

Kecerdasan Buatan

Eri Sasmita Susanto, M.Kom

Dosen Informatika Universitas Teknologi Sumbawa



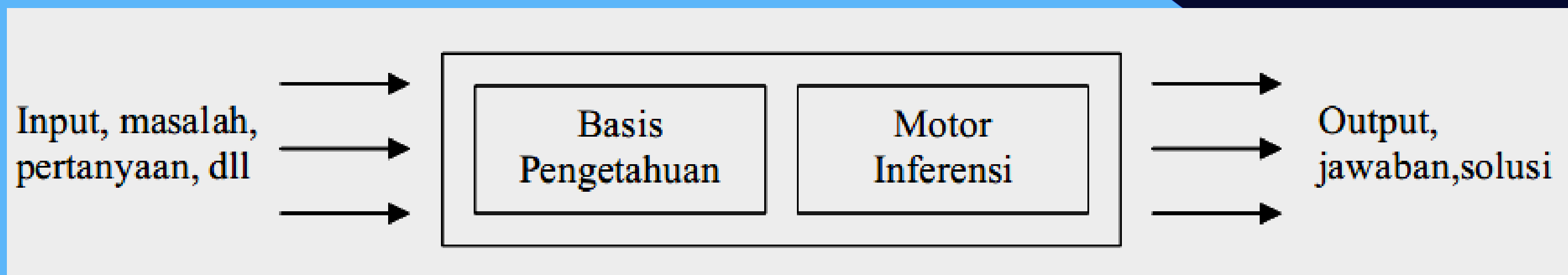
Masalah, Ruang Masalah dan Pencarian

Sistem Kecerdasan Buatan

2 bagian utama kecerdasan buatan

basis pengetahuan (*knowledge base*): berisi fakta fakta, teori, pemikiran & hubungan antara satu dengan lainnya

motor inferensi (*inference engine*): kemampuan menarik kesimpulan berdasarkan basis pengetahuan



Sistem untuk Penyelesaian Masalah

- **MENDEFINISIKAN MASALAH DENGAN TEPAT.**
 - Pendefinisian ini mencakup spesifikasi yang tepat mengenai keadaan awal dan solusi yang diharapkan.
- **MENGANALISIS MASALAH TERSEBUT SERTA Mencari beberapa teknik penyelesaian masalah yang sesuai.**
- **MEREPRESENTASIKAN PENGETAHUAN YANG PERLU UNTUK MENYELESAIKAN MASALAH TERSEBUT.**
- **MEMILIH TEKNIK PENYELESAIAN MASALAH YANG TERBAIK**

