



# ARTIFICIAL INTELLIGENCE

---

Team teaching: Sri Winiarti, Andri Pranolo, dan Anna Hendri Sj

Andri Pranolo

- ▶ W : [apranolo.tif.uad.ac.id](mailto:apranolo.tif.uad.ac.id)
- ▶ M : 081392554050
- ▶ E : [andri.pranolo@tif.uad.ac.id](mailto:andri.pranolo@tif.uad.ac.id)

# POKOK BAHASAN

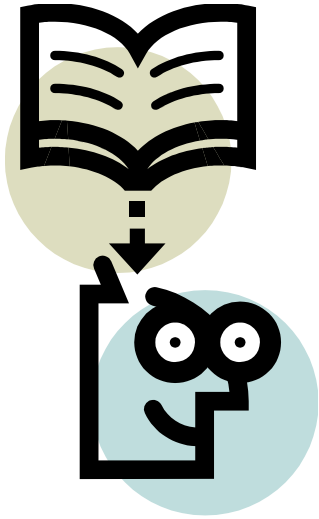
---



1. Masalah, Ruang Keadaan, dan Pencarian
2. Representasi Pengetahuan
3. Metode Inferensi
4. Penalaran (Penentuan Ketidakpastian dan Keyakinan)
5. Sistem Pakar
6. Pengolahan Bahasa Alami
7. Jaringan Syaraf Tiruan
8. Logika Fuzzy
9. Algoritma Genetika

# PERTEMUAN 2, 11 SEPTEMBER 2014

---



## Pokok Bahasan :

Masalah, Ruang Keadaan, dan Pencarian

## Outcome:

Mahasiswa dapat menganalisis masalah dan dapat menyelesaikan ruang masalah dengan metode pencarian buta

## Referensi:

- [1] Kusumadewi, S. *Artificial Intelligence: Teknik dan Aplikasinya*, Graha Ilmu, Yogyakarta, 2003
- [2] Kristanto, A. *Kecerdasan Buatan*, Grha Ilmu, Yogyakarta, 2004
- [3] S. Russell, and P Norvig, “*Artificial Intelligence: A Modern Approach*, 3rd Edition”, Prentice Hall, 2010, ISBN-13: 978-0-13-606733-7
- [4] Herawati, S. TT. Masalah, Ruang Keadaan, dan Pencarian  
<https://zheira83.files.wordpress.com/2014/08/pertemuan-2-masalah-ruang-keadaan-dan-pencarian.ppt>

# 1. MASALAH

---

Masalah dalam kecerdasan buatan adalah **masalah yang dapat dikonversi ke dalam ruang keadaan** (mempunyai keadaan awal dan keadaan tujuan) menggunakan sekumpulan aturan tertentu.

# MASALAH

---



- ✗ 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 Petani menyebrangkan serigala, sayur akan dimakan kambing
- ✗ Jika Petani menyebrangkan sayur maka kambing akan dimakan serigala

# IDENTIFIKASI RUANG KEADAAN

---

- ✗ Permasalahan dilambangkan dengan (**x**: petani, **y** : sayuran, **z**: kambing, **w**: serigala).
- ✗ Keadaan awal : Daerah asal= (**P,Sy,K,Sg**), dan Daerah seberang=(0,0,0,0).
- ✗ Keadaan awal: Daerah asal= (0,0,0,0), dan Daerah seberang=(**P,Sy,K,Sg**).

