

PERTEMUAN 3

METODE SEARCHING

A. Tujuan Pembelajaran

Pada pertemuan ini akan dijelaskan tentang “*blind search*” dan “*heuristic search*” dengan iterasi. Anda harus mampu menjelaskan pencarian buta (*blind search*).

B. Uraian Materi

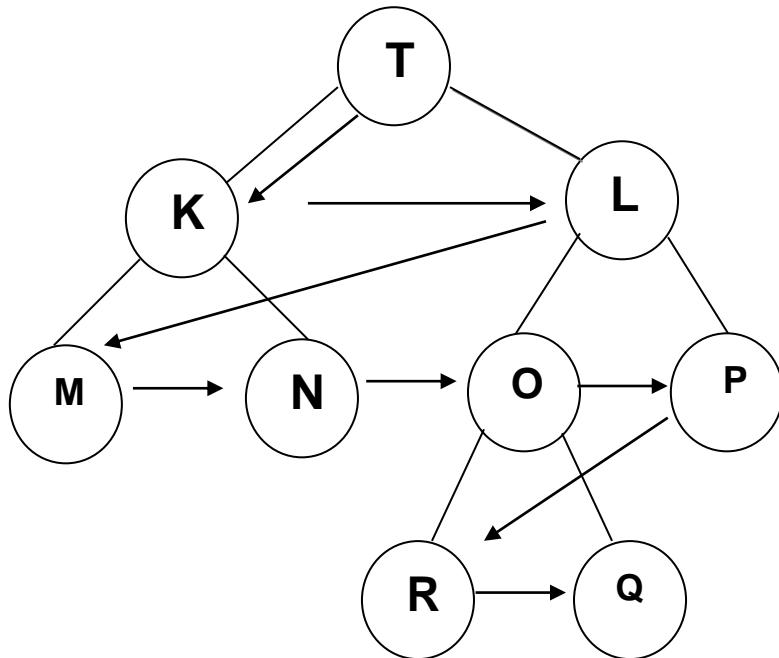
1. Pencarian buta (*blind search*) dengan iterasi

Blind search atau Un-informed search merupakan pencarian untuk melacak ruang keadaan. Metode pencarian secara Blind search terdiri dari beberapa yaitu metode “Breadth First Search(BFS)”, “Dept First Search(DFS)”, “Depth limited search (DLS)”, “Uniform Cost Search (UCS)”, “Iterative deepening search (IDS)”, “Bi-Directional search (BDS)”. Metode Serching secara heuristik terdiri dari generate atau pembangkit dan test atau uji, pendakian bukit atau hill climbing, Simulate annealing (SA), Best first search (pencarian terbaik lebih dahulu), greedy best first search, A* (A bintang), Iterative deepening A* (IDA*), Simplified memory bounded A* (SMA*), Bi-Directional A*(BDA*), Modified Bi-Directional A*(MBDA*), Dynamic weighting A* (DWA*).

Metode *Breadth-first search* (BFS) merupakan bagian dari *blind* yang artinya sebelum mengunjungi level $n+1$ maka yang terlebih dahulu dilakukan adalah mengunjungi level n node. Pencarian dilakukan pada semua simpul dalam setiap level berurutan dari kiri ke kanan. Jika pada level tertentu tidak ditemukan solusi maka dilanjutkan pada level berikutnya. Solusi yang diambil adalah solusi yang terbaik. Dengan kata lain solusi yang complete dan optimal. Hal ini dilakukan dengan menelusuri satu-persatu sampai level yang paling bawah (Suyanto, 2007). Simpul yang bersebelahan dengan simpul awal akan dicari terlebih dahulu (Kristanto A,2004).

