**LAPORAN PRAKTIKUM**

**“Pertemuan ke-9:Post Test - Pohon”**

Diajukan untuk memenuhi salah satu praktikum Mata Struktur Data Informatika yang di ampu oleh:

 Dr., Ardiansyah, S.T., M.Cs.

Disusun Oleh:

Mohammad Farid Hendianto 2200018401

A /Selasa 10.30 – 13.30 Lab. Komputasi Dasar

**PROGRAM STUDI INFORMATIKA**

**UNIVERSITAS AHMAD DAHLAN**

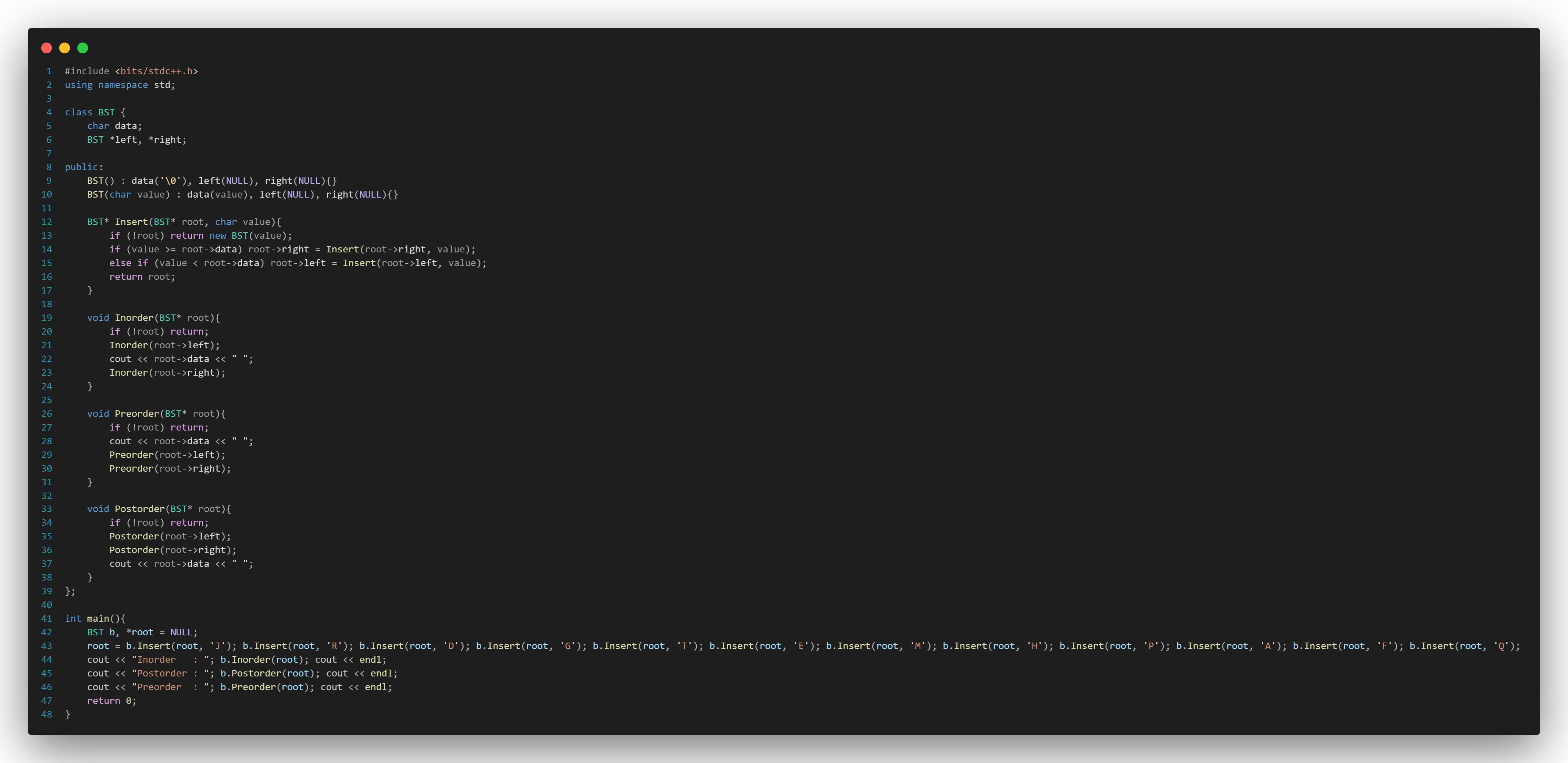
**FAKULTAS TEKNOLOGI INDUSTRI**

**TAHUN 2023**

Dari hasil pretes di atas, buatlah main function (lihat contoh main function di atas). Cocokkan

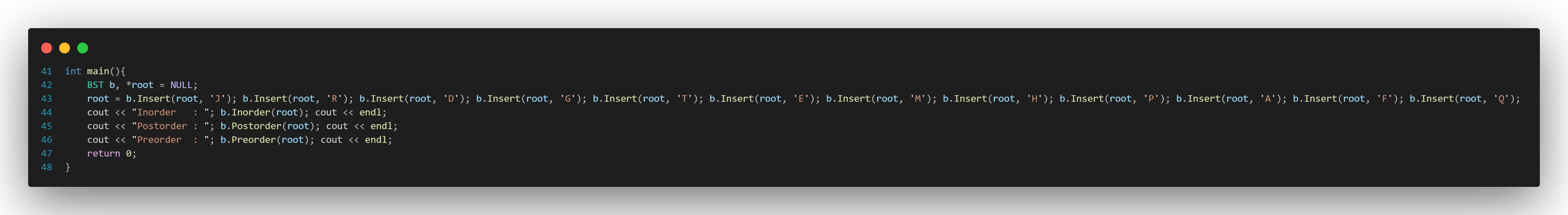
hasil penelusuran Inorder, Postorder, dan Preordernya dari pretes anda.

Source Kode:

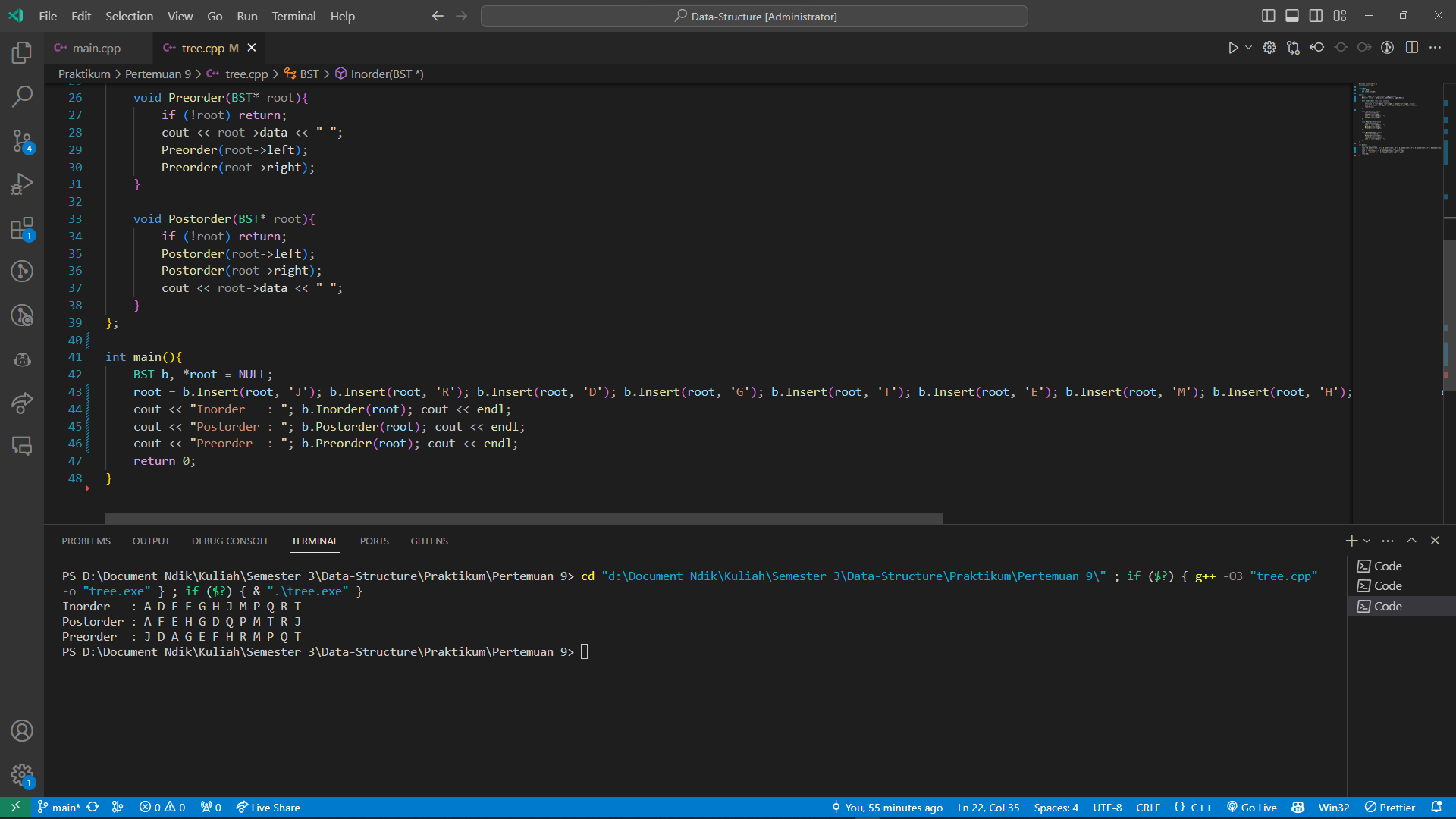


Source code

Fungsi main dan Program jalan (Screenshot input output) (Skor 40)



Int main



Output program jalan

Fungsi dan output inorder (Skor 15)



Fungsi inorder

OUTPUT

Inorder : A D E F G H J M P Q R T

Fungsi dan output postorder (Skor 15)

