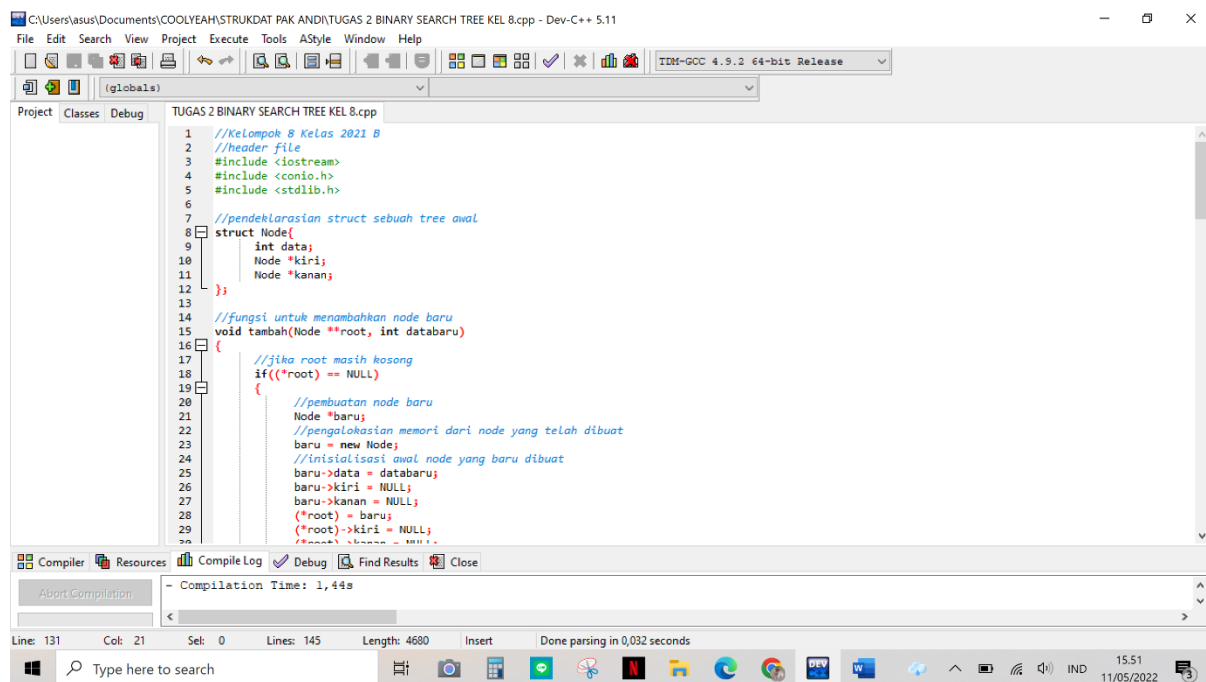


NAMA ANGGOTA :

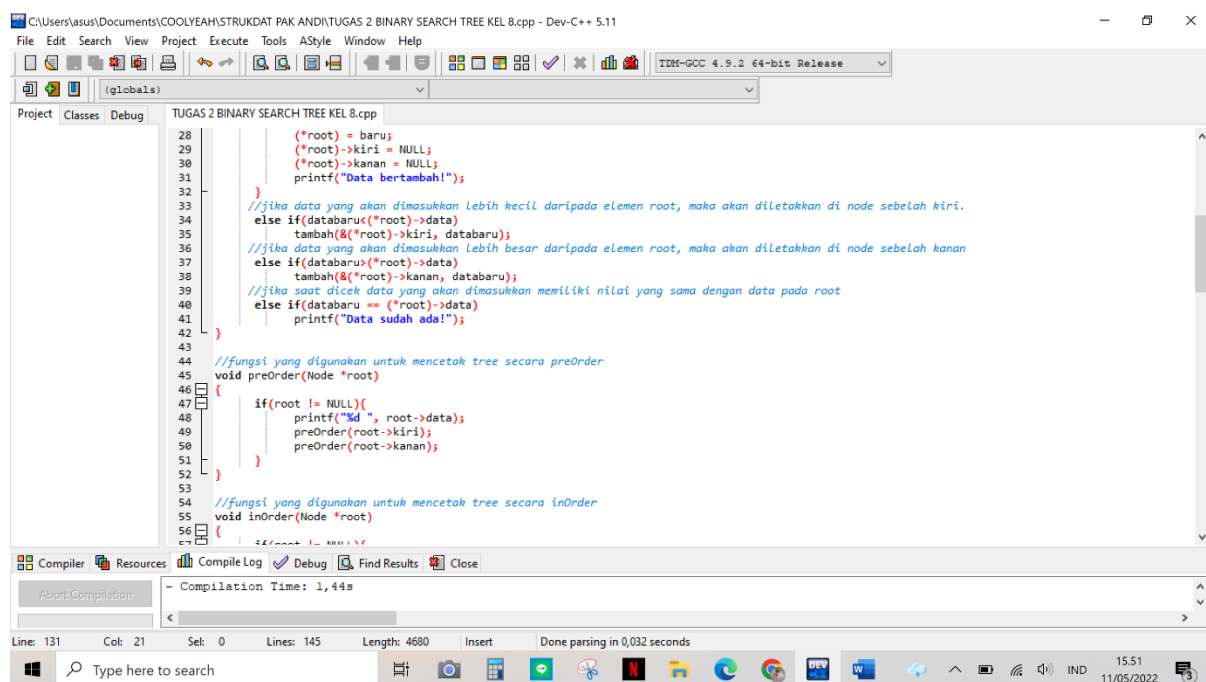
1. AISYAH AQILAH RV (21091397002)
2. RAHMADITYA PLI (21091397036)
3. ZABRINA RULIF A (21091397056)
4. AL ZUCHRI MQ (21091397062)
5. RIFQI YUDO D (21091397074)

