

Masalah, Ruang Keadaan, Pencarian

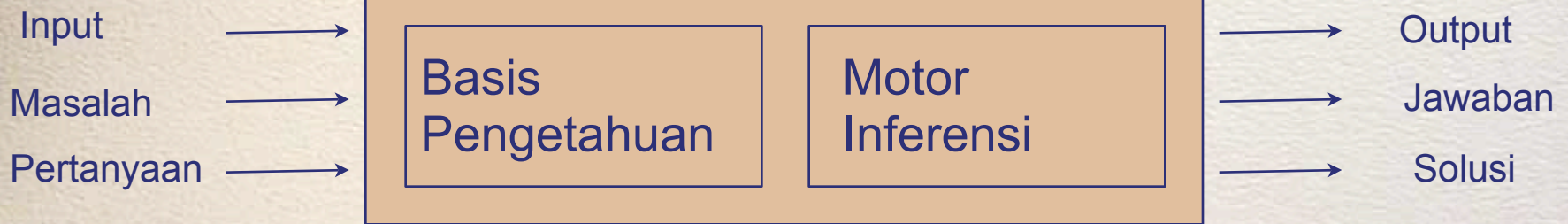
Kecerdasan Buatan

Pertemuan 2

Yudianto Sujana

Sistem AI

Komputer



Masalah

- Untuk membangun sistem yang mampu menyelesaikan masalah, perlu dipertimbangkan 4 hal:
 - Mendefinisikan masalah dengan tepat
 - Spesifikasi yang tepat mengenai keadaan awal
 - Solusi yang diharapkan
 - Menganalisis masalah serta mencari beberapa teknik penyelesaian masalah yang sesuai
 - Merepresentasikan pengetahuan yang perlu untuk menyelesaikan masalah
 - Memilih teknik penyelesaian masalah yang terbaik

Masalah Sebagai Ruang Keadaan

- Misalkan permasalahan yang dihadapi adalah Permainan Catur
- Maka harus ditentukan
 - Posisi awal pada papan catur
 - Aturan-aturan untuk melakukan gerakan secara legal
 - Tujuan (goal)

Posisi awal pada papan catur

- Posisi awal selalu sama



Aturan-aturan untuk melakukan gerakan secara legal

- Aturan-aturan sangat berguna untuk menentukan gerakan suatu bidak
- Untuk mempermudah
 - huruf (a,b,c,d,e,f,g,h) horizontal
 - angka (1,2,3,4,5,6,7,8) vertikal
- Contoh
 - bidak (e,2) ke (e,4)

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)

Aturan-aturan untuk melakukan gerakan secara legal



Tujuan (goal)

