetapkan satu atau lebih tujuan (goal state)
- 4. Menetapkan kumpulan aturan

Ada beberapa cara untuk merepresentasikan Ruang Keadaan, antara lain :

GRAPH KEADAAN

Graph terdiri dari node-node yang menunjukkan keadaan yaitu keadaan awal dan keadaan baru yang akan dicapai dengan menggunakan operator. Node-node dalam graph keadaan saling dihubungkan dengan menggunakan arc (busur) yang diberi panah untuk menunjukkan arah dari suatu keadaan ke keadaan berikutnya.



Graph keadaan dengan node M menunjukkan keadaan awal, node T adalah tujuan. Ada 4 lintasan dari M ke T :

- M-A-B-C-E-T
- M-A-B-C-E-H-T
- M-D-C-E-T
- M-D-C-E-H-T

Lintasan buntu atau lintasan yang tidak sampai ke tujuan :

- M-A-B-C-E-F-G
- M-A-B-C-E-I-J
- M-D-C-E-F-G
- M-D-C-E-I-J
- M-D-I-J

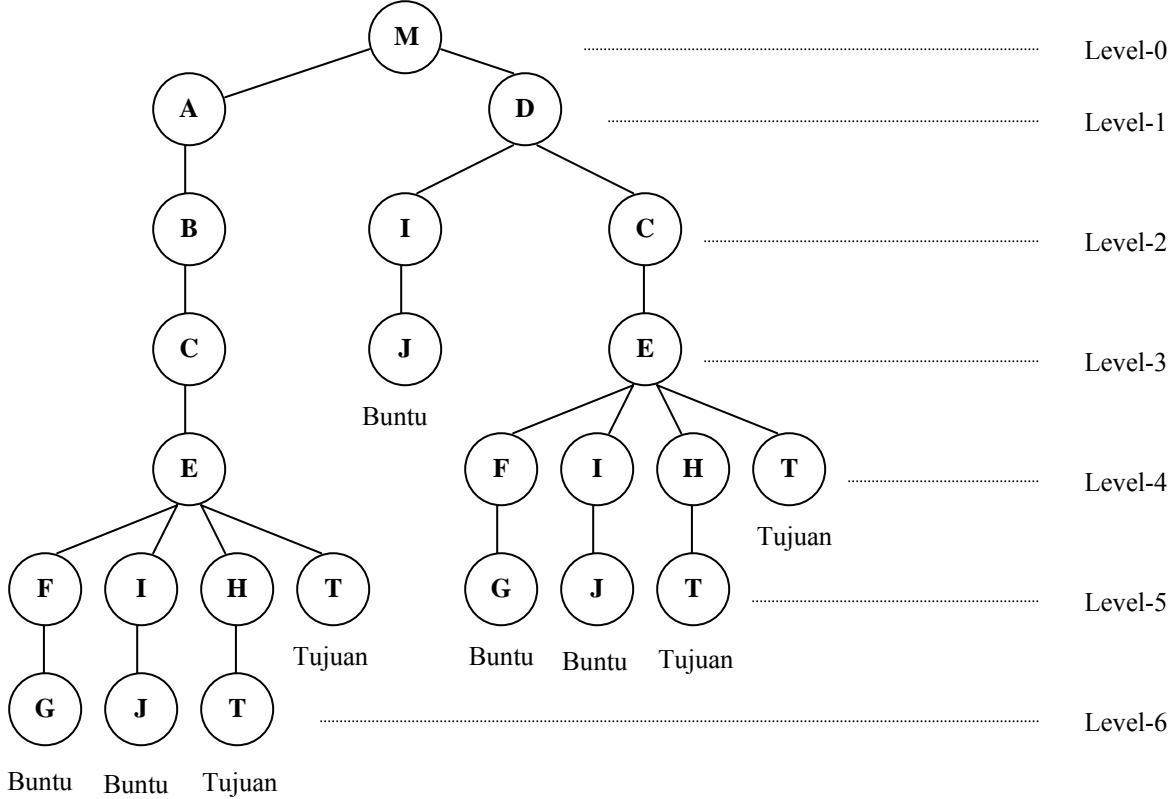
POHON PELACAkan / PENCARIAN

Struktur pohon digunakan untuk menggambarkan keadaan secara hirarkis. Node yg terletak pada level-0 disebut 'akar'.

