

MODUL 3

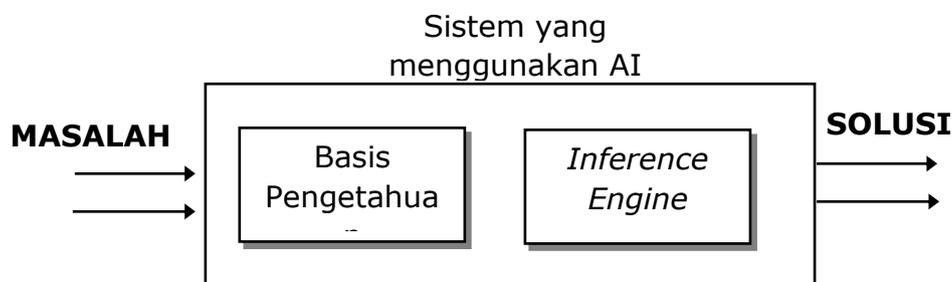
MASALAH DAN RUANG KEADAAN

3.1. Tujuan

Mahasiswa diharapkan mampu merepresentasikan suatu masalah ke dalam bentuk solusi. Meningkatkan pemahaman mahasiswa terhadap topologi tree sebagai salah satu solusi dalam memecahkan permasalahan Kecerdasan Buatan.

3.2. Dasar Teori

✚ Pada sistem yang menggunakan kecerdasan buatan, akan mencoba untuk memberikan output berupa solusi dari suatu masalah berdasarkan kumpulan pengetahuan yang ada (Gambar 3.1).



Gambar 3.1. Sistem yang Menggunakan Kecerdasan Buatan

✚ Pada sistem harus dilengkapi dengan sekumpulan pengetahuan yang ada pada basis pengetahuan. Sistem harus memiliki *inference engine* 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.

3.3 Mendefinisikan Masalah Sebagai Suatu Ruang Keadaan

✚ **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.

✚ Pada Permainan Catur, harus ditentukan :

1. Posisi awal pada papan catur;

Semua bidak diletakkan di atas papan catur dalam 2 sisi, yaitu kubu putih dan kubu hitam (Gambar 3.2).



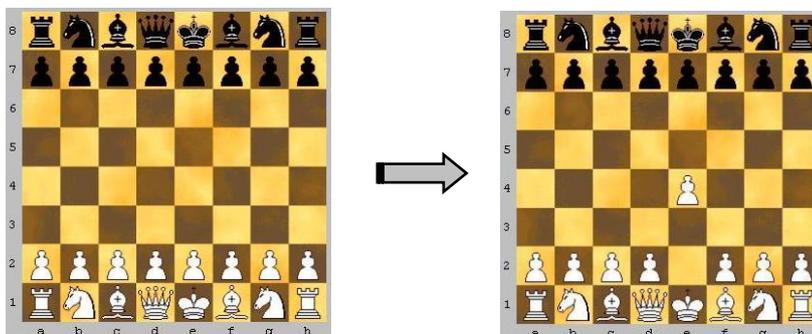
Gambar 3.2. Keadaan Awal Permainan Catur

2. Aturan-aturan untuk melakukan gerakan secara legal:

Aturan-aturan ini sangat berguna untuk menentukan gerakan suatu bidak, yaitu melangkah dari satu keadaan ke keadaan lain. Misalkan 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)**

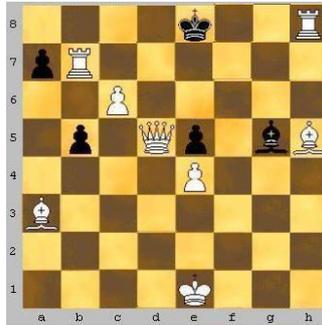
Seperti terlihat pada Gambar 3.3.



Gambar 3.3. Gerakan bidak catur

3. Tujuan (Goal)

Tujuan yang ingin dicapai adalah posisi pada papan catur yang menunjukkan kemenangan seseorang terhadap lawannya. yaitu posisi Raja yang sudah tidak dapat bergerak lagi. Gambar 3.4 merupakan salah satu contoh tujuan telah tercapai, yaitu Raja pada bidak hitam sudah tidak dapat bergerak lagi.



Gambar 3.4. Salah Satu Raja Mati

✚ Sehingga secara umum, untuk mendeskripsikan masalah dengan baik, harus:

1. Mendefinisikan suatu ruang keadaan;
2. Menetapkan satu atau lebih keadaan awal;
3. Menetapkan satu atau lebih tujuan;
4. Menetapkan kumpulan aturan.