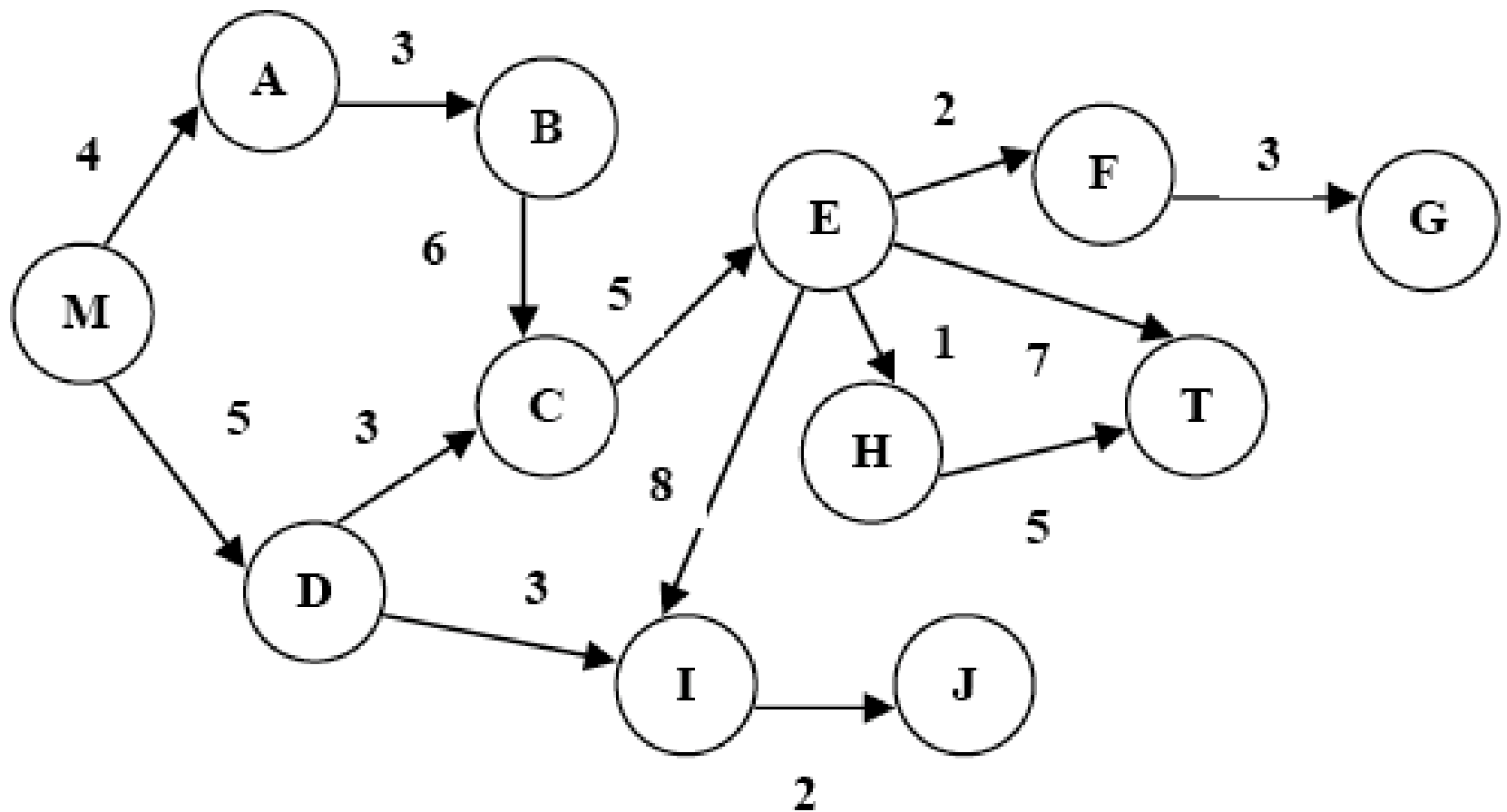
- Tujuan yang ingin dicapai adalah posisi pada papan catur yang menunjukkan kemenangan seseorang terhadap lawannya
- Ditandai dengan posisi Raja yang sudah tidak dapat bergerak lagi

Ruang Keadaan (State Space)

- Suatu ruang yang berisi semua keadaan yang mungkin
- Sehingga secara umum, untuk mendeskripsikan masalah dengan baik, harus:
 - Mendefinisikan suatu ruang keadaan
 - Menetapkan satu atau lebih keadaan awal
 - Menetapkan satu atau lebih tujuan
 - Menetapkan kumpulan aturan
- Ada beberapa cara untuk merepresentasikan Ruang Keadaan