# ATURAN-ATURAN (RULES)

---

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

# SOLUSI

Daerah Asal	Daerah Seberang	Aturan yg digunakan
(P,Sy,K,Sg)	(0,0,0,0)	1
(0,Sy,0,Sg)	(P,0,K,0)	7
(P,Sy,0,Sg)	(0,0,K,0)	3
(0,Sy,0,0)	(P,0,K,Sg)	4
(P,Sy,K,0)	(0,0,0,Sg)	2
(0,0,K,0)	(P,Sy,0,Sg)	7
(P,0,K,0)	(0,Sy,0,Sg)	1
(0,0,0,0)	(P,Sy,K,Sg)	Solusi



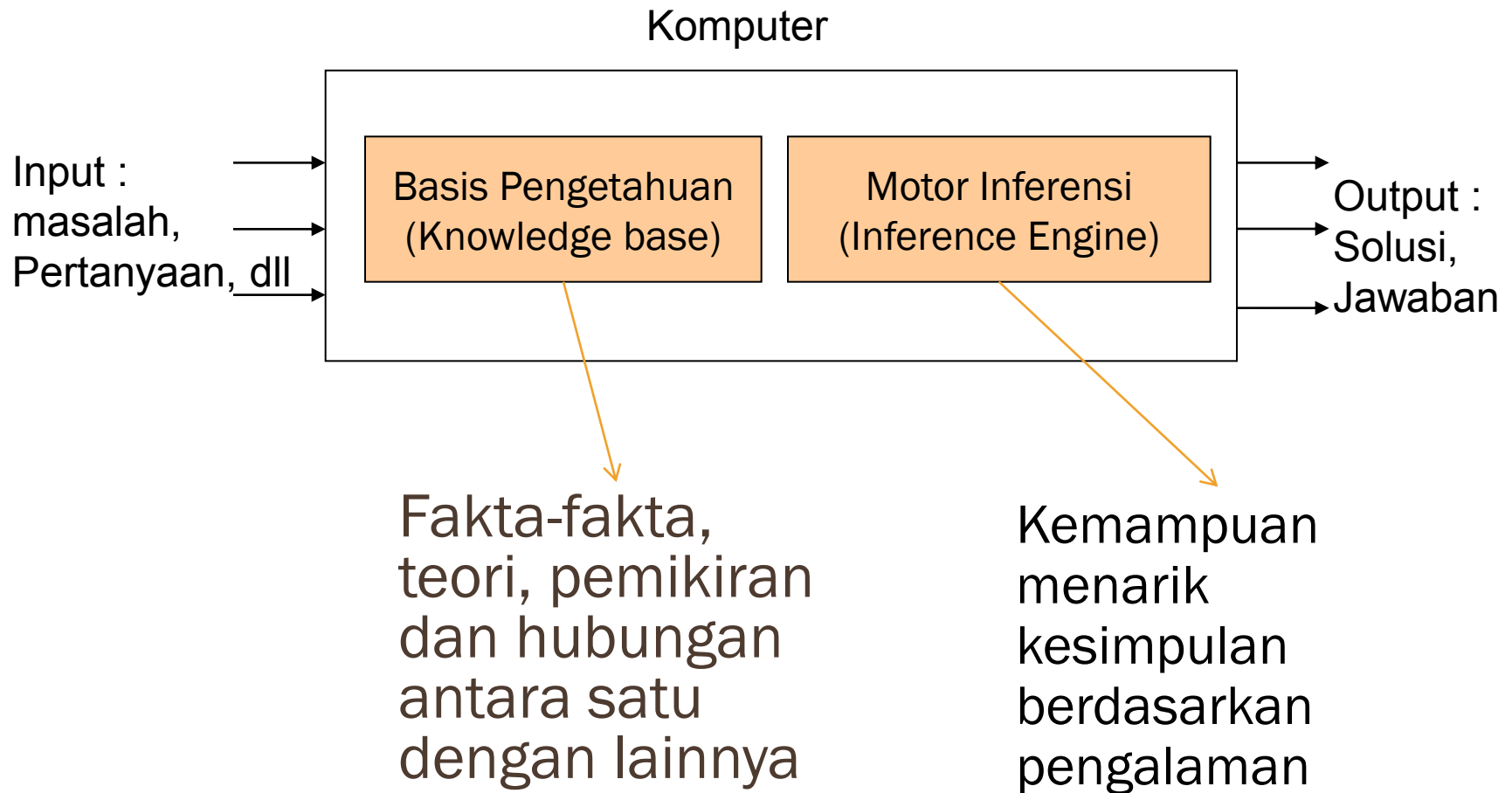
---

4 hal yang perlu diperhatikan untuk **membangun suatu sistem** yang mampu menyelesaikan masalah, yaitu :

