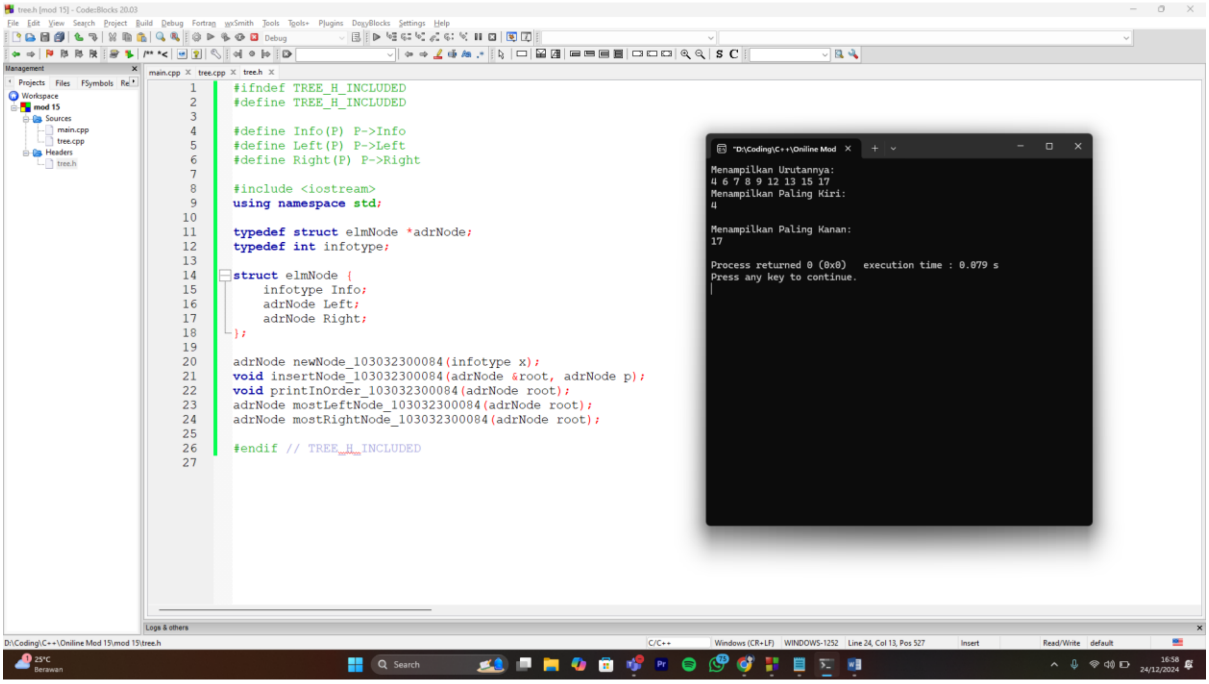
Nama : Delkano Marzuki Berutu

NIM : 103032300084

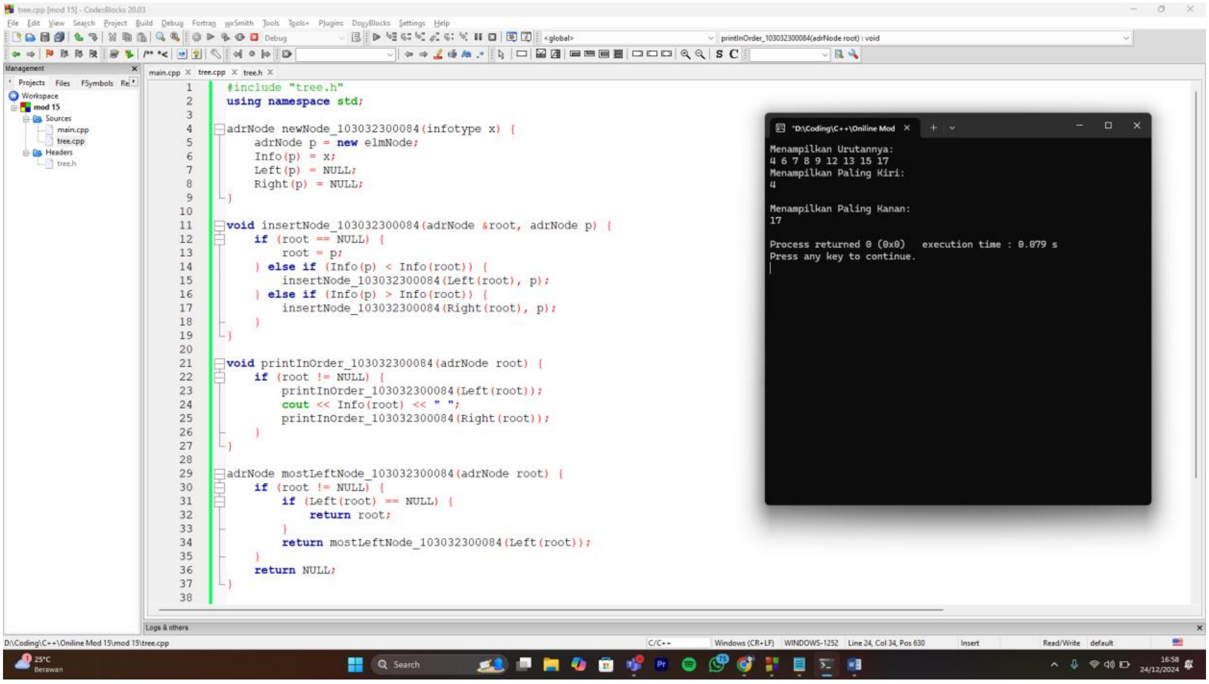
tree.h



```cpp
#ifndef TREE_H_INCLUDED
#define TREE_H_INCLUDED

#define Info(P) P->Info
#define Left(P) P->Left
#define Right(P) P->Right

#include <iostream>
using namespace std;

typedef struct elmNode *adrNode;
typedef int infotype;

struct elmNode {
    infotype Info;
    adrNode Left;
    adrNode Right;
};

adrNode newNode_103032300084(infotype x);
