**LAPORAN AKHIR PRAKTIKUM**

Mata Praktikum : Perancangan Analisis Algoritma

Kelas : 3IA24

Praktikum ke- : 4

Tanggal : 10 November 2023

Materi : Algoritma DFS & BFS

NPM : 50421704

Nama : Juan Samuel Christopher

Ketua Asisten : Sherina Permata

Nama Asisten :

Paraf Asisten :

Jumlah Lembar : 8

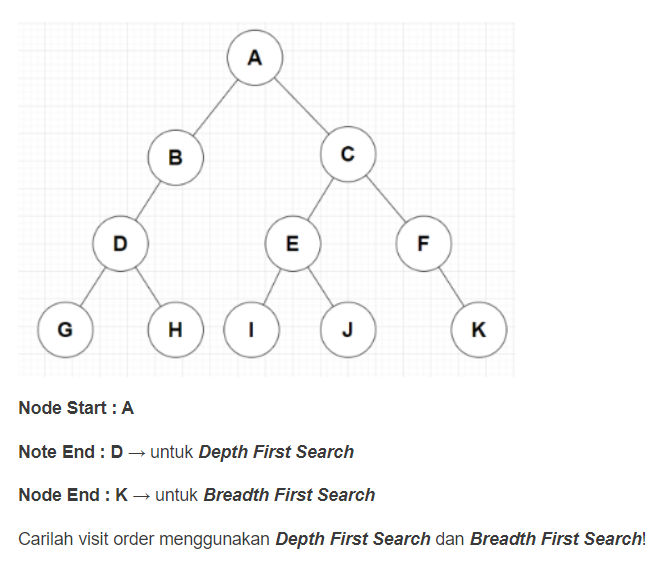


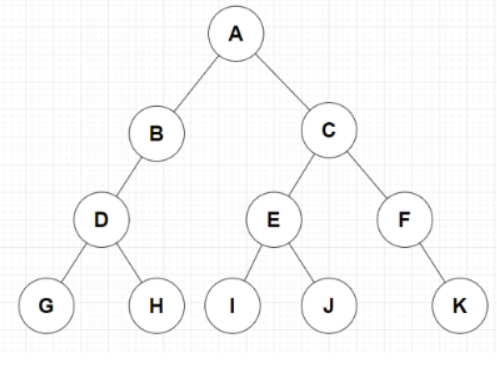
**LABORATORIUM TEKNIK INFORMATIKA**

**UNIVERSITAS GUNADARMA**

**2023**

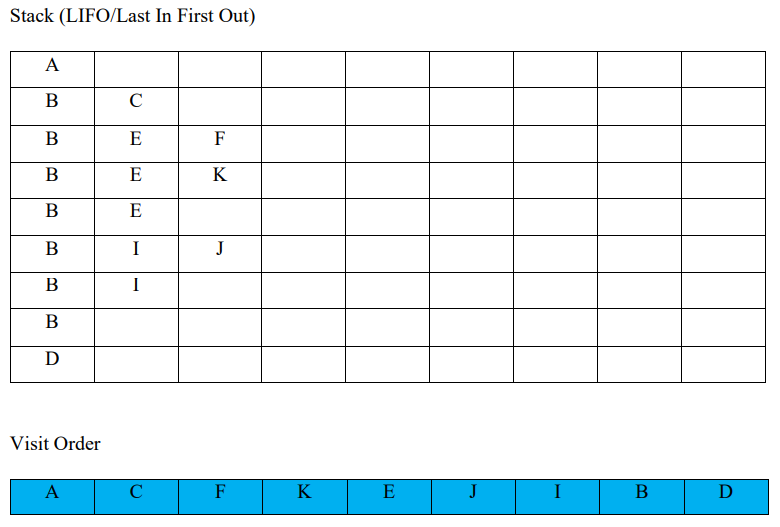
**SOAL**

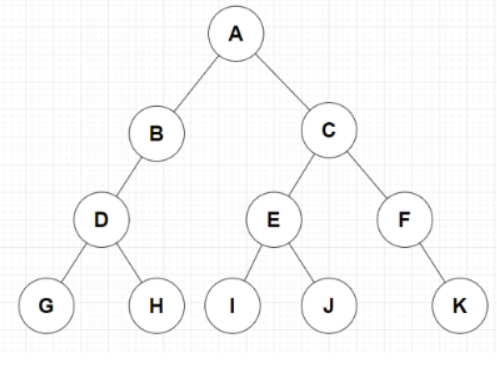
****

**Depth First Search**

**Node Start : A**

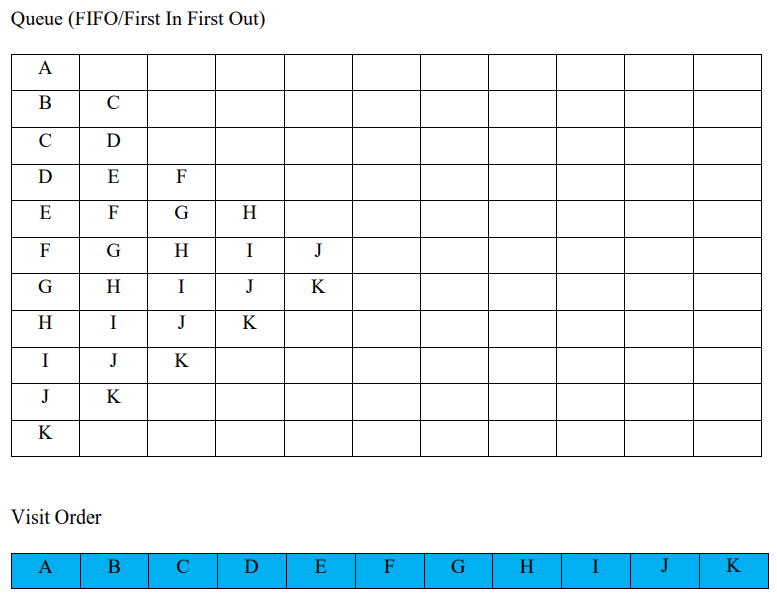
**Node End : D**

****

**Breadth First Search**

**Node Start : A**

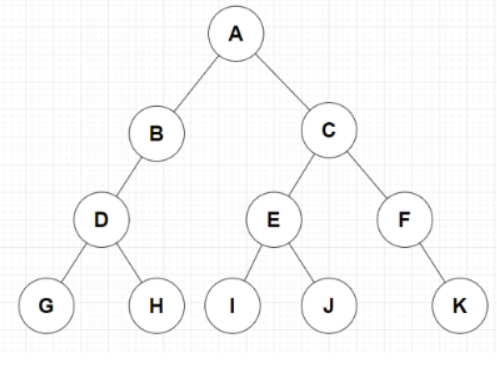
**Node End : K**

****

**LOGIKA**

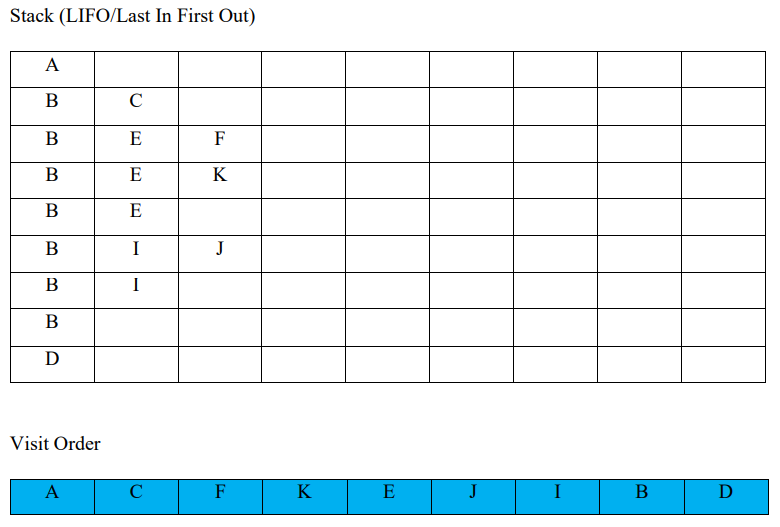
Pada pertemuan ke -4 ini kita belajar tentang Depth First Search dan Breadth First Search.

**LISTING**

**Depth First Search**

**Node Start : A**

**Node End : D**

****

Depth First Search (DFS) adalah algoritma untuk melakukan penelusuran atau pencarian dalam tree atau graph. Proses pencarian dimulai dari node awal dan kemudian bergerak ke node-node tetangganya. Node-node tetangga yang belum dikunjungi akan ditambahkan ke stack. Proses ini akan terus berulang hingga semua node dalam tree atau graph telah dikunjungi.

DFS menggunakan stack untuk menyimpan node-node yang belum dikunjungi. Stack adalah struktur data yang mengikuti prinsip LIFO (Last In First Out), artinya node yang paling terakhir dimasukkan ke stack akan dikeluarkan paling pertama.

Langkah-langkah pengerjaan DFS adalah Masukkan node awal ke stack. Selama stack tidak kosong Keluarkan node dari stack. Periksa apakah node tersebut merupakan node tujuan. Jika ya, maka pencarian selesai. Tambahkan semua node tetangga dari node tersebut ke stack.