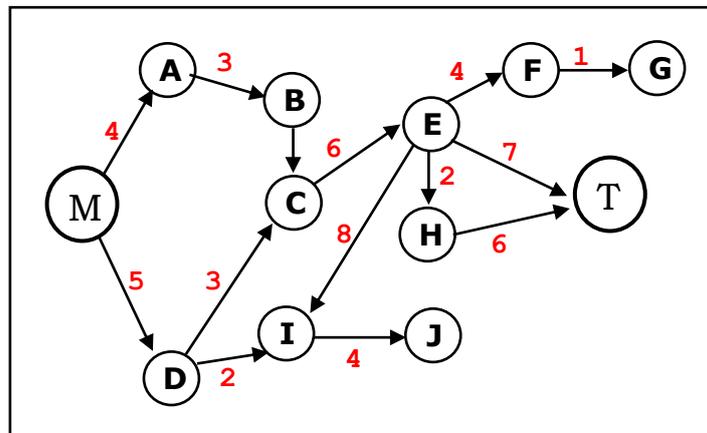
3.4 Representasi Ruang Keadaan

✚ Ada beberapa cara untuk merepresentasikan Ruang Keadaan, antara lain:

1. 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.
- Pada Gambar 2.5 menunjukkan graph berarah dengan node M menunjukkan keadaan awal, dan node T adalah tujuan. Pada Gambar 2.5 tersebut, kita dapat melihat ada lintasan 4 dari M ke T, yaitu:
 - ✓ M-A-B-C-E-T
 - ✓ M-A-B-C-E-H-T
 - ✓ M-D-C-E-T
 - ✓ M-D-C-E-H-T

- Pada graph ini, ada juga lintasan yang tidak sampai ke tujuan atau menemui jalan buntu, yaitu:
 - ✓ 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
- Gambar 3.5, tanpa mempertimbangkan arah, akan didapat siklus: D-C-E-I-D, node-node ini akan selalu berulang.

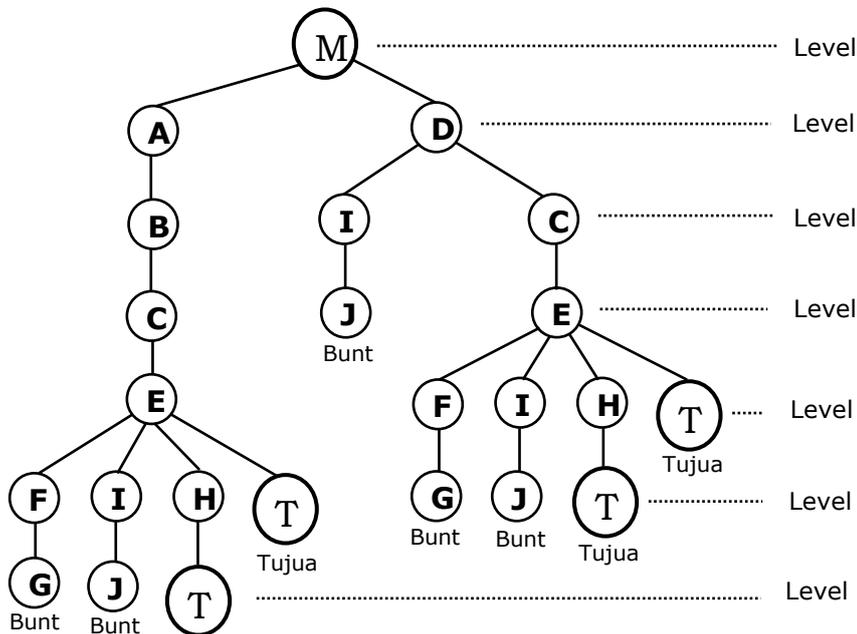


Gambar 3.5. Graph Keadaan

2. Pohon Pelacakan

- Untuk menghindari kemungkinan adanya proses pelacakan suatu node secara berulang, maka digunakan struktur pohon.
- Struktur pohon digunakan untuk menggambarkan keadaan secara hirarkis.
- Pohon juga terdiri-dari beberapa node. Node yang terletak pada level-0 disebut dengan nama “akar”. Node akar menunjukkan keadaan awal yang biasanya merupakan topik atau obyek. Node akar ini terletak pada level ke nol.
- Node akar memiliki beberapa percabangan yang terdiri-atas beberapa node successor yang sering disebut dengan nama “anak” dan merupakan node-node perantara. Namun jika dilakukan pencarian mundur, maka dapat dikatakan bahwa node tersebut memiliki predecessor.
- Node-node yang tidak memiliki anak sering disebut dengan nama node “daun” yang menunjukkan akhir dari suatu pencarian, dapat berupa tujuan yang diharapkan (*goal*) atau jalan buntu (*dead end*). Gambar 3.6 menunjukkan pohon pencarian untuk graph pada gambar 3.5 dengan 6 level.

