**TUGAS PRAKTIKUM**

**STRUKTUR DATA & ALGORITMA**

**“Tree”**



**DISUSUN OLEH**

**Kristian Budi Pradana Putra (M0520043)**

**Kelas B**

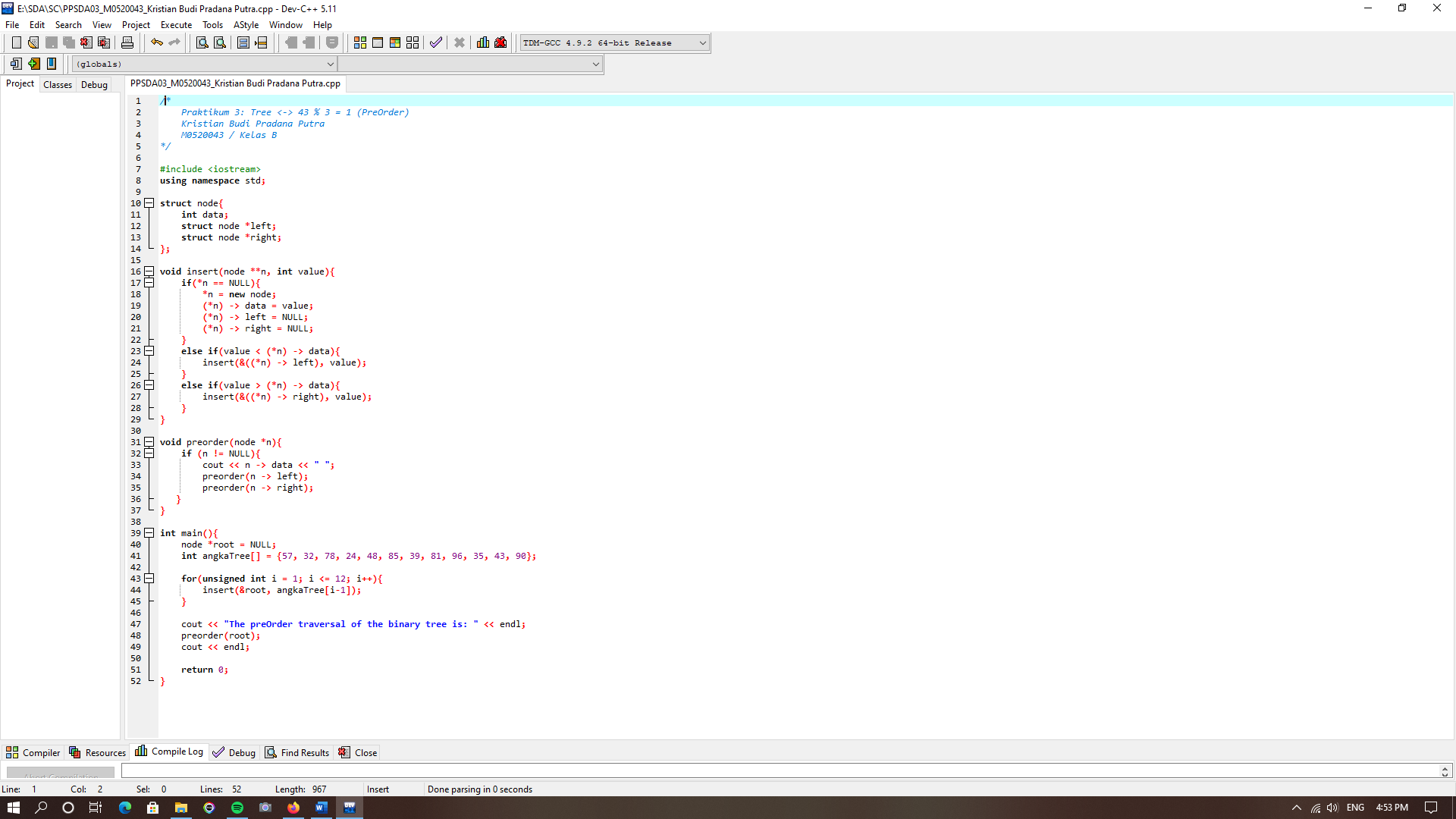
**PROGRAM STUDI INFORMATIKA**

**FAKULTAS MATEMATIKA & ILMU PENGETAHUAN ALAM**

**UNIVERSITAS SEBELAS MARET**

**2021**

**Membuat Program *Binary Tree* dengan *Tree Traversal PreOrder***



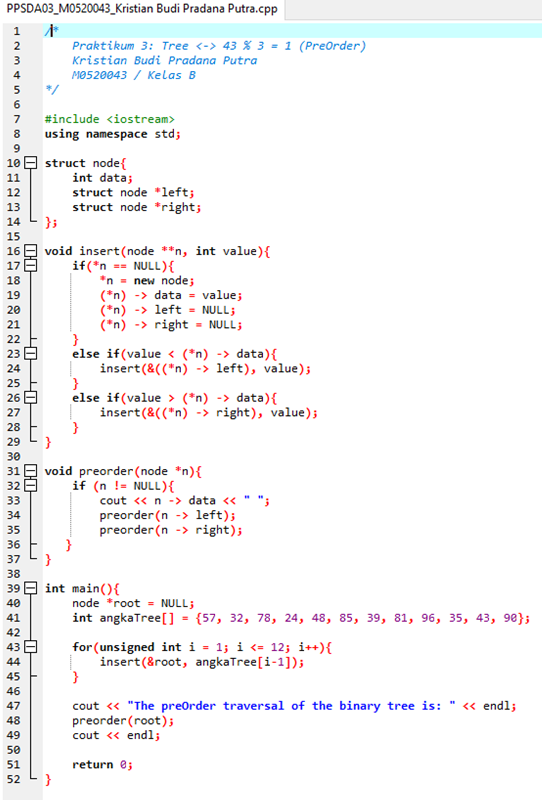
1. **Penjelasan Singkat**

Tree atau pohon merupakan jenis struktur data yang mengandung *node-node* mulai dari *root node* (akar) hingga *leaf node* (daun). Pada praktikum 3 kali ini, saya diminta membuat program *binary tree* dengan jenis *traversal preorder*. Program *binary tree* sendiri merupakan sebuah struktur data *tree* yang tiap-tiap *node*-nya memiliki *child* maksimum 2. Ciri-ciri *binary tree* diantaranya:

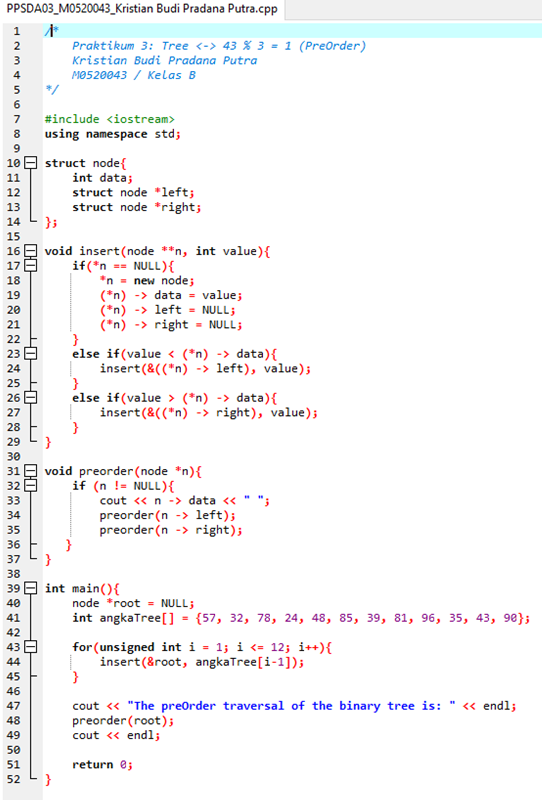
1. Nilai pada *child* bagian kiri lebih kecil dari nilai pada *child* kanan,
2. Nilai pada *child* bagian kanan lebih besar dari nilai pada *child* kiri,
3. Memfasilitasi penghapusan duplikat jika ada nilai yang sama, dan
4. Pencarian yang cepat (untuk *balanced tree*, maksimum perbandingan log n).

Untuk jenis *traversal preorder*, dalam mengurutkan tiap *node*-nya dimulai dari *root* (akar) → *left* (kiri) → *right* (kanan).

