Nama : Muhammad Alvin Faa'iz

NIM : 103012400229

Kelas : IF-48-11

Header

header.h | sources.cpp | main.cpp

```cpp
#ifndef HEADER_H_INCLUDED
#define HEADER_H_INCLUDED

typedef int infotype;

typedef struct node *adrNode;

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

void createTree(adrNode &root);
adrNode createNode(infotype x);
void insertNode(adrNode &root, adrNode p);
void displayTree(adrNode root);
infotype getMinNode(adrNode root);
infotype getMaxNode(adrNode root);


#endif // HEADER_H_INCLUDED
```

Sources

header.h | sources.cpp | main.cpp

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

using namespace std;

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

adrNode createNode(infotype x){
    adrNode p = new node;

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

    return p;
}

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

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

int getMinNode(adrNode root){
    adrNode p = root;

    while(p != nullptr && p->left != nullptr){
        p = p->left;
    }

    return p->info;
}

int getMaxNode(adrNode root){
    adrNode p = root;

    while(p != nullptr && p->right != nullptr){
        p = p->right;
    }

    return p->info;
}
```

Main

header.h | sources.cpp | main.cpp

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

using namespace std;

int main()
{
    adrNode root, p;
    int i, x;
    createTree(root);

    for (i = 0; i < 9; i++){
        cout << "Masukkan node: ";
        cin >> x;
        p = createNode(x);
        insertNode(root, p);
    }

    cout << endl;
    cout << "Node dalam BST : ";
    displayTree(root);
    cout << endl;
    cout << "Node terkecil dalam BST: " << getMinNode(root) << endl;
    cout << "Node terbesar dalam BST: " << getMaxNode(root);
    return 0;
}
```

Output

```
Masukkan node: 8
Masukkan node: 6
Masukkan node: 15
Masukkan node: 4
Masukkan node: 7
Masukkan node: 12
Masukkan node: 17
Masukkan node: 9
Masukkan node: 13

Node dalam BST : 4 6 7 8 9 12 13 15 17
Node terkecil dalam BST: 4
Node terbesar dalam BST: 17
```