Graph Keadaan

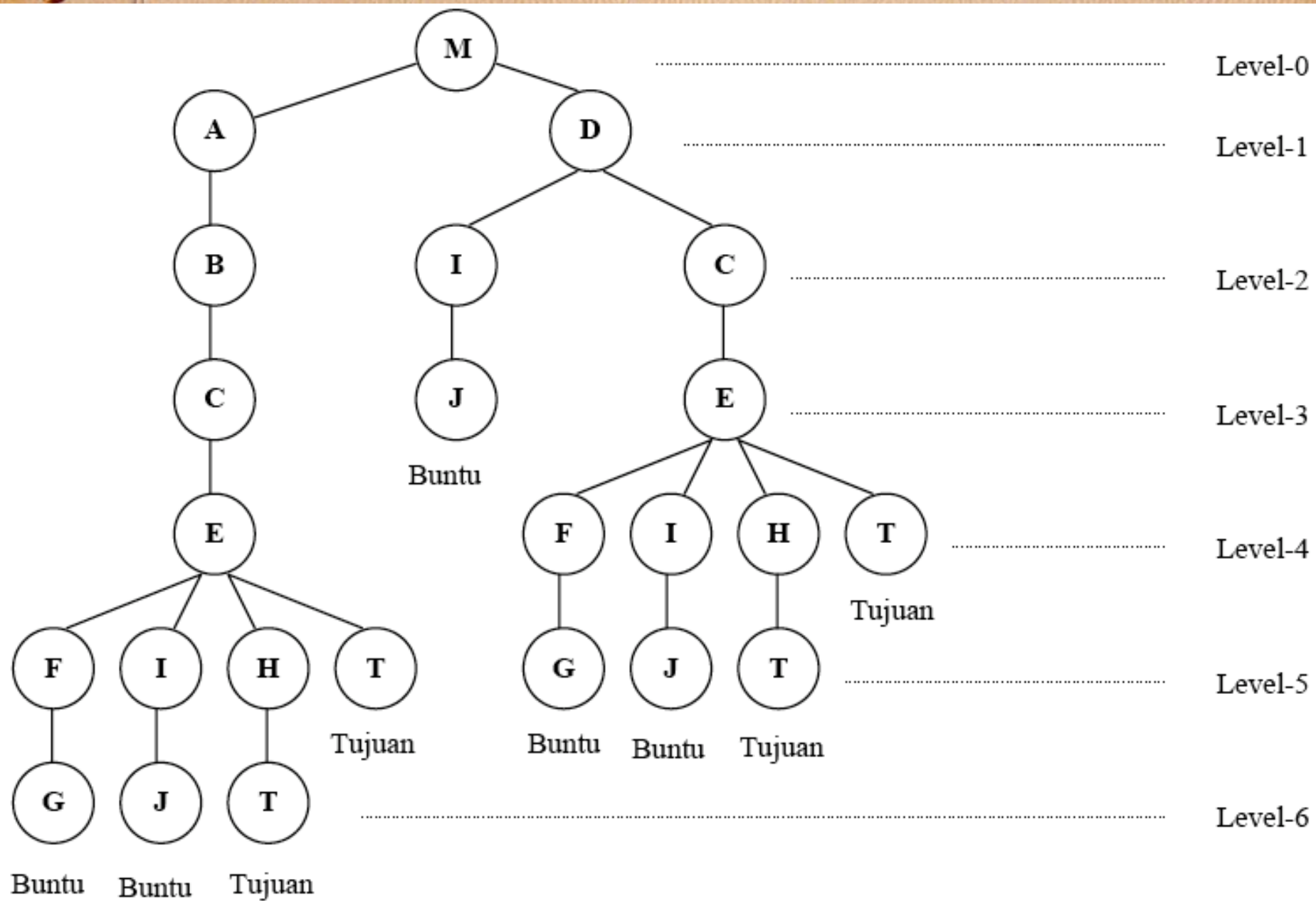
- Terdiri dari node-node yang menunjukkan keadaan yaitu keadaan awal dan keadaan baru yang akan dicapai dengan menggunakan operator
- Node-node saling dihubungkan dengan menggunakan arc (busur) yang diberi panah untuk menunjukkan arah