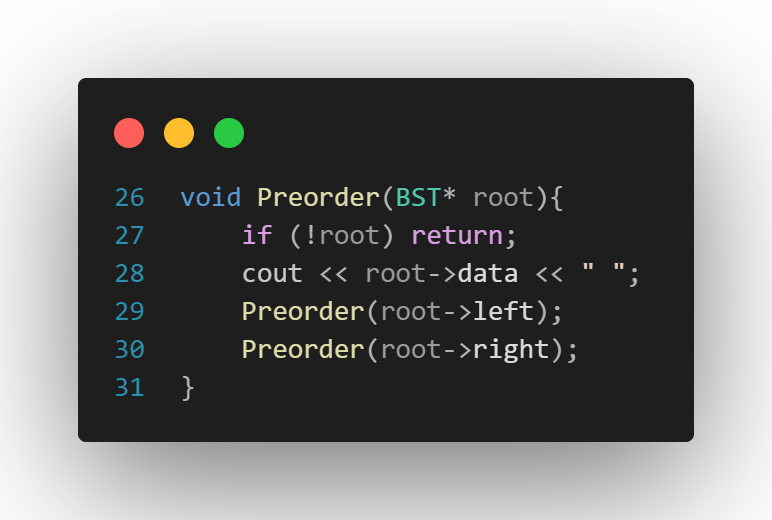


Fungsi postorder

OUTPUT

Postorder : A F E H G D Q P M T R J

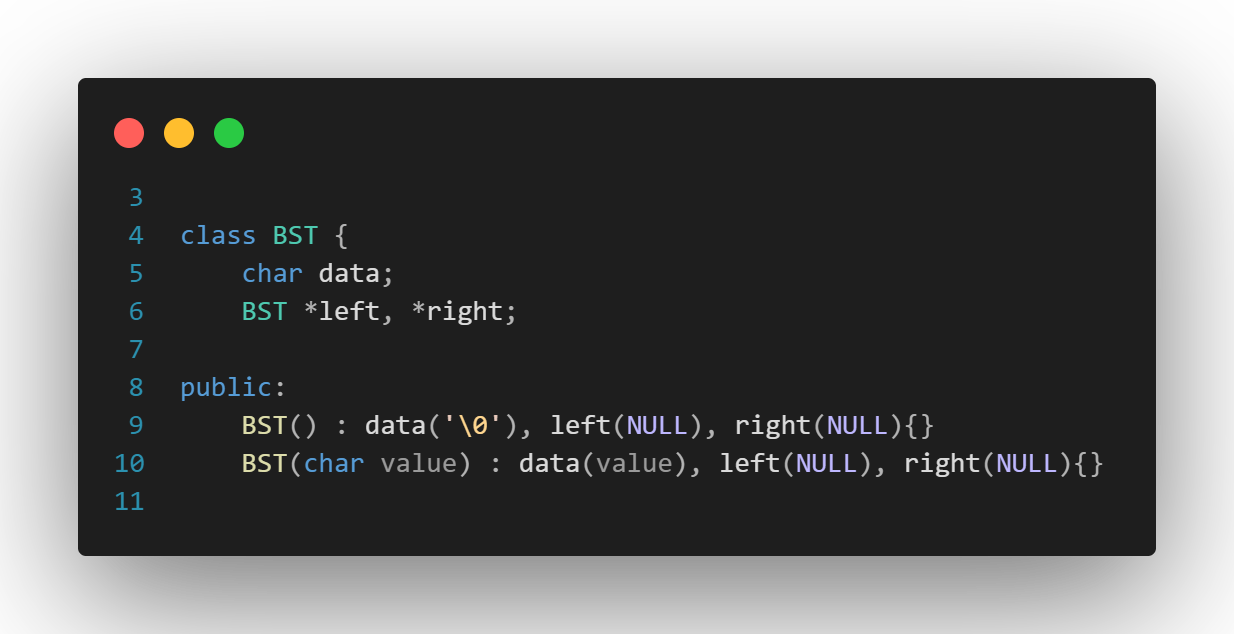
Fungsi dan output preorder (Skor 15)



OUTPUT

Preorder : J D A G E F H R M P Q T

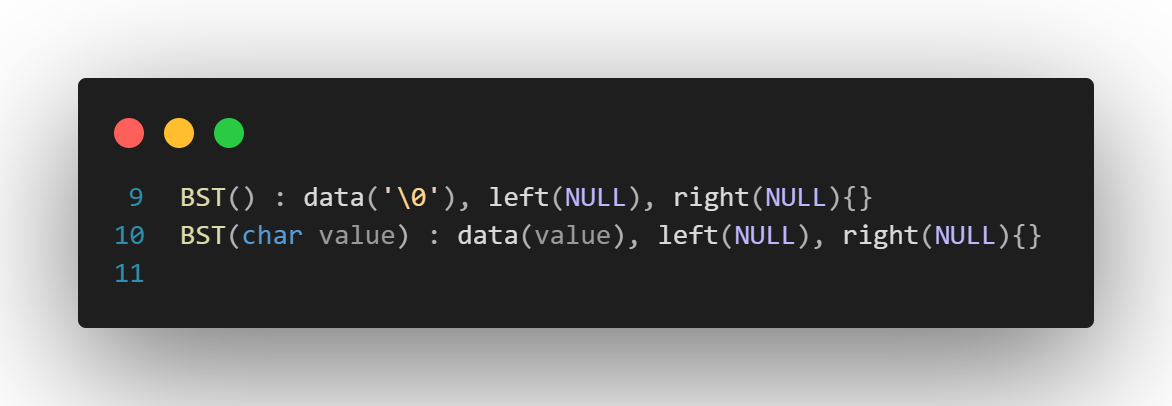
Class



Penggunaan class BST (Binary Search Tree)