void insertNode_103032300084(adrNode &root, adrNode p);
void printInOrder_103032300084(adrNode root);
adrNode mostLeftNode_103032300084(adrNode root);
adrNode mostRightNode_103032300084(adrNode root);

#endif // TREE_H_INCLUDED
```

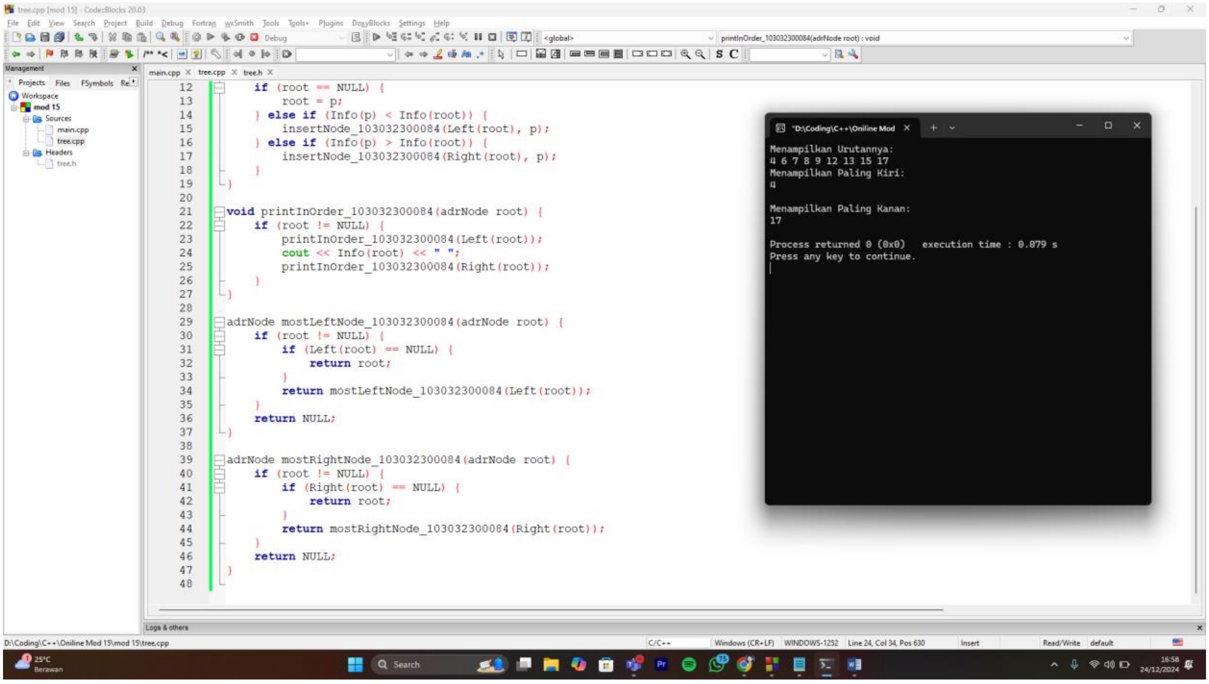
```
Menampilkan Urutannya:
4 6 7 8 9 12 13 15 17
Menampilkan Paling Kiri:
4

Menampilkan Paling Kanan:
17

Process returned 0 (0x0)   execution time : 0.079 s
Press any key to continue.
```

