**Laporan Praktikum Alogaritma dan Struktur Data**

**Jobsheet 14 :** **TREE**



**Nama: Aqil Rahmat Alifiandi NIM: 2341760099**

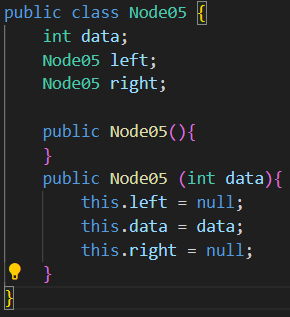
**Kelas: SIB 1E**

**PROGRAM STUDI SISTEM INFORMASI BISNIS JURUSAN TEKNOLOGI INFORMASI POLITEKNIK NEGERI MALANG**

**2023/2024**

**Praktikum 1**

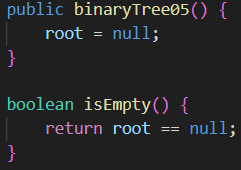
1. Buatlah class Node.05, BinaryTreeNoAbsen dan BinaryTreeMainNoAbsen
2. Di dalam class Node, tambahkan atribut data, left dan right, serta konstruktor default dan berparameter.



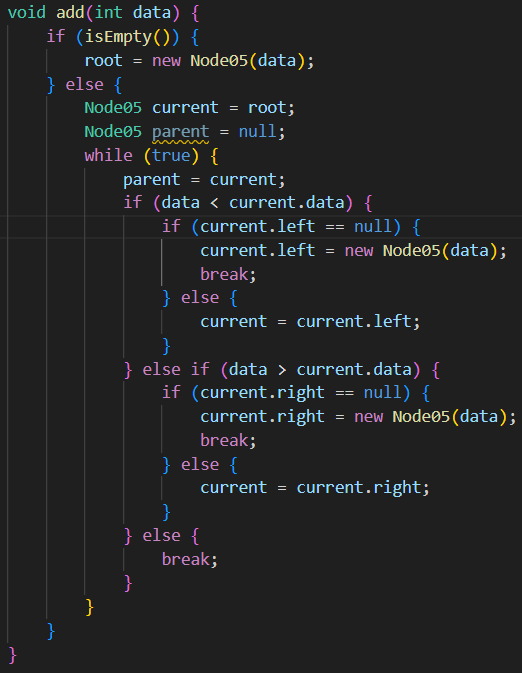
1. Di dalam class BinaryTreeNoAbsen, tambahkan atribut root.



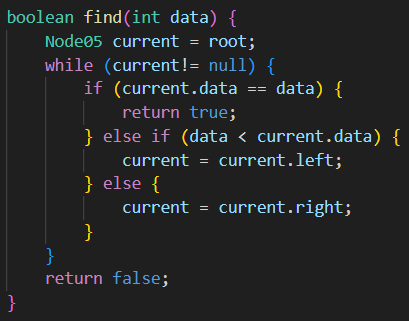
1. Tambahkan konstruktor default dan method isEmpty() di dalam class BinaryTree05



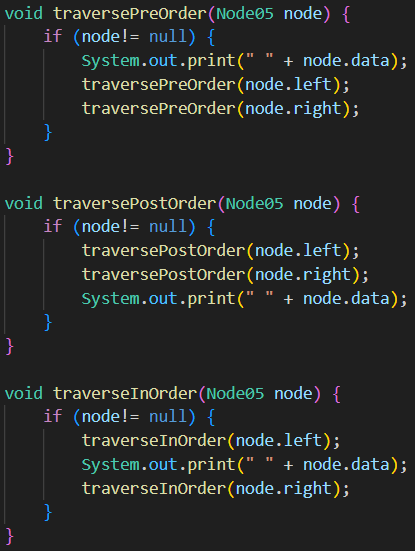
1. Tambahkan method add() di dalam class BinaryTreeNoAbsen. Di bawah ini proses penambahan node tidak dilakukan secara rekursif, agar lebih mudah dilihat alur proses penambahan node dalam tree. Sebenarnya, jika dilakukan dengan proses rekursif, penulisan kode akan lebih efisien.