Node akar : menunjukkan keadaan awal & memiliki beberapa percabangan yang terdiri atas beberapa node yg disebut 'anak'.

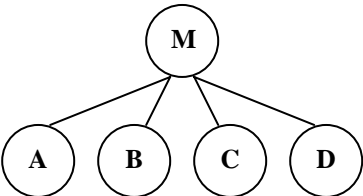
Node-node yg tidak memiliki anak disebut 'daun' menunjukkan akhir dari suatu pencarian, dapat berupa tujuan yang diharapkan (goal) atau jalan buntu (dead end).

Gambar berikut menunjukkan pohon pencarian untuk graph keadaan dengan 6 level.

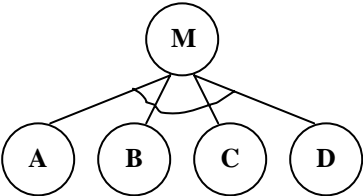


POHON AND/OR

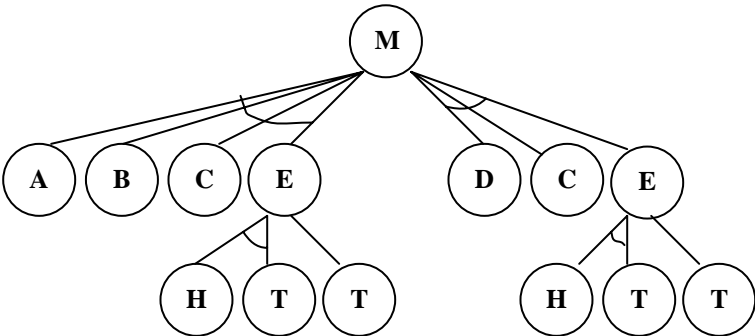
Masalah M dicari solusinya dengan 4 kemungkinan yaitu A OR B OR C OR D.



Masalah M hanya dapat diselesaikan dengan A AND B AND C AND D



Contoh : Dengan menggunakan pohon AND/OR tujuan yang dicapai pada pohon di Gambar sebelumnya bisa dipersingkat hanya sampai level-2 saja.



Contoh 1 : Masalah EMBER

Ada 2 ember masing-masing berkapasitas 4 galon (ember A) dan 3 galon (ember B). Ada pompa air yg akan digunakan untuk mengisi air pada ember tersebut. Bagaimana dapat mengisi tepat 2 galon air ke dalam ember berkapasitas 4 galon?

Penyelesaian :

- 1. Identifikasi ruang keadaan (state space)
Permasalahan ini dapat digambarkan sebagai himpunan pasangan bilangan bulat :
x = jumlah air yg diisikan ke ember 4 galon (ember A)
y = jumlah air yg diisikan ke ember 3 galon (ember B)
Ruang keadaan = (x,y) sedemikian hingga $x \in \{0,1,2,3,4\}$ dan $y \in \{0,1,2,3\}$
- 2. Keadaan awal & tujuan
Keadaan awal : kedua ember kosong = (0,0)
Tujuan : ember 4 galon berisi 2 galon air = (2,n) dengan sembarang n
- 3. Keadaan ember
Keadaan ember bisa digambarkan sebagai berikut :

Keadaan awal	Tujuan ∇				
<div><div></div><div></div><div></div><div></div></div> <div>(0,0)</div>	(1,0)	(2,0)	(3,0)	(4,0)	
(0,1)	(1,1)	(2,1)	(3,1)	(4,1)	
(0,2)	(1,2)	(2,2)	(3,2)	(4,2)	
(0,3)	(1,3)	(2,3)	(3,3)	(4,3)	