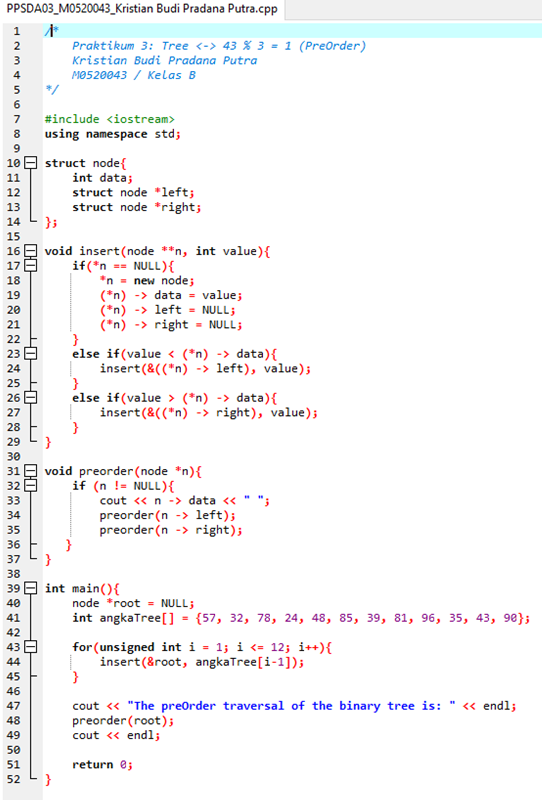
1. **Analisa Kode Sumber**



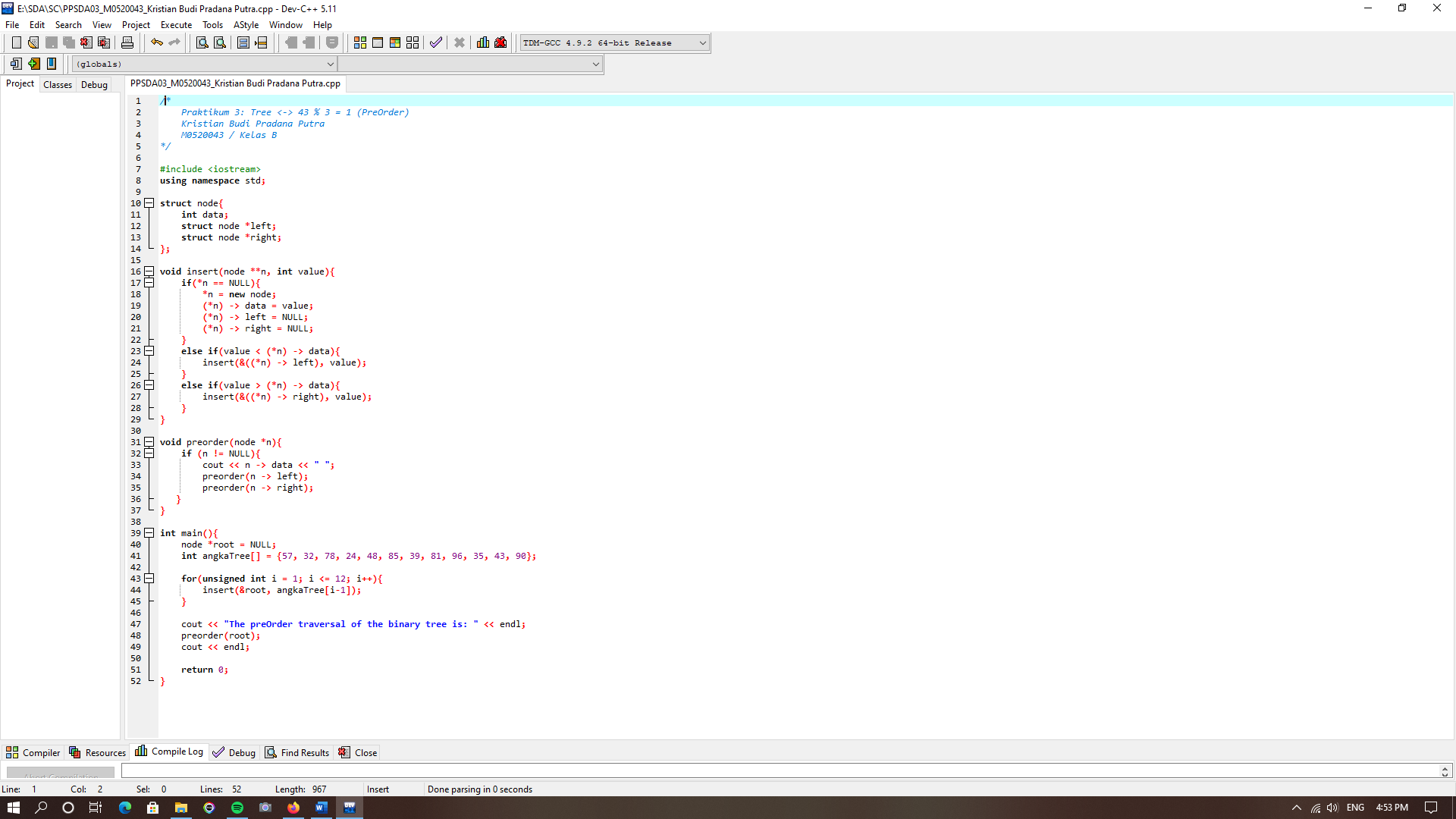
Baris 7 dan 8 merupakan bagian *header* dalam *source code*. Bagian inilah yang akan mengimpor fungsi-fungsi yang sudah didefinisikan pada *header* file, serta mengatur *compiler* agar dapat membaca *library* C++ sehingga nantinya dapat menjalankan *source code* dengan lancar.