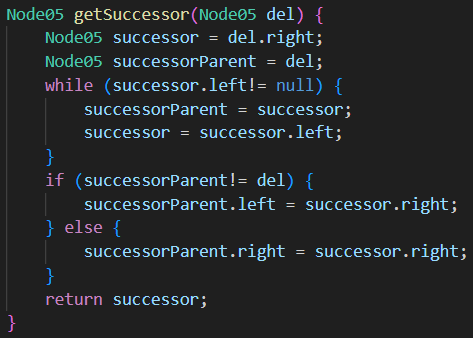
1. Tambahkan method find()



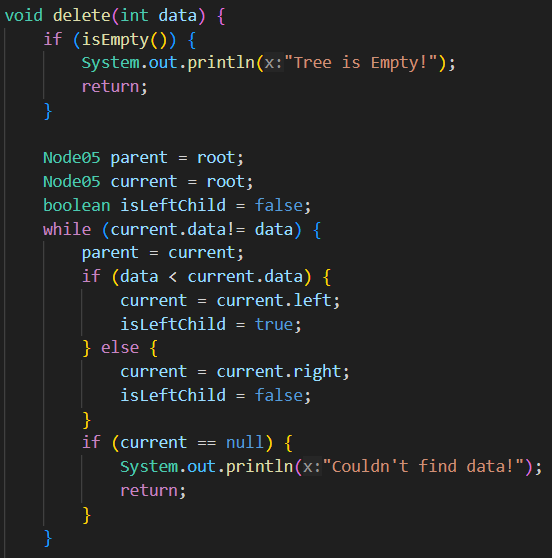
1. Tambahkan method traversePreOrder(), traverseInOrder() dan traversePostOrder(). Method traverse digunakan untuk mengunjungi dan menampilkan node-node dalam tree, baik dalam mode pre-order, in-order maupun post-order



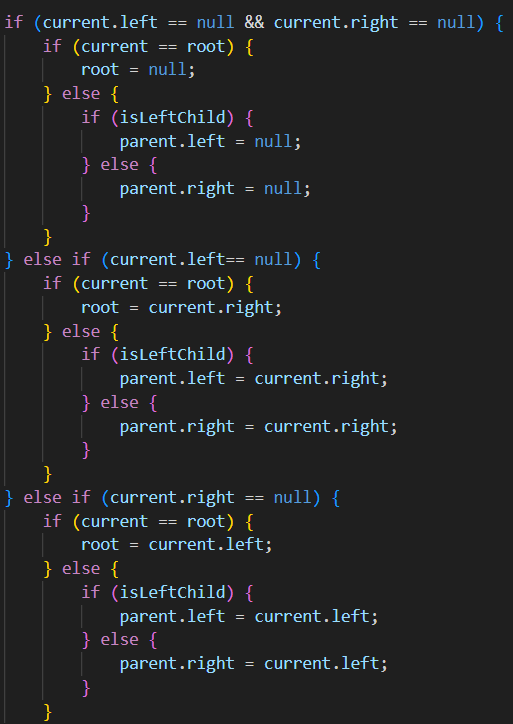
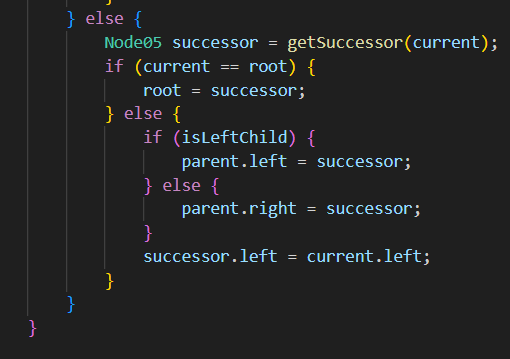
1. Tambahkan method getSuccessor(). Method ini akan digunakan ketika proses penghapusan node yang memiliki 2 child.



1. Tambahkan method delete(). Di dalam method delete tambahkan pengecekan apakah tree kosong, dan jika tidak cari posisi node yang akan di hapus



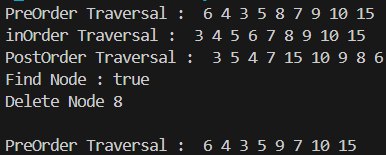
1. Kemudian tambahkan proses penghapusan didalam method delete() terhadap node current yang telah ditemukan.

1. Buka class BinaryTreeMainNoAbsen dan tambahkan method main() kemudian tambahkan kode berikut ini



1. Compile dan jalankan class BinaryTreeMain untuk mendapatkan simulasi jalannya program tree yang telah dibuat



**Pertanyaan**

1. Mengapa dalam binary search tree proses pencarian data bisa lebih efektif dilakukan dibanding binary tree biasa?

**Jawaban:** Karena, pada binary search tree telah ditambahkan aturan baru yaitu, “semua left-child harus lebih kecil dibandingkan right-child dan parentnya” sehingga lebih mudah untuk melakukan pencarian data.