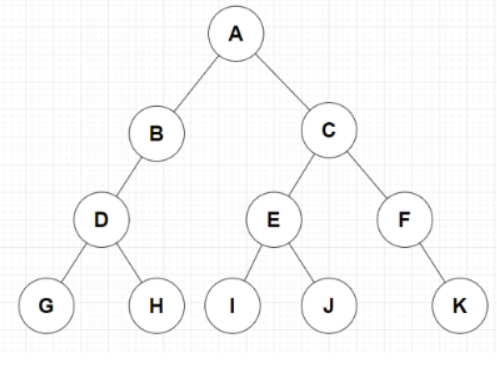
Jadi pada pengerjaan ACT ini diawali dengan memasukkan simpul A ke dalam Stack karena disini simpul dimulai dari A , lalu masukkan A kedalam **Visit Order**, kemudian kunjungi simpul **B** dan **C** karena ini LIFO maka masukkan simpul **C** ke visit order dan simpul B tetap didalam stack Lalu kunjungi simpul E dan simpul F serta masukkan simpul F ke Visit Order Selanjutnya kunjungi simpul K serta masukkan simpul K ke Visit OrderLalu masukkan simpul E ke Visit Order Dilanjut dengan mengunjungi simpul I dan simpul J serta masukkan simpul J ke Visit Order Setelah itu masukkan simpul I ke dalam Visit Order Lalu masukkan simpul B ke dalam Visit Order. Terakhir, karena node end nya adalah di simpul D maka kunjungi simpul D dan masukkan simpul D ke dalam Visit Order.

**OUTPUT**

Visit Order DFS

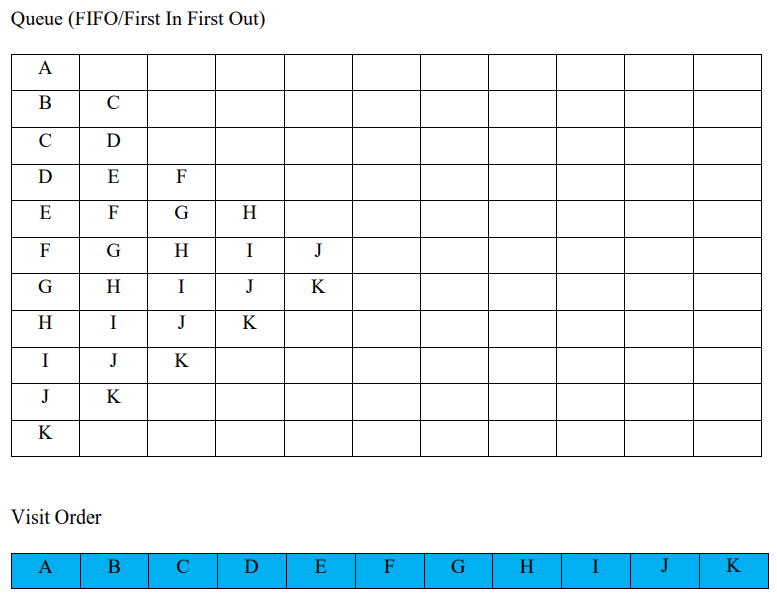


**LISTING**

**Breadth First Search**

**Node Start : A**

**Node End : K**

****

Breadth First Search (BFS) adalah algoritma untuk melakukan penelusuran atau pencarian dalam tree atau graph. Proses pencarian dimulai dari node awal dan kemudian bergerak ke node-node tetangganya. Node-node tetangga yang belum dikunjungi akan ditambahkan ke queue. Proses ini akan terus berulang hingga semua node dalam tree atau graph telah dikunjungi.

BFS menggunakan queue untuk menyimpan node-node yang belum dikunjungi. Queue adalah struktur data yang mengikuti prinsip FIFO (First In First Out), artinya node yang paling pertama dimasukkan ke queue akan dikeluarkan paling pertama.

langkah-langkah pengerjaan BFS adalah Masukkan node awal ke queue. Selama queue tidak kosong: Keluarkan node dari queue. Periksa apakah node tersebut merupakan node tujuan. Jika ya, maka pencarian selesai. ambahkan semua node tetangga dari node tersebut ke queue.

Maka pada pengerjaan ACT 4 ini Diawali dengan memasukkan simpul A kedalam Stack karena simpul dimulai dari A dan masukkan ke Visit Order. Setelah itu kunjungi simpul B dan simpul C serta masukkan simpul B ke Visit Order Selanjutnya kunjungi simpul D serta masukkan simpul C ke Visit Order Selanjutnya kunjungi simpul D serta masukkan simpul C ke Visit Order Dilanjut dengan mengunjungi simpul G dan simpul H serta masukkan simpul E ke Visit Order Setelah itu kunjungi simpul I dan simpul J serta masukkan simpul F ke Visit Order Lalu kunjungi simpul K serta masukkan simpul G ke Visit Order Lalu kunjungi simpul K serta masukkan simpul G ke Visit Order Selanjutnya masukkan simpul I ke Visit Order Lalu masukkan simpul J ke Visit Order Terakhir, karena node end berada di simpul K maka masukkan simpul K ke Visit Order

**OUTPUT**

Visit Order BFS

****