****

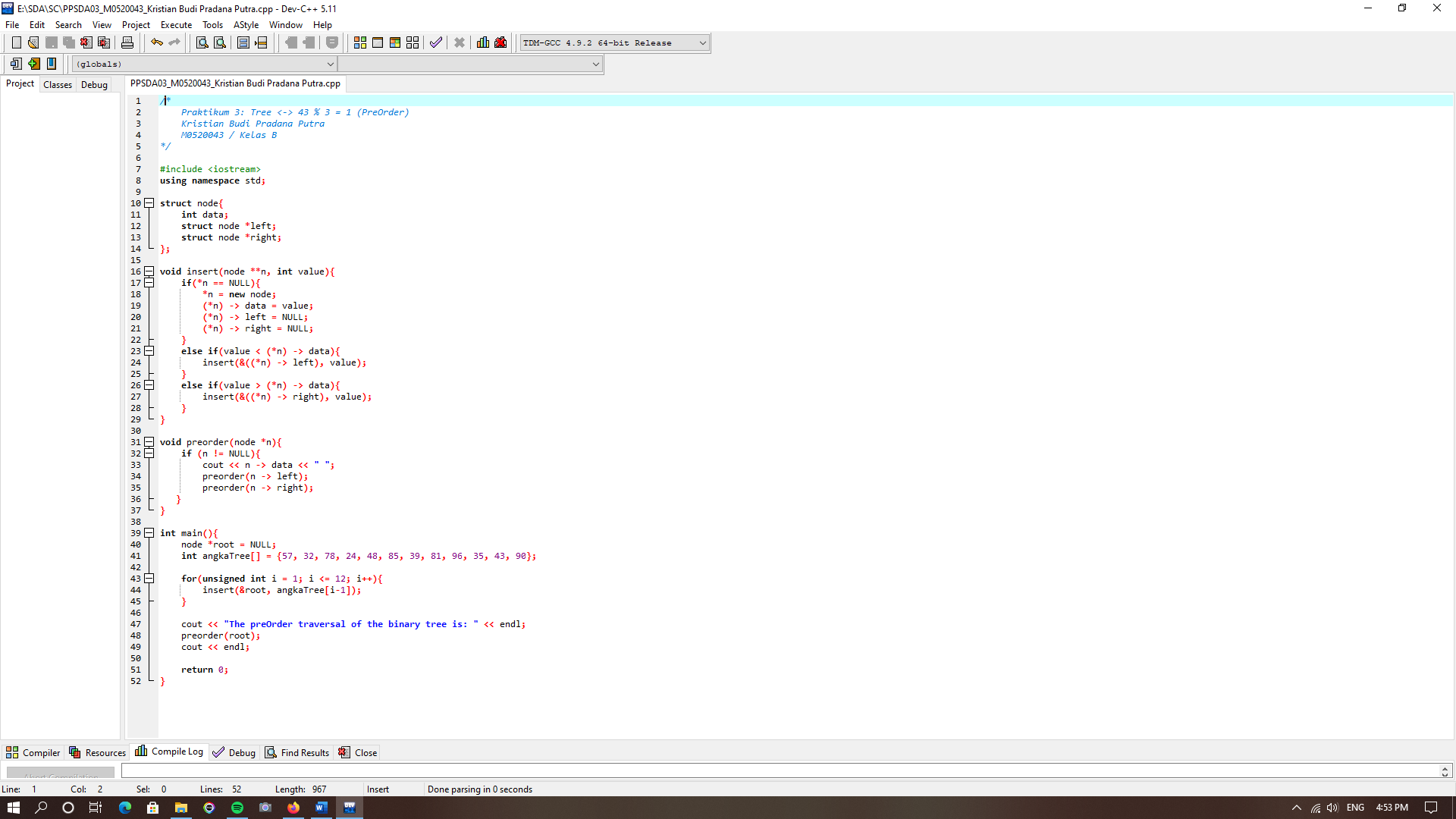
Baris 10-14 merupakan bagian ‘*self referential structure*’ yang di dalamnya terdiri atas pointer-pointer dari ‘struct *node*’. Baris 11 merupakan tipe data dan variabel yang nantinya akan dipakai untuk mewakili *value* dari *node*. Kemudian, terdapat baris 12 yang merupakan *pointer* untuk *child* bagian kiri dari *tree*, sedangkan baris 13 merupakan *pointer* untuk *child* bagian kanan dari *tree*.