- ✗ Mendefinisikan masalah dengan tepat
- ✗ Menganalisis masalah dan memberikan penyelesaian masalah yang sesuai
- ✗ Merepresentasikan pengetahuan untuk merepresentasikan masalah
- ✗ Memilih teknik penyelesaian masalah yang terbaik

## Sistem yang menggunakan AI

---



## 2. RUANG KEADAAN

---

Ruang keadaan (*State Space*) yaitu suatu ruang yang berisi semua keadaan yang mungkin.

Sehingga untuk **mendesripsikan masalah** yang baik, harus :

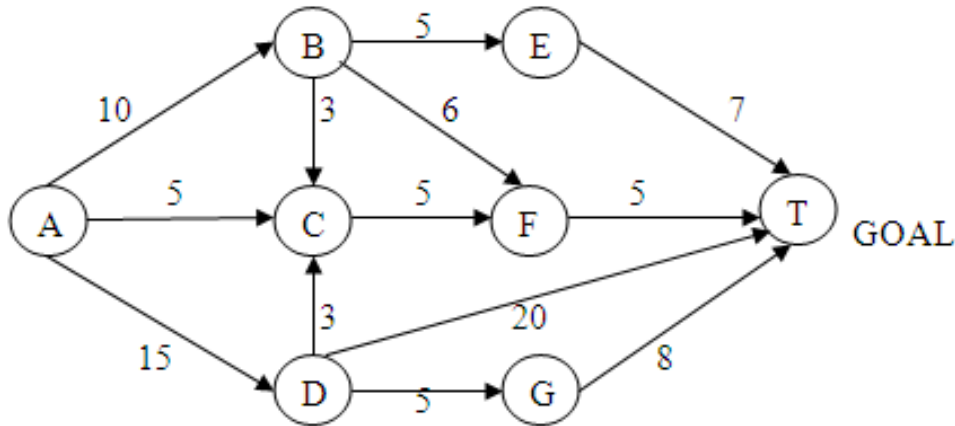
1. Mendefinisikan suatu ruang keadaan
2. Menetapkan satu atau lebih keadaan awal
3. Menetapkan satu atau lebih tujuan
4. Menetapkan kumpulan aturan

---

Ada beberapa cara untuk menyajikan ruang keadaan, antara lain :

