

MASALAH, RUANG KEADAAN, DAN PENCARIAN



Sistem Cerdas dan Pendukung Keputusan

Komputasi Konvensional

- Kita memerintahkan komputer bagaimana menyelesaikan suatu masalah
- Terstruktur dan langkah demi langkah sampai komputer menyelesaikan suatu masalah
- Berdasar suatu algoritme, terusun jelas, kemudian algoritme tersebut di terapkan pada komputer

Cara Software AI bekerja

- Aspek penting dari kecerdasan adalah penyelesaian masalah berbasis tujuan (*goal-based problem solving*)
- Banyak masalah dapat dirumuskan sebagai pencarian sederetan aksi yang mengarah ke tujuan
- AI dapat melakukan penalaran dan menarik kesimpulan dari pengalamannya
- Hal itu dilakukan dengan teknik pelacakan (*searching*) dan pencocokan pola (*pattern matching*)
- Dari informasi awal software AI melacak basis pengetahuan untuk mencari pola-pola kondisi yang spesifik.
- Mencocokkan kriteria yang sesuai dengan basis pengetahuan yang dimilikinya

Pertimbangan pembangunan sistem

- Mendefinisikan masalah dengan tepat
- Menganalisis masalah tersebut serta mencari beberapa teknik penyelesaian
- Merepresentasikan pengetahuan untuk menyelesaikan masalah
- Memilih teknik penyelesaian masalah yang terbaik

Masalah

- Masalah utama dalam membangun sistem berbasis AI adalah bagaimana mengkonversi situasi yang diberikan ke dalam situasi yang lain yang di inginkan menggunakan sekumpulan operasi tertentu
- Ruang keadaan adalah suatu ruang yang berisi semua keadaan yang mungkin

Mendefinisikan masalah sebagai ruang keadaan

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

Problem 1

- Seorang petani akan menyebrangkan seekor kambing, seekor serigala dan sayur mayur dengan sebuah perahu melalui sungai.
- Perahu hanya bisa memuat petani dan satu penumpang lain.
- Jika Sayur dan Kambing ditinggal sendirian maka Sayur akan dimakan Kambing
- Jika Kambing dan Serigala ditinggal sendirian maka Kambing akan dimakan Serigala

- State space identification

Permasalahan ini dapat dilambangkan dengan (Jumlah kambing, jumlah serigala, jumlah sayuran, jumlah boat)

- Keadaan Awal
 1. Daerah asal (1,1,1,1)
 2. Daerah seberang (0,0,0,0)

- Tujuan
 1. Daerah asal (0,0,0,0)
 2. 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

Kambing, Serigala, Sayuran, Boat

Daerah Asal	Seberang	Aturan
(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

Problem 2

- 3 cewek dan 3 kanibal ada di sisi kiri sungai.
- Semuanya harus menyeberang ke sisi kanan sungai, menggunakan *boat yang hanya mampu* dinaiki 2 orang.
- Jumlah kanibal tidak boleh melebihi jumlah cewek, di sisi sungai, kapanpun.
- Bagaimana agar semuanya dapat menyeberang?
- Status:
 - Jumlah cewek di sisi kiri
 - Jumlah kanibal di sisi kiri
 - Di sisi mana *boat berada*.

- *Contoh: Status Awal: (3, 3, kiri)*
- *Operator: Suatu perpindahan yang diwakili oleh jumlah cewek dan jumlah kanibal dalam boat pada satu waktu. Ada 5 kemungkinan:*
 - (2 cewek, 0 kanibal)
 - (1 cewek, 0 kanibal)
 - (1 cewek, 1 kanibal)
 - (0 cewek, 1 kanibal)
 - (0 cewek, 2 kanibal)
- *Goal (Tujuan): (0, 0, kanan)*
- *Biaya jalur: jumlah penyeberangan.*

Problem 3

- Contoh klasik permasalahan dalam AI adalah *masalah 2 ember air*.
- "Diberikan 2 ember air yang berkapasitas 8 liter dan 6 liter. Kita dapat mengisi satu ember dari ember lainnya dan proses penakaran hanya dengan memakai 2 ember tersebut.
- **Bagaimana kita bisa mengisi tepat 4 liter dalam ember 8 liter?**
- Asumsikan tidak boleh ada air yang hilang dalam proses penakaran.