1. Untuk apakah di class Node, kegunaan dari atribut left dan right?

**Jawaban:** berfungsi untuk menyimpan "left child" atau nilai yang lebih kecil dari root (node induk) dan atribut right berfungsi untuk menyimpan "right child" atau nilai yang lebih besar dari root (node induk)

1. **a**. Untuk apakah kegunaan dari atribut root di dalam class BinaryTree?

**Jawaban:** Didalam BinaryTree root digunakan sebagai kepala atau inti, sama dengan head pada linked list yang digunakan sebagai kepala dari setiap linked list atau inti dari sebuah tree.

**b**. Ketika objek tree pertama kali dibuat, apakah nilai dari root?

**Jawaban:** ketika objek tree pertama kali dibuat nilai dari root bernilai null, karena masih belum ada data yang dimasukan

1. Ketika tree masih kosong, dan akan ditambahkan sebuah node baru, proses apa yang akan terjadi?

**Jawaban:** Penambahan node baru yang akan digunakan sebagai root.

1. Perhatikan method add(), di dalamnya terdapat baris program seperti di bawah ini. Jelaskan secara detil untuk apa baris program tersebut?

**if(data<current.data){**

**if(current.left!=null){**

**current = current.left;**

**}else{**

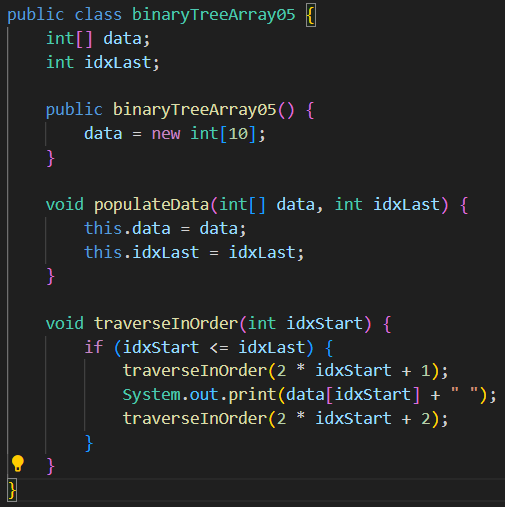
**current.left = new Node(data);**

**break; } }**

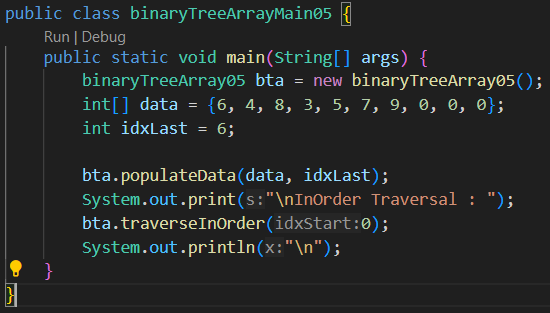
**Jawaban:** untuk mengecek apakah nilai input lebih kecil dari parent atau tidak. Jika iya, dilakukan pengecekan apakah current.left != null atau masih memiliki left child yang dimana memiliki subtree lagi. Jika iya maka dilakukan traversal dengan mengubah nilai current menjadi current.left. lalu, ada pengecekan jika tidak current.left == null atau tidak memiliki subtree atau left child maka posisi current.left tersebut akan menjadi tempat untuk meletakkan data yang diinput.

**Praktikum 2**

1. Buatlah class BinaryTreeArrayNoAbsen dan BinaryTreeArrayMainNoAbsen.
2. Buat atribut data dan idxLast di dalam class BinaryTreeArrayNoAbsen. Buat juga method populateData() dan traverseInOrder().



1. Kemudian dalam class BinaryTreeArrayMain05 buat method main() dan tambahkan kode seperti gambar berikut ini di dalam method Main



1. Jalankan class BinaryTreeArrayMain dan amati hasilnya!



**Pertanyaan**

1. Apakah kegunaan dari atribut data dan idxLast yang ada di class BinaryTreeArray?

**Jawaban:** Atribut data adalah sebuah array yang menyimpan elemen-elemen pohon biner, idxLast menyimpan indeks dari elemen terakhir yang valid dalam array data.

