

LAPORAN PRAKTIKUM

STRUKTUR DATA

MODUL 6

**Disusun Oleh:**

Nama: Juan Axl Ronaldio Zaka Putra

NIM: 220411100066

Kelas: IF 2D

Dosen Pengampu:

Nama: Hermawan, S.T., M.Kom.

NIP: 197908282005011002

Asisten Praktikum:

Nama: Moh. Fadil Abdillah

NIM: 210411100142

PRODI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS TRUNOJOYO MADURA

2023

1. Soal

Buatlah sebuah code yang menampilkan binary tree Pre order dan Post order
(Jika bisa memodif agar dapat menampilkan gambar/struktur tree nya menjadi nilai tambah)

Jawaban

Code Program:

```
#include <iostream>
using namespace std;

class Node
{
public:
    int value;

    Node *left;
    Node *right;
    Node(int value)
    {
        this->value = value;
        this->left = nullptr;
        this->right = nullptr;
    }
};

void inOrderTraversal(Node *root)
{
    if (root == nullptr)
        return;

    inOrderTraversal(root->left);
    cout << root->value << " ";
    inOrderTraversal(root->right);
}

// preOrder Traversal
void preOrderTraversal(Node *root)
{
    if (root == nullptr)
        return;

    cout << root->value << " ";
    preOrderTraversal(root->left);
    preOrderTraversal(root->right);
}

// postOrder Traversal
void postOrderTraversal(Node *root)
{
    if (root == nullptr)
        return;

    postOrderTraversal(root->left);
    postOrderTraversal(root->right);
    cout << root->value << " ";
```

```

}

Node *createBinaryTree()
{
    Node *root = new Node(1);
    return root;
}

void addChildren(Node *root)
{
    Node *leftChild = new Node(2);

    Node *rightChild = new Node(3);
    Node *nodebaru1 = new Node(4);
    Node *nodebaru2 = new Node(5);

    root->left = leftChild;
    root->right = rightChild;
    rightChild->right = nodebaru1;
    rightChild->left = nodebaru2;
}

int main()
{
    Node *root = createBinaryTree();

    addChildren(root);

    cout << "inOrderTraversal:" << endl;
    inOrderTraversal(root);
    cout << "\n" << endl;

    cout << "preOrderTraversal:" << endl;
    preOrderTraversal(root);
    cout << "\n" << endl;

    cout << "postOrderTraversal:" << endl;
    postOrderTraversal(root);
    cout << "\n" << endl;

    delete root->left;
    delete root->right;

    delete root;

    return 0;
}

```

Penjelasan Code Program:

Membuat fungsi void preOrderTraversal(Node *root) , dengan parameter root dari binary tree. Fungsi ini digunakan untuk melakukan traversal pada binary tree secara pre order yaitu mengunjungi root terlebih dahulu lalu menelusuri left child dan terakhir menelusuri right child. Pada fungsi tersebut, pertama lakukan

pengkondisian menggunakan if, jika root bernilai null maka fungsi akan langsung return atau berhenti tanpa mengembalikan apa-apa. Diluar pengkondisian tersebut, mencetak value dari root lalu spasi, berikutnya melakukan rekursif dengan parameter left child dari root, dan terakhir rekursif dengan parameter right child dari root.

Membuat fungsi void postOrderTraversal(Node *root) , dengan parameter root dari binary tree. Fungsi ini digunakan untuk melakukan traversal pada binary tree secara post order yaitu menelusuri left child terlebih dahulu lalu menelusuri right child dan terakhir baru mengunjungi root. Pada fungsi tersebut, pertama lakukan pengkondisian menggunakan if, jika root bernilai null maka fungsi akan langsung return atau berhenti tanpa mengembalikan apa-apa. Diluar pengkondisian tersebut, melakukan rekursif dengan parameter left child dari root, berikutnya rekursif dengan parameter right child dari root, dan terakhir mencetak value dari root lalu spasi.

Hasil Running Program:

```
inOrderTraversal:
2 1 5 3 4

preOrderTraversal:
1 2 3 5 4

postOrderTraversal:
2 5 4 3 1

PS D:\Users\Kuliah\perkuliahan\smt2\Strukturdat> []
```