Langkah penyelesaian:

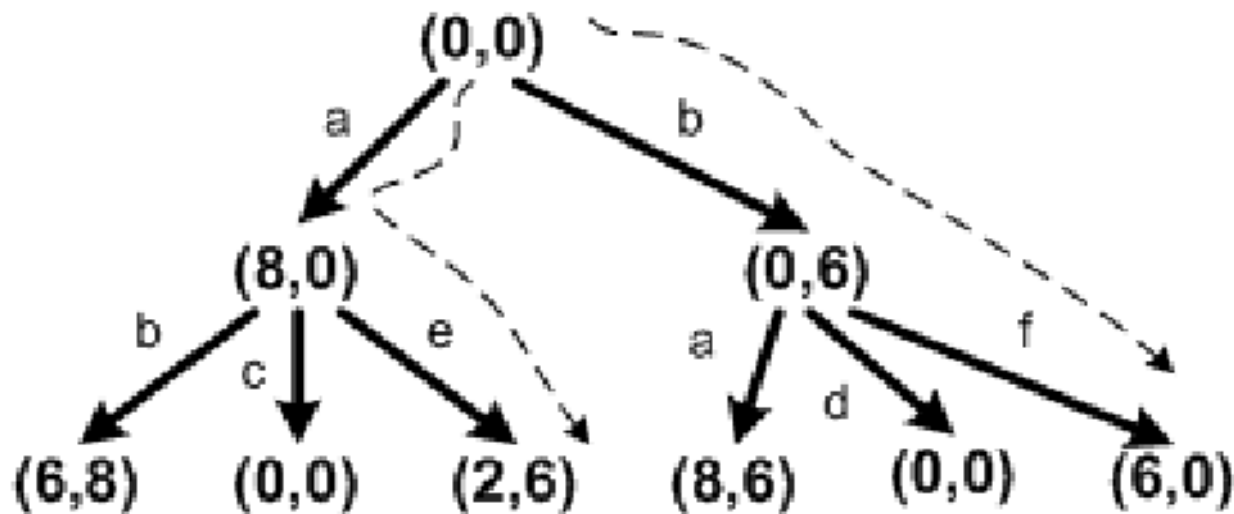
1. Menentukan aksi-aksi (problem space) yang bisa mengubah kondisi pada kedua ember dalam bentuk *rule* atau *tree-diagram*.

Contoh kemungkinan aksi-aksi:

- (a) Isi ember 8 liter.
- (b) Isi ember 6 liter.
- (c) Kosongkan ember 8 liter.
- (d) Kosongkan ember 6 liter.
- (e) Isikan seluruh air dalam ember 8 liter ke 6 liter.
- (f) Isikan seluruh air dalam ember 6 liter ke 8 liter.
- (g) Penuhi ember 8 liter dari 6 liter.
- (h) Penuhi ember 6 liter dari 8 liter.

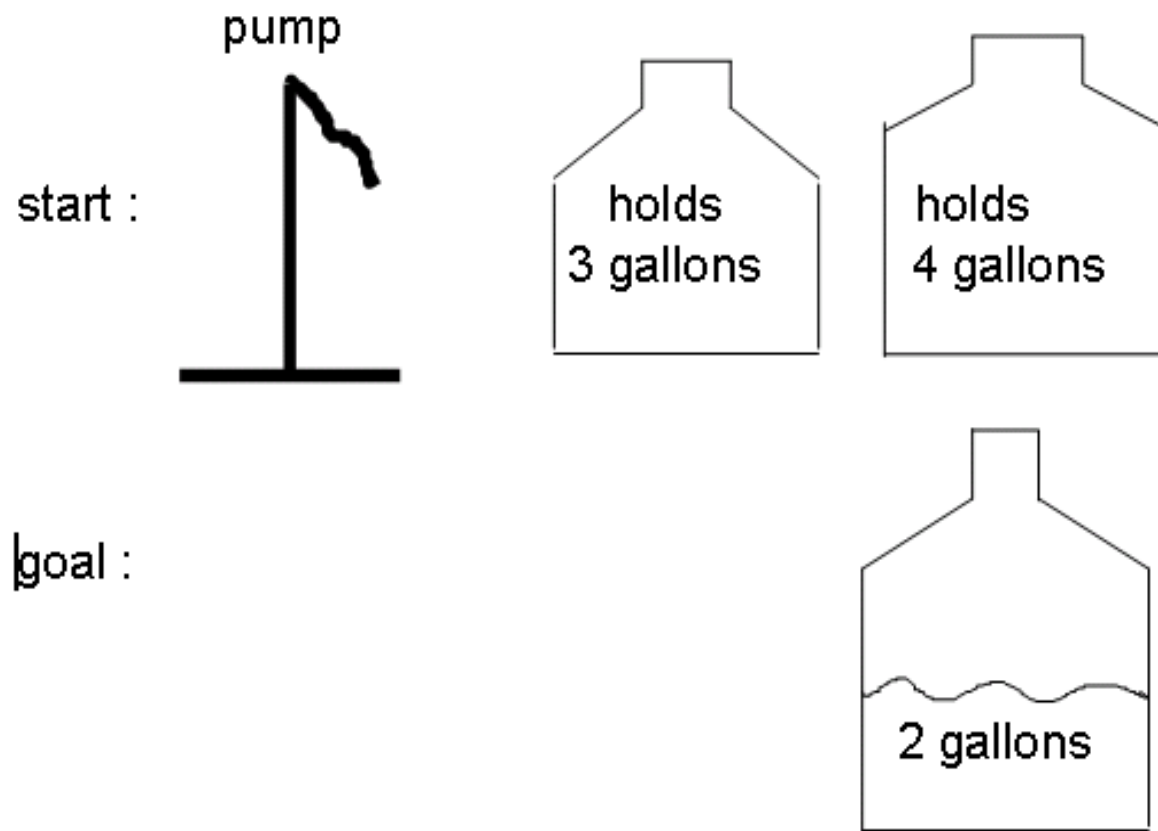
2. Menentukan urutan aksi untuk menghasilkan solusi, seperti:

$(0,0) \xrightarrow{b} (0,6) \xrightarrow{f} (6,0) \xrightarrow{b} (6,6) \xrightarrow{g} (8,4) \xrightarrow{c} (0,4) \xrightarrow{f} (4,0)$



Problem 4

- Bagaimana mendapatkan 2 galon air dari suatu bejana yang berisi maksimum 4 galon dan 3 galon air dalam bejana yang berisi 3 galon



Definisikan masalah

- Identifikasi Ruang Keadaan
Initial State $(x,y) = (0,0)$
X = bejana 4 galon
Y = bejana 3 galon
- Keadaan awal $(0,0)$
- Tujuan: goal state $(0,2)$

Aturan produksi

- Operasi yang mengubah suatu state ke state lainnya
- Semua aturan bisa mungkin karena pemikiran yang berbeda-beda
- Pertanyaannya bagaimana kita tahu bahwa aturan produksi yang kita pakai itu benar atau belum?

1.	(x,y) if $x < 4$	$(4,y)$	Isi penuh jurigen 4 galon
2.	(x,y) if $y < 3$	$(x,3)$	Isi penuh jurigen 3 galon
3.	(x,y) if $x > 0$	$(x-d,y)$	Buang sebagian air dari jurigen 4 galon
4.	(x,y) if $y > 0$	$(x,y-d)$	Buang sebagian air dari jurigen 3 galon
5.	(x,y) if $x > 0$	$(0,y)$	Kosongkan jurigen 4 galon
6.	(x,y) if $y > 0$	$(x,0)$	Kosongkan jurigen 3 galon
7.	(x,y) if $x+y \geq 4$ and $y > 0$	$(4,y-(4-x))$	Tuangkan air dari jurigen 3 galon ke 4 galon sampai jurigen 4 galon penuh
8.	(x,y) if $x+y \geq 3$ and $x > 0$	$(x-(3-y),3)$	Tuangkan air dari jurigen 4 galon ke 3 galon sampai jurigen 3 galon penuh
9.	(x,y) if $x+y \leq 4$ and $y > 0$	$(x+y,0)$	Tuangkan seluruh air dari jurigen 3 galon ke jurigen 4 galon
10.	(x,y) if $x+y \leq 3$ and $x > 0$	$(0,x+y)$	Tuangkan seluruh air dari jurigen 4 galon ke jurigen 3 galon
11.	$(0,2)$	$(2,0)$	Tuangkan 2 galon air dari jurigen 3 galon ke jurigen 4 galon
12.	$(2,y)$	$(0,y)$	Buang 2 galon air dalam jurigen 4 galon sampai habis

Contoh Solusi

metode pencarian yang tepat

Jumlah Air dalam Jurigen 4 galon	Jumlah Air dalam Jurigen 3 galon	Aturan Produksi yang di aplikasikan
0	0	-
0	3	2
3	0	9
3	3	2
4	2	7

Cara Representasi Ruang Keadaan

1. Graph Keadaan

Terdiri dari node-node yang menunjukkan keadaan yaitu keadaan awal dan keadaan baru yang akan dicapai dengan menggunakan operator