1. Apakah kegunaan dari method populateData()?

**Jawaban:** Menggunakan method populateData() dapat dengan mudah mengisi atau memperbarui data pohon biner yang diwakili oleh array, dan menentukan hingga mana elemen-elemen dalam array tersebut valid untuk digunakan dalam operasi pada pohon biner.

1. Apakah kegunaan dari method traverseInOrder()?

**Jawaban:** Kegunaan method traverseInOrder() adalah untuk mengunjungi dan mencetak elemen-elemen dari pohon biner dalam urutan yang teratur, yang sangat berguna untuk analisis data dan verifikasi struktur pohon biner.

1. Jika suatu node binary tree disimpan dalam array indeks 2, maka di indeks berapakah posisi left child dan rigth child masin-masing?

**Jawaban:** Jika suatu node binary tree disimpan dalam array pada indeks 2, maka posisi left child adalah pada indeks 5, dan posisi right child adalah pada indeks 6.

1. Apa kegunaan statement int idxLast = 6 pada praktikum 2 percobaan nomor 4?

**Jawaban:** Statement int idxLast = 6; sangat penting untuk menentukan batas elemen-elemen yang valid dalam array data, memungkinkan traversal yang benar dan efisien pada pohon biner.

**Tugas Praktikum**

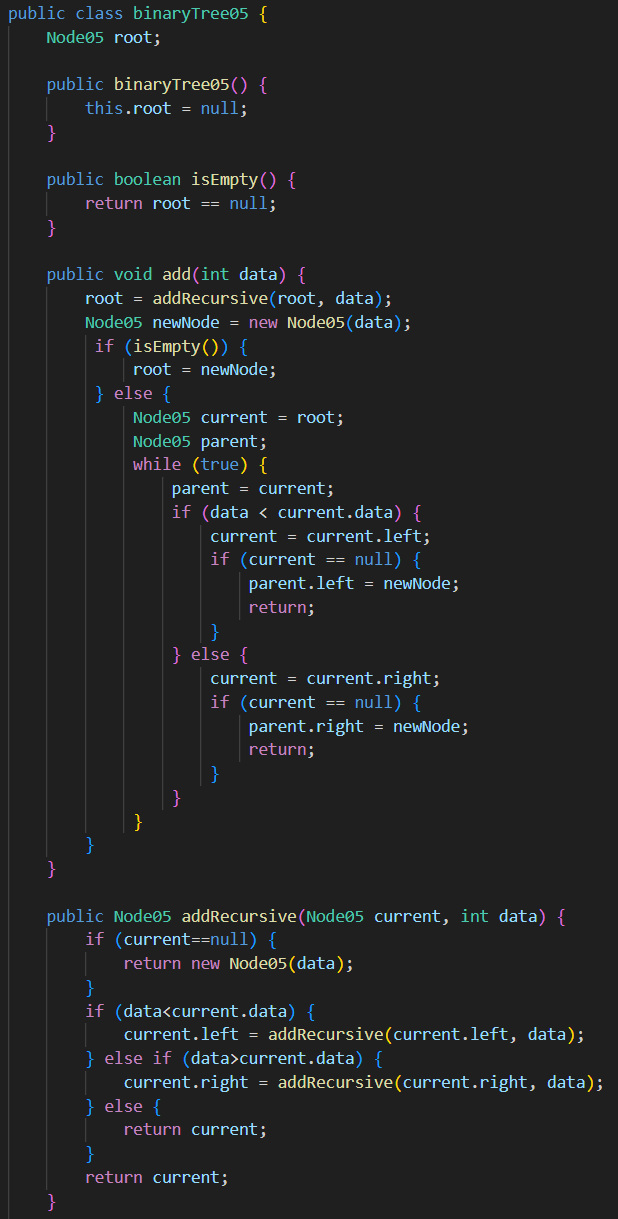
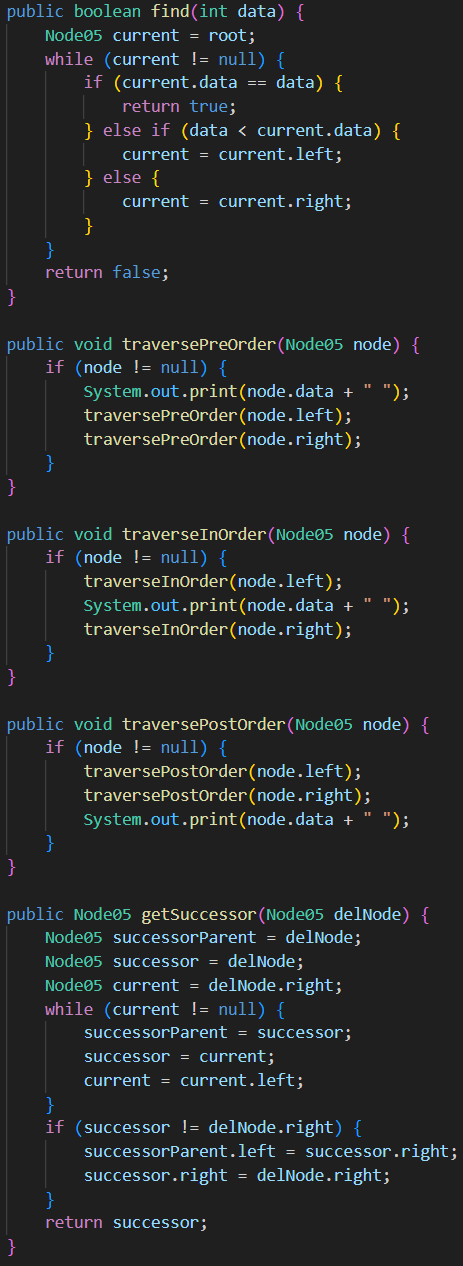
1. Buat method di dalam class BinaryTree yang akan menambahkan node dengan cara rekursif.
2. Buat method di dalam class BinaryTree untuk menampilkan nilai paling kecil dan yang paling besar yang ada di dalam tree.
3. Buat method di dalam class BinaryTree untuk menampilkan data yang ada di leaf.
4. Buat method di dalam class BinaryTree untuk menampilkan berapa jumlah leaf yang ada di dalam tree.
5. Modifikasi class BinaryTreeArray, dan tambahkan :

