

Nama : Monica Tifani Zahara

NRP : 171 111 077

Tugas 6

Breadth First Search

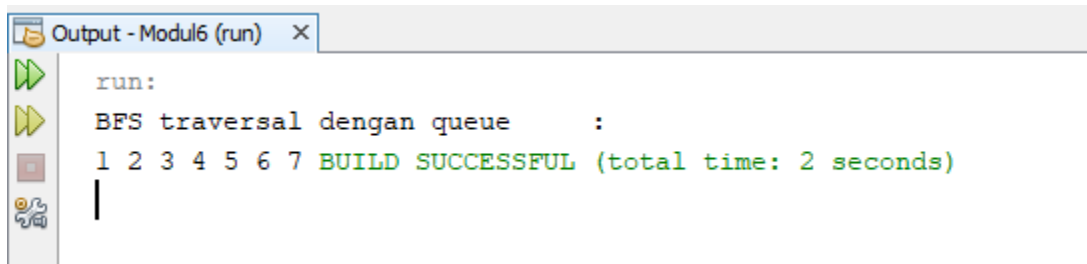
```
1.  /*
2.   * To change this license header, choose License Headers in Project Properties.
3.   * To change this template file, choose Tools | Templates
4.   * and open the template in the editor.
5.   */
6.  package modul6;
7.
8.  import java.util.ArrayList;
9.  import java.util.LinkedList;
10. import java.util.List;
11. import java.util.Queue;
12.
13. /**
14.  *
15.  * @author Monica
16.  */
17. public class BreadthFirstSearch {
18.
19.     private Queue<Node> queue;
20.     static ArrayList<Node> nodenya = new ArrayList<Node>();
21.
22.     static class Node {
23.
24.         int data;
25.         boolean dikunjungi;
26.         List<Node> relasi;
27.
28.         Node(int data) {
29.             this.data = data;
30.             this.relasi = new ArrayList<>();
31.
32.         }
33.
34.         public void addrelasi(Node relasiNode) {
35.             this.relasi.add(relasiNode);
36.         }
37.
38.         public List<Node> getRelasi() {
39.             return relasi;
40.         }
41.
42.         public void setRelasi(List<Node> relasi) {
43.             this.relasi = relasi;
44.         }
45.
46.     }
47.
48.     public BreadthFirstSearch() {
49.         queue = new LinkedList<Node>();
50.     }
51.
```

```

52.     public void B_F_S(Node node) {
53.         queue.add(node);
54.         node.dikunjungi = true;
55.         while (!queue.isEmpty()) {
56.             Node elemen = queue.remove();
57.             System.out.print(elemen.data + " ");
58.
59.             List<Node> relasi = elemen.getRelasi();
60.             for (int i = 0; i < relasi.size(); i++) {
61.                 Node x = relasi.get(i);
62.                 if (x != null && !x.dikunjungi) {
63.                     queue.add(x);
64.                     x.dikunjungi = true;
65.
66.                 }
67.
68.             }
69.         }
70.     }
71.
72.     public static void main(String[] args) {
73.         Node node1 = new Node(1);
74.         Node node2 = new Node(2);
75.         Node node3 = new Node(3);
76.         Node node4 = new Node(4);
77.         Node node5 = new Node(5);
78.         Node node6 = new Node(6);
79.         Node node7 = new Node(7);
80.
81.         node1.addrelasi(node2);
82.         node1.addrelasi(node3);
83.         node2.addrelasi(node4);
84.         node3.addrelasi(node2);
85.         node3.addrelasi(node4);
86.         node3.addrelasi(node5);
87.         node3.addrelasi(node6);
88.         node4.addrelasi(node5);
89.         node5.addrelasi(node7);
90.         node6.addrelasi(node7);
91.
92.         System.out.println("BFS traversal dengan queue      : ");
93.         BreadthFirstSearch BFS = new BreadthFirstSearch();
94.         BFS.B_F_S(node1);
95.
96.     }
97. }