4. Aturan-aturan
- Diasumsikan kita dapat mengisi ember air itu dari pompa air, membuang air dari ember ke luar, menuangkan air dari ember yang satu ke ember yang lain.
- Kita buat beberapa aturan-aturan yang dapat digambarkan sebagai berikut :

Aturan ke-	Jika	Maka
1	(x,y) $x < 4$	$(4,y)$ Isi ember A
2	(x,y) $y < 3$	$(x,3)$ Isi ember B
3	(x,y) $x > 0$	$(x - d,y)$ Tuang sebagian air keluar dari ember A
4	(x,y) $y > 0$	$(x,y - d)$ Tuang sebagian air keluar dari ember B
5	(x,y) $x > 0$	$(0,y)$ Kosongkan ember A dengan membuang airnya
6	(x,y) $y > 0$	$(x,0)$ Kosongkan ember B dengan membuang airnya
7	(x,y) $x+y \geq 4$ dan $y > 0$	$(4,y - (4 - x))$ Tuang air dari ember B ke ember A sampai ember A penuh
8	(x,y) $x+y \geq 3$ dan $x > 0$	$(x - (3 - y),3)$ Tuang air dari ember A ke ember B sampai ember B penuh
9	(x,y) $x+y \leq 4$ dan $y > 0$	$(x+y,0)$ Tuang seluruh air dari ember B ke ember A
10	(x,y) $x+y \leq 3$ dan $x > 0$	$(0,x+y)$ Tuang seluruh air dari ember A ke ember B
11	$(0,2)$	$(2,0)$ Tuang 2 galon air dari ember B ke ember A

5. Representasi ruang keadaan dengan pohon pelacakan
- Pencarian suatu solusi dapat dilukiskan dengan menggunakan pohon. Tiap-tiap node menunjukkan satu keadaan. Jalur dari parent ke child ,menunjukkan 1 operasi. Tiap node memiliki node child yg menunjukkan keadaan yg dapat dicapai oleh parent.

Solusi yg ditemukan :

