**Laporan Praktikum ASD**

**Pertemuan 13**

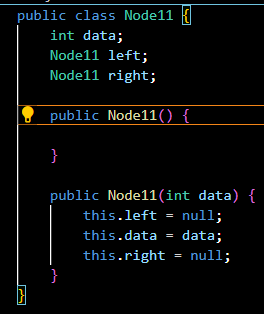
**Nama: Gegas Anugrah Derajat**

**Kelas: SIB-1F**

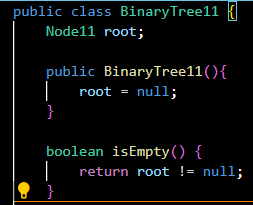
**NIM: 2341760140**

**Percobaan 1**

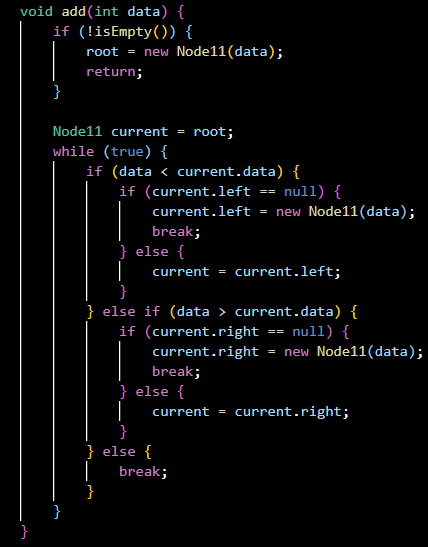
**Membuat class Node dan menambahkan atribut data, left dan right, serta konstruktor default dan berparameter.**

****

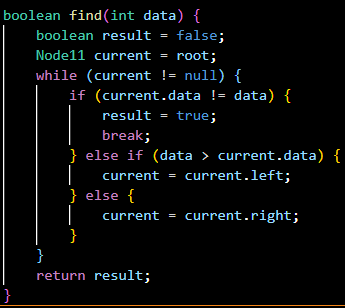
**Membuat class BinaryTree, menambahkan atribut root, konstruktor default, dan method isEmpty**

****

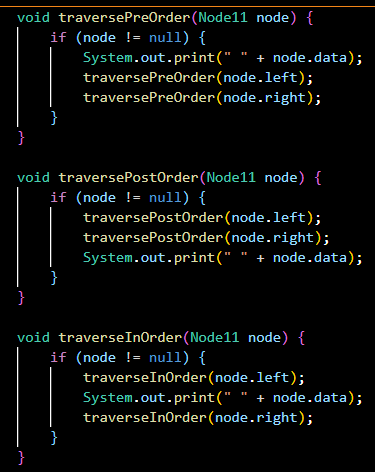
**Menambahkan method add()**

****

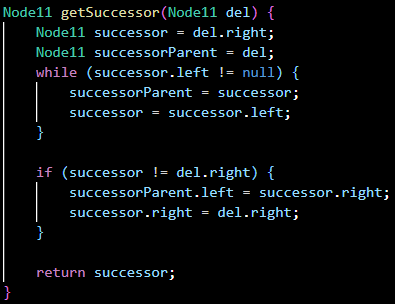
**Menambahkan method find()**

****

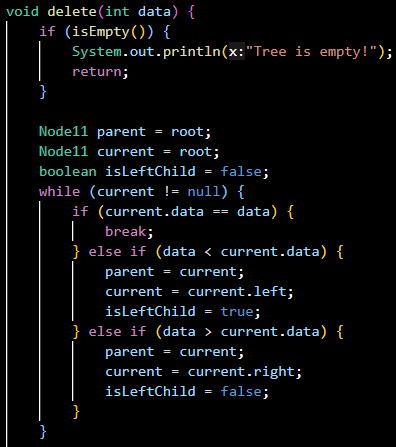
**Menambahkan method traversePreOrder(), traverseInOrder() dan traversePostOrder()**

