

Nama : Frederico Steven Kwok

NPM : 242310037

Kelas : TI-24-PA

Matkul : Lab. Desain dan Analisis Algoritma

Github : [https://github.com/RicoSteven120206/Desain\\_Analisis\\_Algoritma.git](https://github.com/RicoSteven120206/Desain_Analisis_Algoritma.git)

Tugas : Buatlah program sebuah C++ mengenai konsep penerapan tree dengan kriteria input bilangan ke dalam program 12, 16, 20, 24, 32, dan dapat mengurutkan bilangan yang diinput secara Pre-order, In-order dan Post-order. Hal. 65 Tugas dan Latihan

#### 1. Screenshot Source Code

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  class Node {
6  public:
7      int data;
8      Node* left;
9      Node* right;
10
11      Node(int value) {
12          data = value;
13          left = NULL;
14          right = NULL;
15      }
16 };
17
18 class Tree {
19 public:
20     Node* root;
21     Tree() {
22         root = NULL;
23     }
24
25     void insert(int value) {
26         root = insert(root, value);
27     }
28
29     Node* insert (Node* node, int value) {
30         if (node == NULL) {
31             node = new Node(value);
32         } else if (value <= node->data) {
33             node -> left = insert(node->left, value);
34         } else {
35             node -> right = insert(node->right, value);
36         }
37
38         return node;
39     }
40 }
```

41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61

```
void preorder() {  
    preorder(root);  
}  
  
void preorder(Node* node) {  
    if(node == NULL) return;  
    cout << node->data << " ";  
    preorder(node->left);  
    preorder(node->right);  
}  
  
void inorder() {  
    inorder(root);  
}  
  
void inorder(Node* node) {  
    if (node == NULL) return;  
    inorder(node->left);  
    cout<<node->data<<" ";  
    inorder(node->right);  
}
```

62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85

```
void postorder() {  
    postorder(root);  
}  
  
void postorder(Node* node) {  
    if(node == NULL) return;  
    postorder(node->left);  
    postorder(node->right);  
    cout << node->data << " ";  
}  
  
void printTree(int mode) {  
    if (mode == 1) {  
        cout<<"Pre-order : ";  
        preorder(root);  
        cout << endl;  
    }  
  
    else if (mode == 2) {  
        cout<<"In-order : ";  
        inorder(root);  
        cout << endl;  
    }  
}
```

```

86
87
88         else if (mode == 3) {
89             cout<<"Post-order : ";
90             postorder(root);
91             cout << endl;
92         }
93     };
94
95     int main(){
96         Tree tree;
97         int n;
98
99         cout<<"Masukkan jumlah simpul : ";
100        cin>>n;
101
102        for(int i = 0; i < n; i++) {
103            int data;
104            cout<<"Masukkan Simpul "<<i+1<<" : ";
105            cin >> data;
106            tree.insert(data);
107        }
108
109        cout<<"\n=====\\n";
110        tree.printTree(1);
111        cout<<endl;
112        tree.printTree(2);
113        cout<<endl;
114        tree.printTree(3);
115        cout<<endl;
116
117        return 0;
118    }

```

## 2. Screenshot Output

Jika simpul yang diinput secara berurut.

```

Masukkan jumlah simpul : 5
Masukkan Simpul 1 : 12
Masukkan Simpul 2 : 16
Masukkan Simpul 3 : 20
Masukkan Simpul 4 : 24
Masukkan Simpul 5 : 32

=====
Pre-order      : 12 16 20 24 32
In-order       : 12 16 20 24 32
Post-order     : 32 24 20 16 12

```

Jika simpul yang diinput secara acak.

```
Masukkan jumlah simpul : 5
Masukkan Simpul 1 : 12
Masukkan Simpul 2 : 32
Masukkan Simpul 3 : 20
Masukkan Simpul 4 : 16
Masukkan Simpul 5 : 24

=====
Pre-order      : 12 32 20 16 24

In-order       : 12 16 20 24 32

Post-order     : 16 24 20 32 12
```

3. Screenshot Jam Pengerjaan

