Nama : Nabiilah Nur Fauziyyah

NPM : 2310631170105

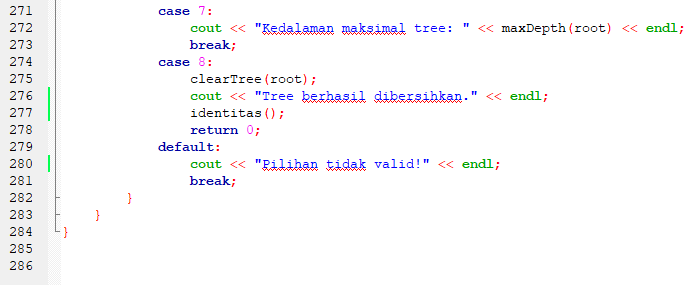
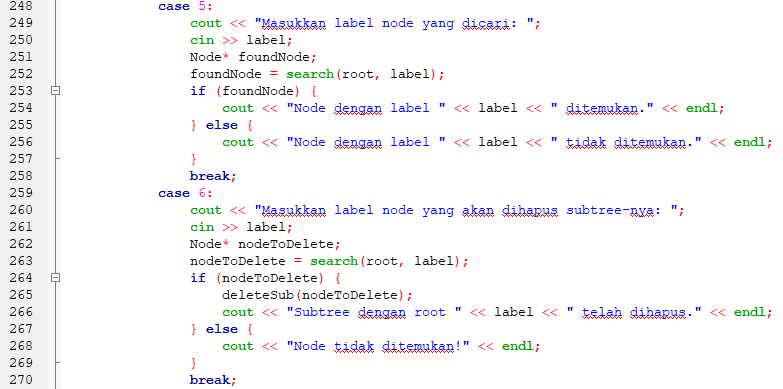
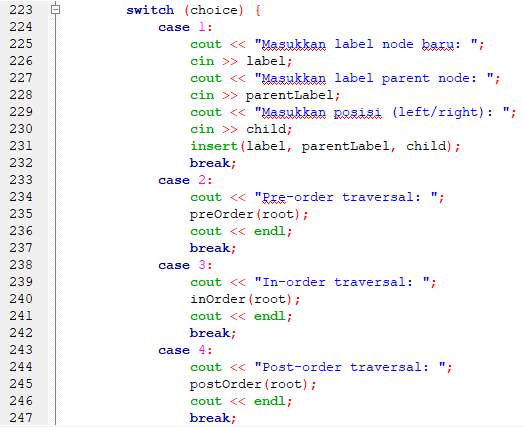