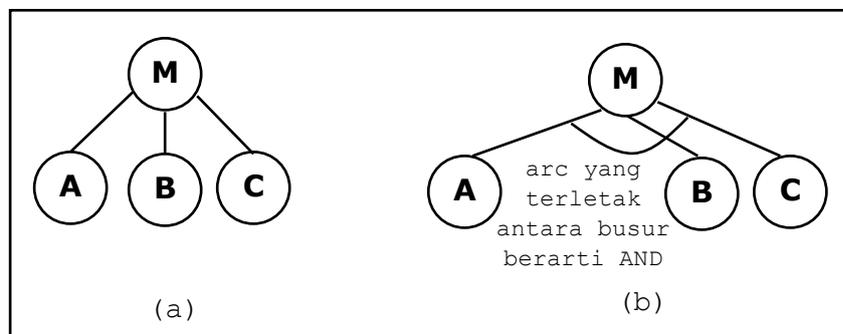
- Pada Gambar 3.6 di bawah ini, sudah tidak terlihat lagi adanya siklus, karena setiap node tidak diperbolehkan memiliki cabang kembali ke node dengan level yang lebih rendah



Gambar 3.6. Struktur Pohon

3. Pohon AND/OR

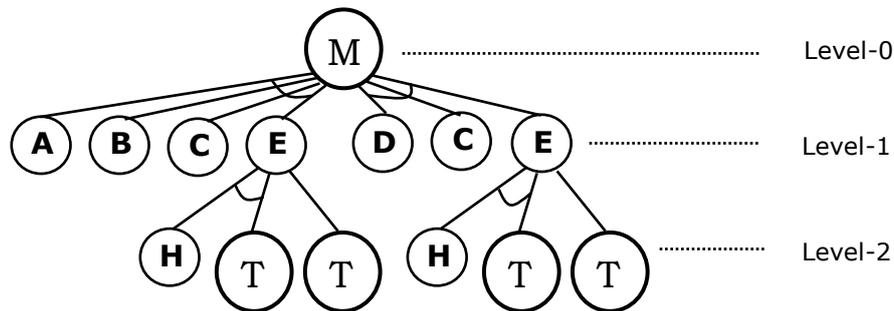
- Pada Gambar 3.7a terlihat ada suatu masalah M yang hendak dicari solusinya dengan 3 kemungkinan yaitu A, B atau C. Artinya, masalah M bisa diselesaikan jika salah satu dari subgoal A, B, atau C tidak terpecahkan.
- Lain halnya dengan gambar 3.7b, masalah M hanya dapat diselesaikan dengan A AND B AND C. Dengan kata lain, untuk memecahkan masalah M, maka harus dipecahkan subgoal A, B dan C terlebih dahulu. Pohon semacam ini disebut dengan Pohon AND/OR.



Gambar 3.7. Node AND/OR

- Gambar 3.8 memperlihatkan pencapaian tujuan pada graph Gambar 2.5 dengan menggunakan Pohon AND/OR. Dengan menggunakan pohon AND/OR, tujuan yang

dicapai pada pohon (Gambar 3.6) sampai pada level-6 bisa dipersingkat hanya sampai pada level-2 saja.



Gambar 3.8. Pohon AND/OR

3.5. Contoh Kasus

✚ 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 :

1. 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.

2. Keadaan awal & tujuan

Keadaan awal, pada kedua daerah:

- a. Daerah asal: (1,1,1,1)
- b. Daerah seberang: (0,0,0,0)

Tujuan, pada kedua daerah:

- a. Daerah asal: (0,0,0,0)
- b. Daerah seberang: (1,1,1,1)

3. Aturan-aturan

Aturan-aturan dapat digambarkan seperti pada Tabel 3.1.

Tabel 3.1. Aturan-aturan masalah teko air

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

4. Solusi

Salah satu solusi yang bisa ditemukan dapat dilihat pada Tabel 3.2.

Tabel 3.2. Contoh solusi masalah petani, kambing, sayuran, dan serigala

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

Tugas

Pelajari dan analisa kasus petani, kambing, sayuran, dan serigala. Buatlah mekanisme pembuatan struktur agent.