

Laporan BST

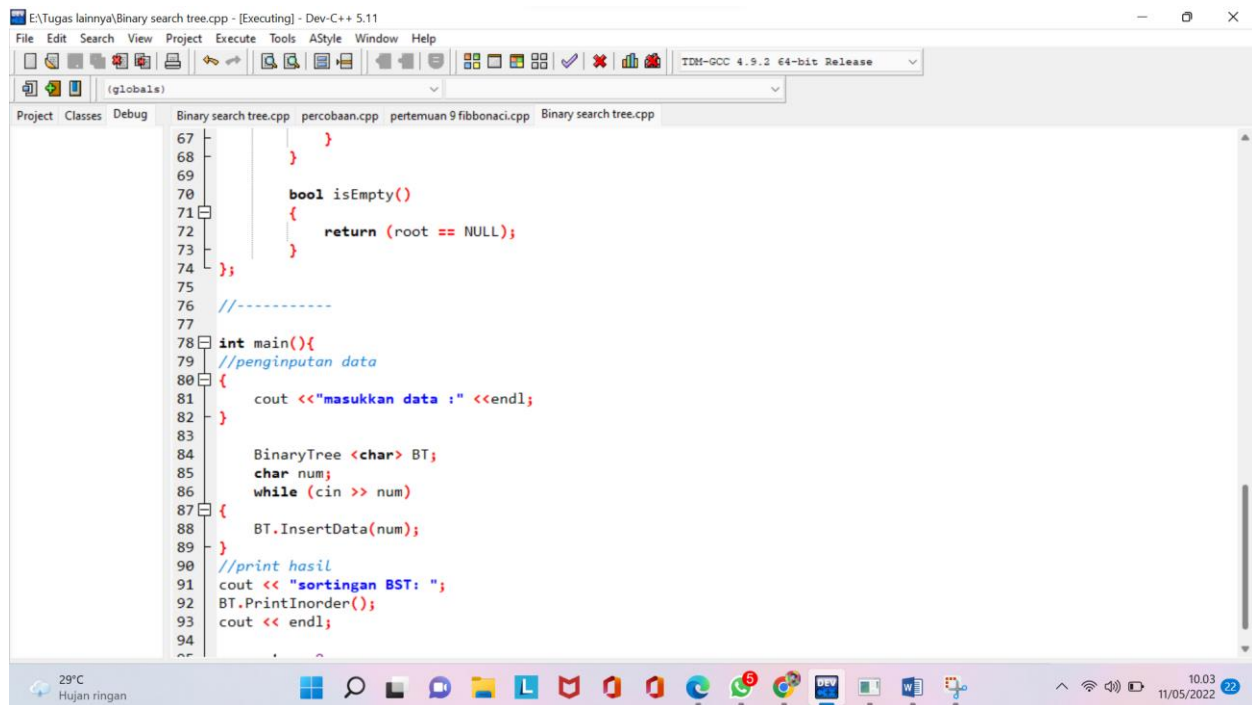
Codingan

```
E:\Tugas lainnya\Binary search tree.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug Binary search tree.cpp percobaan.cpp pertemuan 9 fibonacci.cpp Binary search tree.cpp

1 #include <iostream>
2 using namespace std;
3 //menggunakan tipe data class untuk menyimpan banyak fungsi
4 template <class T>
5 class BinaryTree //fungsi bst
6 {
7     //struct node
8     private:
9         struct TreeNode
10        {
11            TreeNode *left;
12            TreeNode *right;
13            T data;
14        };
15        TreeNode *root;
16
17    public:
18        //---membuat fungsi BST--
19        BinaryTree()
20        {
21            root = NULL;
22        }
23
24        void Inorder(TreeNode *n)
25        {
26            if(n != NULL)
27            {
28                Inorder(n -> left);
29            }
30        }
31    };
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
E:\Tugas lainnya\Binary search tree.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug Binary search tree.cpp percobaan.cpp pertemuan 9 fibonacci.cpp Binary search tree.cpp

28 Inorder(n -> left);
29 cout<< n -> data;
30 Inorder(n -> right);
31 }
32 }
33
34 void PrintInorder()
35 {
36     Inorder(root);
37 }
38
39 void InsertData(T data)
40 {
41     TreeNode *t = new TreeNode;
42     TreeNode *parent;
43     t -> data = data;
44     t -> left = NULL;
45     t -> right = NULL;
46     parent = NULL;
47
48     if (isEmpty())
49     {
50         root = t;
51     }
52     else
53     {
54         TreeNode *curr;
55         curr = root;
56         while(curr)
57         {
58             if (curr -> left == NULL)
59             {
60                 curr -> left = t;
61                 break;
62             }
63             else if (curr -> right == NULL)
64             {
65                 curr -> right = t;
66                 break;
67             }
68             else
69             {
70                 curr = curr -> left;
71             }
72         }
73     }
74 }
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```



```
67     }
68 }
69
70 bool isEmpty()
71 {
72     return (root == NULL);
73 }
74 };
75
76 //-----
77
78 int main(){
79     //penginputan data
80     {
81         cout << "masukkan data : " << endl;
82     }
83
84     BinaryTree <char> BT;
85     char num;
86     while (cin >> num)
87     {
88         BT.InsertData(num);
89     }
90     //print hasil
91     cout << "sortingan BST: ";
92     BT.PrintInorder();
93     cout << endl;
94 }
```

Output

E:\Tugas lainnya\Binary search tree.exe

```
masukkan data :
4 6 2 8 9
^Z
sortingan BST: 24689

-----
Process exited after 13.73 seconds with return value 0
Press any key to continue . . .
```