Pada kasus ini seperti contoh pada Gambar 3.1 yaitu searching diawali node awal atau akar (node T) teruske tahapan atau level pertama yaitu K dari kiri(left) ke kanan(right) yaitu L, Setelah itu beralih ke tahapan berikutnya atau ke level selanjutnya, mulai dari kiri(left) M ke kanan(right) N sampai ditemukanya *goal* (node Q). Gambar 3.1 merupakan metode *Breadth-first search* (BFS) yang diadopsi dari buku kecerdasan buatan oleh Sutoju, Mulyanto E, Suhartono V

(2011). Aturan dalam metode ini yaitu Dalam algoritma BFS sebagai anak(child) yang sudah didatangi kemudian disimpan pada antrian. Antrian ini digunakan sebagai arah simpul tetangga yang hendak dikunjungi sesuai dengan urutan.



Gambar 3.1 Metode Breadth-first search (BFS)

Untuk memperjelas cara kerja algoritma BFS beserta antrean yang digunakanya, berikut urutan BFS algorithm.

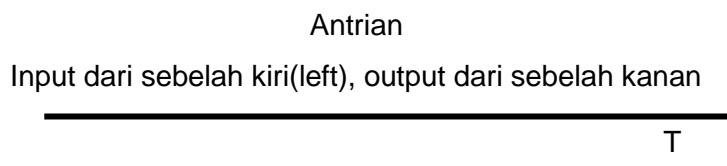
- a. Masukan node Akar kedalam antrian
- b. Dapatkan titik untuk mengawali suatu antrian, setelah itu cek apakah terdapat suatu penyelesaian
- c. Apabila terdapat titik solusi, maka searching berakhir
- d. Apabila titik penyelesaian, input semua titik pada antrian
- e. Apabila antrian tidak terisi atau kosong dan tiap titik telah diperiksa maka searching selesai.
- f. dari poin 2
- g. Keuntungan Menjamin ditemukanya solusi yang paling baik (*Complete* dan *Optimal*)
- h. Kelemahan Karena BSF harus menyimpan semua node yang dibangkitkan maka metode ini membutuhkan memori dan waktu yang cukup banyak
Jika Queue tidak kosong, ulangi pencarian mulai

Contoh 1

Misalkan diketahui pohon pelacakan seperti Gambar 3.1 implementasikan algoritma BFS agar mendapatkan penyelesaian mulai dari titik P hingga titik Q melalui beberapa iterasi yaitu:

Iterasi ke 1

Input titik T Pada antrian



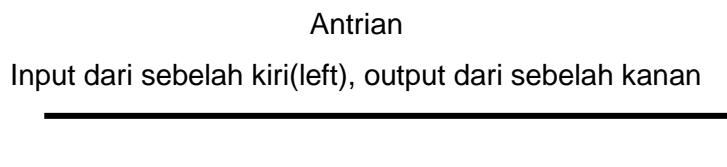
Gambar 3.2 Stack *Breadth-first search (BFS)* iterasi 1 tahap 1

Representasi Ruang Keadaan :



Gambar 3.3 Repesentasi keadaan *Breadth-first search (BFS)* iterasi 1

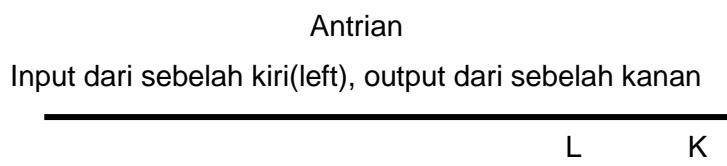
Keluarkan T dari antrian kemudian periksa T merupakan tujuan(Goal)



