TP_MOD_15_103032330095_GENA DARMA

1. Tree.h

main.cpp | Tree.h | Tree.cpp

```
#ifndef TREE_H_INCLUDED
#define TREE_H_INCLUDED

#include <iostream>

using namespace std;

typedef int infotype;

typedef struct Node *adrNode;

struct Node {
    infotype info;
    adrNode left;
    adrNode right;
};

adrNode newNode_103032330095(infotype x);

void insertNode_103032330095(adrNode &root, adrNode p);

void deleteNode_103032330095(adrNode &root, adrNode &p);

void printInOrder_103032330095(adrNode root);

adrNode findMin_103032330095(adrNode root);

adrNode findNode_103032330095(adrNode root, infotype x);

#endif // TREE_H_INCLUDED

```

2. Tree.cpp

main.cpp | Tree.h | Tree.cpp

```cpp
#include "Tree.h"

adrNode newNode_103032330095(infotype x){
    adrNode p = new Node;
    p->info = x;
    p->left = NULL;
    p->right = NULL;
    return p;
}

void insertNode_103032330095(adrNode &root, adrNode p){
    if (root == NULL){
        root = p;
    } else {
        if (root->info > p->info) {
            insertNode_103032330095(root->left, p);
        } else if (root->info < p->info){
            insertNode_103032330095(root->right, p);
        } else {
            cout << endl << "Node Duplicate" << endl;
        }
    }
}

```

main.cpp | Tree.h | Tree.cpp

```
25  void deleteNode_103032330095(adrNode &root, adrNode &p){
26      adrNode temp;
27      if (root == NULL){
28          cout << "Node tidak ketemu!" << endl;
29      } else if (root->info > p->info){
30          deleteNode_103032330095(root->left, p);
31      } else if (root->info < p->info) {
32          deleteNode_103032330095(root->right, p);
33      } else {
34          if (root->left == NULL && root->right == NULL){
35              delete root;
36              root = NULL;
37          } else if (root->right == NULL){
38              temp = root;
39              root = root->left;
40              delete temp;
41          } else if (root->left == NULL){
42              temp = root;
43              root = root->right;
44              delete temp;
45          } else {
46              temp = findMin_103032330095(root->right);
47              root->info = temp->info;
48              deleteNode_103032330095(root->right, temp);
49          }
50      }
51  }
52
53  void printInOrder_103032330095(adrNode root){
54      if (root != NULL){
55          printInOrder_103032330095(root->left);
56          cout << root->info << " ";
57          printInOrder_103032330095(root->right);
58      }
59  }
60
```

```
61  adrNode findMin_103032330095(adrNode root){
62      if (root->left == NULL){
63          return root;
64      } else {
65          return findMin_103032330095(root->left);
66
67      }
68  }
69
70  adrNode findNode_103032330095(adrNode root, infotype x){
71      if (root->info == x || root == NULL){
72          return root;
73      } else {
74          if (root->info > x){
75              return findNode_103032330095(root->left, x);
76          } else if (root->info < x){
77              return findNode_103032330095(root->right, x);
78          }
79      }
80  }
81
```

3. Main.cpp

main.cpp | Tree.h | Tree.cpp

```cpp
#include "Tree.h"
int main()
{
    int x[9] = {8,6,15,4,7,12,17,9,13};
    int y[9] = {8,9,12,13,15,17,7,6,4};
    int i;
    adrNode root;
    adrNode P;

    root = NULL;

    for (i = 0; i < 9; i++){
        P = newNode_103032330095(x[i]);
        insertNode_103032330095(root, P);
    }
    cout << endl;

    for (i = 0; i < 9; i++){
        if (root != NULL){
            printInOrder_103032330095(root);
            cout << endl;
            P = findNode_103032330095(root, y[i]);
            deleteNode_103032330095(root, P);
        }
    }
    cout << "(kosong)";
    return 0;
}
```

4. Output

```
"D:\KuliahGena\SMT 3\Strukt

4 6 7 8 9 12 13 15 17
4 6 7 9 12 13 15 17
4 6 7 12 13 15 17
4 6 7 13 15 17
4 6 7 15 17
4 6 7 17
4 6 7
4 6
4
(kosong)
Process returned 0 (0x0)   execution time : 0.026 s
Press any key to continue.
```