tree.cpp



```cpp
#include "tree.h"
using namespace std;

adrNode newNode_103032300084(infotype x) {
    adrNode p = new elmNode;
    Info(p) = x;
    Left(p) = NULL;
    Right(p) = NULL;
}

void insertNode_103032300084(adrNode &root, adrNode p) {
    if (root == NULL) {
        root = p;
    } else if (Info(p) < Info(root)) {
        insertNode_103032300084(Left(root), p);
    } else if (Info(p) > Info(root)) {
        insertNode_103032300084(Right(root), p);
    }
}

void printInOrder_103032300084(adrNode root) {
    if (root != NULL) {
        printInOrder_103032300084(Left(root));
        cout << Info(root) << " ";
        printInOrder_103032300084(Right(root));
    }
}

adrNode mostLeftNode_103032300084(adrNode root) {
    if (root != NULL) {
        if (Left(root) == NULL) {
            return root;
        }
        return mostLeftNode_103032300084(Left(root));
    }
    return NULL;
}
```

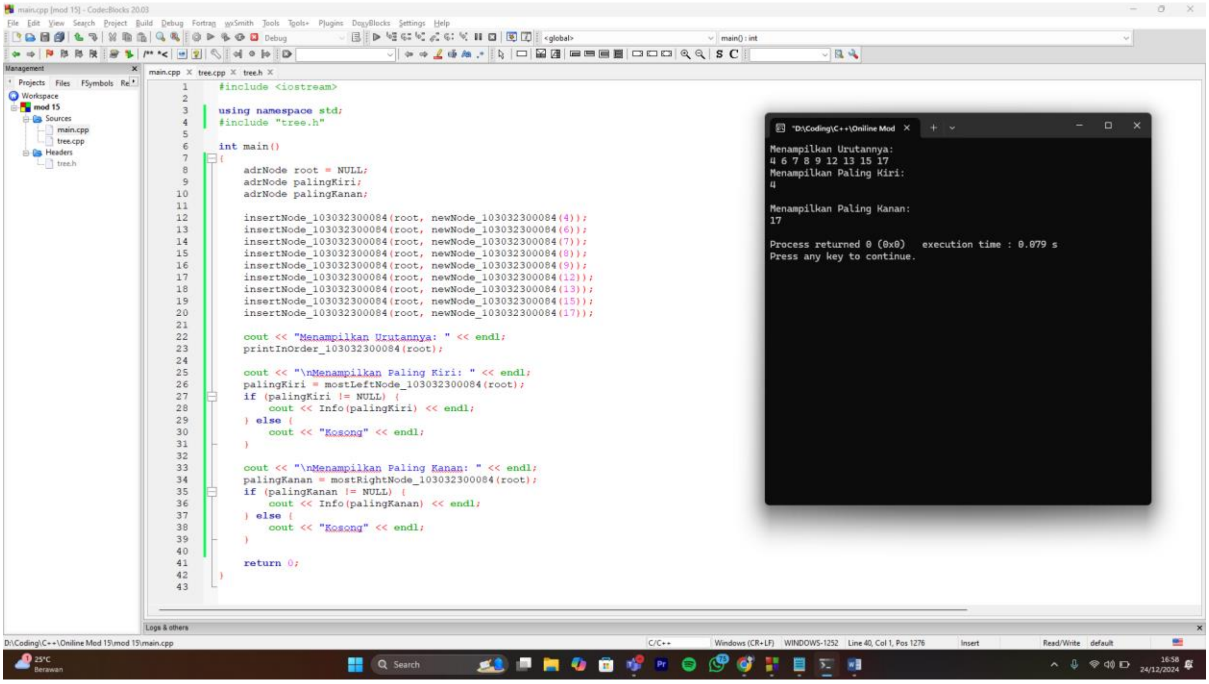
```
Menampilkan Urutannya:
4 6 7 8 9 12 13 15 17
Menampilkan Paling Kiri:
4

Menampilkan Paling Kanan:
17

Process returned 0 (0x0)   execution time : 0.079 s
Press any key to continue.
```


