Made Naradeon Handika Pramesta
103032300101

Main.cpp

tree.h | tree.cpp | main.cpp | ostream

```cpp
#include <iostream>
#include "tree.h"

using namespace std;

int main()
{
    printf("================================================================================\n");
    adrNode root = NULL;
    int x[9] = {5,3,9,10,4,7,1,8,6};
    for (int i = 0; i < 9; i++) {
        cout << x[i] << " ";
        insertNode(root, newNode(x[i]));
    }
    cout << "\n\nPre Order\t: ";
    printPreOrder(root);
    void printPreOrder(adrNode root)t: ";
    printDecendant(root, 9);
    cout << "\n\nSum of BST Info\t\t: ";
    cout << sumNode(root);
    cout << "\nNumber of Leaves\t: ";
    cout << countLeaves(root);
    cout << "\nHeight of Tree\t\t: ";
    cout << heightTree(root);
    printf("\n================================================================================");
    return 0;
}

```

Tree.cpp

tree.h | tree.cpp | main.cpp | ostream

```cpp
#include "tree.h"

adrNode newNode(int x) {
    adrNode p = new node;
    p->info = x;
    p->left = NULL;
    p->right = NULL;
    return p;
}

adrNode findNode(adrNode root, int x) {
    if (root == NULL || root->info == x) {
        return root;
    }
    if (x < root->info) {
        return findNode(root->left, x);
    } else {
        return findNode(root->right, x);
    }
}

void insertNode(adrNode &root, adrNode p){
    if (root == NULL) {
        root = p;
    } else if (p->info < root->info) {
        insertNode(root->left, p);
    } else if (p->info > root->info) {
```

Made Naradeon Handika Pramesta
103032300101

tree.h tree.cpp main.cpp ostream

```
28              insertNode(root->right, p);
29          }
30      }
31
32  void printPreOrder(adrNode root) {
33      if (root != NULL) {
34          cout << root->info << " ";
35          printPreOrder(root->left);
36          printPreOrder(root->right);
37      }
38  }
39
40  void printDecendant(adrNode root, int x) {
41      adrNode node = findNode(root, x);
42      if (node != NULL) {
43          printPreOrder(node->left);
44          printPreOrder(node->right);
45      } else {
46          cout << "Node Tidak Ditemukan" << endl;
47      }
48  }
49
50  int sumNode(adrNode root) {
51      if (root == NULL) {
52          return 0;
53      }
54      return root->info + sumNode(root->left) + sumNode(root->right);
```

tree.h tree.cpp main.cpp ostream

```
49
50  int sumNode(adrNode root) {
51      if (root == NULL) {
52          return 0;
53      }
54      return root->info + sumNode(root->left) + sumNode(root->right);
55  }
56
57  int countLeaves(adrNode root) {
58      if (root == NULL) {
59          return 0;
60      }
61      if (root->left == NULL && root->right == NULL) {
62          return 1;
63      }
64      return countLeaves(root->left) + countLeaves(root->right);
65  }
66
67  int heightTree(adrNode root) {
68      if (root == NULL) {
69          return -1;
70      }
71      int leftHeight = heightTree(root->left);
72      int rightHeight = heightTree(root->right);
73      return 1 + max(leftHeight, rightHeight);
74  }
75
```

Tree.h

Made Naradeon Handika Pramesta
103032300101

tree.h | tree.cpp | main.cpp | ostream

```h
#ifndef TREE_H_INCLUDED
#define TREE_H_INCLUDED

#include <iostream>

using namespace std;

typedef struct node* adrNode;

struct node {
    adrNode right;
    adrNode left;
    int info;
};

adrNode newNode(int x);
adrNode findNode(adrNode root, int x);
void insertNode(adrNode &root, adrNode p);
void printPreOrder(adrNode root);
void printDecendant(adrNode root, int x);
int sumNode(adrNode root);
int countLeaves(adrNode root);
int heightTree(adrNode root);


#endif // TREE_H_INCLUDED

```

Hasil:

```
"D:\Kuliahan banget\Semester 3 Masa ga ip 4 lagi\STD KHS\TP\TP 13\bin\Debug\TP 13.exe"
================================================================================
5 3 9 10 4 7 1 8 6

Pre Order       : 5 3 1 4 9 7 6 8 10

Decendent of Node 9     : 7 6 8 10

Sum of BST Info         : 53
Number of Leaves        : 5
Height of Tree          : 3
================================================================================
Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
```