# LAPORAN PRAKTIKUM
# STRUKTUR DATA NON LINIER
# MODUL II

Dosen Pengampu

JB. Budi Darmawan S.T., M.Sc.


DISUSUN OLEH

**Ignatius Aryajulio Prananta**

**245314005**

**PROGRAM STUDI INFORMATIKA**

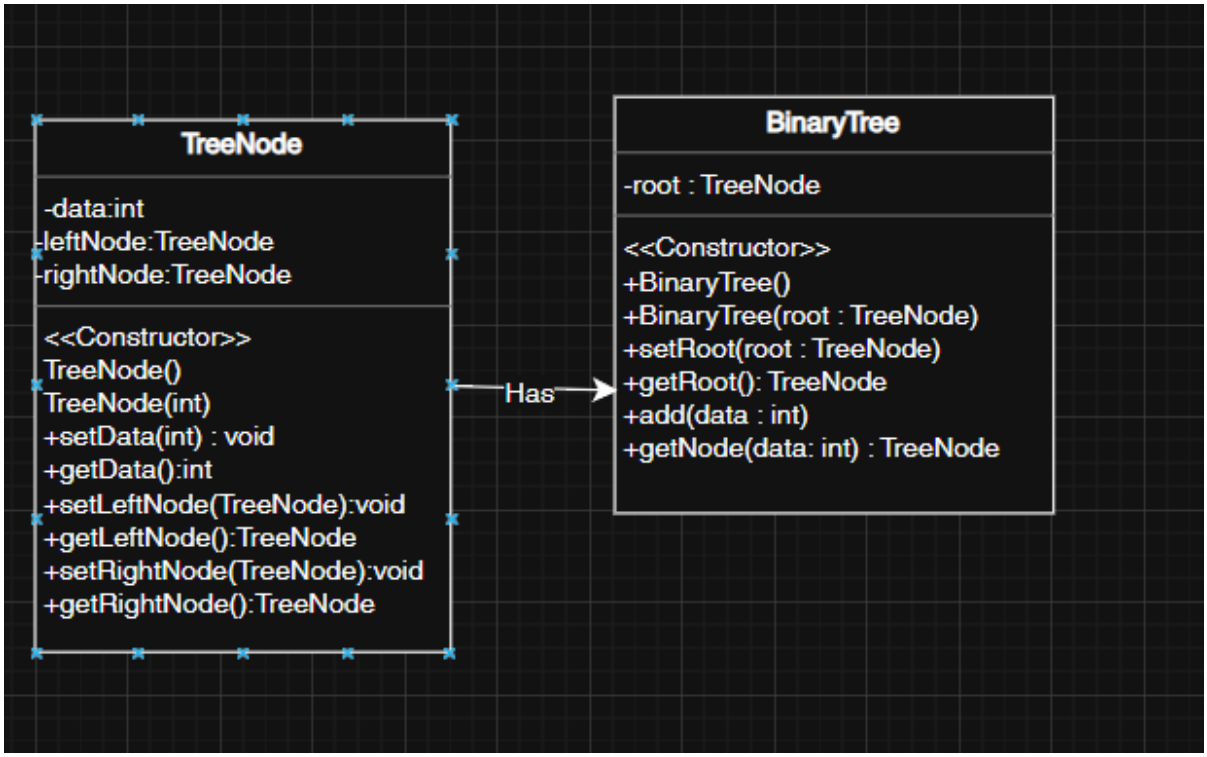
**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS SANATA DHARMA**