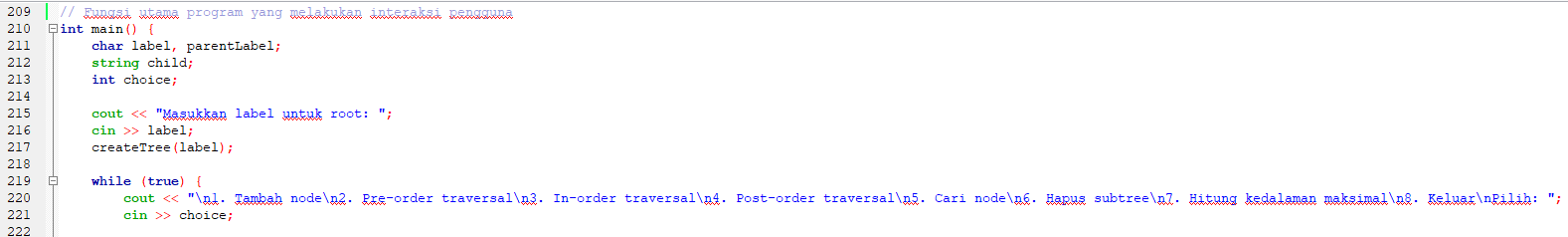
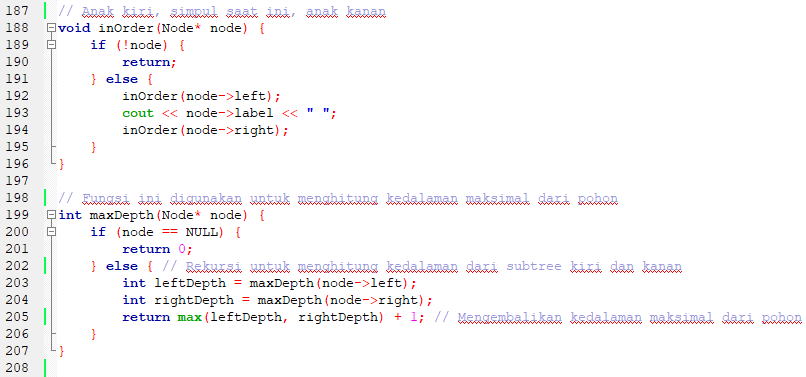
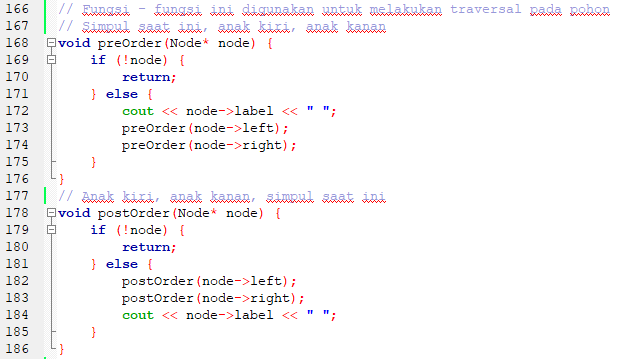
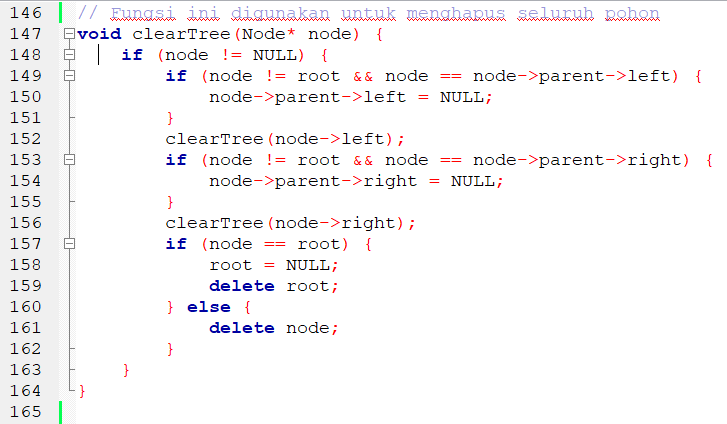
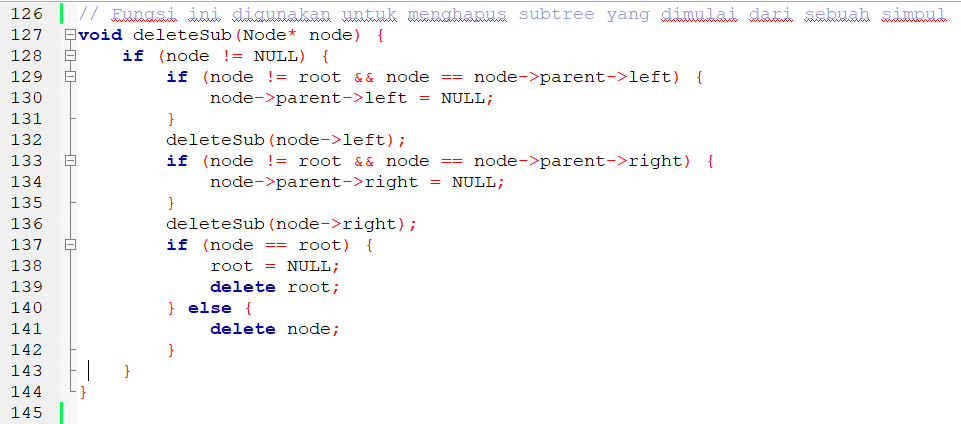
