STRUKTUR
DATA

Nama Kelompok 6 :

1. **Sayydina Aulia (034)**
2. **Dian Nur Safitri (044)**
3. **Ananda Putri R (046)**
4. **Talhah Alfian Padma (054)**
5. **Bagas Syafiq P (064)**

D4 Manajemen Informatika 2021 B

- Membuat Fungsi untuk membangun sebuah binary search tree

C:\Users\user\Downloads\binary search tree.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug | binary search tree.cpp

```cpp
#include<iostream>
#define SPACE 10

using namespace std;
//mendeklarasikan class
class TreeNode {
  public:
    int value;
  TreeNode * left;
  TreeNode * right;

  TreeNode() {
    value = 0;
    left = NULL;
    right = NULL;
  }
  TreeNode(int v) {
    value = v;
    left = NULL;
    right = NULL;
  }
};

class BST {
  public:
    TreeNode * root;
  BST() {
    root = NULL;
  }
  bool isTreeEmpty() {
    if (root == NULL) {
      return true;
    } else {
      return false;
    }
  }
// fungsi untuk menambahkan nilai ke tree
  void insertNode(TreeNode * new_node) {
    if (root == NULL) {
```

Compiler Resources Compile Log Debug Find Results

Line: 1 Col: 1 Sel: 0 Lines: 263 Length: 6394 Insert Done parsing in 0,469 seconds

C:\Users\user\Downloads\binary search tree.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug | binary search tree.cpp

```cpp
  void insertNode(TreeNode * new_node) {
    if (root == NULL) {
      root = new_node;
      cout << "Value Inserted as root node!" << endl;
    } else {
      TreeNode * temp = root;
      while (temp != NULL) {
        if (new_node -> value == temp -> value) {
          cout << "Nilai telah ada," <<
            "masukkan nilai yang lain" << endl;
          return;
        } else if ((new_node -> value < temp -> value) && (temp -> left == NULL)) {
          temp -> left = new_node;
          cout << "nilai dimasukkan ke kiri" << endl;
          break;
        } else if (new_node -> value < temp -> value) {
          temp = temp -> left;
        } else if ((new_node -> value > temp -> value) && (temp -> right == NULL)) {
          temp -> right = new_node;
          cout << "nilai dimasukkan ke kanan" << endl;
          break;
        } else {
          temp = temp -> right;
        }
      }
    }
  }
  TreeNode* insertRecursive(TreeNode *r, TreeNode *new_node)
  {
      if(r==NULL)
      {
          r=new_node;
          cout <<"sukses memasukkan nilai"<<endl;
          return r;
      }

      if(new_node->value < r->value)
      {
          r->left = insertRecursive(r->left,new_node);
```

Compiler Resources Compile Log Debug Find Results

Line: 1 Col: 1 Sel: 0 Lines: 263 Length: 6394 Insert Done parsing in 0,469 seconds

C:\Users\user\Downloads\binary search tree.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug | binary search tree.cpp

```cpp
        }
        else if (new_node->value > r->value)
        {
            r->right = insertRecursive(r->right,new_node);
        }
       else
       {
            cout << "tidak boleh ada nilai yang sama" << endl;
            return r;
        }
        return r;
    }

  void print2D(TreeNode * r, int space) {
    if (r == NULL) // Base case
      return;
    space += SPACE; // meningkatkan jarak diantara angka
    print2D(r -> right, space); // memproses children atau subtree disebelah kanan terlebih dahulu
    cout << endl;
    for (int i = SPACE; i < space; i++)
      cout << " ";
    cout << r -> value << "\n";
    print2D(r -> left, space); // memproses children disebelah kiri
  }

  void printPreorder(TreeNode * r) //mengurutkan dari(nilai sekarang, kiri, kanan)
  {
    if (r == NULL)
      return;
    /* print nilai pertama terlebih dahulu*/
    cout << r -> value << " ";
    /* rekursif dari kiri*/
    printPreorder(r -> left);
    /* lalu dilanjutkan rekursif dari kanan */
    printPreorder(r -> right);
  }

  void printInorder(TreeNode * r) //mengurutkan dari  (kiri, nilai sekarang, kanan)
  {
```

Compiler Resources Compile Log Debug Find Results

Line: 1 Col: 1 Sel: 0 Lines: 263 Length: 6394 Insert Done parsing in 0,469 seconds