Ruang Masalah / Ruang Keadaan

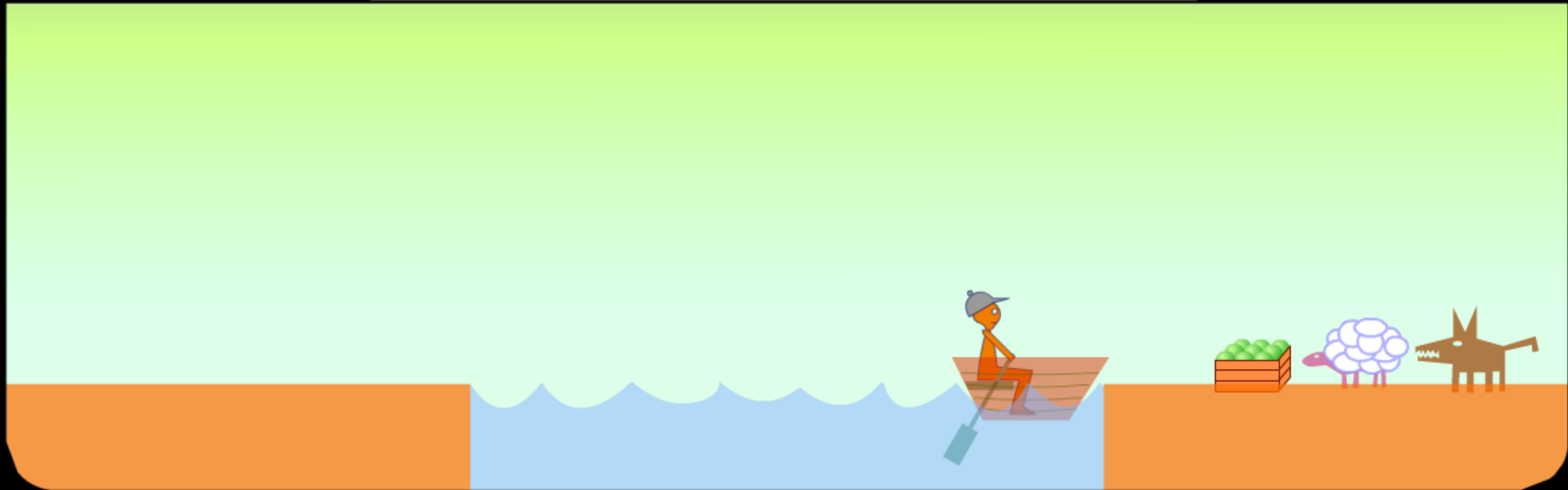
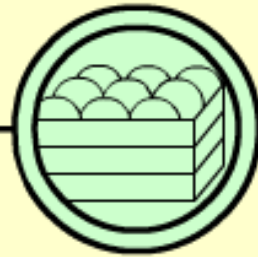
- Suatu ruang yang berisi semua keadaan yang mungkin
- Contoh : Kita dapat memulai bermain catur dengan menempatkan diri pada keadaan awal, kemudian bergerak dari satu keadaan ke keadaan yang lain sesuai dengan aturan yang ada, dan mengakhiri permainan jika salah satu telah mencapai tujuan

Ruang Masalah / Ruang Keadaan

- **MENDEFINISIKAN SUATU RUANG KEADAAN (STATE SPACE)**
- **MENETAPKAN SATU ATAU LEBIH KEADAAN AWAL (INITIAL STATE)**
- **MENETAPKAN SATU ATAU LEBIH TUJUAN (GOAL STATE)**
- **MENETAPKAN KUMPULAN ATURAN**

© 2002 Plastelina Games
All Rights Reserved!

A product of **PLASTELINA.NET**



Contoh : Masalah PETANI

- Identifikasi ruang keadaan
 - PERMASALAHAN INI DAPAT DILAMBANGKAN DENGAN (KAMBING, SERIGALA, SAYURAN, PERAHU).
 - CONTOH : DAERAH ASAL $(0,1,1,1)$ = DAERAH ASAL TIDAK ADA KAMBING, ADA SERIGALA, ADA SAYURAN, ADA PERAHU
- 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 = $(1,1,1,1)$