Gambar 3.4 Stack *Breadth-first search (BFS)* iterasi 1 tahap 2

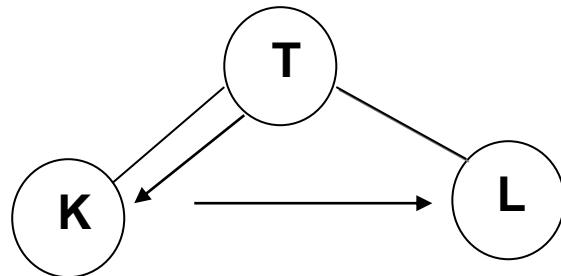
Ternyata T ≠ Goal

T punya anak K dan L masukan pada antrian



Gambar 3.5 Stack *Breadth-first search (BFS)* iterasi 1 tahap 3

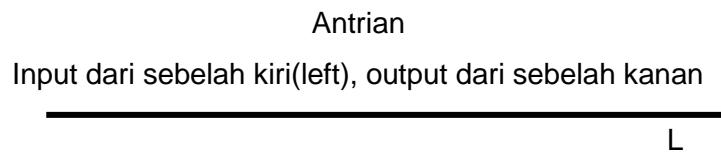
Representasi Ruang Keadaan



Gambar 3.6 Repesentasi keadaan *BreadthFirstSearch (BFS)*
Tahapan(iterasi) 2

Iterasi ke-2

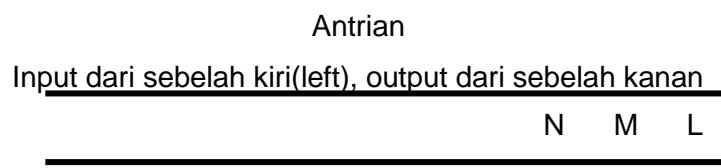
Lempar K dari Antrian periksa K merupakan Tujuan Goal



Gambar 3.7 Stack *Breadth-first search (BFS)* iterasi 2 tahap 1

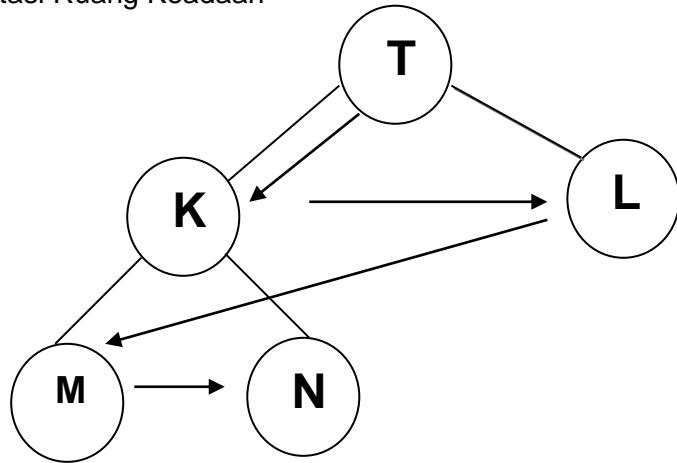
Ternyata K ≠ Goal

K mempunyai anak(child) M serta N input pada antrian

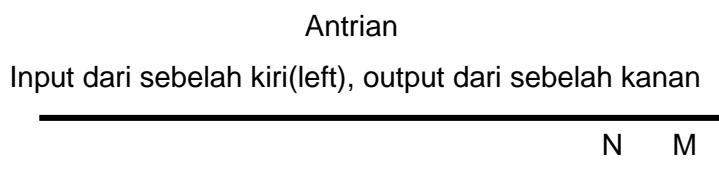


Gambar 3.8 Stack *Breadth-first search (BFS)* iterasi 2 tahap 2

Representasi Ruang Keadaan

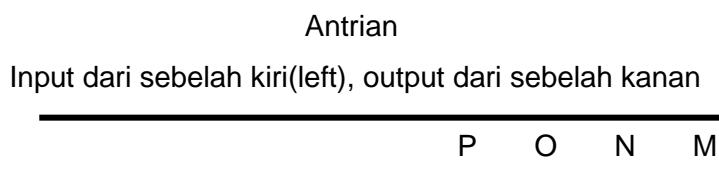
**Gambar 3.9** Repesentasi keadaan *Breadth-first search (BFS)* iterasi 3**Iterasi ke-3**

Lempar L dari Antrian lalu periksa L merupakan tjuan(Goal)

