

PERTEMUAN 4

DEPTH FIRST SEARCH

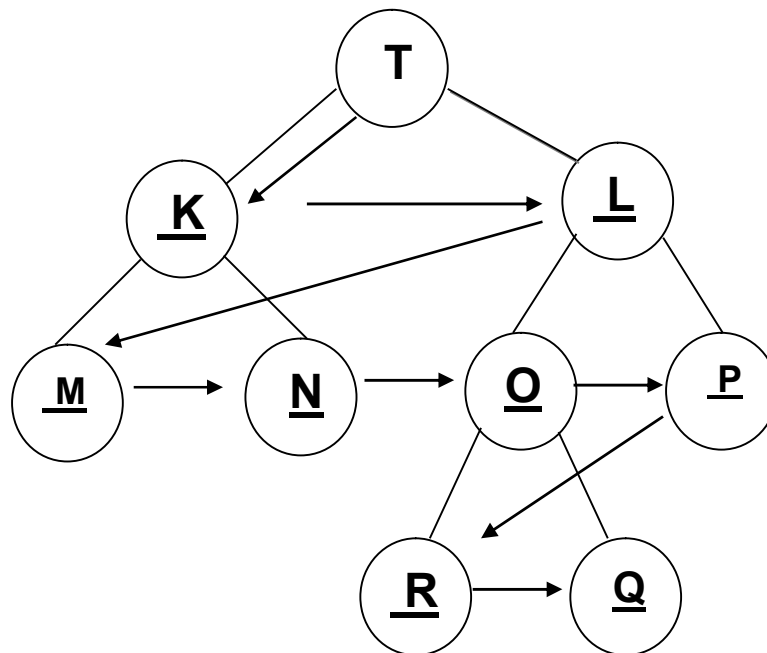
A. Tujuan Pembelajaran

Pada pertemuan ini akan dijelaskan penggunaan *depth first search*. Setelah menyelesaikan materi pada pertemuan ini, mahasiswa mampu memahami pencarian mendalam pertama (*Depth-first search*).

B. Uraian Materi

1. Memahami pencarian mendalam pertama (*Depth-first search*)

Depth-first search merupakan bagian dari blind search. Proses ini diulangi terus hingga ditemukannya goal (node G). Pada pencarian ini dilakukan searching pada semua anak(child) terlebih dahulu baru setelah itu dilakukan pencarian pada nodes atau tahap yang selevel. Searching diawali pada nodes T(akar) ke level di atasnya. Metode ini juga dilakukan dengan menelusuri setiap kemungkinan *path*.



Gambar 4.1 *Depth first search*

Gambar 4.1 merupakan Depth First Search (DFS) . Proses pada DFS algorithm ini Stack atau tumpukan digunakan untuk menyimpan titik(node) yang telah dikunjungi. *Algorithm DFS* mengatur stack melalui urutan misalnya masuk

terakhir keluar pertama. Tahapan proses Depth First Search DFS:

- Input Titik atau nodesAkar pada stack
- Gunakan nodes yang paling atas, setelah itu periksa apakah merupakan suatu penyelesaian.
- Apabila nodes termasuk penyelesaian maka pencarian berakhir
- Apabila nodes tidak termasuk penyelesaian maka input semua nodes child(anak) pada stack
- Apabila nodes tidak berisi atau kosong, maka periksa seluruh nodes, searching berakhir
- Coba kembali mulai dari tahap ke-2

Kelebihan :

- Hanya nodes yang aktif yang selalu disimpan pada stack sehingga memerlukan memori yang kecil.
- Proses ini tidak cukup melelahkan karena tidak mesti memeriksa seluruh nodes.

Kekurangan:

- Terkadang Goal(tujuan)tidak sesuai.
- Untuk setiap searching selalu mendapat 1 penyelesaian

Misalkan diketahui pohon pelacakan seperti Gambar 4.1 proses algorithm DFS terhadap permasalahan mulai T nodes (Start) hingga Q tujuan(Goal).