Contoh : Masalah PETANI

- 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

Contoh : Masalah PETANI

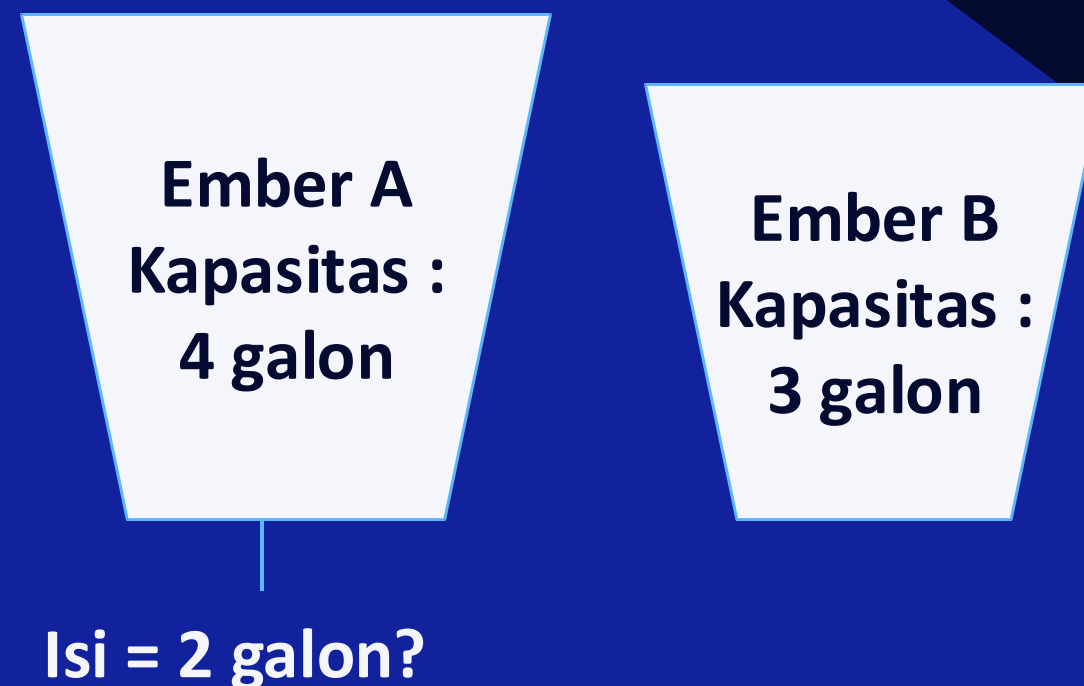
- SOLUSI

(kambing,serigala,sayuran,perahu)

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

Contoh : 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?



Contoh : Masalah Ember

- Penyelesaian :
- Identifikasi ruang keadaan (state space)
 - PERMASALAHAN INI DAPAT DIGAMBARAKAN 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 $Y \in \{0,1,2,3\}$
- Keadaan awal & tujuan
 - KEADAAN AWAL : KEDUA EMBER KOSONG = $(0,0)$
 - TUJUAN : EMBER 4 GALON BERISI 2 GALON AIR = $(2,N)$ DENGAN SEMBARANG N
- Keadaan ember
- Keadaan ember bisa digambarkan sebagai berikut :

Contoh : Masalah Ember

- **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.