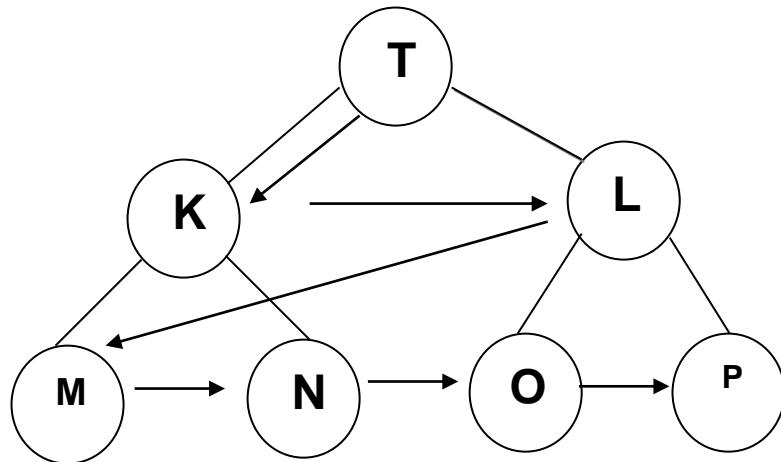
**Gambar 3.10** Stack *Breadth-first search (BFS)* iterasi 3 tahap 1

Ternyata L \neq Goal

L mempunyai anak(child) O serta P input pada antrian

**Gambar 3.11** Stack *Breadth-first search (BFS)* iterasi 3 tahap 2

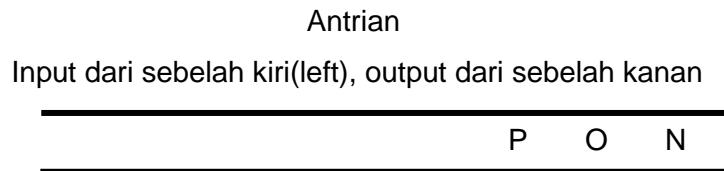
Representasi Ruang Keadaan



Gambar 3.12 Representasi keadaan Breadth-first search (BFS) iterasi 4

Iterasi ke-4

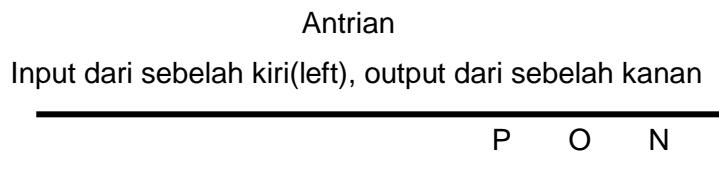
Lempar M dari Antrian lalu periksa M merupakan tujuan(Goal)