Baris 16-29 merupakan fungsi yang digunakan untuk memasukkan *node* ke dalam *tree*. Pada fungsi ini, apabila *node-node tree* yang ada masih kosong, maka akan dibentuk *node* ‘n’ yang nantinya akan memesan/mengalokasikan memori menggunakan fungsi ‘new’. Baris 19-21 akan berjalan setelah *node* ‘n’ memiliki nilai/isi dan memorinya sudah teralokasi. Untuk nilai dari data ‘value’ pada akan disimpan pada data dari *node* ‘n’. ‘NULL’ akan disimpan pada *pointer* kiri dan *pointer* kanan dari *node* ‘n’. Baris 23 dan 24 akan ter-*trigger* setelah selesai menjalankan baris sebelumnya. Disitu, data yang dimasukkan adalah data yang nilainya lebih kecil dari data yang sebelumnya dimasukkan ke dalam *node*. Baris 26 dan 27 juga akan ter-*trigger* setelah menjalankan baris sebelum baris 23 dan 24. Disitu, data yang dimasukkan adalah data yang nilainya lebih besar dari data sebelumnya yang dimasukkan ke dalam *node*.

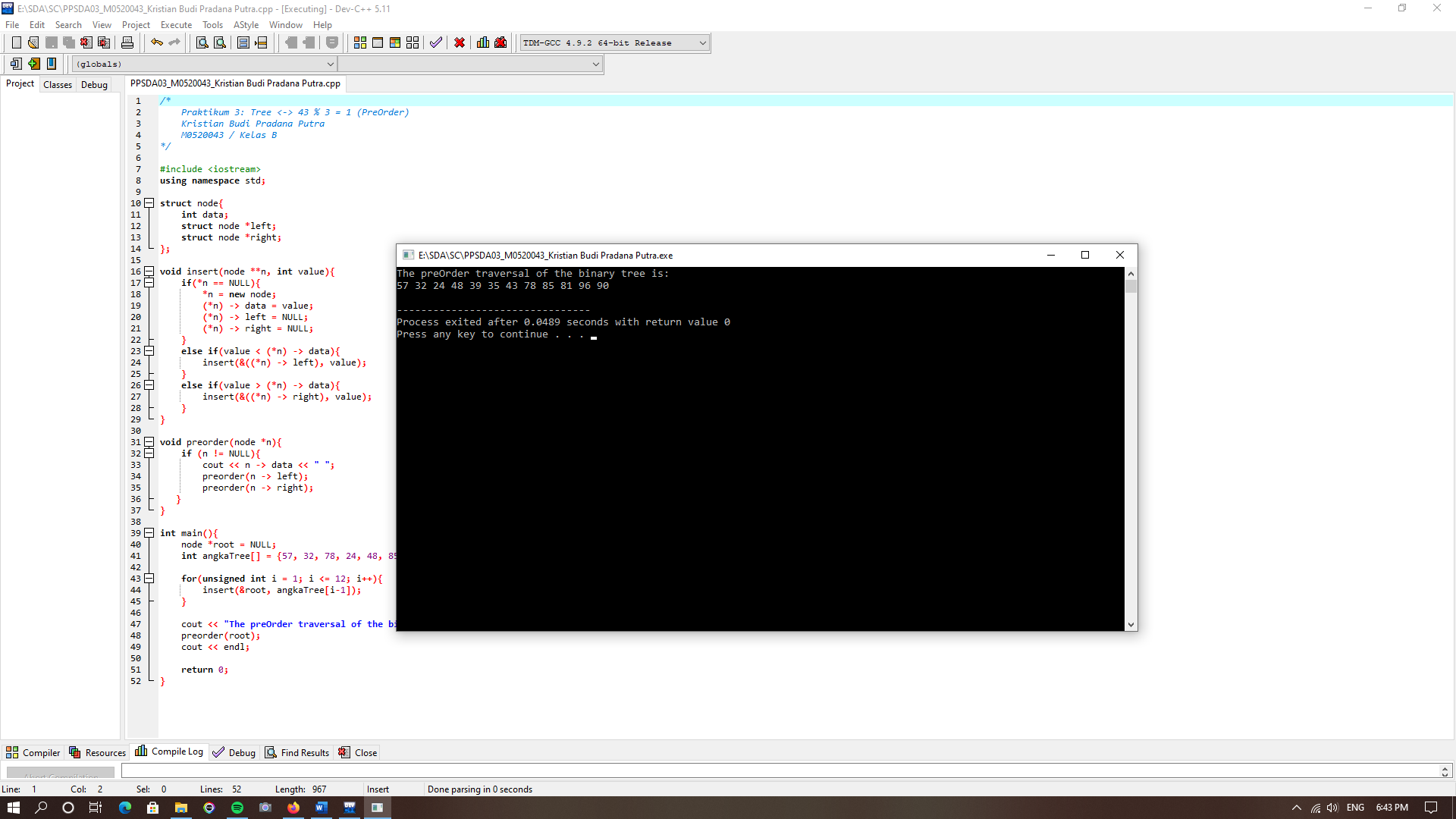


Baris 31-37 terdapat fungsi ‘preorder’ yang sesuai dengan bagian saya pada tugas praktikum 3 kali ini. Pada baris inilah akan dimulai proses *preorder traversal* dari *tree* yang sudah tidak kosong tadi. Fungsi ‘preorder’ mengambil *node* ‘n’ dari *binary tree* sebagai sebuah argumen dan mencetak/menampilkan elemen yang ada dari *tree* dalam bentuk *preorder*. Fungsi ‘preorder’ ini menggunakan fungsi rekursif.



Baris 39-52 merupakan fungsi main, yang mana program-program di dalamnya akan di eksekusi pertama kali. Pada fungsi ‘main’, *root node* pertama kali didefinisikan sebagai NULL atau *empty* (baris 40). Baris 41 menampilkan data *array* bernama ‘angkaTree’ yang didalamnya berisi angka-angka sesuai dengan *tree* pada gambar soal yang diberikan. Nah, semua *node* yang *value*/angkanya sudah didefinisikan dalam data *array* ‘angkaTree’ tadi dimasukkan ke dalam *binary tree* menggunakan *for loop*, lalu terjadi pemanggilan fungsi *insert* (seperti yang ada pada baris ke 43 dan 44). Lalu, akhirnya pada baris 47-49, fungsi ‘preorder()’ dipanggil menggunakan *root* *node*, dan semua nilai/*value* *tree* akan ditampilkan dalam bentuk *preorder* menggunakan ‘cout’ (*traverse the tree preorder*).

1. **Hasil *Output***



Hasil *output* yang ditampilkan pada *compiler* sudah sesuai dengan jawaban yang ada pada slide ppt yang diberikan dalam material “Week 3 Tree”.