Contoh : Masalah Ember

ke-		
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

Contoh : Masalah Ember

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

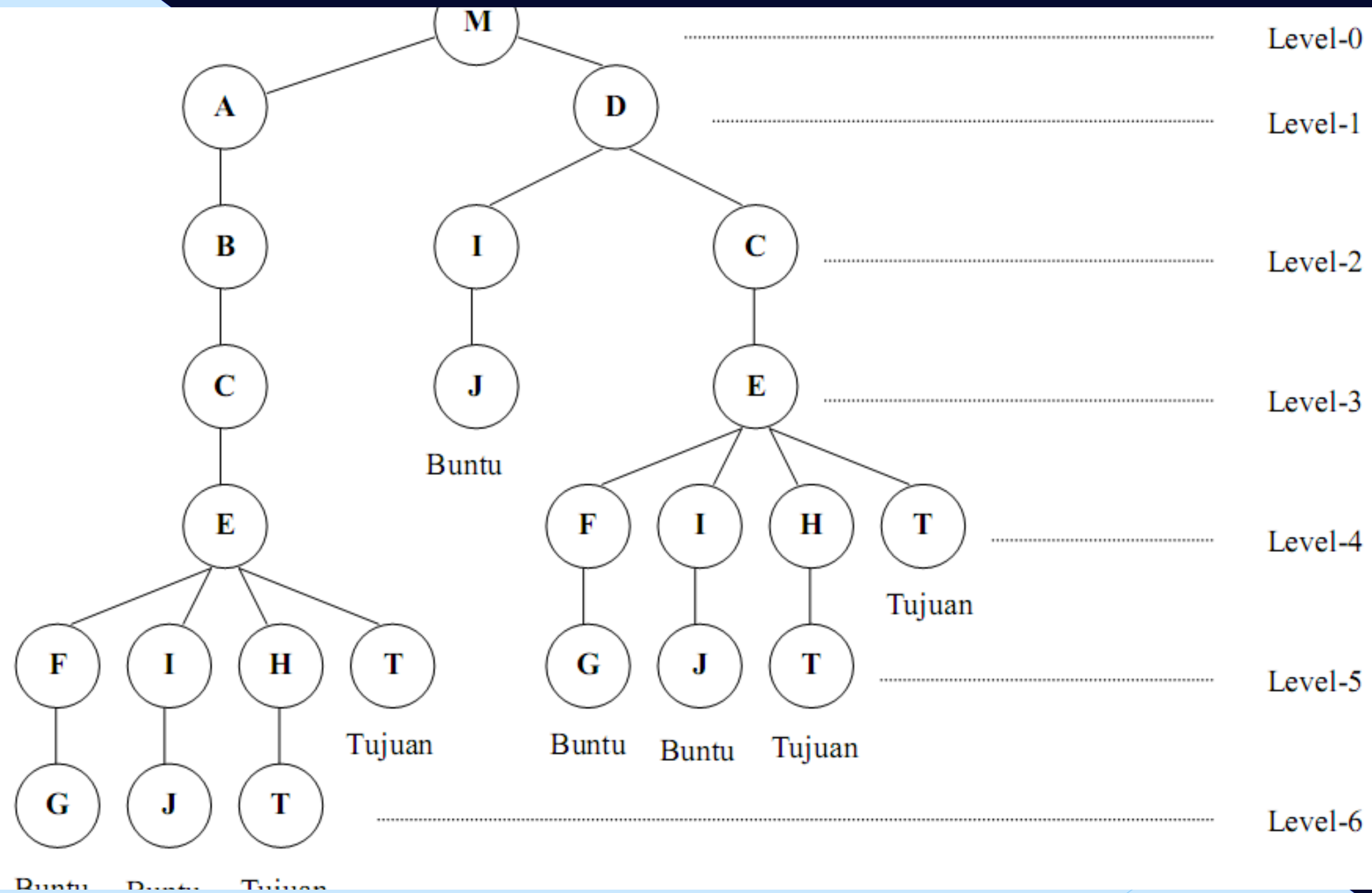
Representasi Ruang Keadaan

- **GRAPH KEADAAN**
 - node-node keadaan awal dan keadaan baru yang akan dicapai dengan menggunakan operator.
 - Node-node saling dihubungkan dengan menggunakan arc (busur) yang diberi panah untuk menunjukkan arah dari suatu keadaan ke keadaan berikutnya.

Pohon Pelacakan / Pencarian

- MENGGAMBARKAN KEADAAN SECARA HIRARKIS
- NODE PADA LEVEL-0 DISEBUT 'AKAR/ROOT' - MENUNJUKKAN KEADAAN AWAL & MEMILIKI BEBERAPA PERCABANGAN YANG TERDIRI ATAS BEBERAPA NODE YG DISEBUT 'ANAK/CHILD'
- NODE YG TIDAK MEMILIKI ANAK DISEBUT 'DAUN/LEAF' - MENUNJUKKAN AKHIR DARI SUATU PENCARIAN, DAPAT BERUPA TUJUAN YANG DIHARAPKAN (GOAL) ATAU JALAN BUNTU (DEAD END).

Pohon Pelacakan / Pencarian



Metode Pelacakan / Pencarian

- Hal penting dalam menentukan keberhasilan sistem cerdas adalah kesuksesan dalam pencarian.
- Pencarian = suatu proses mencari solusi dari suatu permasalahan melalui sekumpulan kemungkinan ruang keadaan (state space).
- Ruang keadaan = merupakan suatu ruang yang berisi semua keadaan yang mungkin.

