Panji Iman Baskoro
171111023
Praktikum Progdas 2

Modul 6

coba6.java

```
public class coba6 {
```

```
1.import java.util.Scanner;
2.
3.public class coba6 {
4.
5.  public static void main(String[] args) {
6.    Scanner pot = new Scanner(System.in);
7.    System.out.println("Masukkan root nodenya");
8.    int kolo = pot.nextInt();
9.    Tree t = new Tree(new TreeNode(kolo));
10.    System.out.println("berapa childnya?");
11.
12.    int yolo = pot.nextInt();
13.    for(int y = 0;y<yolo;y++){
14.      System.out.println("masukkan node Child");
15.      int polo = pot.nextInt();
16.      System.out.println("masukkan distance Child");
17.      int lolo = pot.nextInt();
18.      t.root.add_child(new TreeNode(polo), lolo);
19.    }
20.    t.print();
21.    pot.close();
22.  }
23.
24.}
```

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 {
```

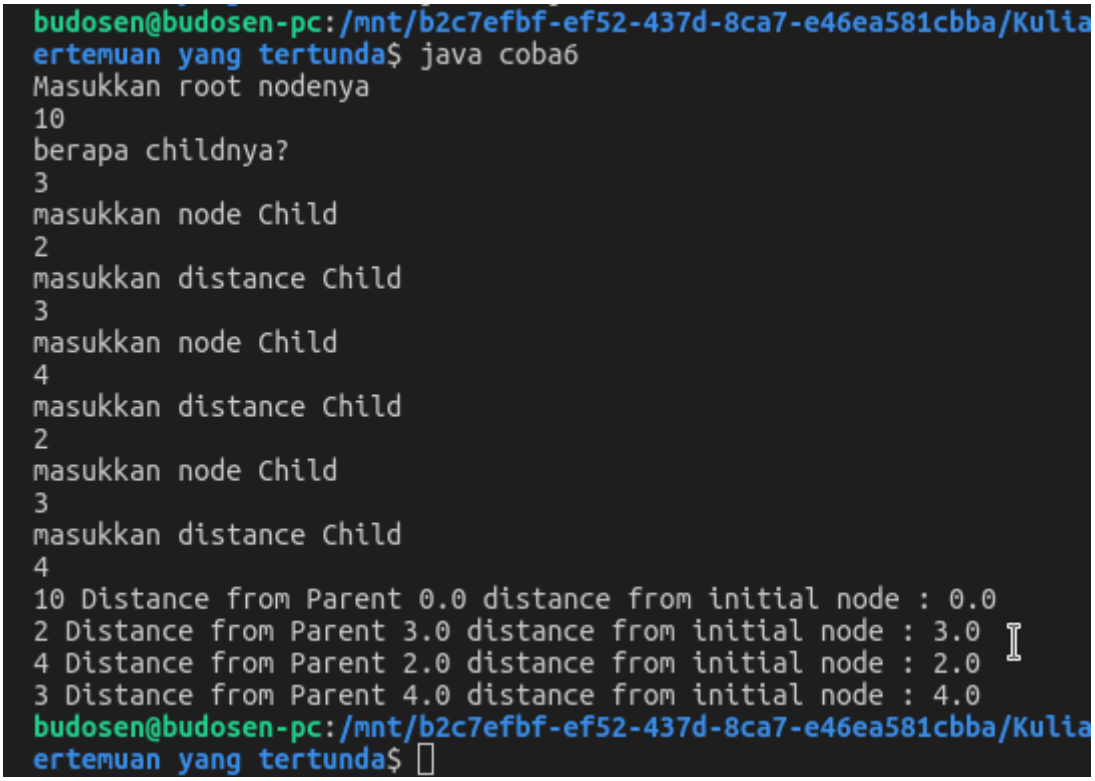
```
      this.root.print();
    }
  }
}
```

TreeNode.java

```
import java.util.ArrayList;

public class TreeNode {
  TreeNode parent;
  double distance;
  ArrayList<TreeNode> children;
  int data;

  public TreeNode(int new_data) {
    this.data = new_data;
    this.parent = null;
     this.distance = 0.0;
     this.children = new ArrayList<TreeNode>();
  }

  void set_parent(TreeNode new_parent, double distance) {
    this.parent = new_parent;
    this.distance = distance;
    if (this.parent != null) {
      parent.children.add(this);
    }
  }

  void set_parent(TreeNode new_parent) {
    this.set_parent(new_parent, 0);
  }

  void add_child(TreeNode new_child, double distance) {
    new_child.set_parent(this);
    new_child.distance = distance;
  }

  /* Simply remove child from this node's children */
  void remove_child(TreeNode child) {
    this.children.remove(child);
  }

  void print(String spaces, double distance) {
    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 :



Terimakasih