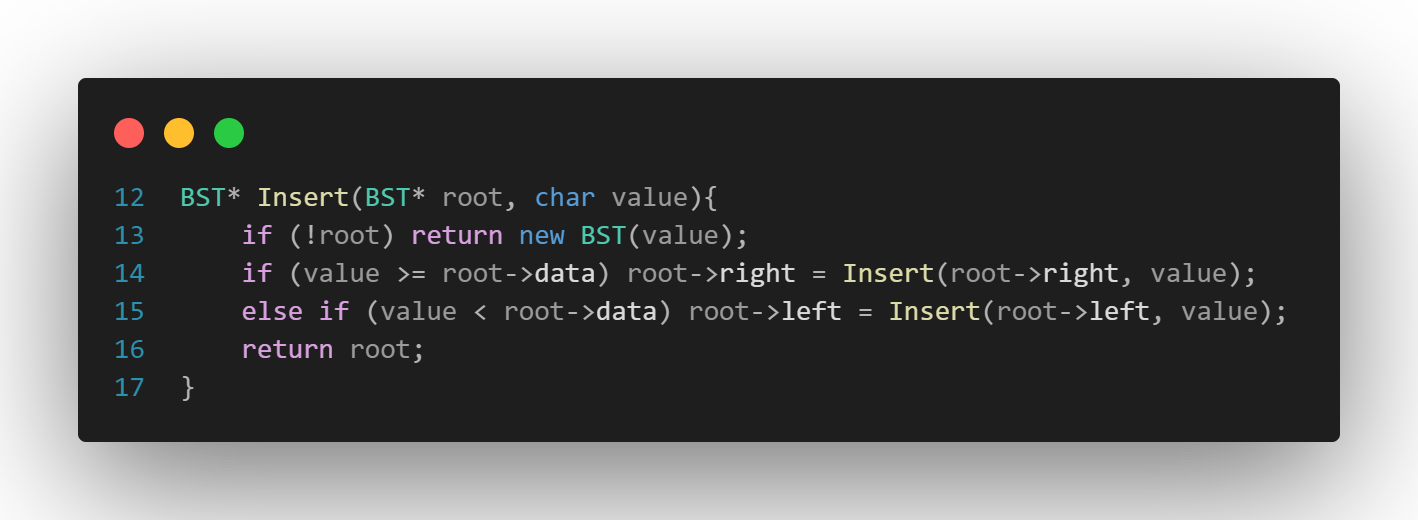
Penjelasan program:

Kode ini mendefinisikan struktur data pohon biner pencarian (BST/ binary search tree) dan beberapa operasi yang dapat dilakukan pada BST.