2. Pohon pelacakan

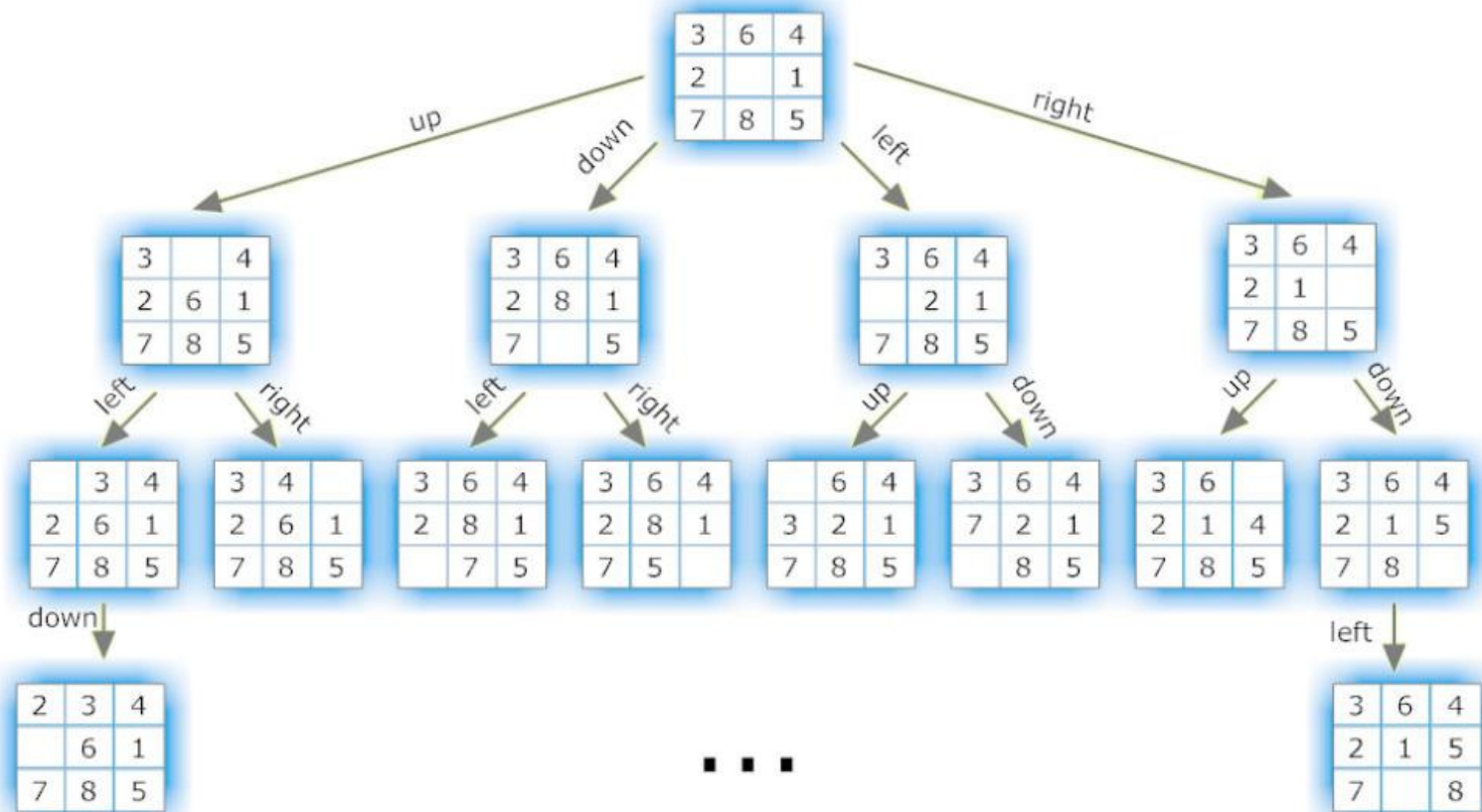
Struktur pohon digunakan untuk menggambarkan keadaan secara hirarkis. Terdiri dari beberapa node dan memiliki level

Pencarian Berbasis Pohon

- Himpunan semua jalur dalam ruang status dapat digambarkan sebagai graf *node-node yang terhubung*.
- Jejak jelajah pencarian dapat membentuk *tree (pohon)*
- Istilah penting:
 - *Root node (akar): mewakili node awal pencarian;*
 - *Leaf node (daun): node berhenti, tanpa anak;*
 - *Ancestor/descendant: node A adalah ancestor B jika A adalah induk B atau A adalah nenek dari induknya B. Jika A adalah ancestor B, maka B dikatakan descendant (keturunan) dari A;*
 - *Branching factor: jumlah anak maksimum dari suatu node daun dalam pohon pencarian;*
 - *Path (jalur): jalur dalam pohon pencarian yang mewakili jalur lengkap jika itu bermula dengan node awal dan berakhir dengan node tujuan. Jika tidak, disebut jalur *partial*.*

- Suatu node dapat ditampilkan oleh struktur data berikut:
 - Suatu deskripsi status;
 - Suatu pointer ke induk dari node;
 - Kedalaman dari node;
 - Operator yang membangkitkan node ini;
 - Biaya jalur (jumlah biaya operator) diperoleh dari status awal (mulai).
- **Kerugian:** dapat secara berulang mengunjungi node yang sama.
- **Solusi:** simpan semua node yang telah dikunjungi tetapi perlu tambahan memory.