C:\Users\user\Downloads\binary search tree.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug | binary search tree.cpp

```cpp

  void printInorder(TreeNode * r) //mengurutkan dari  (kiri, nilai sekarang, kanan)
  {
    if (r == NULL)
      return;
    /* rekursif dari kiri */
    printInorder(r -> left);
    /* print data yang sekarang*/
    cout << r -> value << " ";
    /* rekursif dari kanan*/
    printInorder(r -> right);
  }
  void printPostorder(TreeNode * r) //mengurutkan dari (kiri, kanan, Root)
  {
    if (r == NULL)
      return;
    // rekursif dari kiri
    printPostorder(r -> left);
    // rekursif dari kanan
    printPostorder(r -> right);
    // print data yang sekarang
    cout << r -> value << " ";
  }

  TreeNode * iterativeSearch(int v) {
    if (root == NULL) {
      return root;
    } else {
      TreeNode * temp = root;
      while (temp != NULL) {
        if (v == temp -> value) {
          return temp;
        } else if (v < temp -> value) {
          temp = temp -> left;
        } else {
          temp = temp -> right;
        }
      }
      return NULL;
```

Compiler Resources Compile Log Debug Find Results

Line: 1 Col: 1 Sel: 0 Lines: 263 Length: 6394 Insert Done parsing in 0,469 seconds

C:\Users\user\Downloads\binary search tree.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug

binary search tree.cpp

```cpp
      }
      return NULL;
    }
  }

  TreeNode * recursiveSearch(TreeNode * r, int val) {
    if (r == NULL || r -> value == val)
      return r;

    else if (val < r -> value)
      return recursiveSearch(r -> left, val);

    else
      return recursiveSearch(r -> right, val);
  }

  int height(TreeNode * r) {
    if (r == NULL)
      return -1;
    else {
      /* menghitung tinggi setiap subtree */
      int lheight = height(r -> left);
      int rheight = height(r -> right);

      /* gunakan yang terbesar */
      if (lheight > rheight)
        return (lheight + 1);
      else return (rheight + 1);
    }
  }

  /* Print angka/nilai pada tempat yang telah ditentukan*/
  void printGivenLevel(TreeNode * r, int level) {
    if (r == NULL)
      return;
    else if (level == 0)
      cout << r -> value << " ";
    else  level > 0;
    {
```

Compiler Resources Compile Log Debug Find Results

Line: 1 Col: 1 Sel: 0 Lines: 263 Length: 6394 Insert Done parsing in 0,469 seconds

C:\Users\user\Downloads\binary search tree.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug

binary search tree.cpp

```cpp
    else  level > 0;
    {
      printGivenLevel(r -> left, level - 1);
      printGivenLevel(r -> right, level - 1);
    }
  }
  void printLevelOrderBFS(TreeNode * r) {
    int h = height(r);
    for (int i = 0; i <= h; i++)
      printGivenLevel(r, i);
  }

  TreeNode * minValueNode(TreeNode * node) {
    TreeNode * current = node;
    /* loop down to find the leftmost leaf */
    while (current -> left != NULL) {
      current = current -> left;
    }
    return current;
  }



};

int main() {
  BST obj;
  int option, val;

  do {
    cout << "What operation do you want to perform? " <<
      " Select Option number. Enter 0 to exit." << endl;
    cout << "1. Insert Node" << endl;
    cout << "2. Print/Traversal BST values" << endl;
    cout << "0. Exit Program" << endl;

    cin >> option;
    TreeNode n1;
    TreeNode * new_node = new TreeNode();
```

Compiler Resources Compile Log Debug Find Results

Line: 1 Col: 1 Sel: 0 Lines: 263 Length: 6394 Insert Done parsing in 0,469 seconds

C:\Users\user\Downloads\binary search tree.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug

binary search tree.cpp

```
225         TreeNode * new_node = new TreeNode();
226
227         switch (option) {
228         case 0:
229           break;
230         case 1:
231             cout <<"INSERT"<<endl;
232               cout <<"Masukkan nilai satu persatu: ";
233               cin >> val;
234               new_node->value = val;
235               obj.root= obj.insertRecursive(obj.root,new_node);
236               obj.insertNode(new_node);
237               cout<<endl;
238                 break;
239
240
241         case 2:
242           cout << "PRINT 2D: " << endl;
243           obj.print2D(obj.root, 5);
244           cout << endl;
245           cout << "Print Level Order BFS: \n";
246           obj.printLevelOrderBFS(obj.root);
247           cout << endl;
248           cout <<"PRE-ORDER: ";
249           obj.printPreorder(obj.root);
250           cout<<endl;
251           cout <<"IN-ORDER: ";
252           obj.printInorder(obj.root);
253           cout<<endl;
254           cout <<"POST-ORDER: ";
255           obj.printPostorder(obj.root);
256           break;
257
258         }
259
260       } while (option != 0);
261
262       return 0;
263     }
```

