

Panji Iman Baskoro  
171111023  
Praktikum Progdas 2

## Modul 6

coba6.java

```
public class coba6 {  
    1.  
    2.     public static void main(String[] args) {  
    3.         Tree t = new Tree(new TreeNode(1));  
    4.         t.root.add_child(new TreeNode(2), 1);  
    5.         t.root.add_child(new TreeNode(3), 1);  
    6.         t.root.add_child(new TreeNode(4), 2);  
    7.         t.root.children.get(0).add_child(new TreeNode(5), 1);  
    8.         t.root.children.get(2).add_child(new TreeNode(6), 1);  
    9.         t.root.children.get(2).add_child(new TreeNode(7), 2);  
    10.        t.print();  
    11.    }  
    12.  
    13.}
```

Tree .java

```
public class Tree {  
    1.     TreeNode root;  
    2.  
    3.     public Tree() {  
    4.         this.root = null;  
    5.     }  
    6.  
    7.     public Tree(TreeNode root) {  
    8.         this.root = root;  
    9.     }  
    10.  
    11.     void print() {  
    12.         if (this.root == null) {  
    13.             System.out.println();  
    14.         } else {  
    15.             this.root.print();  
    16.         }  
    17.     }  
    18.}
```

## TreeNode.java

```
import java.util.ArrayList;
```

```
1. public class TreeNode {
2.     TreeNode parent;
3.     double distance;
4.     ArrayList<TreeNode> children;
5.     int data;
6.
7.     public TreeNode(int new_data) {
8.         this.data = new_data;
9.         this.parent = null;
10.        this.distance = 0.0;
11.        this.children = new ArrayList<TreeNode>();
12.    }
13.
14.    void set_parent(TreeNode new_parent, double distance) {
15.        this.parent = new_parent;
16.        this.distance = distance;
17.        if (this.parent != null) {
18.            parent.children.add(this);
19.        }
20.    }
21.
22.    void set_parent(TreeNode new_parent) {
23.        this.set_parent(new_parent, 0);
24.    }
25.
26.    void add_child(TreeNode new_child, double distance) {
27.        new_child.set_parent(this);
28.        new_child.distance = distance;
29.    }
30.
31.    /* Simply remove child from this node's children */
32.    void remove_child(TreeNode child) {
33.        this.children.remove(child);
34.    }
35.
36.    void print(String spaces, double distance) {
37.        System.out.println(data+" Distance from Parent "+this.distance+ " distance from initial node : "+(distance+this.distance));
38.        for (int i = 0; i < this.children.size(); i++) {
39.            this.children.get(i).print(" ", this.distance);
40.        }
41.    }
42.
43.    void print() {
44.        this.print("", 0);
45.    }
46.}
```

output :

```
budosen@budosen-pc:/mnt/b2c7efbf-ef52-437d-8ca7-e46ea581cbba/Kuliah/materiku
ertemuan yang tertunda$ javac *.java
budosen@budosen-pc:/mnt/b2c7efbf-ef52-437d-8ca7-e46ea581cbba/Kuliah/materiku
ertemuan yang tertunda$ java coba6
1 Distance from Parent 0.0 distance from initial node : 0.0
2 Distance from Parent 1.0 distance from initial node : 1.0
5 Distance from Parent 1.0 distance from initial node : 2.0
3 Distance from Parent 1.0 distance from initial node : 1.0
4 Distance from Parent 2.0 distance from initial node : 2.0
6 Distance from Parent 1.0 distance from initial node : 3.0
7 Distance from Parent 2.0 distance from initial node : 4.0
```

Terimakasih