Contoh



Pencarian Berbasis Graf

- Ruang status diwakili oleh graf $G(V,E)$, dimana V adalah himpunan node dan E adalah himpunan vertex dari satu node ke node lain.
- Informasi pada tiap node:
 - Suatu deskripsi status;
 - Induk dari node;
 - Operator yang membangkitkan node tersebut dari induknya;
 - Informasi lain.

- **Status awal:** Node yang ditunjuk sebagai node awal.
- **Ruang status:** Awalnya, node awal S , ditulis $V=\{S\}$. Kemudian S diexpand dan node yang dibangkitkannya ditambahkan ke V dan E . Proses ini berlanjut sampai tujuan ditemukan;
- **Jalur :** setiap node mewakili solusi parsial dari node awal untuk node yang diberikan. Dari node ini ada banyak jalur yang mungkin yang mempunyai jalur parsial ini sebagai prefix-nya;
- **Biaya jalur:** jumlah biaya vertex pada jalur solusi;
- **Uji tujuan:** Uji diterapkan terhadap suatu status untuk menentukan jika node terkait adalah tujuan & memenuhi semua kondisi tujuan;
- **Solusi:** sederetan operator yang dikaitkan dengan suatu jalur dalam ruang status dari node awal sampai node tujuan

Algoritma Pencarian Dasar

- Pencarian_Umum(masalah, strategi)
- Gunakan status awal dari masalah untuk mengawali pohon pencarian

Loop

If tidak ada node yang akan *diexpand*

Then return GAGAL;

Berdasarkan pada strategi, pilih node untuk perluasan;

terapkan uji tujuan;

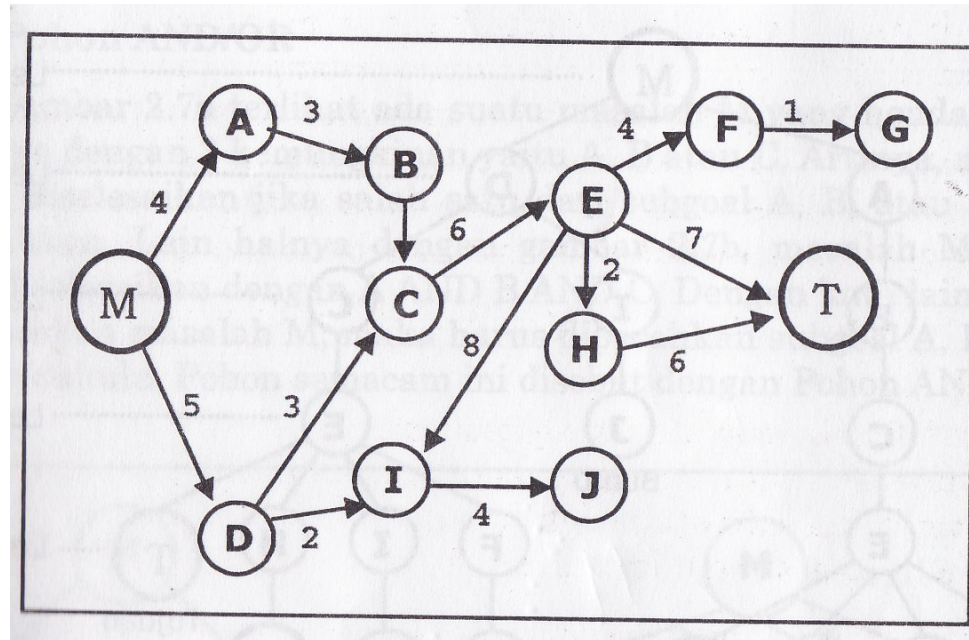
If node adalah status tujuan

then return SOLUSI

Else *expand node tersebut dan tambahkan node-node yang dihasilkan dan vertex ke pohon pencarian.*

End;

Contoh 1



- Tentukan keadaan awal , misal : M.
- Tentukan keadaan akhir : T