Gambar 3.13 Stack Breadth-first search (BFS) iterasi 4 tahap 1

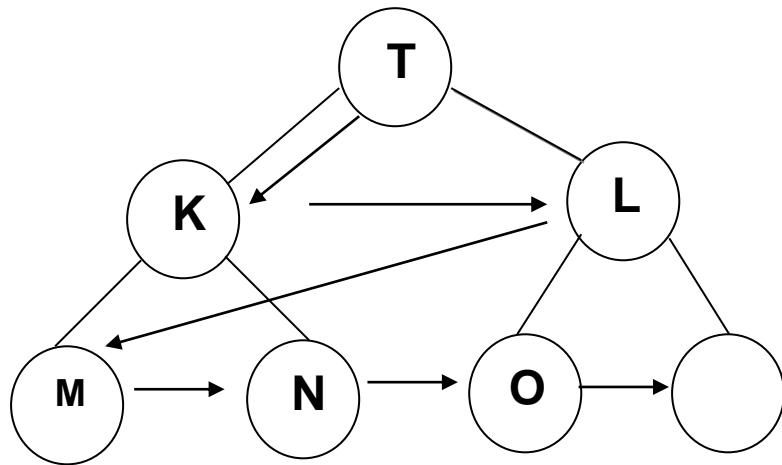
Ternyata $M \neq Goal$

M bukan memiliki anak(child) sehingga tidak di input pada antrian



Gambar 3.14 Stack Breadth-first search (BFS) iterasi 4 tahap 2

Representasi Ruang Keadaan

**Gambar 3.15** Representasi pengetahuan *Breadth-first search (BFS)* iterasi 5**Iterasi ke 5**

Lempar N pada antrian lalu periksa N merupakan (tujuan)Goal

Antrian

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

P O

Gambar 3.16 Stack *Breadth-first search (BFS)* iterasi 5 tahap 1

Ternyata N ≠ Goal

N bukan memiliki anak(child), lalu dikeluarkan dari antrian

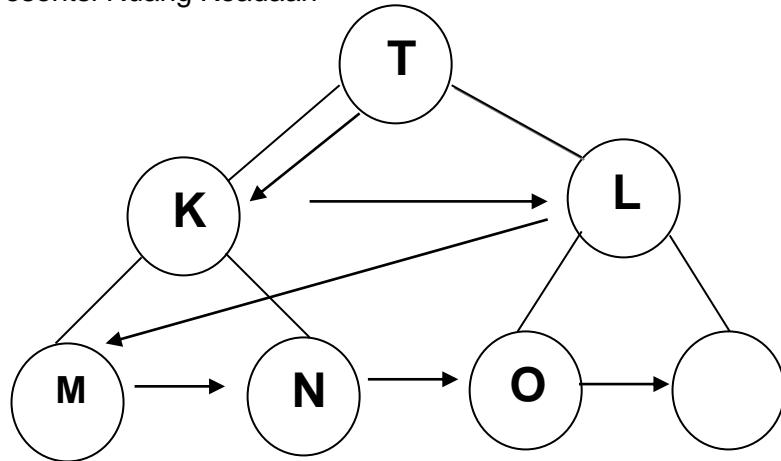
Antrian

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

P O

Gambar 3.17 Stack *Breadth-first search (BFS)* iterasi 5 tahap 2

Reperesensi Ruang Keadaan

**Gambar 3.18** Representasi pengetahuan Breadth-first search (BFS) iterasi 6**Iterasi ke-6**

Lempar O dari Antrian kemudian periksa O merupakan Tujuan(Goal)

Antrian

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

P

Gambar 3.19 Stack Breadth-first search (BFS) iterasi 6 tahap 1

Ternyata O \neq Goal

O mempunyai anak(child) R serta Q, input pada Antrian

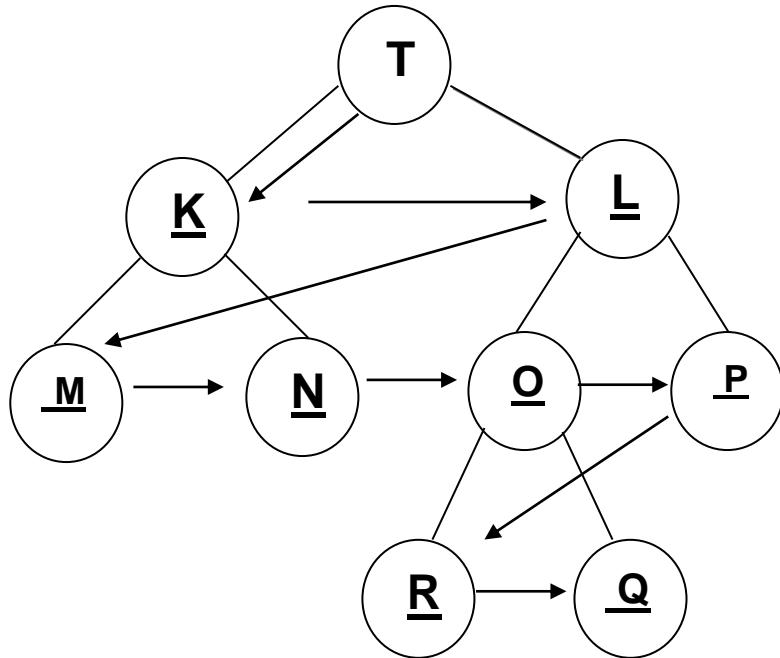
Antrian

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

R Q P

Gambar 3.20 Stack Breadth-first search (BFS) iterasi 6 tahap 2

Representasi Ruang Keadaan

**Gambar 3.21** Representasi pengetahuan Breadth-first search (BFS) iterasi 7**Iterasi ke-7**

Pada iterasi ke-7 sudah terdapat banyak anak(child). Titik awal yaitu T sementara child(anak) terdiri dari titik atau huruf yang diberi garis bawah seperti K,L,M,N,O dan lain-lain.