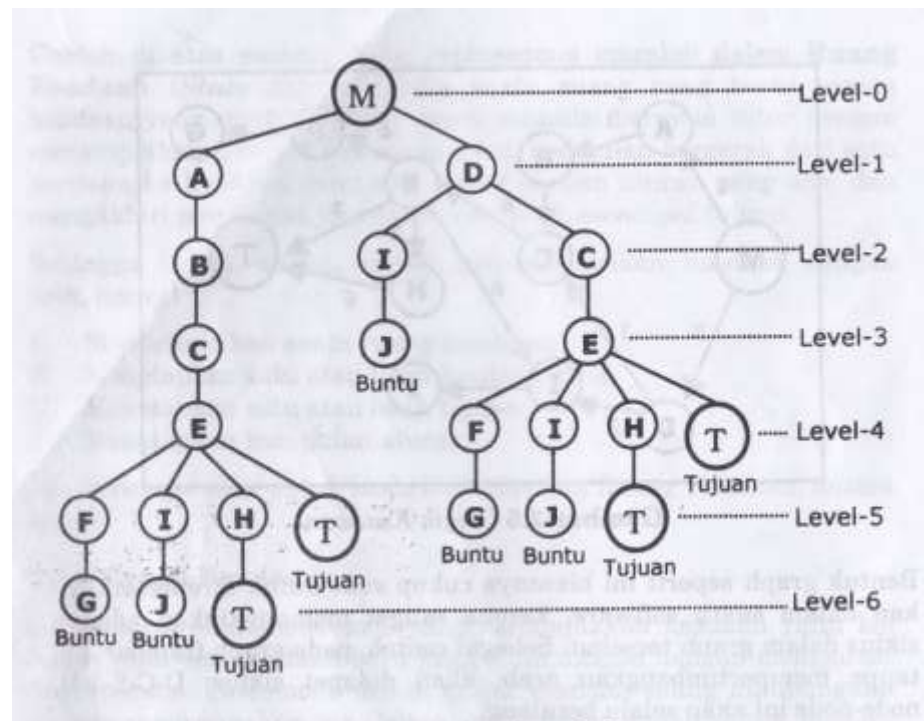
- ✕ Graph Keadaan
- ✕ Pohon pelacakan

# GRAPH KEADAAN



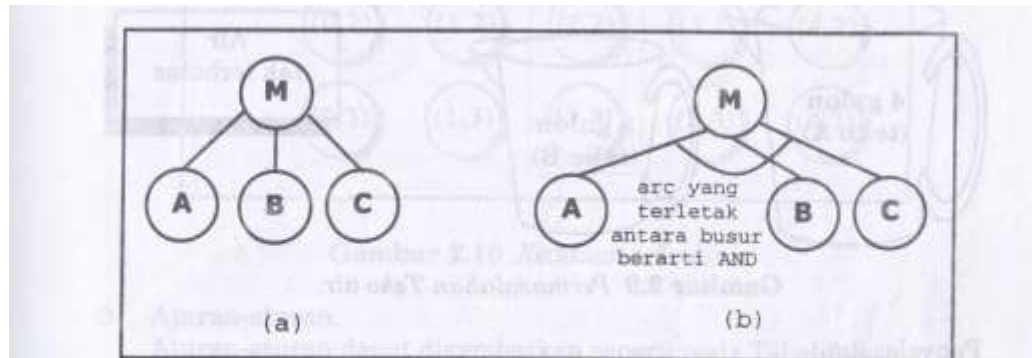
Dalam graph disamping kota yang dilaju adalah T, sehingga jalur yang dilewati dicari yang paling optimal yaitu : A – C – F – T = 15

# POHON PELACAKAN



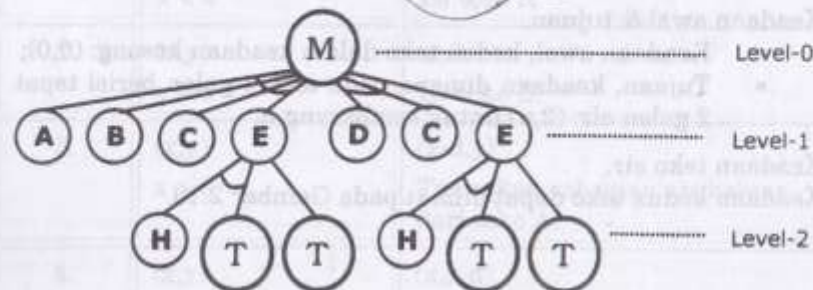
Ruang keadaan dalam *bentuk* graph pada umumnya dibuat untuk kasus-kasus yang menggambarkan solusi secara grafis, sedangkan pohon AND/ OR biasanya digunakan untuk kasus diagnosa. Namun kadang kala dapat digunakan untuk menggambarkan struktur pelacakan dari sistem komputer yang menggunakan konsep AI.