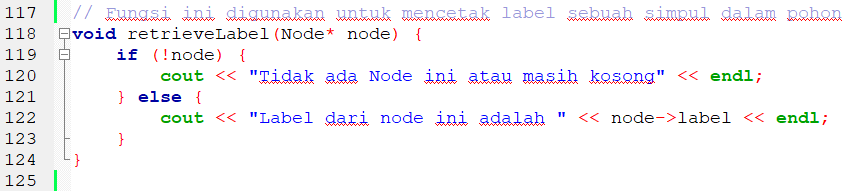
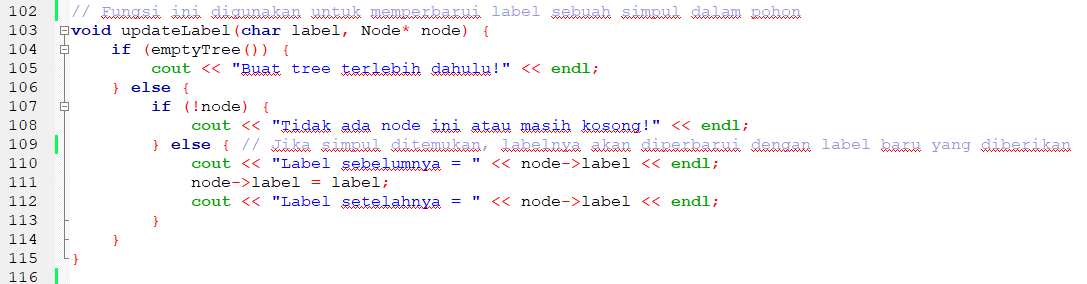
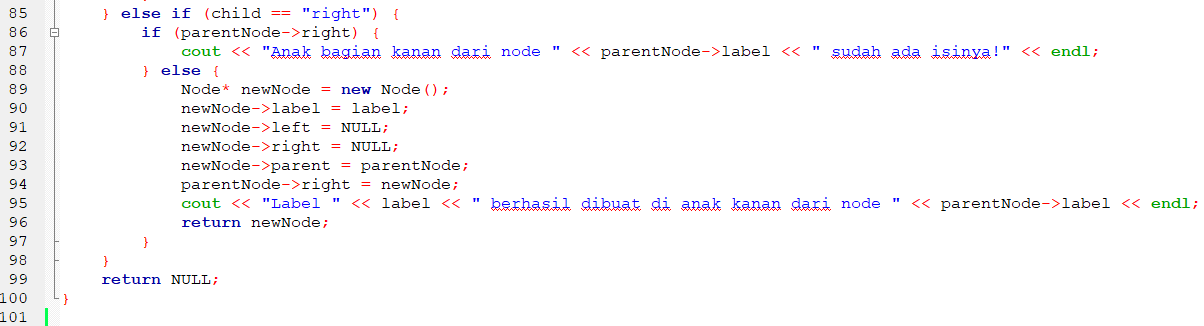
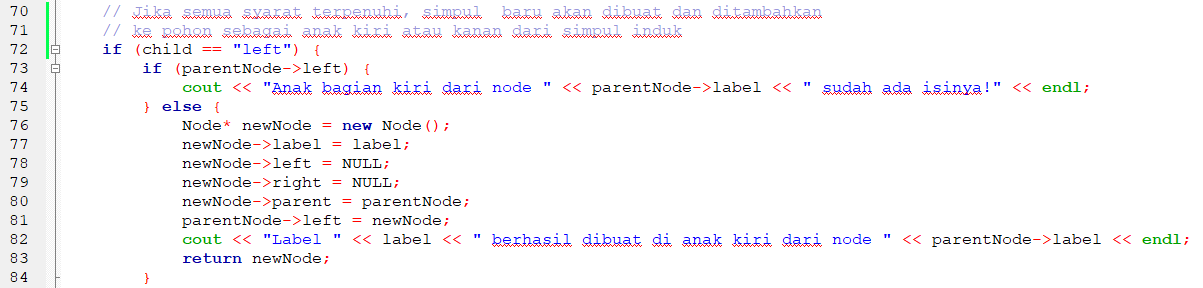