Interaksi ke-1

Input nodes T pada Stack setelah itu perlu ditandai

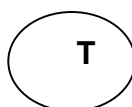
Tumpukan

Masuk dari pintu kiri, keluar lewat pintu kiri



Gambar 4.2 Stack *Depth first search* iterasi 1 tahap 1

Representasi keadaan :



Gambar 4.3 Representasi pengetahuan iterasi 1

Keluarkan **T** dari *Stack* dan cek.

Stack

Masuk dari pintu kiri, keluar lewat pintu kiri



Gambar 4.4 *Stack Depth first search* iterasi 1 tahap 2

Ternyata **T** \neq Goal

T memiliki anak(child) **K** serta **L**, setelah itu perlu ditandai, input nodes **K**, **L** pada tumpukan.

Stack

Masuk dari pintu kiri, keluar lewat pintu kiri

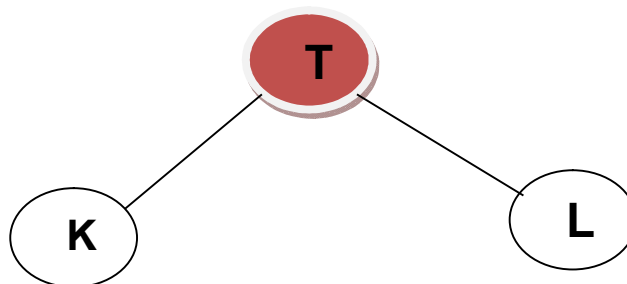


Gambar 4.5 *Stack Depth first search* iterasi 1 tahap 3

Karena **T** memiliki child(anak), input **T** sebagai penyelesaian sementara:

PenyelesaianSementara = [**T**]

Representasi suatu keadaan :



Gambar 4.6 Representasi pengetahuan iterasi 2

Iterasi ke-2 :

Stack

Masuk dari pintu kiri, keluar lewat pintu kiri



Gambar 4.7 Stack Depth first search iterasi 2 tahap 1

Solusi sementara = [T]

Keluarkan **K** dari Stack dan cek

Stack

Masuk dari pintu kiri, keluar lewat pintu kiri



Gambar 4.8 Stack Depth first search iterasi 2 tahap 2

Ternyata **K** ≠ Goal

K memiliki anak(child) **M** serta **N**, setelah itu perlu ditandai, input nodes M,N pada tumpukan

Stack(tumpukan)

Masuk dari pintu kiri, keluar lewat pintu kiri

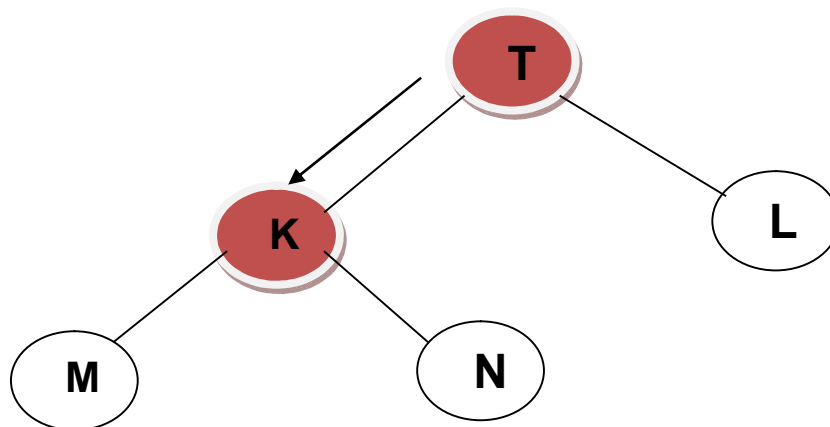


Gambar 4.9 Stack Depth first search iterasi 2 tahap 3

Karena K memiliki child(anak), input **K** sebagai penyelesaian sementara :

PenyelesaianSementara =[T K]

Representasi suatu keadaan :