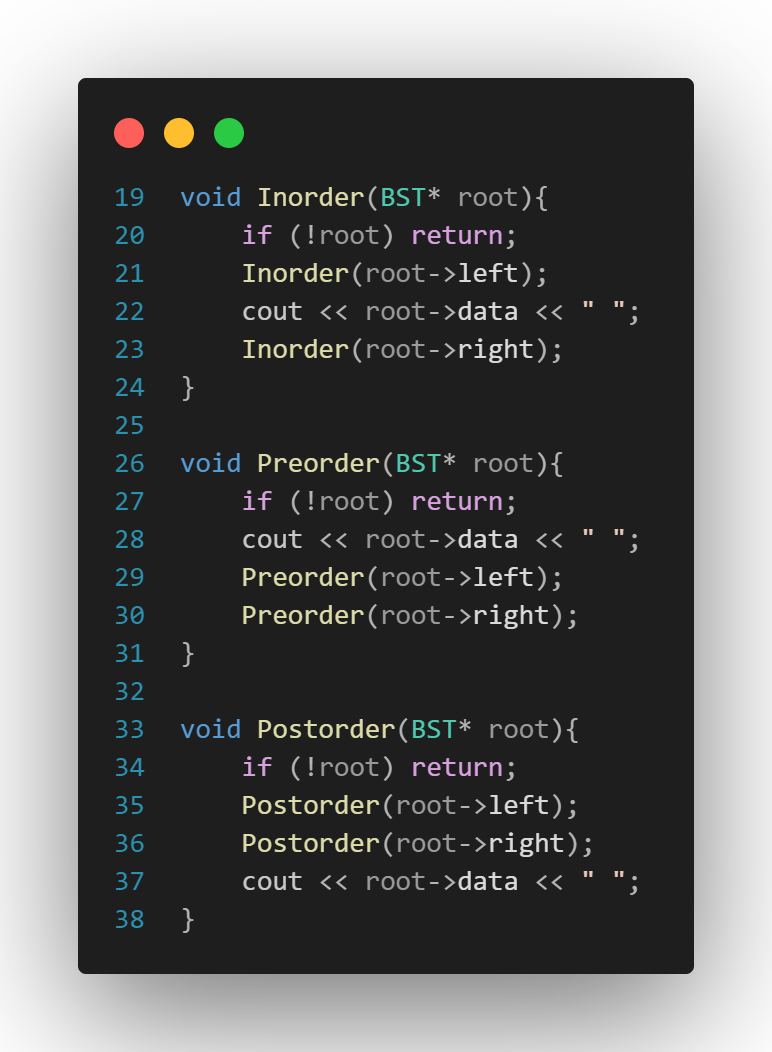
constructor

`BST() : data('\0'), left(NULL), right(NULL){}` dan `BST(char value) : data(value), left(NULL), right(NULL){}` adalah konstruktor untuk kelas BST. Konstruktor pertama membuat BST kosong, sedangkan konstruktor kedua membuat BST dengan nilai awal.



Fungsi insert

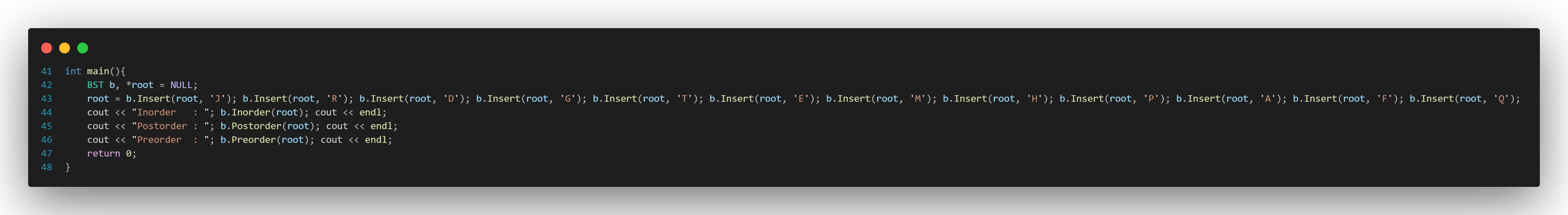
`BST\* Insert(BST\* root, char value)` adalah fungsi untuk memasukkan elemen baru ke dalam BST. Jika BST kosong, fungsi ini akan membuat BST baru. Jika tidak, fungsi ini akan memasukkan elemen baru ke sub-pohon kiri jika elemen tersebut lebih kecil dari root, atau ke sub-pohon kanan jika elemen tersebut lebih besar atau sama dengan root.



Fungsi Inorder,Preorder, dan Postorder

`void Inorder(BST\* root)`, `void Preorder(BST\* root)`, dan `void Postorder(BST\* root)` adalah fungsi untuk melakukan penjelajahan pohon (tree traversal) dalam urutan inorder, preorder, dan postorder.

* Dalam penjelajahan inorder, fungsi ini akan mengunjungi sub-pohon kiri, root, dan sub-pohon kanan.
* Dalam penjelajahan preorder, fungsi ini akan mengunjungi root, sub-pohon kiri, dan sub-pohon kanan.
* Dalam penjelajahan postorder, fungsi ini akan mengunjungi sub-pohon kiri, sub-pohon kanan, dan root.



Fungsi utama

`int main()` adalah fungsi utama yang membuat BST dan melakukan beberapa operasi pada BST tersebut, termasuk memasukkan beberapa elemen dan melakukan penjelajahan pohon.