Jawaban Tugas Pendahuluan Modul 11

Nama: Sheren Aulia Azahra

NIM: 103032400036

Kelas: IT-48-05

File tree.h

h tree.cpp main.cpp

```
#ifndef TREE_H_INCLUDED
#define TREE_H_INCLUDED

typedef int infotype;
typedef struct Node *adrNode;
struct Node {
    infotype info;
    adrNode left;
    adrNode right;
} ;

void createTree_103032400036(adrNode &root);
adrNode createNode_103032400036(infotype x);
void insertNode_103032400036(adrNode &root, adrNode p);
adrNode searchNode_103032400036(adrNode root, infotype x);
void displayTree_103032400036(adrNode root);
int countNodes_103032400036(adrNode root);
infotype getMinValue_103032400036(adrNode root);
infotype getMaxValue_103032400036(adrNode root);
#endif // TREE_H_INCLUDED

```

File tree.cpp

tree.h tree.cpp main.cpp

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

using namespace std;

void createTree_103032400036(adrNode &root){
    root = nullptr;
}

adrNode createNode_103032400036(infotype x){
    adrNode p;

    p = new Node;
    p->info = x;
    p->left = nullptr;
    p->right = nullptr;

    return p;
}

void insertNode_103032400036(adrNode &root, adrNode p){
    if (root == nullptr) {
        root = p;
    } else if (p->info < root->info) {
        insertNode_103032400036(root->left, p);
    } else {
        insertNode_103032400036(root->right,p);
    }
}

adrNode searchNode_103032400036(adrNode root, infotype x){
    if (root == nullptr){
        return nullptr;
    } else if (x < root->info) {
        return searchNode_103032400036(root->left, x);
    } else if (x > root->info) {
        return searchNode_103032400036(root->right, x);
    }
    return root;
}

void  displayTree_103032400036(adrNode root){
    if (root != nullptr){
        displayTree_103032400036(root->left);
        cout << root->info << " ";
        displayTree_103032400036(root->right);
    }
}

