Panji Iman Baskoro
171111023

Tugas Pertemuan 6

BreadthFirstSearch.java

```
import java.util.List;
```

```
1.import java.util.LinkedList;
2.import java.util.Queue;
3.import java.util.ArrayList;
4.
5.public class BreadthFirstSearch {
6.
7.  private Queue<Node> queue;
8.  static ArrayList<Node> nodenya = new ArrayList<Node>();
9.
10.
11.  public BreadthFirstSearch() {
12.    queue = new LinkedList<Node>();
13.  }
14.
15.  public void B_F_S(Node node) {
16.    queue.add(node);
17.    node.visited = true;
18.    while (!queue.isEmpty()) {
19.      Node elemen = queue.remove();
20.      System.out.print(elemen.data + " ");
21.
22.      List<Node> relasi = elemen.getRelasi();
23.      for (int i = 0; i < relasi.size(); i++) {
24.        Node x = relasi.get(i);
25.        if (x != null && !x.visited) {
26.          queue.add(x);
27.          x.visited = true;
28.
29.        }
30.
31.      }
32.    }
33.  }
34.  public static void main(String[] args) {
35.    Node node1 = new Node(1);
36.    Node node2 = new Node(2);
37.    Node node3 = new Node(3);
38.    Node node4 = new Node(4);
39.    Node node5 = new Node(5);
40.    Node node6 = new Node(6);
41.    Node node7 = new Node(7);
42.
43.    node1.addrelasi(node2);
44.    node1.addrelasi(node3);
```

```
    node2.addrelasi(node4);
    node3.addrelasi(node2);
    node3.addrelasi(node4);
    node3.addrelasi(node5);
    node3.addrelasi(node6);
    node4.addrelasi(node5);
    node5.addrelasi(node7);
    node6.addrelasi(node7);

    System.out.println("BFS traversal dengan queue     : ");
    BreadthFirstSearch BFS = new BreadthFirstSearch();
    BFS.B_F_S(node1);
    System.out.println("\nTerimakasih");

  }
}
```

DepthFirstSearch.java

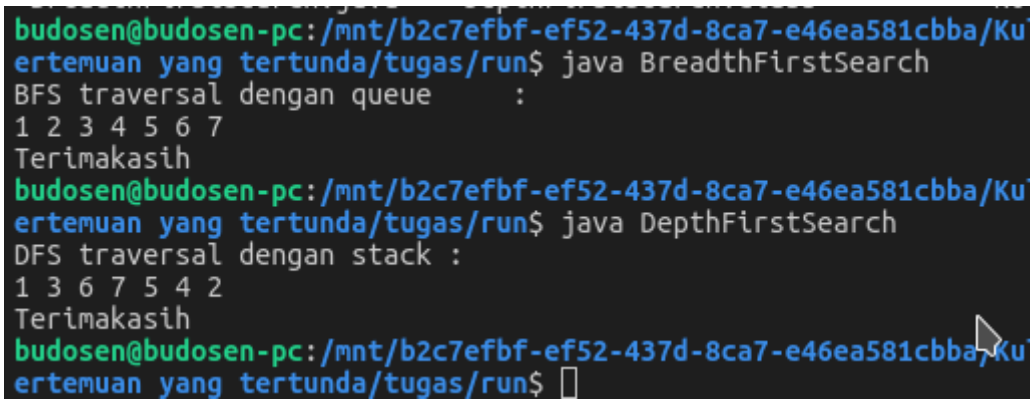
```
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;
import java.util.Stack;
public class DepthFirstSearch {
   public void DepthFirstSearch(Node node){
     Stack<Node> lifo = new Stack<Node>();
     lifo.add(node);
     node.visited= true;
     while(!lifo.isEmpty()){
        Node elemen = lifo.pop();
        System.out.print(elemen.data+" ");
        List<Node> relasi = elemen.getRelasi();
        for (int i = 0; i < relasi.size(); i++) {
          Node y = relasi.get(i);
          if(y!= null && !y.visited){
            lifo.add(y);
            y.visited=true;
          }
        }
     }
  }
   public static void main(String[] args) {
     Node node1 = new Node(1);
     Node node2 = new Node(2);
     Node node3 = new Node(3);
     Node node4 = new Node(4);
     Node node5 = new Node(5);
     Node node6 = new Node(6);
     Node node7 = new Node(7);

     node1.addrelasi(node2);
     node1.addrelasi(node3);
```

```
        node2.addrelasi(node4);
        node3.addrelasi(node2);
        node3.addrelasi(node4);
        node3.addrelasi(node5);
        node3.addrelasi(node6);
        node4.addrelasi(node5);
        node5.addrelasi(node7);
        node6.addrelasi(node7);

        System.out.println("DFS traversal dengan stack : ");
        DepthFirstSearch DFS = new DepthFirstSearch();
        DFS.DepthFirstSearch(node1);
        System.out.println("\nTerimakasih");
    }
}
```

Node.java

```
import java.util.List;
```

```
import java.util.ArrayList;

public class Node {

  int data;
  boolean visited;
  List<Node> relasi;

  Node(int data) {
     this.data = data;
     this.relasi = new ArrayList<>();

   }

   public void addrelasi(Node relasiNode) {
     this.relasi.add(relasiNode);
   }

   public List<Node> getRelasi() {
     return relasi;
   }

   public void setRelasi(List<Node> relasi) {
     this.relasi = relasi;
   }

}
```


budosen@budosen-pc:/mnt/b2c7efbf-ef52-437d-8ca7-e46ea581cbba/Ku
ertemuan yang tertunda/tugas/run$ java BreadthFirstSearch
BFS traversal dengan queue :
1 2 3 4 5 6 7
Terimakasih
budosen@budosen-pc:/mnt/b2c7efbf-ef52-437d-8ca7-e46ea581cbba/Ku
ertemuan yang tertunda/tugas/run$ java DepthFirstSearch
DFS traversal dengan stack :
1 3 6 7 5 4 2
Terimakasih
budosen@budosen-pc:/mnt/b2c7efbf-ef52-437d-8ca7-e46ea581cbba/Ku
ertemuan yang tertunda/tugas/run$