Lempar P dari Antrian lalu periksa P merupakan Tujuan(Goal)

Antrian

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

R Q

Gambar 3.22 Stack Breadth-first search (BFS) iterasi 7 tahap 1

Ternyata $P \neq Goal$

P tidak Memiliki anak(child), lalu tidak bisa disertakan pada antrian

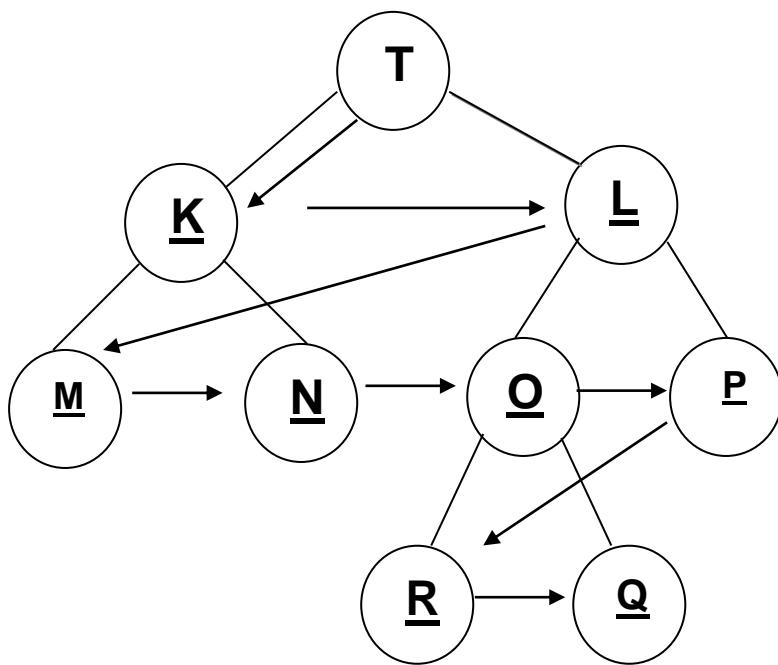
Antrian

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

R Q

Gambar 3.23 Stack Breadth-first search (BFS) iterasi 7 tahap 2

Representasi Ruang Keadaan



Gambar 3.24 Representasi pengetahuan Metode Breadth-first search (BFS)
iterasi 8

Iterasi ke 8

Lempar R dari Antrian lalu periksa R merupakan Tujuan(Goal)