Gambar 4.10 Representasi pengetahuan iterasi 3

Iterasi ke-3*Stack*

Masuk dari pintu kiri, keluar lewat pintu kiri

**Gambar 4.11** *Stack Depth first search* iterasi 3 tahap 1Keluarkan **M** dari *Stack* dan cek.*Stack*

Masuk dari pintu kiri, keluar lewat pintu kiri

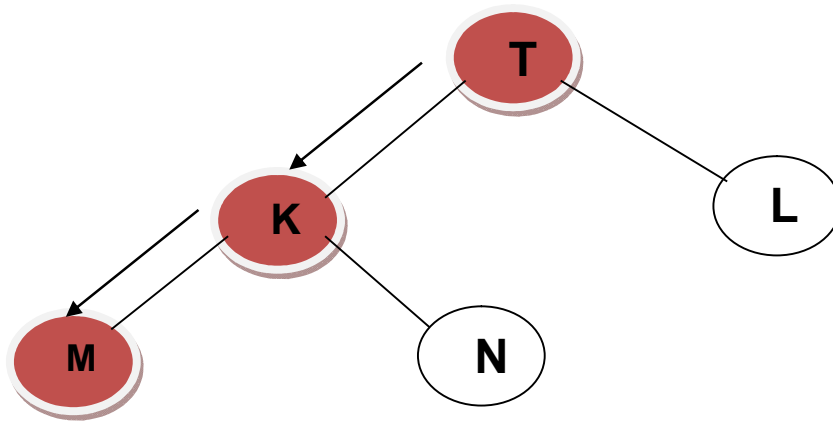
**Gambar 4.12** *Stack Depth first search* iterasi 3 tahap 2Ternyata **M** \neq Goal**M** tidak memiliki child(anak), sehingga tidak terdapat sesuatu yang di input pada *Stack*.*Stack*

Input dari sebelah kiri(left), output dari sebelah kanan

**Gambar 4.13** *Stack Depth first search* iterasi 3 tahap 3Karena **M** tidak memiliki child(anak), sehingga **M** tidak bisa di input sebagai penyelesaian sementara.

PenyelesaianSementara = [T K]

Representasi keadaan :



Gambar 4.14 Representasi pengetahuan iterasi 4

Iterasi ke-4 :

Stack

Masuk dari pintu kiri, keluar lewat pintu kiri



Gambar 4.15 Stack Depth first search iterasi 4 tahap 1

Solusi sementara = [T K]

Keluarkan **N** dari *Stack* dan cek

Stack

Masuk dari pintu kiri, keluar lewat pintu kiri



Gambar 4.16 Stack Depth first search iterasi 4 tahap 2

Ternyata **N** ≠ Goal

N Tidak memiliki anak(child), sehingga tidak ada yang di input pada *Stack*

Stack

Input dari sebelah kiri(left), output dari sebelah kanan

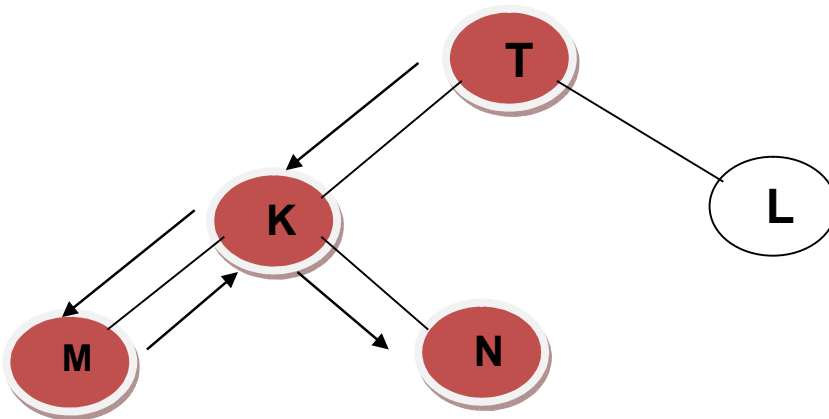


Gambar 4.17 Stack *Depth first search* iterasi 4 tahap 3

Karena **N** tidak memiliki anak(child), sehingga **N** tidak di input sebagai penyelesaian sementara:

PenyelesaianSementara =[**T K**]

Representasi keadaan :



Gambar 4.18 Representasi pengetahuan iterasi 5

Iterasi ke-5 :

Stack

Masuk dari pintu kiri, keluar lewat pintu kiri



Gambar 4.19 Stack *Depth first search* iterasi 5 tahap 1

Solusi sementara = [**T K**]

Keluarkan yang diberitanda pada *Stack* lalu pakailah untuk menghilangkan penyelesaian sementara sementara 1 huruf.