```

Output



```

Output - Modul6 (run) x
run:
BFS traversal dengan queue      :
1 2 3 4 5 6 7 BUILD SUCCESSFUL (total time: 2 seconds)

```

Depth First Search

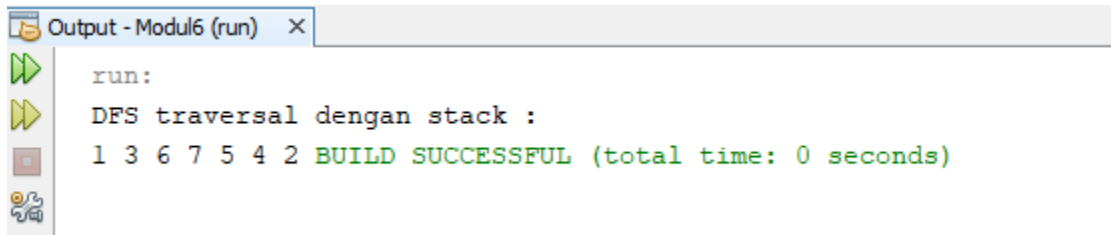
```
1.  /*
2.  * To change this license header, choose License Headers in Project Properties.
3.  * To change this template file, choose Tools | Templates
4.  * and open the template in the editor.
5.  */
6.  package modul6;
7.  import java.util.ArrayList;
8.  import java.util.LinkedList;
9.  import java.util.List;
10. import java.util.Stack;
11. /**
12.  *
13.  * @author Monica
14.  */
15. public class DepthFirstSearch {
16.     static class Node {
17.
18.         int data;
19.         boolean dikunjungi;
20.         List<Node> relasi;
21.
22.         Node(int data) {
23.             this.data = data;
24.             this.relasi = new ArrayList<>();
25.
26.         }
27.
28.         public void addrelasi(Node relasiNode) {
29.             this.relasi.add(relasiNode);
30.         }
31.
32.         public List<Node> getRelasi() {
33.             return relasi;
34.         }
35.
36.         public void setRelasi(List<Node> relasi) {
37.             this.relasi = relasi;
38.         }
39.     }
40.
41.     public void DepthFirstSearch(Node node){
42.         Stack<Node> lifo = new Stack<Node>();
43.         lifo.add(node);
44.         node.dikunjungi= true;
45.         while(!lifo.isEmpty()){
46.             Node elemen = lifo.pop();
47.             System.out.print(elemen.data+" ");
48.             List<Node> relasi = elemen.getRelasi();
49.             for (int i = 0; i < relasi.size(); i++) {
50.                 Node y = relasi.get(i);
51.                 if(y!= null && !y.dikunjungi){
52.                     lifo.add(y);
53.                     y.dikunjungi=true;
54.                 }
55.             }
56.         }
57.     }
58.     public static void main(String[] args) {
```

```

59.      Node node1 = new Node(1);
60.      Node node2 = new Node(2);
61.      Node node3 = new Node(3);
62.      Node node4 = new Node(4);
63.      Node node5 = new Node(5);
64.      Node node6 = new Node(6);
65.      Node node7 = new Node(7);
66.
67.      node1.addrelasi(node2);
68.      node1.addrelasi(node3);
69.      node2.addrelasi(node4);
70.      node3.addrelasi(node2);
71.      node3.addrelasi(node4);
72.      node3.addrelasi(node5);
73.      node3.addrelasi(node6);
74.      node4.addrelasi(node5);
75.      node5.addrelasi(node7);
76.      node6.addrelasi(node7);
77.
78.      System.out.println("DFS traversal dengan stack : ");
79.      DepthFirstSearch DFS = new DepthFirstSearch();
80.      DFS.DepthFirstSearch(node1);
81.  }
82. }

```

Output



```

Output - Modul6 (run) X
run:
DFS traversal dengan stack :
1 3 6 7 5 4 2 BUILD SUCCESSFUL (total time: 0 seconds)

```