Antrian

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

Q

Gambar 3.25 Stack Breadth-first search (BFS) iterasi 8 tahap 1

Ternyata R ≠ Goal

R tidak Memiliki child(anak), sehingga tidak bisa disertakan pada antrian

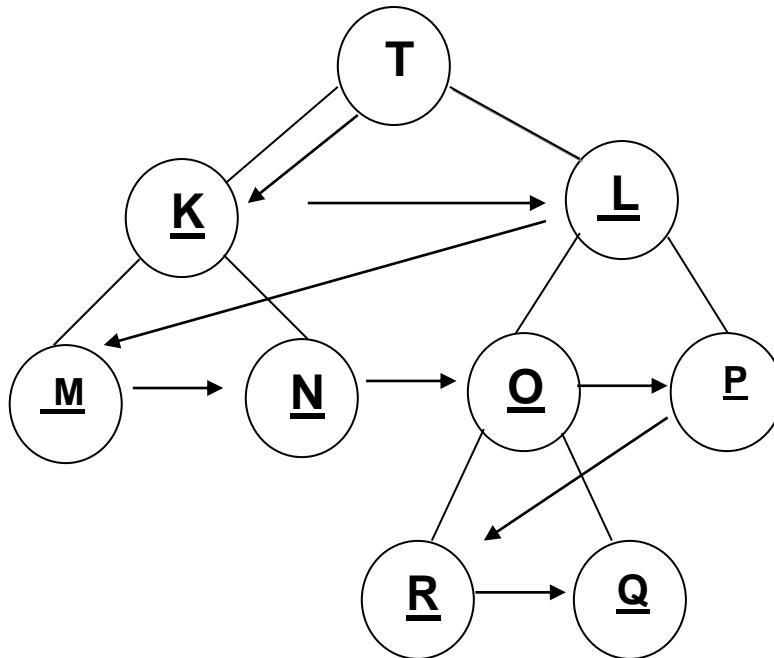
Antrian

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

Q

Gambar 3.26 Stack Breadth-first search (BFS) iterasi 8 tahap 2

Representasi Ruang Keadaan



Gambar 3.27 Representasi pengetahuan Metode Breadth-first search (BFS)
iterasi 9

Iterasi ke 9

Lempar Q dari Antrian lalu periksa Q merupakan tujuan(Goal)

Antrian

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

Gambar 3.28 Stack Breadth-first search (BFS) iterasi 9 tahap 1

Ternyata Q = Goal

Pencarian di hentikan

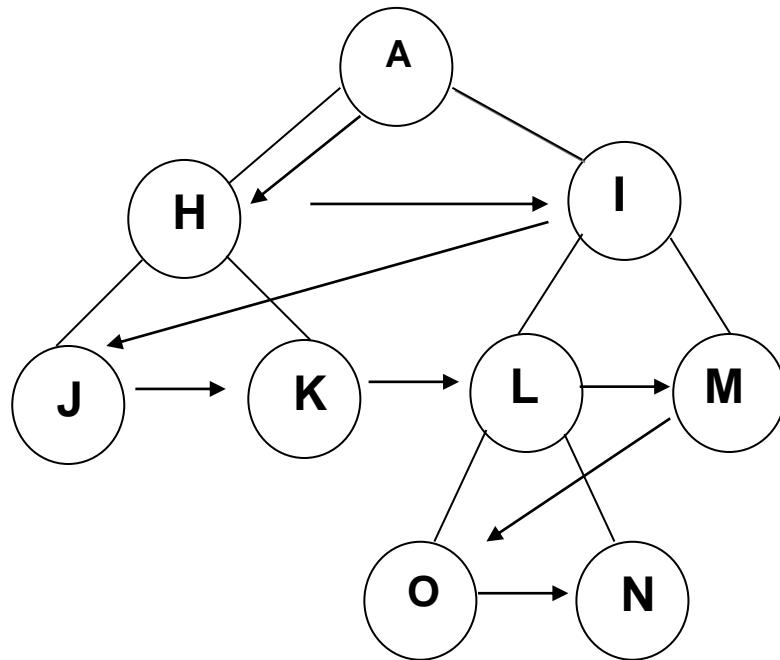
Penyelesaian Masalah :

Q anaknya(child) O, lalu O anaknya(child) L, serta L anaknya(child) T

T merupakan titik asal(akar) sehingga searching penyelesaian berhenti dan penyelesaiannya ialah : T-L-O-R

C. Soal Latihan/ Tugas

1. Ilustrasikan gambar di bawah ini dengan metode *breadth first search (BFS)* melalui 8 iterasi?



2. Carilah 2 jurnal lain yang berhubungan dengan Pencarian (searching)?

D. Referensi

- Prasetyo Y, Dewa SW, Udjulawa D. [Tahun tidak diketahui]. Penerapan algoritma Breadth first search untuk rancang bangun edugame icon perangkat teknologi informasi berbasis android. [jurnal dan volume tidak diketahui]
- Kristanto A. 2004. *Kecerdasan buatan*. Graha Ilmu: Yogyakarta.
- Sutojo T, Mulyanto E, Suhartono V.2011. *kecerdasan buatan*. ANDI. Yogyakarta.
- Suyanto.2007. *Artificial intelligence searching, reasoning, planning dan learning. Informatika Bandung*. Bandung
- Wijaya E. 2013. *Analisis Penggunaan Algoritma Breadth First Search Dalam Konsep Artificial Intellegencia*. Jurnal TIME , Vol. II No 2 : 18-26