Stack

Masuk dari pintu kiri, keluar lewat pintu kiri



Gambar 4.20 Stack *Depth first search* iterasi 5 tahap 2

Solusi sementara = [T K] = [T]

Interasi -6

Stack

Masuk dari pintu kiri, keluar dari pintu kiri

Masuk dari pintu kiri, keluar lewat pintu kiri



Gambar 4.21 Stack *Depth first search* iterasi 6 tahap 1

Solusi sementara = [T]

Keluarkan L dari stack dan cek

Stack

Masuk dari pintu kiri, keluar lewat pintu kiri



Gambar 4.22 Stack *Depth first search* iterasi 6 tahap 2

Ternyata $L \neq \text{Goal}$

L memiliki anak(child) O dan P, perlu ditandai serta input nodes O serta P pada tumpukan.

Stack

Masuk dari pintu kiri, keluar lewat pintu kiri

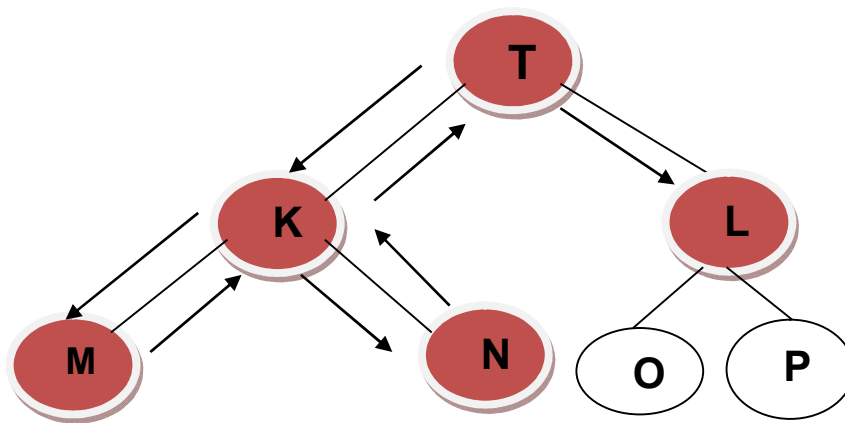


Gambar 4.23 Stack *Depth first search* iterasi 6 tahap 3

Karena L punya anak, maka L dimasukkan ke solusi sementara:

Solusi sementara = [T L]

Representasi keadaan:



Gambar 4.24 Representasi pengetahuan iterasi 7

Iterasi 7

Stack

Masuk dari pintu kiri, keluar lewat pintuk iri



Gambar 4.25 Stack Depth first search iterasi 7 tahap 1

Solusi sementara = [T L]

Keluarkan O dari stack dan cek

Stack

Masuk dari pintu kiri, keluar lewat pintu kiri



Gambar 4.26 Stack Depth first search iterasi 7 tahap 2

Ternyata $O \neq \text{Goal}$

O memiliki anak(child) R serta Q, perlu ditandai lalu input nodes R serta Q pada tumpukan.