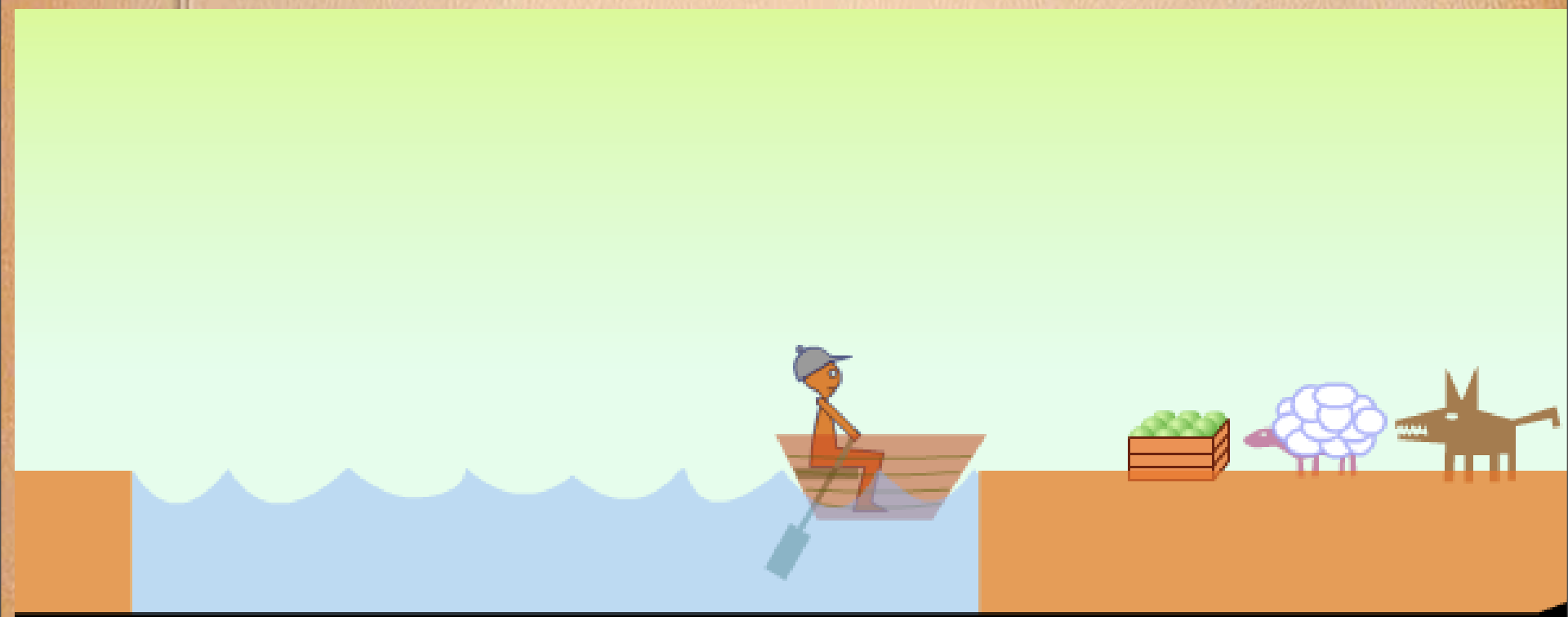


- Contoh lintasan dari M ke T
 - M-A-B-C-E-T, M-A-B-C-E-H-T, M-D-C-E-T, M-D-C-E-H-T
- Yang tidak sampai
 - M-A-B-C-E-F-G, M-A-B-C-E-I-J, M-D-C-E-F-G, M-D-C-E-I-J, M-D-I-J

Pohon Pelacakan

- Struktur pohon digunakan untuk menggambarkan keadaan secara hirarkis
- Terdiri dari beberapa node, level 0 disebut akar atau keadaan awal
- 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).





Contoh

- **Petani, Sayur, Kambing dan Serigala**
- Seorang petani akan menyeberangkan seekor kambing, seekor serigala, dan sayur-sayuran dengan sebuah boat yang melalui sungai.
- Boat hanya bisa memuat petani dan satu penumpang yang lain (kambing, serigala atau sayur-sayuran).
- Jika ditinggalkan oleh petani tersebut, maka sayur-sayuran akan dimakan oleh kambing, dan kambing akan dimakan oleh serigala.

Penyelesaian masalah secara umum

- Mendefinisikan suatu ruang keadaan;
- Menetapkan satu atau lebih keadaan awal;
- Menetapkan satu atau lebih tujuan;
- Menetapkan kumpulan aturan.

Penyelesaian

- Identifikasi ruang keadaan
 - Permasalahan ini dapat dilambangkan dengan (JumlahKambing, JumlahSerigala, JumlahSayuran, JumlahBoat).
 - Sebagai contoh: Daerah asal (0,1,1,1) berarti pada daerah asal tidak ada kambing, ada serigala, ada sayuran, dan ada boat.
- Keadaan Awal
 - Daerah asal: (1,1,1,1)
 - Daerah seberang: (0,0,0,0)
 - Tujuan
 - Daerah asal: (0,0,0,0)
 - Daerah seberang: (1,1,1,1)

Aturan-aturan

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

Penyelesaian

Daerah Asal	Daerah Seberang	Aturan yang 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