Made Naradeon Handika Pramesta 103032300101

Main.cpp

```
tree.cpp X tree.h X main.cpp X
     1
          #include <iostream>
         #include "tree.h"
    3
     4
         using namespace std;
     5
         int main()
     6
     7
        ₽{
    8
              int x[9] = {8,6,15,4,7,12,17,9,13};
    9
              int y[9] = {8,9,12,13,15,17,7,6,4};
   10
             adrNode root = NULL;
   11
            for (int i=0; i<9; i++) {
   12
                 cout << x[i] << " ";
   13
   14
            cout << endl;
             adrNode P = NULL;
   15
   16
           for (int i=0; i<9; i++) {
   17
                  insertNode_103032300101(root, newNode_103032300101(x[i]));
   18
   19
             cout << endl;
   20
             for (int i=0; i<10; i++) {
                 if (root != NULL) {
   21
   22
                     printInOrder_103032300101(root);
   23
                     cout << endl;</pre>
   24
                     P = findNode 103032300101(root, y[i]);
   25
                     DeleteNode 103032300101(root, P);
   26
                 }else{
   27
                      cout << "(kosong)";
    28
    29
    30
              return 0;
    31
          }
    32
```

Tree.cpp

```
tree.cpp X tree.h X main.cpp X
          #include "tree.h"
     2
     3
         □adrNode newNode_103032300101(infotype x) {
              adrNode n = new node;
     4
     5
              n->info = x;
     6
              n->left = NULL;
     7
              n->right = NULL;
     8
              return n;
         L3
     9
    10
         □adrNode findNode 103032300101(adrNode root, infotype x) {
    11
    12
              if (root == NULL || root->info == x) {
    13
                   return root;
    14
              } else {
    15
                   if (root->info > x) {
    16
                      return findNode_103032300101(root->left, x);
    17
                   } else if (root->info < x) {</pre>
    18
                      return findNode_103032300101(root->right, x);
    19
    20
         L,
    21

□void insertNode 103032300101 (adrNode &root, adrNode p) {

    22
    23
              if (root == NULL) {
    24
                   root = p;
    25
               } else if (p->info < root->info) {
    26
                   insertNode_103032300101(root->left, p);
    27
               } else if (p->info > root->info) {
    28
                   insertNode_103032300101(root->right, p);
    29
         L<sub>3</sub>
    30
```

```
tree.cpp X tree.h X main.cpp X
    28
                   insertNode 103032300101(root->right, p);
    29
         L,
    30
         □void DeleteNode_103032300101(adrNode &root, adrNode &p) {
    31
    32
              adrNode temp;
              if (root == NULL) {
    33
    34
                  cout << "Node not found" << endl;</pre>
    35
              }else if (p->info < root->info) {
    36
                  DeleteNode_103032300101(root->left, p);
    37
              }else if (p->info > root->info) {
    38
                  DeleteNode 103032300101(root->right, p);
    39
              }else{
    40
                  if (root->left == NULL && root->right == NULL) {
    41
                      delete root;
                      root = NULL;
    42
                   }else if (root->left == NULL) {
    43
    44
                       temp = root;
    45
                      root = root->right;
    46
                      delete temp;
    47
                   }else if (root->right == NULL) {
    48
                       temp = root;
    49
                       root = root->left;
    50
                       delete temp;
    51
                  }else{
    52
                       temp = findMin 103032300101(root->right);
    53
                       root->info = temp->info;
    54
                      DeleteNode 103032300101(root->right, temp);
    55
                   }
    56
              }
    57
```

```
tree.cpp X tree.h X main.cpp X
                      root = root->right;
    46
                      delete temp;
    47
                  }else if (root->right == NULL) {
    48
                      temp = root;
    49
                      root = root->left;
    50
                      delete temp;
    51
                  }else{
    52
                      temp = findMin 103032300101(root->right);
    53
                      root->info = temp->info;
    54
                      DeleteNode 103032300101 (root->right, temp);
    55
                  }
    56
    57

─void printInOrder 103032300101 (adrNode root) {
    58
    59
              if (root != NULL) {
    60
                  printInOrder 103032300101(root->left);
    61
                  cout << root->info << " ";
    62
                  printInOrder_103032300101(root->right);
    63
         L
    64
    65
         □adrNode findMin_103032300101(adrNode root) {
    66
              if (root == NULL) {
    67
                  return NULL;
    68
    69
              while (root->left != NULL) {
    70
                  root = root->left;
    71
    72
              return root;
    73
          }
    74
```

Tree.h

```
tree.cpp X tree.h X main.cpp X
          #ifndef TREE H INCLUDED
          #define TREE H INCLUDED
     2
          #include <iostream>
     3
     4
     5
         using namespace std;
     6
     7
         typedef int infotype;
     8
         typedef struct node* adrNode;
     9
    10 | struct node {
    11
              infotype info;
    12
              adrNode left;
              adrNode right;
    13
        L};
    14
    15
         adrNode newNode 103032300101(infotype x);
    16
         adrNode findNode 103032300101 (adrNode root, infotype x);
    17
         void insertNode_103032300101(adrNode &root, adrNode p);
    18
    19
         void DeleteNode 103032300101(adrNode &root, adrNode &p);
         void printInOrder 103032300101(adrNode root);
    20
    21
         adrNode findMin 103032300101(adrNode root);
    22
    23
    24
        #endif // TREE_H_INCLUDED
    25
```

Hasil:

```
8 6 15 4 7 12 17 9 13

4 6 7 8 9 12 13 15 17

4 6 7 9 12 13 15 17

4 6 7 13 15 17

4 6 7 15 17

4 6 7 17

4 6 7 17

4 6 7 17

Process returned 0 (0x0) execution time : 0.042 s

Press any key to continue.
```