Compiler Resources Compile Log Debug Find Results

Line: 1 Col: 1 Sel: 0 Lines: 263 Length: 6394 Insert Done parsing in 0,469 seconds

- Input : jumlah data dalam tree

  Data data node dalam tree yang sudah di sorting

  Contoh input : 2 5 6 8 9 10 11

- Output (hasil kodingan)
  Print menggunakan BFS

C:\Users\user\Downloads\binary search tree.exe

```
What operation do you want to perform?  Select Option number. Enter 0 to exit.
1. Insert Node
2. Print/Traversal BST values
0. Exit Program
1
INSERT
Masukkan nilai satu persatu: 2
sukses memasukkan nilai
Nilai telah ada,masukkan nilai yang lain

What operation do you want to perform?  Select Option number. Enter 0 to exit.
1. Insert Node
2. Print/Traversal BST values
0. Exit Program
1
INSERT
Masukkan nilai satu persatu: 5
sukses memasukkan nilai
Nilai telah ada,masukkan nilai yang lain

What operation do you want to perform?  Select Option number. Enter 0 to exit.
1. Insert Node
2. Print/Traversal BST values
0. Exit Program
1
INSERT
Masukkan nilai satu persatu: 6
sukses memasukkan nilai
Nilai telah ada,masukkan nilai yang lain

What operation do you want to perform?  Select Option number. Enter 0 to exit.
1. Insert Node
2. Print/Traversal BST values
0. Exit Program
1
INSERT
Masukkan nilai satu persatu: 8
sukses memasukkan nilai
Nilai telah ada,masukkan nilai yang lain

What operation do you want to perform?  Select Option number. Enter 0 to exit.
1. Insert Node
2. Print/Traversal BST values
0. Exit Program
1
INSERT
```

C:\Users\user\Downloads\binary search tree.exe

```
1
INSERT
Masukkan nilai satu persatu: 9
sukses memasukkan nilai
Nilai telah ada,masukkan nilai yang lain

What operation do you want to perform?  Select Option number. Enter 0 to exit.
1. Insert Node
2. Print/Traversal BST values
0. Exit Program
1
INSERT
Masukkan nilai satu persatu: 10
sukses memasukkan nilai
Nilai telah ada,masukkan nilai yang lain

What operation do you want to perform?  Select Option number. Enter 0 to exit.
1. Insert Node
2. Print/Traversal BST values
0. Exit Program
1
INSERT
Masukkan nilai satu persatu: 11
sukses memasukkan nilai
Nilai telah ada,masukkan nilai yang lain

What operation do you want to perform?  Select Option number. Enter 0 to exit.
1. Insert Node
2. Print/Traversal BST values
0. Exit Program
2
PRINT 2D:

                                                            11

                                                    10

                                            9

                                    8

                            6

                    5

            2
```

Type here to search 32°C 12:57 11/05/2022

C:\Users\user\Downloads\binary search tree.exe

```
2. Print/Traversal BST values
0. Exit Program
1
INSERT
Masukkan nilai satu persatu: 11
sukses memasukkan nilai
Nilai telah ada,masukkan nilai yang lain

What operation do you want to perform?  Select Option number. Enter 0 to exit.
1. Insert Node
2. Print/Traversal BST values
0. Exit Program
2
PRINT 2D:

                                                            11

                                                    10

                                            9

                                    8

                            6

                    5

            2

Print Level Order BFS:
2 5 6 8 9 10 11
PRE-ORDER: 2 5 6 8 9 10 11
IN-ORDER: 2 5 6 8 9 10 11
POST-ORDER: 11 10 9 8 6 5 2 What operation do you want to perform?  Select Option number. Enter 0 to exit.
1. Insert Node
2. Print/Traversal BST values
0. Exit Program
```

Type here to search 32°C 12:58 11/05/2022