Raditya Aydin

IF4509

1301213292

Tree.h

Tree.h | Tree.cpp | main.cpp

```
1  #ifndef TREE_H_INCLUDED
2  #define TREE_H_INCLUDED
3  #include <iostream>
4
5  #define info(P) (P)->info
6  #define Lchild(P) (P) -> leftChild
7  #define Rchild(P) (P)-> rightChild
8
9  using namespace std;
10
11 typedef struct node *adrNode;
12 typedef int infotype;
13
14 struct node{
15     infotype info;
16     adrNode leftChild;
17     adrNode rightChild;
18 };
19
20
21 adrNode newNode_1301213292(infotype x);
22 adrNode findNode_1301213292(adrNode root, infotype x);
23 void insertNode_1301213292(adrNode &root, adrNode p);
24 void printPreOrder_1301213292(adrNode root);
25 void printDescendant_1301213292(adrNode root, infotype x);
26 int sumNode_1301213292(adrNode root);
27 int countLeaves_1301213292(adrNode root);
28 int heightTree_1301213292(adrNode root);
29
30 #endif // TREE_H_INCLUDED
```

Tree.cpp

Tree.h | Tree.cpp | main.cpp

```cpp
#include "Tree.h"

adrNode newNode_1301213292(infotype x){
    adrNode p = new node;
    info(p) = x;
    Lchild(p) = NULL;
    Rchild(p) = NULL;
    return p;
}

adrNode findNode_1301213292(adrNode root, infotype x){
    if(root == NULL){
        return NULL;
    }else{
        if(info(root) == x){
            return root;
        }else if(x < info(root)){
            findNode_1301213292(Lchild(root), x);
        }else{
            findNode_1301213292(Rchild(root), x);
        }
    }
}

void insertNode_1301213292(adrNode &root, adrNode p){
    if(root == NULL){
        root = p;
    }else{
        if(info(p) < info(root)){
            insertNode_1301213292(Lchild(root), p);
        }else if(info(p) > info(root)){
            insertNode_1301213292(Rchild(root), p);
        }else{
            cout << "Data sudah terdaftar" << endl;
        }
    }
}
```

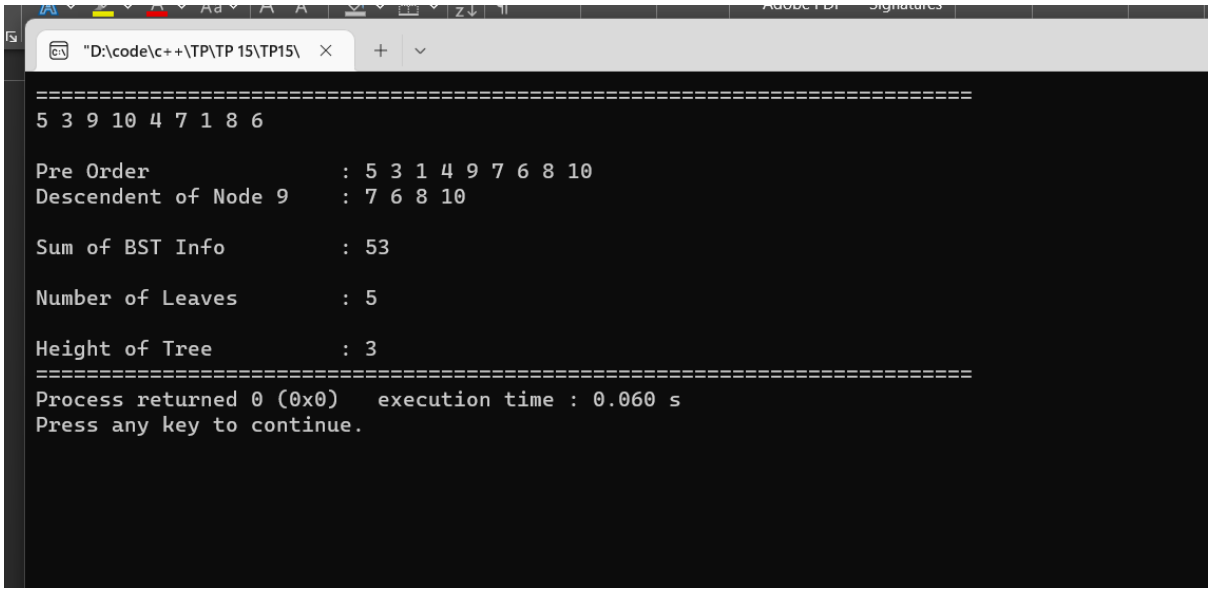
```
39  void printPreOrder_1301213292(adrNode root){
40      if(root != NULL){
41          cout << info(root) << " ";
42          printPreOrder_1301213292(Lchild(root));
43          printPreOrder_1301213292(Rchild(root));
44      }
45  }
46
47  void printDescendant_1301213292(adrNode root, infotype x){
48      adrNode target = findNode_1301213292(root, x);
49      if(root == NULL){
50          cout << "Tree kosong" << endl;
51      }
52      printPreOrder_1301213292(Lchild(target));
53      printPreOrder_1301213292(Rchild(target));
54  }
55
56  int sumNode_1301213292(adrNode root){
57      if(root == NULL){
58          return 0;
59      }
60      return info(root) + sumNode_1301213292(Lchild(root)) + sumNode_1301213292(Rchild(root));
61  }
62
63  int countLeaves_1301213292(adrNode root){
64      if(root == NULL){
65          return 0;
66      }
67      if(Lchild(root) == NULL && Rchild(root) == NULL){
68          return 1;
69      }
70      return countLeaves_1301213292(Lchild(root)) + countLeaves_1301213292(Rchild(root));
71  }
72
73  int heightTree_1301213292(adrNode root){
74      if(root == NULL){
75          return -1;
76      }
77      int kiri = heightTree_1301213292(Lchild(root));
78      int kanan = heightTree_1301213292(Rchild(root));
79      return max(kiri, kanan) + 1;
80  }
81
```

Main.cpp

Tree.h | Tree.cpp | main.cpp

```
1   lude <iostream>
2   lude "Tree.h"
3
4   g namespace std;
5
6   main()
7   {
8   cout << "================================================================================"<<endl;
9   adrNode root = NULL;
10  int x[9] = {5,3,9,10,4,7,1,8,6};
11
12  for(int i = 0; i < 9; i++){
13      cout << x[i] << " ";
14  }
15
16  for(int i=0; i<9; i++){
17     adrNode nodeBaru = newNode_1301213292(x[i]);
18     insertNode_1301213292(root, nodeBaru);
19  }
20
21  cout <<"\n\nPre Order\t\t: ";
22  printPreOrder_1301213292(root);
23
24  cout <<"\nDescendent of Node 9\t: ";
25  printDescendant_1301213292(root, 9);
26
27  cout <<"\n\nSum of BST Info\t\t: " << sumNode_1301213292(root)<<endl;
28
29  cout <<"\nNumber of Leaves\t: " << countLeaves_1301213292(root)<<endl;
30
31  cout <<"\nHeight of Tree\t\t: "<< heightTree_1301213292(root)<<endl;
32
33  cout<< "================================================================================";
34  return 0;
35
36  }
```

Output


"D:\code\c++\TP\TP 15\TP15\
5 3 9 10 4 7 1 8 6
Pre Order : 5 3 1 4 9 7 6 8 10
Descendent 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 : 0.060 s
Press any key to continue.