**YOGYAKARTA**

**2025**

## A. DIAGRAM UML



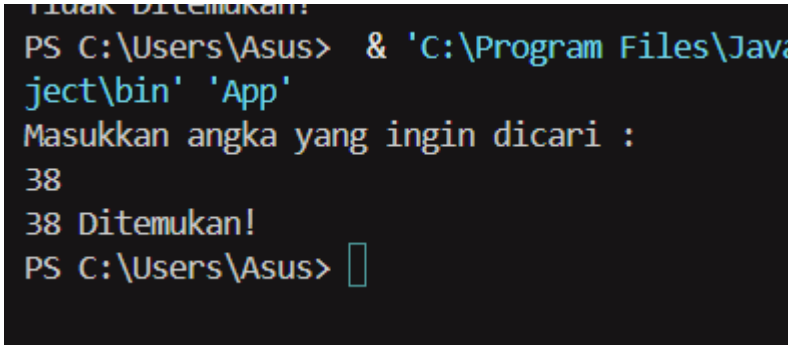

## B. SOURCE CODE

```java
public class BinaryTree {
   private TreeNode root;

   public BinaryTree(){
    root = null;
   }

    public BinaryTree(TreeNode root) {
        this.root = null;
    }
    public TreeNode getRoot(){
        return root;
    }

    public void setRoot(TreeNode root){
        this.root = root;
    }

    public void add (int data){
        if (root == null) {
            root = new TreeNode(data);
        }else{
            TreeNode bantu = root;
        while(true) {
            if (data < bantu.data) {
                if (bantu.leftNode == null) {
                    bantu.leftNode = new TreeNode(data);
                    return;
                } else{
                    bantu = bantu.leftNode;
                }
            }
                else if (data > bantu.data) {
                    if (bantu.rightNode == null) {
                    bantu.rightNode = new TreeNode(data);
                    return;
                } else{
                    bantu = bantu.rightNode;
                }
            } else {
                return;
            }
        }
    }
        }



    public TreeNode getNode(int data) {
        TreeNode bantu = root;
        while (bantu != null) {
            if (data == bantu.data) {
                return bantu;
            } else {
                if (data < bantu.data) {
                    bantu = bantu.leftNode;
                } else {
                    if (data > bantu.data) {
                        bantu = bantu.rightNode;
                    }
                }
            }
        }
        return null;
    }
}


// Setter untuk root
```

```java
public class TreeNode{
    int data;
    TreeNode leftNode;
    TreeNode rightNode;

    public TreeNode(int data){
        this.data=data;
        this.leftNode = null;
        this.rightNode = null;
    }
    public int getData() {
        return this.data;
    }


    public void setData(int data) {
        this.data = data;
    }


    public TreeNode getLeftNode() {
        return this.leftNode;
    }


    public void setLeftNode(TreeNode leftNode) {
        this.leftNode = leftNode;
    }


    public TreeNode getRightNode() {
        return this.rightNode;
    }


    public void setRightNode(TreeNode rightNode) {
        this.rightNode = rightNode;
    }
}
```

```java
import java.util.*;
public class App {
    public static void main(String[] args) throws Exception {
        Scanner input = new Scanner(System.in);

        // Membuat objek BinaryTree
        BinaryTree myTree = new BinaryTree();
        myTree.add(42);
        myTree.add(21);
        myTree.add(38);
        myTree.add(27);
        myTree.add(71);
        myTree.add(82);
        myTree.add(55);
        myTree.add(63);
        myTree.add(6);
        myTree.add(2);
        myTree.add(40);
        myTree.add(12);


        System.out.println("Masukkan angka yang ingin dicari : ");
         int angka = input.nextInt();
        if (myTree.getNode(angka)!= null) {
            System.out.println(angka + " Ditemukan!");
        } else {
            System.out.println("Tidak Ditemukan!");
        }









    }
}
```

## C. OUTPUT


PS C:\Users\Asus> & 'C:\Program Files\Java
ject\bin' 'App'
Masukkan angka yang ingin dicari :
38
38 Ditemukan!
PS C:\Users\Asus>

## D. ANALISA

Diagram UML diatas terdapat dua kelas utama, yaitu TreeNode dan Binary Tree. Pada kelas TreeNode menggambarkan simpul(node) dalam sebuah struktur pohon biner. Setiap simpul menyimpan sebuah data bertipe integer serta referensi ke anak kiri(leftNode) dan juga anak kanan(rightNode). Kelas ini dilengkapi dengan kosntruktor untuk inisialisasi node, serta metode getter setter yang memungkinkan pengaksesan maupun pengubahan nilai data maupun relasi antar node.

Lainnya, pada kelas BinaryTree berfungsi sebagai pengelola struktur pohon biner secara keseluruhan. Attribut utama dari kelas ini adalah root, yang merepresntasikan node akar dari pohon. Kelas ini menyediakan konstruktor kosong maupun yang dengan parameter root, serta metode untuk mengatur nilai root dan mengambil nilai dari root. Selain itu kelas ini juga memiliki metode add(int data) untuk menambahkan simpul baru sesuai dengan aturan pohon biner pencarian, dan metode getNode(int data) untuk mencati simpul tertentu berdasarkan nilai yang tersimpan

## E. REFERENSI

(Optional, jika ada silahkan cantumkan)