## LAPORAN KELOMPOK 8 KELAS 2021 B "BINARY SEARCH TREE"

Codingan :



```
1 //Kelompok 8 Kelas 2021 B
2 //header file
3 #include <iostream>
4 #include <conio.h>
5 #include <stdlib.h>
6
7 //pendeclarasian struct sebuah tree awal
8 struct Node{
9     int data;
10    Node *kiri;
11    Node *kanan;
12 };
13
14 //fungsi untuk menambahkan node baru
15 void tambah(Node **root, int databaru)
16 {
17     //jika root masih kosong
18     if((*root) == NULL)
19     {
20         //pembuatan node baru
21         Node *baru;
22         //pengalokasian memori dari node yang telah dibuat
23         baru = new Node;
24         //inisialisasi awal node yang baru dibuat
25         baru->data = databaru;
26         baru->kiri = NULL;
27         baru->kanan = NULL;
28         (*root) = baru;
29         (*root)->kiri = NULL;
30         (*root)->kanan = NULL;
```



```
28         (*root) = baru;
29         (*root)->kiri = NULL;
30         (*root)->kanan = NULL;
31         printf("Data bertambah!");
32     }
33     //jika data yang akan dimasukkan lebih kecil daripada elemen root, maka akan diletakkan di node sebelah kiri.
34     else if(databaru < (*root)->data)
35     {
36         tambah(&(*root)->kiri, databaru);
37     }
38     //jika data yang akan dimasukkan lebih besar daripada elemen root, maka akan diletakkan di node sebelah kanan
39     else if(databaru > (*root)->data)
40     {
41         tambah(&(*root)->kanan, databaru);
42     }
43     //jika saat dicek data yang akan dimasukkan memiliki nilai yang sama dengan data pada root
44     else if(databaru == (*root)->data)
45     {
46         printf("Data sudah ada!");
47     }
48 }
49
50 //fungsi yang digunakan untuk mencetak tree secara preOrder
51 void preOrder(Node *root)
52 {
53     if(root != NULL){
54         printf("%d ", root->data);
55         preOrder(root->kiri);
56         preOrder(root->kanan);
57     }
58 }
59
60 //fungsi yang digunakan untuk mencetak tree secara inOrder
61 void inOrder(Node *root)
62 {
63     if(root != NULL){
64         inOrder(root->kiri);
65         printf("%d ", root->data);
66         inOrder(root->kanan);
67     }
68 }
```



C:\Users\asus\Documents\COOLYEAH\STRUKDAT PAK ANDI\TUGAS 2 BINARY SEARCH TREE KEL 8.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

(globals)

Project Classes Debug TUGAS 2 BINARY SEARCH TREE KEL 8.cpp

```
97 {
98     //jika pil bernilai 1
99     case 1 :
100         printf("\nINPUT : ");
101         printf("\n-----\n");
102         printf("\ndata baru : ");
103         scanf("%d", &data);
104         //panggil fungsi untuk menambah node yang berisi data pada tree
105         tambah(&pohon, data);
106         break;
107
108     //jika pil bernilai 2
109     case 2 :
110         printf("\nOUTPUT PRE ORDER : ");
111         printf("\n-----\n");
112         if(pohon!=NULL)
113             //panggil fungsi untuk mencetak data secara preOrder
114             preOrder(pohon);
115         else
116             printf("Masih kosong!");
117         break;
118
119     //jika pil bernilai 3
120     case 3 :
121         printf("\nOUTPUT IN ORDER : ");
122         printf("\n-----\n");
123         if(pohon!=NULL)
124             //panggil fungsi untuk mencetak data secara inOrder
125             inOrder(pohon);
126 }
```