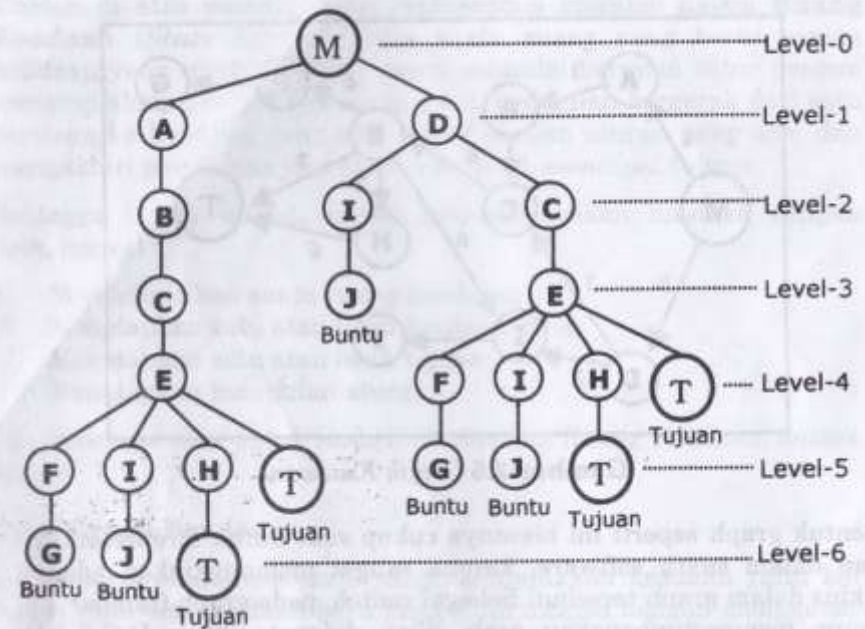
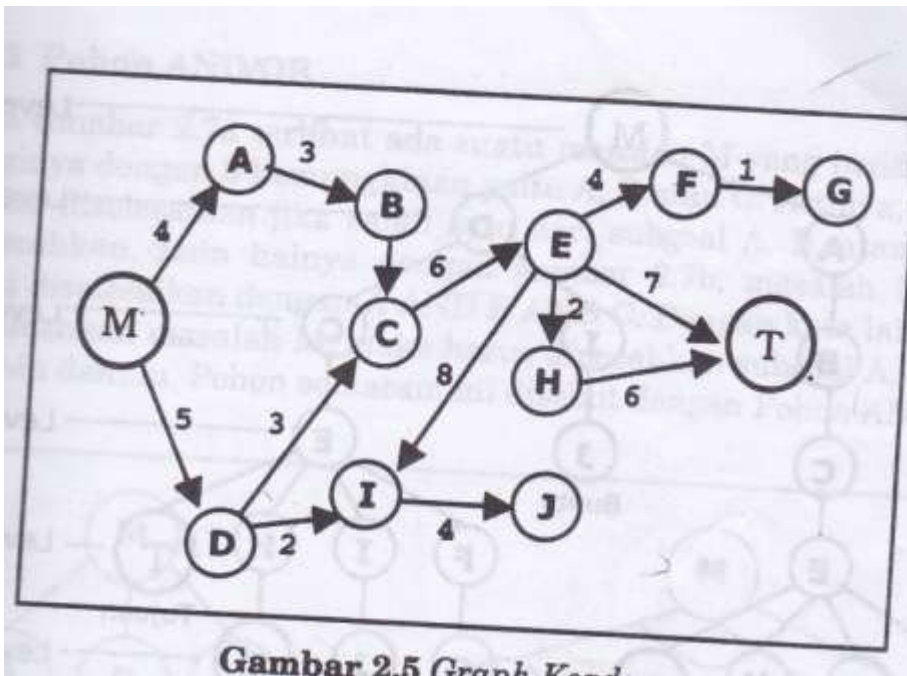
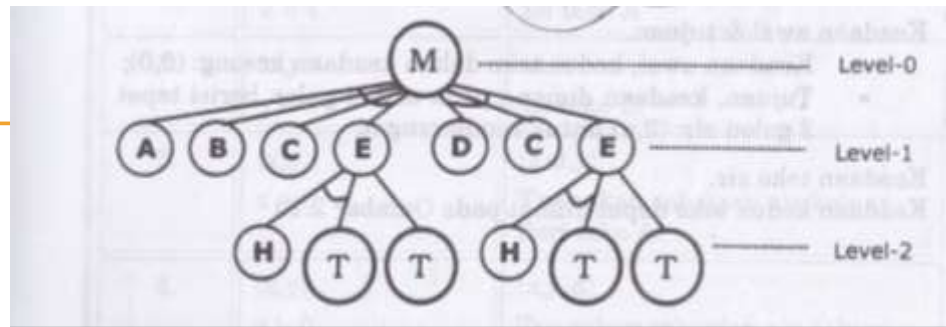
## POHON AND/OR



