*Implementasi Struktur Data*

*Genap 2021/2022*

**Jurnal 09: Binary Tree and Tree Traversal**

**Implementation**

Nama : Alfian Mohammad Rizki

NIM : 6706213005

Kelas : D3 RPLA 45-04

Kerjakanlah soal-soal berikut. Penjelasan yang diminta cukup dengan menggunakan narasi dituliskan pada Word/ sejenisnya. Diharapkan penulisan kode program sesuai dengan Java coding style. Kumpulkan kode program, penjelasan, dan screenshot hasil jalannya program.

Bacalah Modul 09: Sorting dan Searching. Tuliskan kembali kode soal yang ada pada sub-bab 9.3.1 (Pembuatan Node Tree), 9.3.2 (Pembuatan Binary Tree), dan 9.3.3 (Pembuatan tree) dan jalankan program untuk melihat hasilnya. Kemudian, ubahlah program pada sub-bab 9.3.3 agar dapat menerima masukan berupa integer.

Masukkan data berikut sesuai dengan urutannya:

23 – 10 – 5 – 8 – 11 – 4 – 7

Jalankan program. Kemudian, kumpulkan project program, screenshot hasil jalannya program, penjelasan program, beserta gambar pohon yang terbentuk sesuai dengan masukan tersebut. Hal yang harus dijelaskan adalah sebagai berikut:

a. Penjelasan koding

b. Gambar dari pohon yang terbentuk sesuai dengan masukan pada program

c. Penjelasan mengenai mengapa pohon yang terbentuk seperti itu, juga penjelasan mengenai

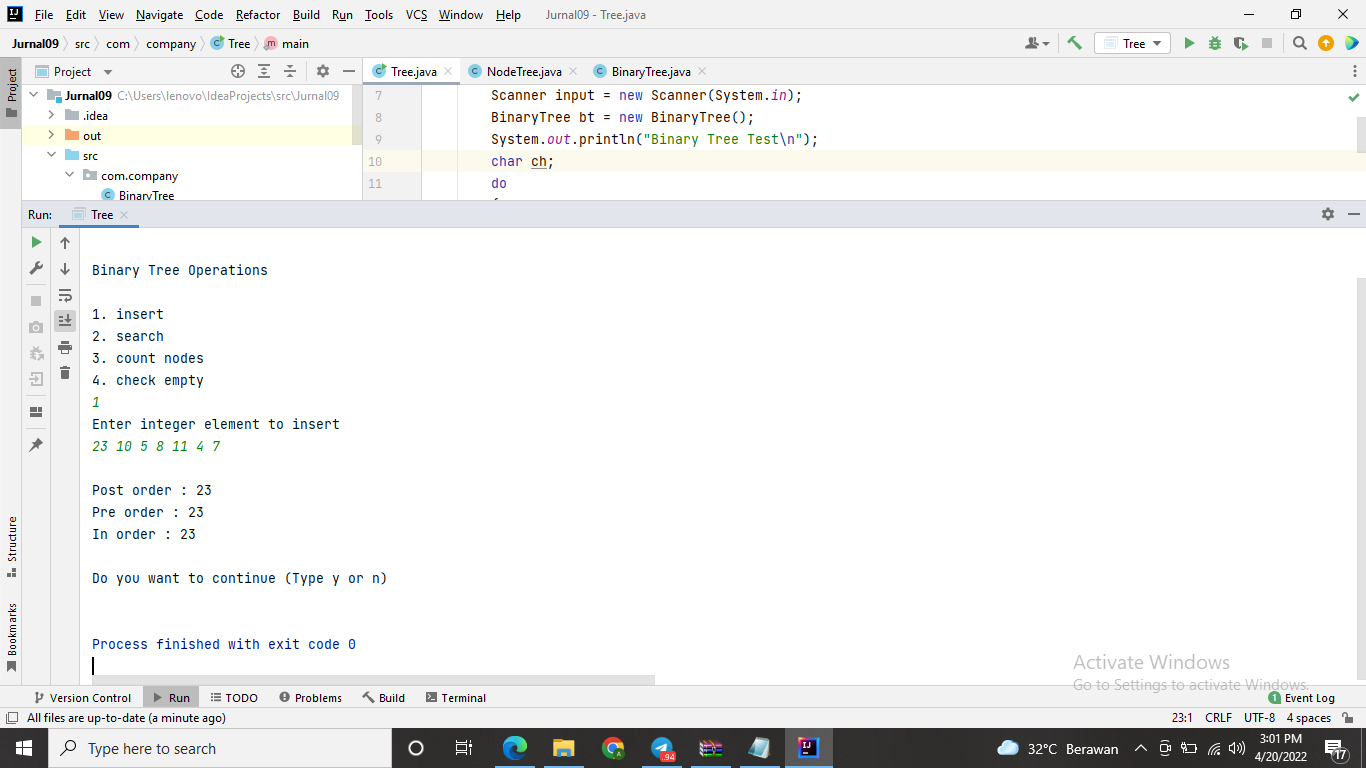
traversal preorder, inorder, dan postorder.