tree.cpp [mod 15] - Code::Blocks 20.03
main.cpp
tree.cpp
tree.h
D:\Coding\C++\Oniline Mod 15\mod 15\tree.cpp

```cpp
    if (root == NULL) {
        root = p;
    } else if (Info(p) < Info(root)) {
        insertNode_103032300084(Left(root), p);
    } else if (Info(p) > Info(root)) {
        insertNode_103032300084(Right(root), p);
    }
}

void printInOrder_103032300084(adrNode root) {
    if (root != NULL) {
        printInOrder_103032300084(Left(root));
        cout << Info(root) << " ";
        printInOrder_103032300084(Right(root));
    }
}

adrNode mostLeftNode_103032300084(adrNode root) {
    if (root != NULL) {
        if (Left(root) == NULL) {
            return root;
        }
        return mostLeftNode_103032300084(Left(root));
    }
    return NULL;
}

adrNode mostRightNode_103032300084(adrNode root) {
    if (root != NULL) {
        if (Right(root) == NULL) {
            return root;
        }
        return mostRightNode_103032300084(Right(root));
    }
    return NULL;
}
```

```
Menampilkan Urutannya:
4 6 7 8 9 12 13 15 17
Menampilkan Paling Kiri:
4

Menampilkan Paling Kanan:
17

Process returned 0 (0x0)   execution time : 0.079 s
Press any key to continue.
```

main.cpp


main.cpp [mod 15] - Code::Blocks 20.03
main.cpp
tree.cpp
tree.h
D:\Coding\C++\Oniline Mod 15\mod 15\main.cpp

```cpp
#include <iostream>

using namespace std;
#include "tree.h"

int main()
{
    adrNode root = NULL;
    adrNode palingKiri;
    adrNode palingKanan;

    insertNode_103032300084(root, newNode_103032300084(4));
    insertNode_103032300084(root, newNode_103032300084(6));
    insertNode_103032300084(root, newNode_103032300084(7));
    insertNode_103032300084(root, newNode_103032300084(8));
    insertNode_103032300084(root, newNode_103032300084(9));
    insertNode_103032300084(root, newNode_103032300084(12));
    insertNode_103032300084(root, newNode_103032300084(13));
    insertNode_103032300084(root, newNode_103032300084(15));
    insertNode_103032300084(root, newNode_103032300084(17));

    cout << "Menampilkan Urutannya: " << endl;
    printInOrder_103032300084(root);

    cout << "\nMenampilkan Paling Kiri: " << endl;
    palingKiri = mostLeftNode_103032300084(root);
    if (palingKiri != NULL) {
        cout << Info(palingKiri) << endl;
    } else {
        cout << "Kosong" << endl;
    }

    cout << "\nMenampilkan Paling Kanan: " << endl;
    palingKanan = mostRightNode_103032300084(root);
    if (palingKanan != NULL) {
        cout << Info(palingKanan) << endl;
    } else {
        cout << "Kosong" << endl;
    }

    return 0;
}
```

```
Menampilkan Urutannya:
4 6 7 8 9 12 13 15 17
Menampilkan Paling Kiri:
4

Menampilkan Paling Kanan:
17

Process returned 0 (0x0)   execution time : 0.079 s
Press any key to continue.
```