**Gambar 2.7 Node AND/OR**

Gambar 2.8 memperlihatkan pencapaian tujuan pada graph Gambar 2.5 dengan menggunakan Pohon AND/OR. Dengan menggunakan pohon AND/OR, tujuan yang dicapai pada pohon (Gambar 2.6) sampai pada level-6 bisa dipersingkat hanya sampai pada level-2 saja.







# METODE PENCARIAN DAN PELACAKAN

---

## 1. Pencarian Buta

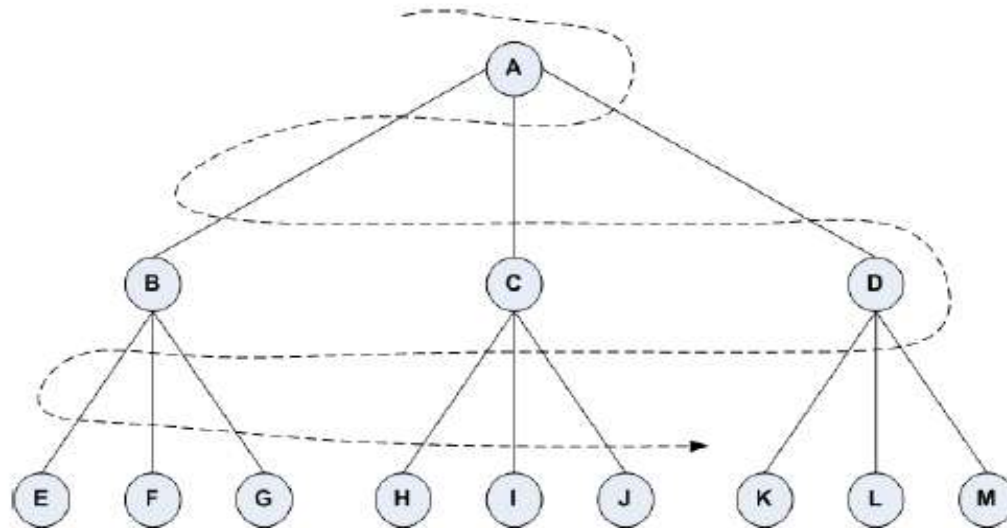
- × Breadth-First Search (Pencarian Melebar Pertama)
- × Depth-First Search (Pencarian Mendalam Pertama)
- + Depth-Limited Search (DLS)
- + Uniform Cost Search (UCS)
- + Iterative-Deepening Search (IDS)
- + Bi-Directional Search (BDS)