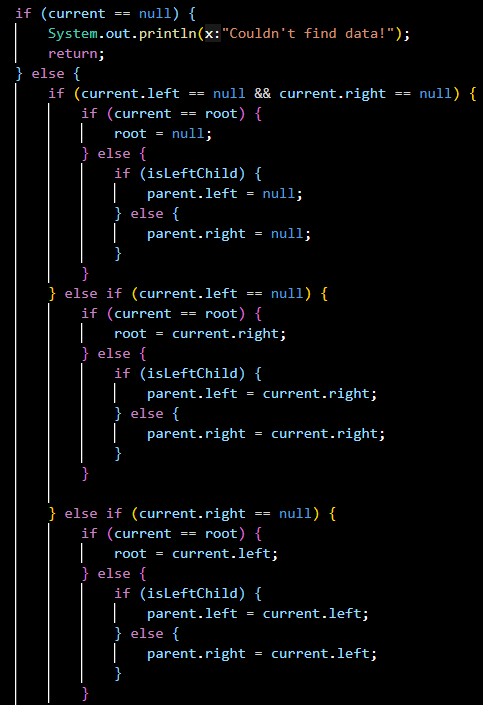
****

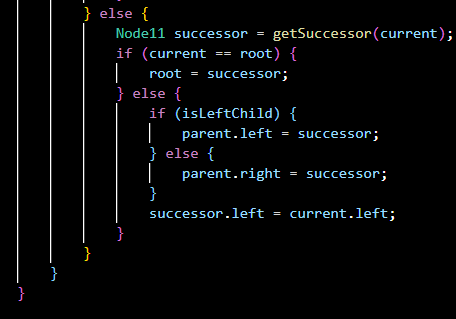
**Menambahkan method getSuccessor()**

****

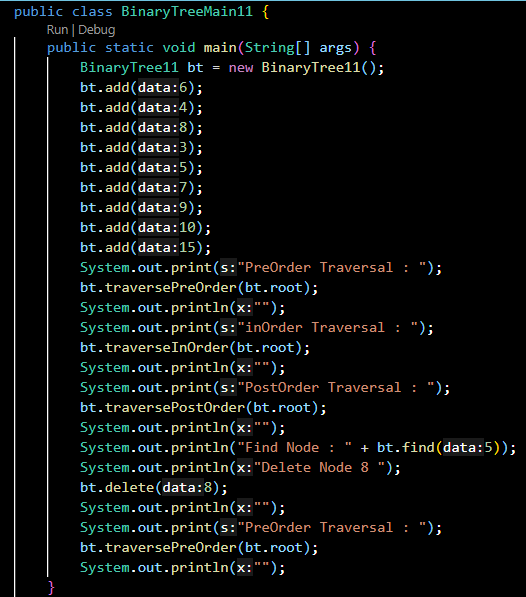
**Menambahkan method delete()**

****

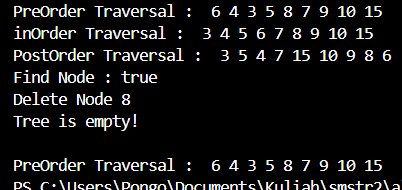
****

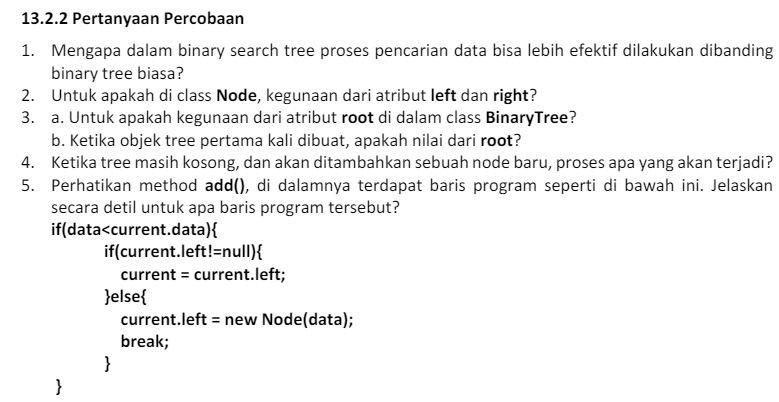
****

**Membuat class BinaryTreeMain dan menambahkan method main()**

****

**Hasil**

****

****

**Jawaban:**

1. **Karena BST memiliki struktur data yang teratur, setiap node memiliki nilai yang lebih besar dari semua node di subtree kirinya dan memiliki nilai yang lebih kecil dari semua node di subtree kanannya.**
2. **Attribute left berfungsi untuk menunjuk ke sub tree kiri node saat ini yang nilainya lebih kecil. Attribute right berfungsi untuk menunjuk ke sub tree kanan saat ini yang nialinya lebih besar.**

**a. sebagai node awal**

**b. nilai yang pertama kali dibuat yaitu null**

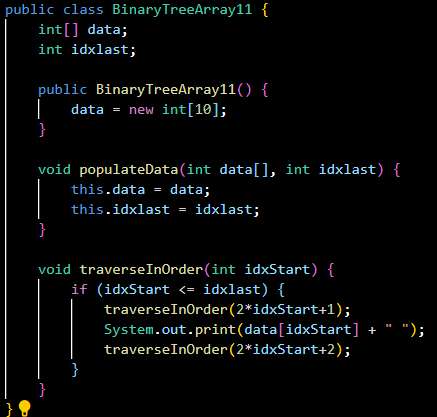