Stack

Masuk dari pintu kiri, keluar lewat pintu kiri

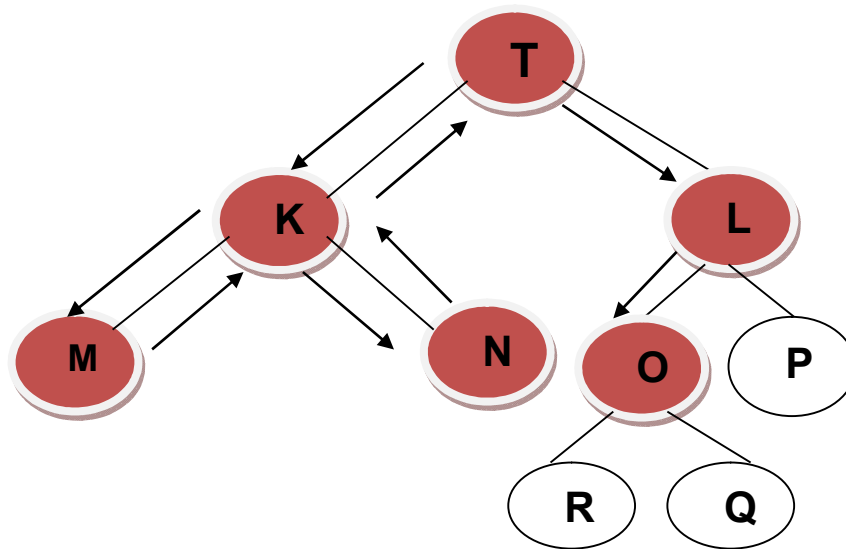


Gambar 4.27 Stack Depth first search iterasi 7 tahap 3

Karena O memiliki anak(child), lalu O diinput sebagai penyelesaian sementara:

Penyelesaian Sementara = [T L O]

Reprepsi keadaan



Gambar 4.28 Representasi keadaan iterasi 8

Iterasi 8

Stack

Masuk dari pintu kiri, keluar lewat pintu kiri



Gambar 4.29 Stack Depth first search iterasi 8 tahap 1

Solusi sementara = [T L O]

Keluarkan R dari stack dancek

Stack

Masuk dari pintu kiri, keluar lewat pintu kiri



Gambar 4.30 Stack Depth first search iterasi 8 tahap 2

Ternyata R ≠ Goal

R tidak memiliki child(anaksehingga R tidak di input pada stack

Stack

Input dari sebelah kiri(left), output dari sebelah kanan

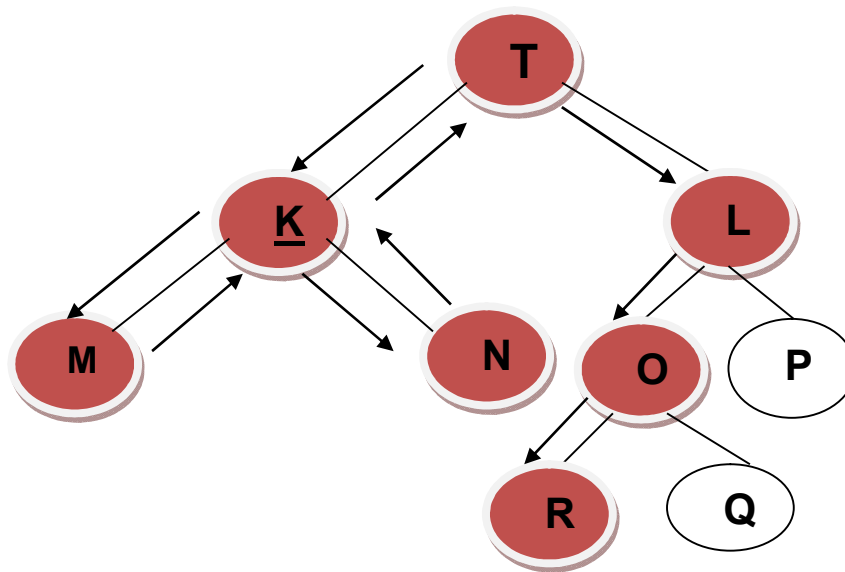


Gambar 4.31 Stack *Depth first search* iterasi 8 tahap 3

Karena R tidak memiliki anak(child), Sehingga R diinput sebagai penyelesaian sementara:

PenyelesaianSementara = [T L O]

Representasi keadaan :



Gambar 4.32 Representasi keadaan iterasi 9

Iterasi ke-9

Stack

Masuk dari pintu kiri, keluar lewat pintu kiri



Gambar 4.33 Stack *Depth first search* iterasi 9 tahap 1

Solusi sementara = [T L O]