## 2. Pencarian Heuristik

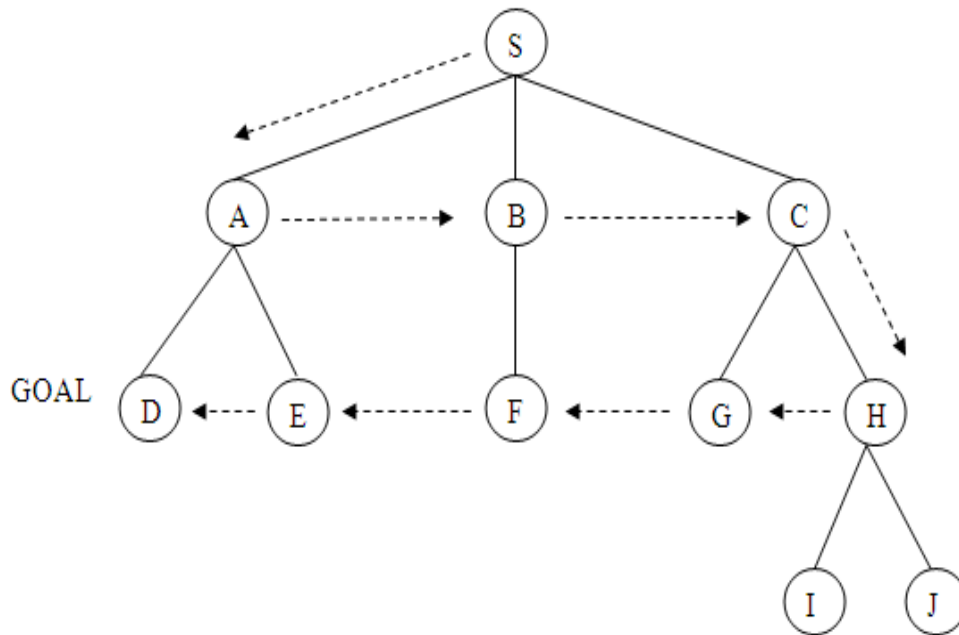
- × Generate and Test (Pembangkitan dan Pengujian)
- × Hill Climbing (Pendakian Bukit)
- × Best-First Search (Pencarian Terbaik Pertama)
- × Simulated Annealing
- × Greedy Best First Search
- × DLL

# BREADTH-FIRST SEARCH

*Breadth-first search (BFS)* melakukan proses searching pada semua node yang berada pada level atau hirarki yang sama terlebih dahulu sebelum melanjutkan proses searching pada node di level berikutnya.



# CONTOH BREADTH FIRST SEARCH



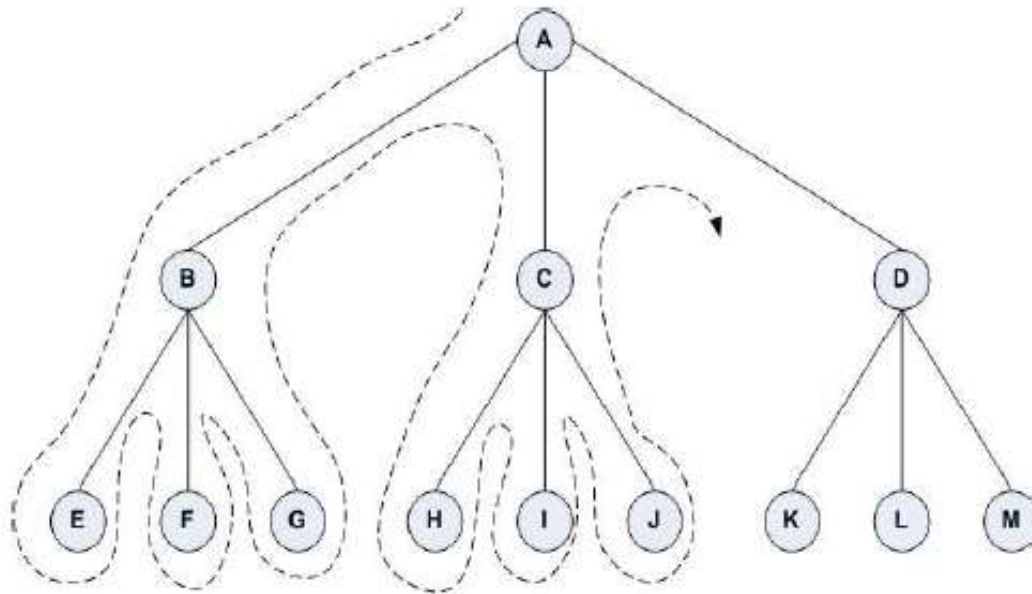
Dari gambar disamping maka solusinya adalah:

S – A – B – C – H – G – F – E – D

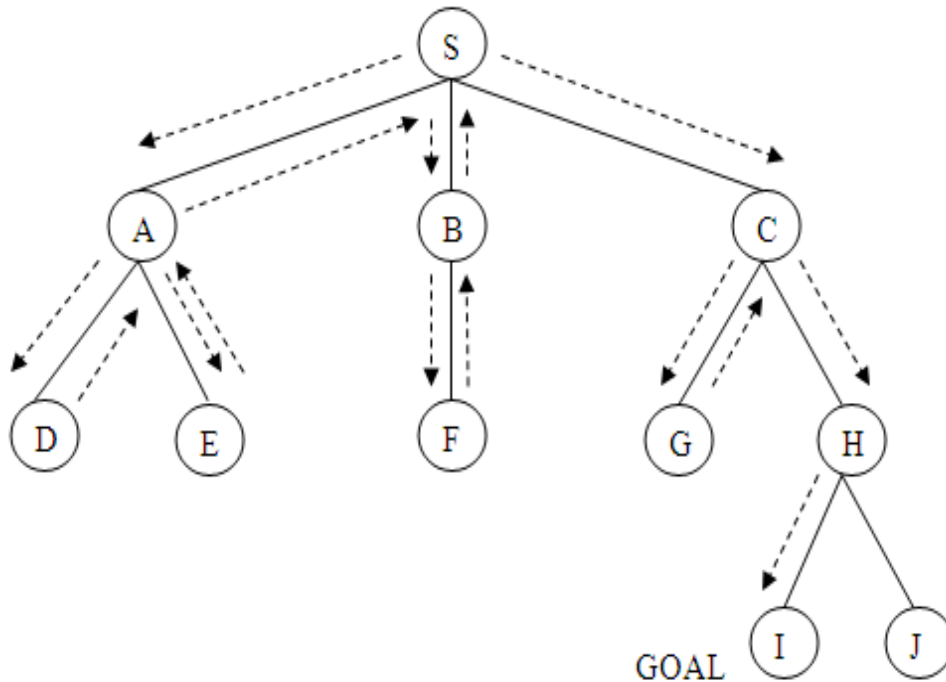
Prosesnya sama-sama lama dengan Depth First Search tetapi tingkat kesalahannya lebih rendah.

# DEPTH-FIRST SEARCH

Pencarian yang dilakukan pada semua anaknya sebelum dilakukan pencarian ke node-node yang selevel. Pencarian dimulai dari node akar ke level yang lebih tinggi. Proses ini diulangi terus hingga ditemukan solusinya.



# CONTOH DEPTH FIRST SEARCH

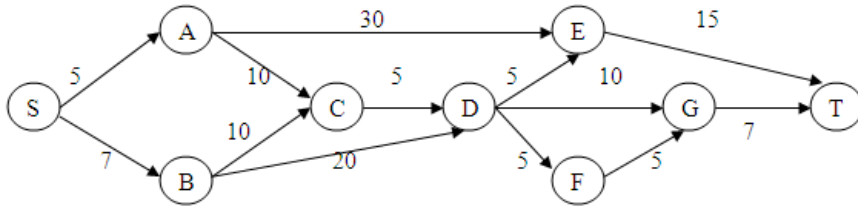


Dari gambar disamping  
maka solusinya adalah :  
S-A-D-A-E-A-S-B-F-B-S-C-  
G-C-H-I

Pada metode ini  
membutuhkan waktu  
yang lama tetapi tingkat  
kesalahannya kecil.

# TUGAS

---



Ubahlah graph disamping ini, kedalam pohon pelacakan

# TERIMA KASIH

---