Compiler Resources Compile Log Debug Find Results Close

About Compilation

Compilation Time: 1,44s

Line: 131 Col: 21 Sel: 0 Lines: 145 Length: 4680 Insert Done parsing in 0,032 seconds

Type here to search

15:52 11/05/2022

C:\Users\asus\Documents\COOLYEAH\STRUKDAT PAK ANDI\TUGAS 2 BINARY SEARCH TREE KEL 8.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

(globals)

Project Classes Debug TUGAS 2 BINARY SEARCH TREE KEL 8.cpp

```
117         break;
118
119     //jika pil bernilai 3
120     case 3 :
121         printf("\nOUTPUT IN ORDER : ");
122         printf("\n-----\n");
123         if(pohon!=NULL)
124             //panggil fungsi untuk mencetak data secara inOrder
125             inOrder(pohon);
126         else
127             printf("Masih kosong!");
128         break;
129
130     //jika pil bernilai 4
131     case 4 :
132         printf("\nOUTPUT POST ORDER : ");
133         printf("\n-----\n");
134         if(pohon!=NULL)
135             //panggil fungsi untuk mencetak data secara postOrder
136             postOrder(pohon);
137         else
138             printf("Masih kosong!");
139         break;
140     }
141     getch();
142     while(pil != 5); //akan diulang jika input tidak samadengan 5
143     return EXIT_FAILURE;
144 }
145 }
```

Compiler Resources Compile Log Debug Find Results Close

About Compilation


Compilation Time: 1,44s

Line: 131 Col: 21 Sel: 0 Lines: 145 Length: 4680 Insert Done parsing in 0,032 seconds

Type here to search

15:52 11/05/2022


Input:

 C:\Users\asus\Documents\COOLYEAH\ST

```
#PROGRAM TREE C++#
=====

MENU
----
1. Tambah
2. Lihat pre-order
3. Lihat in-order
4. Lihat post-order
5. Exit
Pilihan : 1


INPUT :
-----
Data baru : 7
Data bertambah!
```

 C:\Users\asus\Documents\COOLYEAH\STR

```
#PROGRAM TREE C++#
=====

MENU
----
1. Tambah
2. Lihat pre-order
3. Lihat in-order
4. Lihat post-order
5. Exit
Pilihan : 1


INPUT :
-----
Data baru : 9
Data bertambah!
```

 C:\Users\asus\Documents\COOLYEAH\ST

```
#PROGRAM TREE C++#
=====

MENU
----
1. Tambah
2. Lihat pre-order
3. Lihat in-order
4. Lihat post-order
5. Exit
Pilihan : 1

INPUT :
-----
Data baru : 5
Data bertambah!
```

 C:\Users\asus\Documents\COOLYEAH\STI

```
#PROGRAM TREE C++#
=====

MENU
----
1. Tambah
2. Lihat pre-order
3. Lihat in-order
4. Lihat post-order
5. Exit
Pilihan : 1

INPUT :
-----
Data baru : 2
Data bertambah!
```

C:\Users\asus\Documents\COOLYEAH\

```
#PROGRAM TREE C++#
=====

MENU
----
1. Tambah
2. Lihat pre-order
3. Lihat in-order
4. Lihat post-order
5. Exit
Pilihan : 1

INPUT :
-----
Data baru : 6
Data bertambah!
```

Hasil Output Menu no 2 :


C:\Users\asus\Documents\COOLYEAH\STRUKDAT PAK ANDI\TU

```
#PROGRAM TREE C++#
=====

MENU
----
1. Tambah
2. Lihat pre-order
3. Lihat in-order
4. Lihat post-order
5. Exit
Pilihan : 2

OUTPUT PRE ORDER :
-----
7 5 2 6 9 _
```

Hasil Output Menu no 3 :


 C:\Users\asus\Documents\COOLYEAH\STRUKDAT PAK AN

```
#PROGRAM TREE C++#
=====

MENU
----
1. Tambah
2. Lihat pre-order
3. Lihat in-order
4. Lihat post-order
5. Exit
Pilihan : 3

OUTPUT IN ORDER :
-----
2 5 6 7 9 _
```

Hasil Output Menu no 4 :


 C:\Users\asus\Documents\COOLYEAH\STRUKDAT PAK ANDI\T

```
#PROGRAM TREE C++#
=====

MENU
----
1. Tambah
2. Lihat pre-order
3. Lihat in-order
4. Lihat post-order
5. Exit
Pilihan : 4

OUTPUT POST ORDER :
-----
2 6 5 9 7 _
```

Hasil Output Menu no 5:

 C:\Users\asus\Documents\COOLYEAH\STRUKDAT PAK ANDI\TU

```
#PROGRAM TREE C++#  
=====
```

MENU

----

1. Tambah
2. Lihat pre-order
3. Lihat in-order
4. Lihat post-order
5. Exit

Pilihan : 5