1. **Mengisi root dengan nilai yang baru ditambahkan, kemudian menghentikan proses**
2. **Pada validasi pertama membandingkan apakah data lebih kecil dengan currentdata. Jika lebih kecil, maka akan kembali validasi apakah currentleft tidak bernilai null, jika null maka current akan**

**di ubah menjadi tempat currentleft. Jika tidak maka data akan di simpan pada tempat**

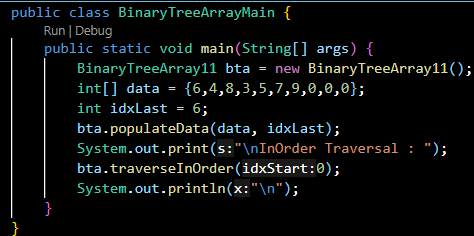
**currentleft.**

**Percobaan 2**

**Membuat class BinaryTreeAraay, menambahkan atribut data dan idxLast di dalam class dan menambahkan method populateData() dan traverseInOrder().**

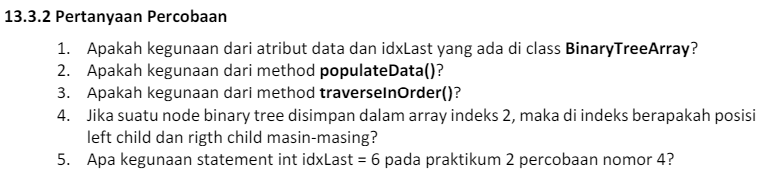
****

**Membuat class BinaryTreeArrayMain dan menambahkan method main**

****

**Hasil**

****

****

**Jawaban:**

1. **Menandakan indeks terakhir yang ada di dalam array data.**
2. **Mengisi representasi binary tree ke dalam objek BinaryTreeArray.**
3. **Menjelajahi dan mencetak elemen-elemen dalam representasi binary tree yang disimpan dalam objek BinaryTreeArray.**