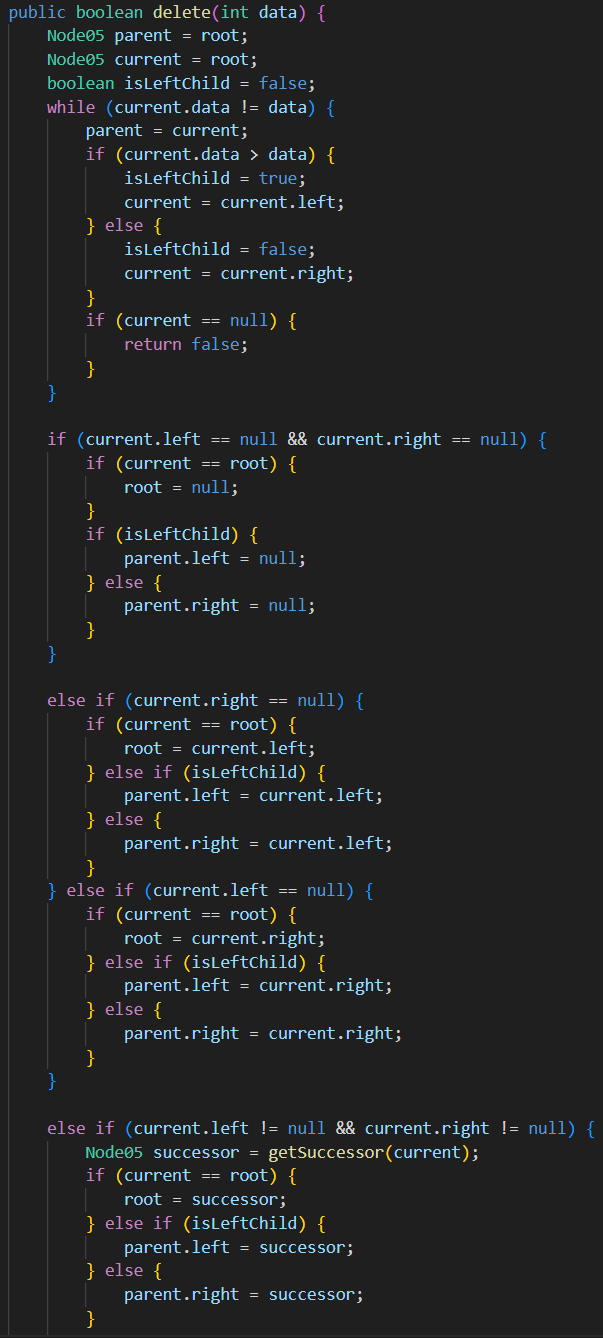
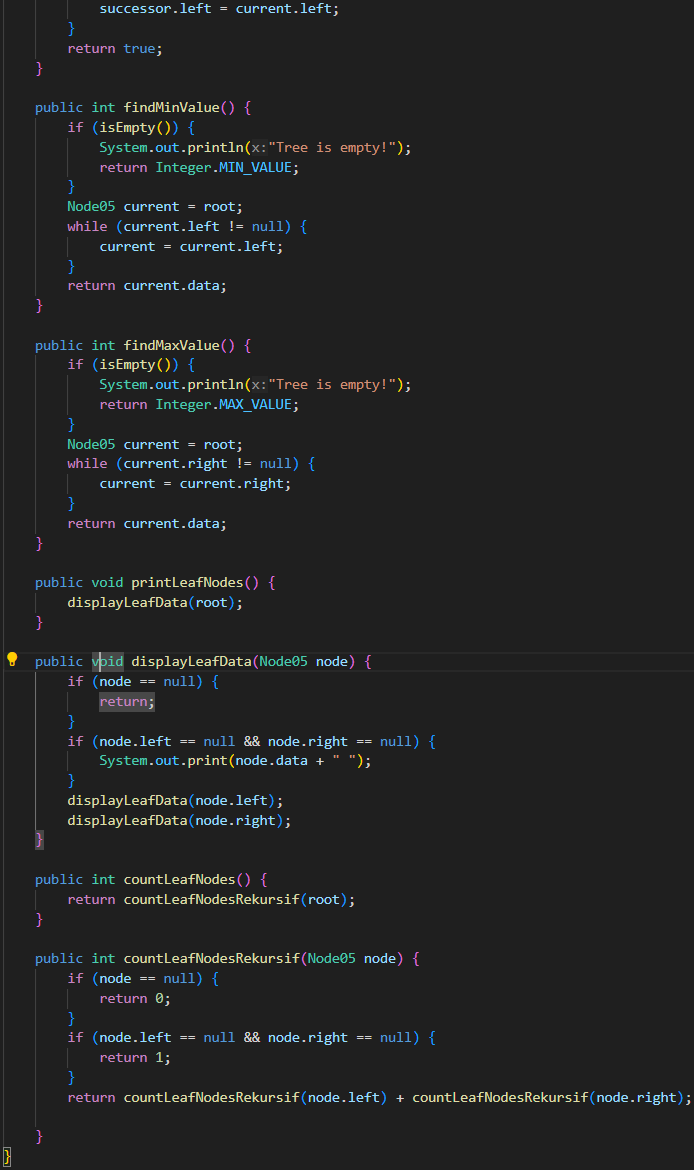
• method add(int data) untuk memasukan data ke dalam tree