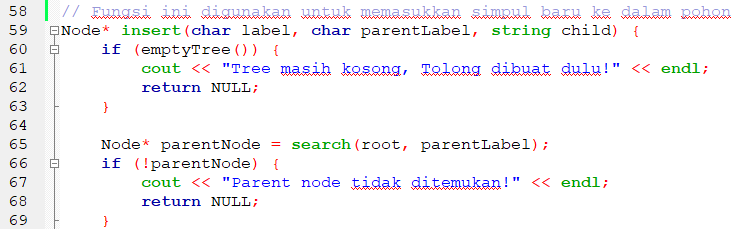
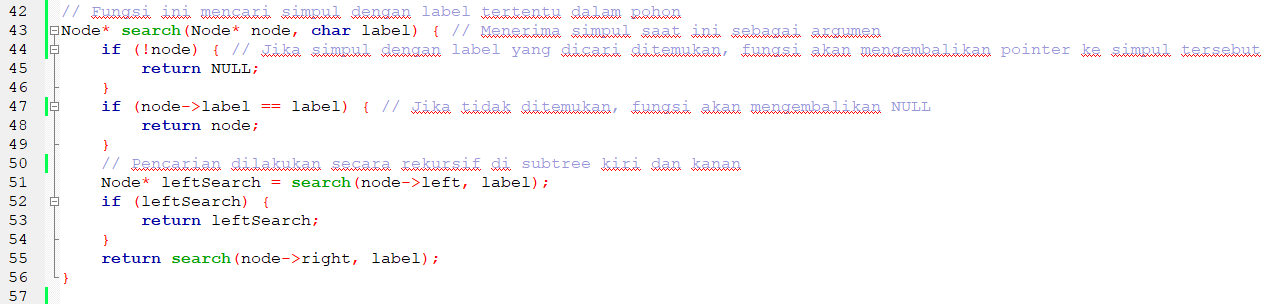
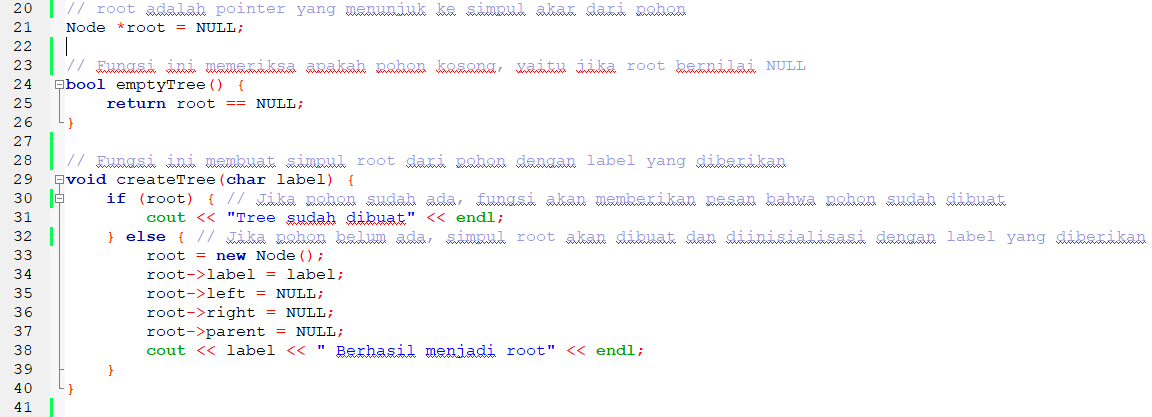
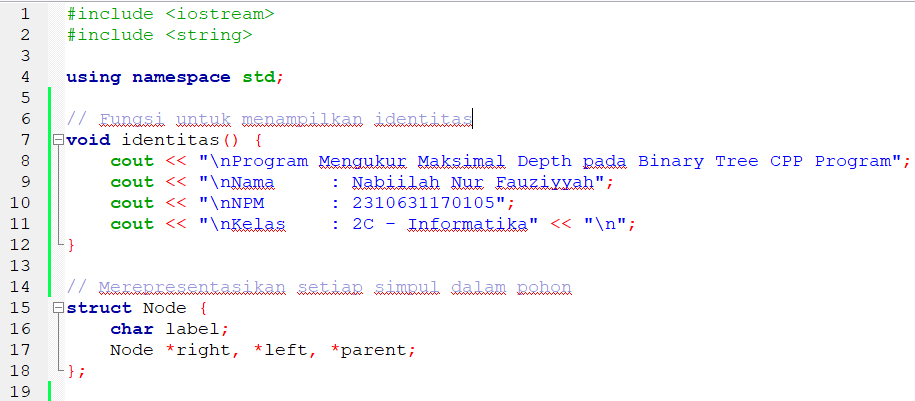
Kelas : 2C – Informatika

