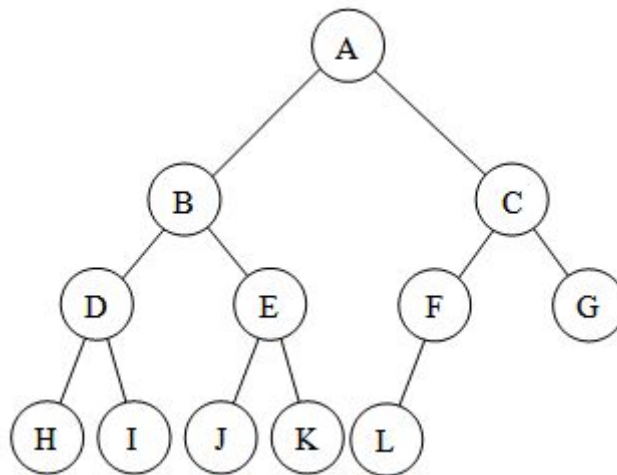


Jurnal 11: Tree

Kerjakanlah soal berikut. Jawaban dinilai berdasarkan dua kriteria: a. Kode sesuai dengan permintaan soal, dan b. Sesi tanya-jawab dengan Asprak. Kerjakan sesuai dengan prinsip Java Coding Style.

Marisa sedang menyelesaikan insiden di bawah tanah. Saat di bawah tanah, ia menemukan banyak sekali cabang yang mengarah lebih dalam. Marisa ingin melakukan mapping terlebih dahulu, dibantu dengan boneka Alice untuk melihat tiap cabang. Jalan ke bawah tanah ini mirip dengan tree, karena itu, Marisa ingin mencetak bentuk tree dengan post/in/level/pre-order traversal, kemudian menampilkan kedalaman tree tersebut.



Cara mencari kedalaman dari tree, berarti kita harus mendapatkan seluruh node yang ada, kemudian kita hitung dia sudah mencapai kedalaman mana saja, bukan? Dan karena kita mencari kedalaman, maka kita mencari node paling bawah dalam tree, ntah itu di sebelah kiri, atau di sebelah kanan. Karena itu kita gunakan fungsi max!

Untuk menghitung kedalaman:

1. Hitung kedalaman subtree kiri.
2. Hitung kedalaman subtree kanan.
3. Kedalaman pohon saat ini = $1 + \max(\text{kedalaman kiri}, \text{kedalaman kanan})$

Ini mirip dengan post-order traversal karena kita melakukan evaluasi kepada node kiri dan kanan terlebih dahulu sebelum kita lanjut ke langkah selanjutnya.

Contoh Output:

```
Inorder Traversal:
H D I B J E K A L F C G
Post Order Traversal:
H I D J K E B L F G C A
Pre Order Traversal:
A B D H I E J K C F L G
Level Order Traversal:
A B C D E F G H I J K L
Total Nodes: 12
Apakah huruf A ada? : true
Kedalaman tree: 4
```

Java Coding Style

Ikuti gaya penulisan berikut di kode program agar mendapat nilai maksimal

1. Setiap nama kelas, variabel dan konstanta harus dapat menggambarkan isinya.

Misal, membuat variabel untuk menyimpan jumlah penumpang di bus.

Deklarasi:

```
int penumpang = 0;           // Bagus, OK
int p = 0;                   // Tidak OK
```

2. Penulisan nama harus menggunakan huruf besar/kecil yang sesuai.

Nama kelas: UpperCamelCase. Misal: PenumpangBus

Nama variabel: lowerCamelCase. Misal: penumpangBus

Nama konstanta: CAPITAL_CASE. Misal: PENUMPANG_BUS

Nama method: lowerCamelCase. Misal: hitungPenumpangBus

3. Penggunaan tab/spasi ketika berbeda blok program, WAJIB masuk 1 tab atau 4 spasi ke dalam.

```
public class PenumpangBus {
    public static void main(String[] args) {
        int penumpang = 0;
        penumpang = penumpang + 5;
        penumpang = penumpang - 4 + 2;
        penumpang = penumpang - 1;
        penumpang = penumpang - 2 + 3;
        penumpang = penumpang - 2 + 5;
        penumpang = penumpang - 1 + 3;
        System.out.println(penumpang);
    }
}
```

4. Gunakan komentar seperlunya.

Javadoc comment (yang warna biru) hanya untuk mengomentari kelas, method dan variable

```
/** Jumlah penumpang di bus */  
int penumpang = 0;  
  
/* Ada 5 orang penumpang naik ke bus */  
penumpang = penumpang + 5;  
  
// Penumpang turun 4 orang, naik 2  
penumpang = penumpang - 4 + 2;
```