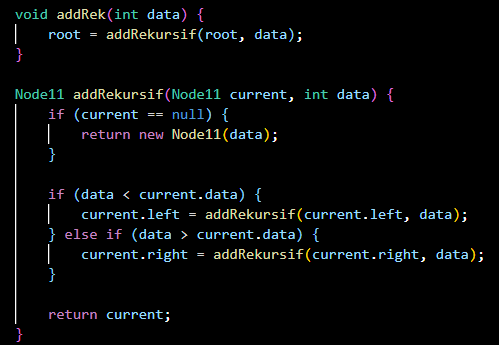
**right child = 2\*2+2 = 6**

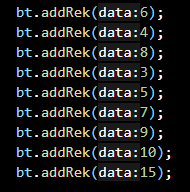
**left child = 2\*2+1 = 5**

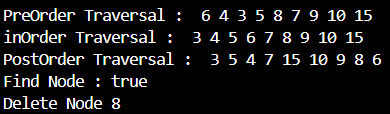
1. **Menentukan batas akhir elemen tree yang tersimpan dalam array data**

**Tugas**

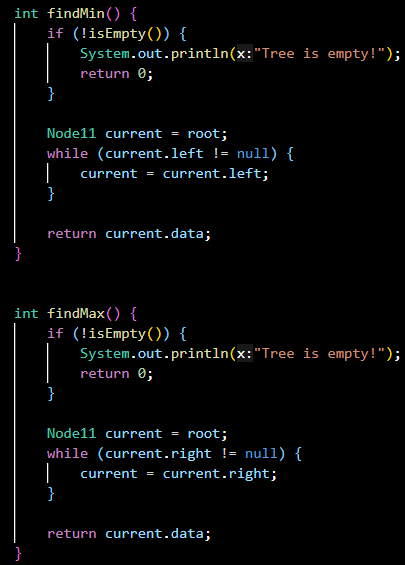
1. **Buat method di dalam class BinaryTree yang akan menambahkan node dengan cara rekursif.**

****



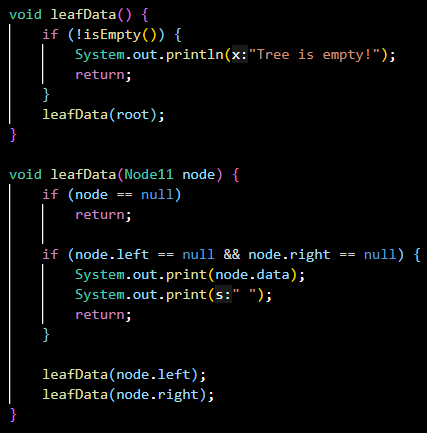


1. **Buat method di dalam class BinaryTree untuk menampilkan nilai paling kecil dan yang paling besar yang ada di dalam tree.**

****

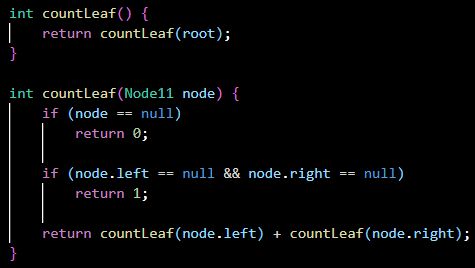


1. **Buat method di dalam class BinaryTree untuk menampilkan data yang ada di leaf.**

****



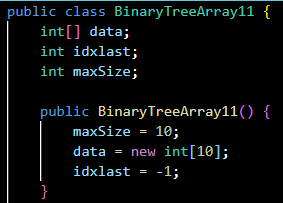
1. **Buat method di dalam class BinaryTree untuk menampilkan berapa jumlah leaf yang ada di dalam tree.**

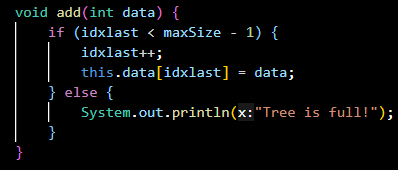
****



**5. Modifikasi class BinaryTreeArray, dan tambahkan :**

**• method add(int data) untuk memasukan data ke dalam tree**

****

****

**• method traversePreOrder() dan traversePostOrder()**

