Nama : Monica Tifani Zahara

NRP : 171 111 077

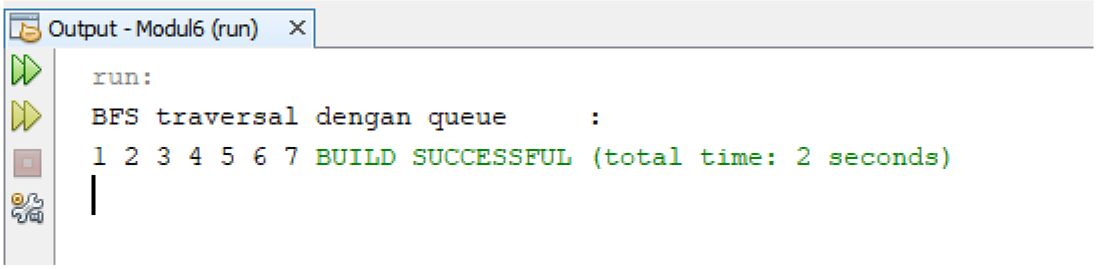
Tugas 6

Breadth First Search

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modul6;

import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;
import java.util.Queue;

/**
 *
 * @author Monica
 */
public class BreadthFirstSearch {

    private Queue<Node> queue;
    static ArrayList<Node> nodenya = new ArrayList<Node>();

    static class Node {

        int data;
        boolean dikunjungi;
        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;
        }

    }

    public BreadthFirstSearch() {
        queue = new LinkedList<Node>();
    }

```

```java
    public void B_F_S(Node node) {
        queue.add(node);
        node.dikunjungi = true;
        while (!queue.isEmpty()) {
            Node elemen = queue.remove();
            System.out.print(elemen.data + " ");

            List<Node> relasi = elemen.getRelasi();
            for (int i = 0; i < relasi.size(); i++) {
                Node x = relasi.get(i);
                if (x != null && !x.dikunjungi) {
                    queue.add(x);
                    x.dikunjungi = 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("BFS traversal dengan queue      : ");
        BreadthFirstSearch BFS = new BreadthFirstSearch();
        BFS.B_F_S(node1);

    }
}
```

Output

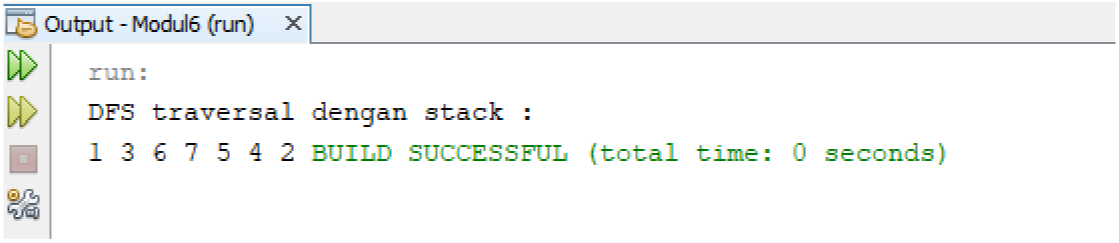


```
run:
BFS traversal dengan queue      :
1 2 3 4 5 6 7 BUILD SUCCESSFUL (total time: 2 seconds)
```

Depth First Search

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modul6;
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;
import java.util.Stack;
/**
 *
 * @author Monica
 */
public class DepthFirstSearch {
    static class Node {

        int data;
        boolean dikunjungi;
        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;
        }

    }
     public void DepthFirstSearch(Node node){
        Stack<Node> lifo = new Stack<Node>();
        lifo.add(node);
        node.dikunjungi= 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.dikunjungi){
                    lifo.add(y);
                    y.dikunjungi=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);
    }
}
```

Output



```
run:
DFS traversal dengan stack :
1 3 6 7 5 4 2 BUILD SUCCESSFUL (total time: 0 seconds)
```