**Noted:**

Seperti biasa, penjelasan harus dilakukan dalam bahasa dan gaya Anda sendiri. **Dilarang** mengambil/ menyontek penjelasan dari sumber lain!

Jawaban :

Output Source Code :



Penjelasan Coding :

Pada jurnal ke 9 ini sudah masuk pada bagian binary tree yang dimana pada jurnal kali ini kita membuat 3 class, bahkan kita hanya tinggal menyalin saja dari modul yang sudah diberikan. 2 class dapat disalin dan tidak usah diganti, sedangkan 1 class lagi boleh disalin dan dilakukan penggantian. Dimana class tersebut terbagi menjadi 3 yaitu class Binary Tree, Node Tree, dan Tree. Pada class node tree berfungsi untuk menampung berbagai tipe data yang dimana dijadikan kelas generic yang akan dibagi lagi menjadi 2 subtree yaitu subtree left dan subtree right. Pada source code nya berisi 2 konstruktor yaitu untuk inisiasi, dan item, setelah itu terdapat function untuk menentukan node kiri atau node kanan, setelah diatur maka kita membuat function get untuk mendapatkan node kiri atau kanan. Jika sudah ditentukan maka kita tinggal memasukkan data dan akan diproses data tersebut masuk kedalam data bagian kiri atau kanan, ketika semua proses sudah selesai maka kita menggunakan return untuk memanggil atau memperoleh nilai tersebut.

Pada bagian class binary search merupakan sebuah kelas generic yang dimana pada class ini memiliki satu atribut yaitu root yang berfungsi sebagai awal tree. Pada source codenya pun menggunakan if untuk menentukan bahwa awalan tree tersebut kosong atau tidak dan tentunya menggunakan konstruktor . jika pada node kiri kosong maka akan dijalankan yang bagian kiri terlebih dahulu ataupun sebaliknya, untuk memasukkan elemen tersebut menggunakan insert sebagai methodnya. Karena berbentuk pohon maka pada bagian ini menggunakan method rekursif. Selain itu pada bagian ini juga menggunakan function untuk in order, post order, dan pre order, dan tak lupa pula untuk menambahkan method count didalamnya untuk menghitung total keseluruhan dari elemen tree yang terbentuk.

Pada langkah terakhir yaitu menggunakan class tree yang berfungsi sebagai class main yang dimana didalam class ini terdapat sebuah menu yang berisi insert elemen, pencarian, melihat jumlah data secara keseluruhan dan mengecek apakah tree tersebut kosong atau tidak.

Saat membuat source code ini kita menambahkan class sebelumnya yang telah kita buat untuk menyempurnakan proses running program ini yaitu melakukan compare dengan class binary tree, dan node tree yang nantinya akan saling berhubungan. Jika semua proses telah selesai maka cetak hasil akhir.

Gambar Pohon yang terbentuk :

Post-order :

In-order (Kiri, Root, Kanan) :

Pre-order (Root, Kiri, Kanan) :