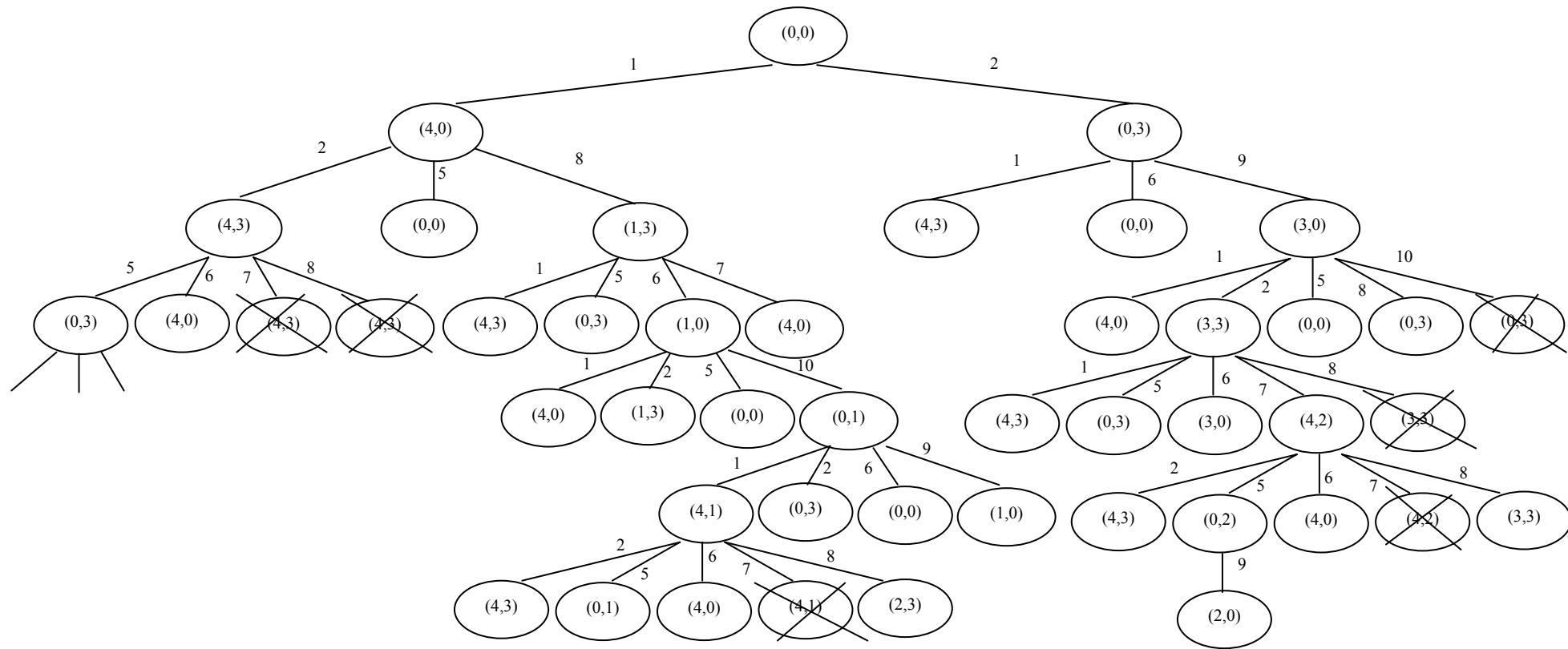
Solusi 1

Isi ember A	Isi ember B	Aturan yg dipakai
0	0	1
4	0	8
1	3	6
1	0	10
0	1	1
4	1	8
2	3	Solusi

Solusi 2

Isi ember A	Isi ember B	Aturan yg dipakai
0	0	2
0	3	9
3	0	2
3	3	7
4	2	5
0	2	9
2	0	Solusi

Representasi ruang keadaan untuk kasus EMBER



Contoh 2 : Masalah PETANI,KAMBING,SERIGALA,SAYURAN,PERAHU

Seorang petani akan menyeberangkan seekor kambing,seekor serigala,sayuran dengan sebuah perahu yg melalui sungai. Perahu hanya bisa memuat petani & satu penumpang yg lain (kambing, serigala, atau sayuran). Jika ditinggalkan petani tersebut, maka sayuran dimakan kambing dan kambing akan dimakan serigala.

Penyelesaian :

1. Identifikasi ruang keadaan
- Permasalahan ini dapat dilambangkan dengan (jumlah kambing,jumlah serigala,jumlah sayuran,jumlah perahu).
- Contoh : daerah asal (0,1,1,1) = daerah asal tidak ada kambing,ada serigala, ada sayuran,ada perahu
2. Keadaan awal & tujuan
- Keadaan awal, pada kedua daerah :
- daerah asal = (1,1,1,1)
- daerah seberang = (0,0,0,0)
- Keadaan tujuan, pada kedua daerah :
- daerah asal = (0,0,0,0)
- daerah seberang = (1,1,1,1)

3. Aturan-aturan

Aturan ke-	Aturan
1	Kambing menyeberang
2	Sayuran menyeberang
3	Serigala menyeberang
4	Kambing kembali
5	Sayuran kembali
6	Serigala kembali
7	Perahu kembali

4. Solusi yg ditemukan

Daerah asal	Daerah seberang	Aturan yg dipakai
(1,1,1,1)	(0,0,0,0)	1
(0,1,1,0)	(1,0,0,1)	7
(0,1,1,1)	(1,0,0,0)	3
(0,0,1,0)	(1,1,0,1)	4
(1,0,1,1)	(0,1,0,0)	2
(1,0,0,0)	(0,1,1,1)	7
(1,0,0,1)	(0,1,1,0)	1
(0,0,0,0)	(1,1,1,1)	Solusi