Kriteria

- COMPLETENESS : APAKAH METODE TERSEBUT MENJAMIN PENEMUAN SOLUSI JIKA SOLUSINYA MEMANG ADA?
- TIME COMPLEXITY : BERAPA LAMA WAKTU YANG DIPERLUKAN?
- SPACE COMPLEXITY : BERAPA BANYAK MEMORI YANG DIPERLUKAN?
- OPTIMALITY : APAKAH METODE TERSEBUT MENJAMIN MENEMUKAN SOLUSI YANG TERBAIK JIKA TERDAPAT BEBERAPA SOLUSI BERBEDA?

Teknik Pencarian

- **PENCARIAN BUTA (BLIND SEARCH) : TIDAK ADA INFORMASI AWAL YANG DIGUNAKAN DALAM PROSES PENCARIAN**
 - Pencarian melebar pertama (Breadth – First Search)
 - Pencarian mendalam pertama (Depth – First Search)
- **PENCARIAN TERBIMBING (HEURISTIC SEARCH) : ADANYA INFORMASI AWAL YANG DIGUNAKAN DALAM PROSES PENCARIAN**
 - Pendakian Bukit (Hill Climbing)
 - Pencarian Terbaik Pertama (Best First Search)

Breadth First Search

- SEMUA NODE PADA LEVEL N AKAN DIKUNJUNGI TERLEBIH DAHULU SEBELUM MENGENJUNGI NODE-NODE PADA LEVEL $N+1$.
- PENCARIAN DIMULAI DARI NODE AKAR TERUS KE LEVEL 1 DARI KIRI KE KANAN, KEMUDIAN BERPINDAH KE LEVEL BERIKUTNYA DARI KIRI KE KANAN HINGGA SOLUSI DITEMUKAN.

Breadth First Search

- KEUNTUNGAN :
 - tidak akan menemui jalan buntu, menjamin ditemukannya solusi (jika solusinya memang ada) dan solusi yang ditemukan pasti yang paling baik
 - jika ada 1 solusi, maka breadth – first search akan menemukannya, jika ada lebih dari 1 solusi, maka solusi minimum akan ditemukan.
 - Kesimpulan : complete dan optimal
- KELEMAHAN :
 - membutuhkan memori yang banyak, karena harus menyimpan semua simpul yang pernah dibangkitkan. Hal ini harus dilakukan agar BFS dapat melakukan penelusuran simpul-simpul sampai di level bawah
 - membutuhkan waktu yang cukup lama

Depth First Search

- **PENCARIAN DILAKUKAN PADA SUATU SIMPUL DALAM SETIAP LEVEL DARI YANG PALING KIRI.**
- **JIKA PADA LEVEL YANG PALING DALAM TIDAK DITEMUKAN SOLUSI, MAKA PENCARIAN DILANJUTKAN PADA SIMPUL SEBELAH KANAN DAN SIMPUL YANG KIRI DAPAT DIHAPUS DARI MEMORI.**
- **JIKA PADA LEVEL YANG PALING DALAM TIDAK DITEMUKAN SOLUSI, MAKA PENCARIAN DILANJUTKAN PADA LEVEL SEBELUMNYA. DEMIKIAN SETERUSNYA SAMPAI DITEMUKAN SOLUSI.**

Depth First Search

- **KEUNTUNGAN :**
 - membutuhkan memori relatif kecil, karena hanya node-node pada lintasan yang aktif saja yang disimpan
 - Secara kebetulan, akan menemukan solusi tanpa harus menguji lebih banyak lagi dalam ruang keadaan, jadi jika solusi yang dicari berada pada level yang dalam dan paling kiri, maka DFS akan menemukannya dengan cepat (waktu cepat)
- **KELEMAHAN :**
 - Memungkinkan tidak ditemukannya tujuan yang diharapkan, karena jika pohon yang dibangkitkan mempunyai level yang sangat dalam (tak terhingga) / tidak complete karena tidak ada jaminan menemukan solusi
 - Hanya mendapat 1 solusi pada setiap pencarian, karena jika terdapat lebih dari satu solusi yang sama tetapi berada pada level yang berbeda, maka DFS tidak menjamin untuk menemukan solusi yang paling baik (tidak optimal).