Tugas Struktur Data

Link Github : https://github.com/fauziyyah22/NabiilahNF\_SD\_Tugas7

1. Buatlah laporan dari source code pada link di bawah ini :

[Mengukur Maximal Depth Pada Binary Tree CPP Program](https://github.com/athyathanasia01/Data-Structure-Assistant-2024/blob/main/7/exercise%207%20-%20tree/exercise7_maxDepth3.cpp)



Penjelasan :

Program ini merupakan implementasi dari struktur data Binary Tree dalam bahasa C++. Program ini memungkinkan pengguna untuk membuat, mengedit, dan menjelajahi pohon biner.

1. Pengguna diminta untuk memasukkan label untuk root tree.
2. Fungsi createTree() kemudian dipanggil untuk membuat simpul root dengan label yang dimasukkan.
3. Program masuk ke dalam loop while (true) yang berjalan tak terbatas, sehingga terus menampilkan menu interaktif kepada pengguna.
4. Pengguna diberikan pilihan untuk melakukan berbagai operasi pada pohon biner, seperti menambahkan node baru, melakukan traversal, mencari node, menghapus subtree, menghitung kedalaman maksimal, dan keluar dari program.
5. Program membaca pilihan yang dimasukkan oleh pengguna menggunakan pernyataan switch.
6. Bergantung pada pilihan yang dipilih, program memanggil fungsi-fungsi yang sesuai untuk menjalankan operasi yang diminta.
7. Terdapat beberapa operasi yang dapat dilakukan:
8. Menambahkan node baru ke pohon dengan memanggil fungsi insert().
9. Melakukan traversal pada pohon menggunakan pre-order, in-order, dan post-order traversal.
10. Mencari node dalam pohon dengan menggunakan fungsi search().
11. Menghapus subtree dari pohon menggunakan fungsi deleteSub().
12. Menghitung kedalaman maksimal pohon dengan memanggil fungsi maxDepth().
13. Keluar dari program dengan membersihkan pohon dan menampilkan identitas pembuat program.
14. Setiap operasi yang dilakukan akan memberikan output yang sesuai kepada pengguna, seperti pesan sukses atau pesan kesalahan.
15. Pengguna dapat terus berinteraksi dengan program sampai memilih untuk keluar.
16. Setelah pengguna memilih untuk keluar dari program, program akan membersihkan pohon dan menampilkan identitas pembuat program sebelum akhirnya berakhir.

Demikianlah alur kerja dari program ini, di mana pengguna dapat berinteraksi dengan pohon biner dan melakukan berbagai operasi yang relevan sesuai dengan kebutuhan mereka.

Hasil dari program tersebut sebagai berikut :