int countNodes_103032400036(adrNode root) {
    if (root == nullptr){
        return 0;
    } else {
        return 1 + countNodes_103032400036(root->left) + countNodes_103032400036(root->right);
    }
}
```

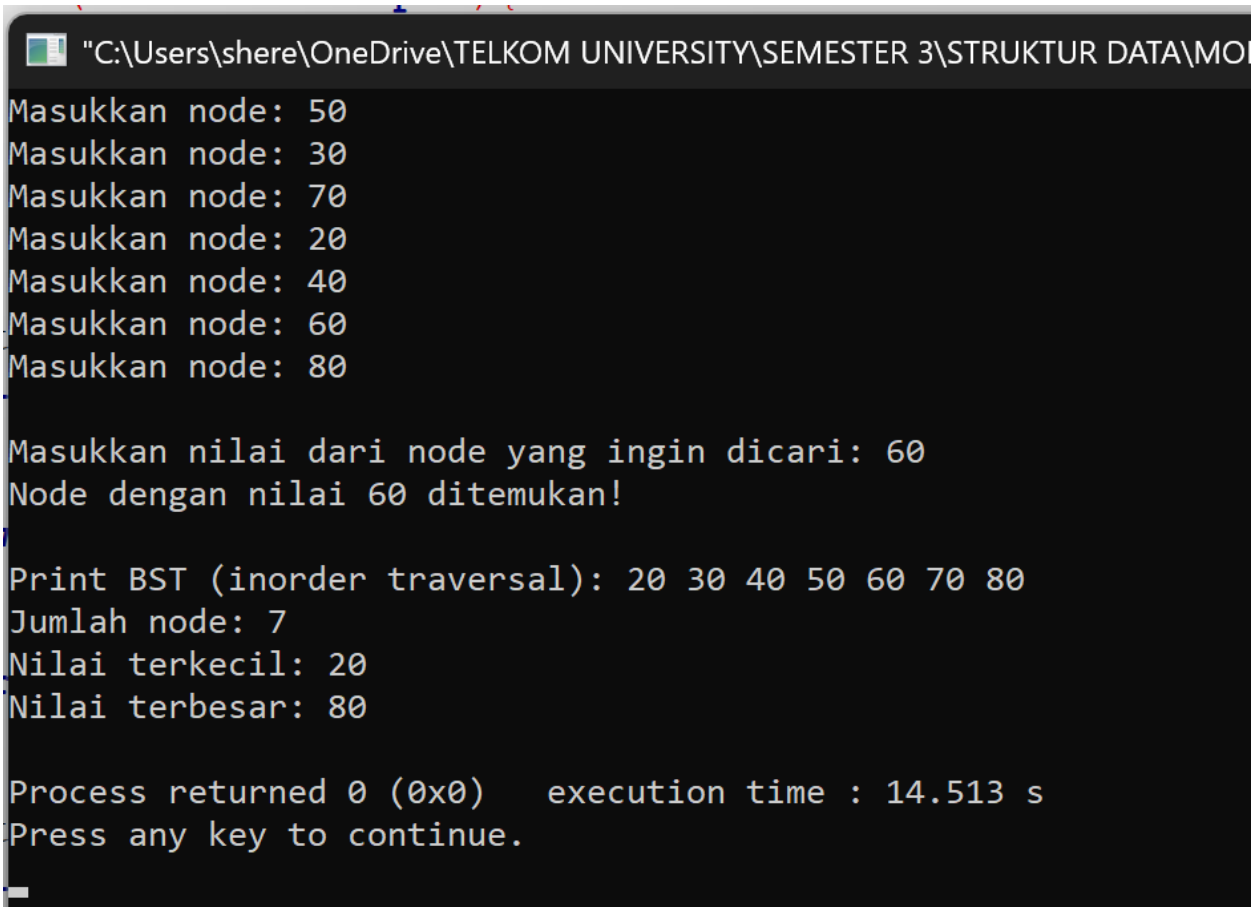
```
57
58  infotype getMinValue_103032400036(adrNode root){
59      if (root == nullptr){
60          return -1;
61      }
62      while (root->left != nullptr){
63          root = root->left;
64      }
65      return root->info;
66  }
67
68  infotype getMaxValue_103032400036(adrNode root){
69      if (root == nullptr){
70          return -1;
71      }
72      while (root->right != nullptr){
73          root = root->right;
74      }
75      return root->info;
76  }
77
```

File main.cpp

tree.h | tree.cpp | main.cpp

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

using namespace std;

int main(){
    adrNode p, root;
    infotype x;
    createTree_103032400036(root);

    for (int i = 0; i < 7; i++){
        cout << "Masukkan node: ";
        cin >> x;
        p = createNode_103032400036(x);
        insertNode_103032400036(root,p);
    }
    cout << endl;

    int cari;
    cout << "Masukkan nilai dari node yang ingin dicari: ";
    cin >> cari;

    adrNode hasil = searchNode_103032400036(root,cari);
    if (hasil != nullptr) {
        cout << "Node dengan nilai " << hasil->info << " ditemukan!" << endl;
    }
    cout << endl;

    cout << "Print BST (inorder traversal): ";
    displayTree_103032400036(root);
    cout << endl;

    cout << "Jumlah node: " << countNodes_103032400036(root) << endl;
    cout << "Nilai terkecil: " << getMinValue_103032400036(root) << endl;
    cout << "Nilai terbesar: " << getMaxValue_103032400036(root) << endl;
    return 0;
}
```

Output


"C:\Users\shere\OneDrive\TELKOM UNIVERSITY\SEMESTER 3\STRUKTUR DATA\MO
Masukkan node: 50
Masukkan node: 30
Masukkan node: 70
Masukkan node: 20
Masukkan node: 40
Masukkan node: 60
Masukkan node: 80
Masukkan nilai dari node yang ingin dicari: 60
Node dengan nilai 60 ditemukan!
Print BST (inorder traversal): 20 30 40 50 60 70 80
Jumlah node: 7
Nilai terkecil: 20
Nilai terbesar: 80
Process returned 0 (0x0) execution time : 14.513 s
Press any key to continue.