Keluarkan Q dari stack dancek

Stack

Masuk dari pintu kiri, keluar lewat pintu kiri

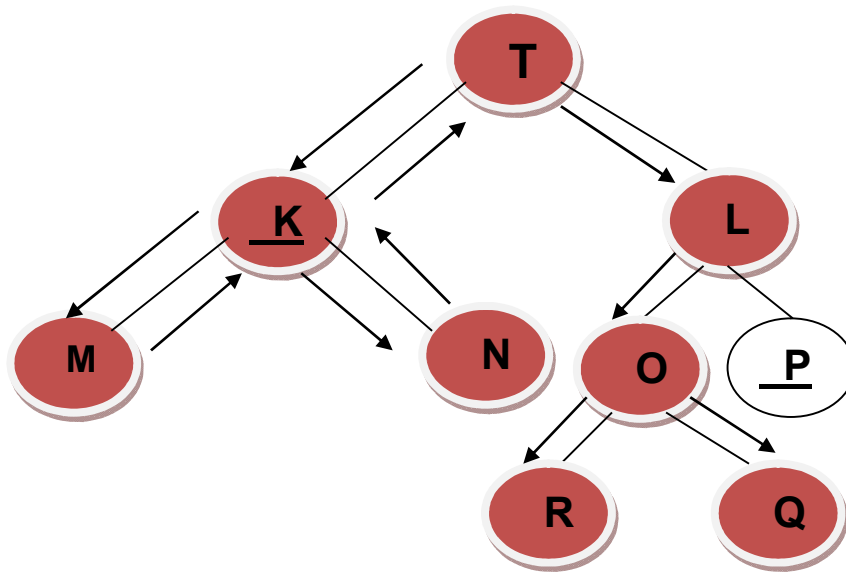


Gambar 4.34 Stack *Depth first search* iterasi 9 tahap 2

Tenyata Q= goal(tujuan), input sebagai penyelesaian sementara serta searching bkhirer

Penyelesaian= [T L O Q]

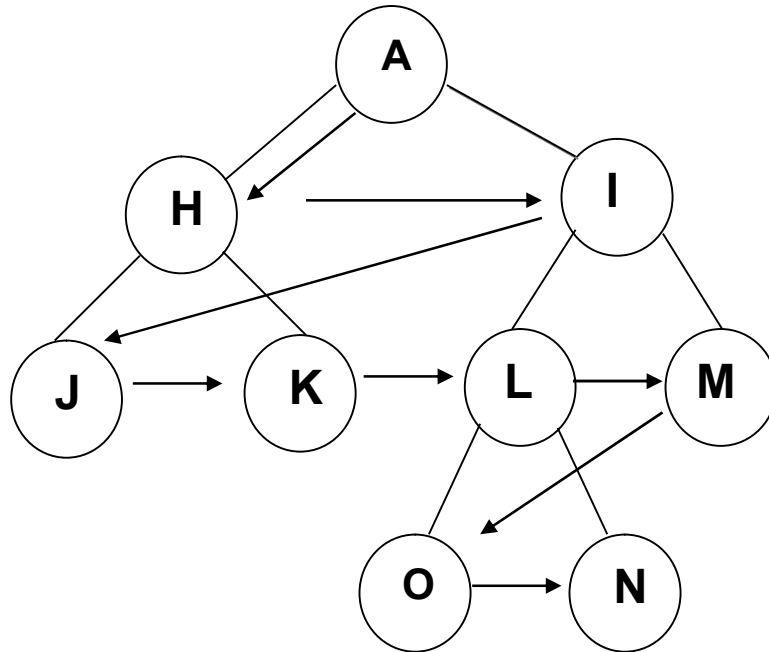
Representasi keadaan :



Gambar 4.35 Representasi keadaan akhir

C. Soal Latihan/ Tugas

1. Ilustrasikan gambar di bawah ini dengan metode *Depth first search (DFS)* dengan beberapa iterasi?

**D. Referensi**

- Siswanto. 2010. Kecerdasan Tiruan. Graha Ilmu edisi 2. Yogyakarta.
- Sutojo, T. Edy ,Mulyanto dan Suhartono,Vincent. 2010. Kecerdasan Buatan. Andi Offset. Yogyakarta.