• method traversePreOrder() dan traversePostOrder(

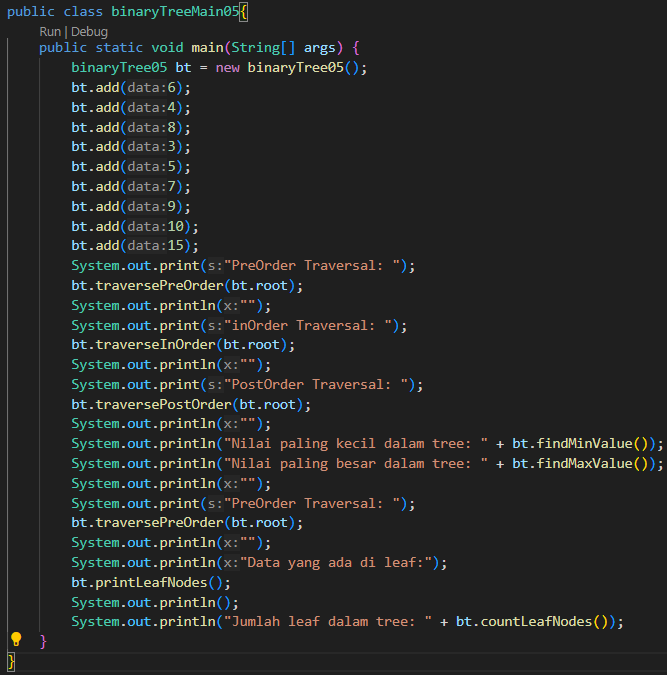
**Jawaban:**

* BinaryTree

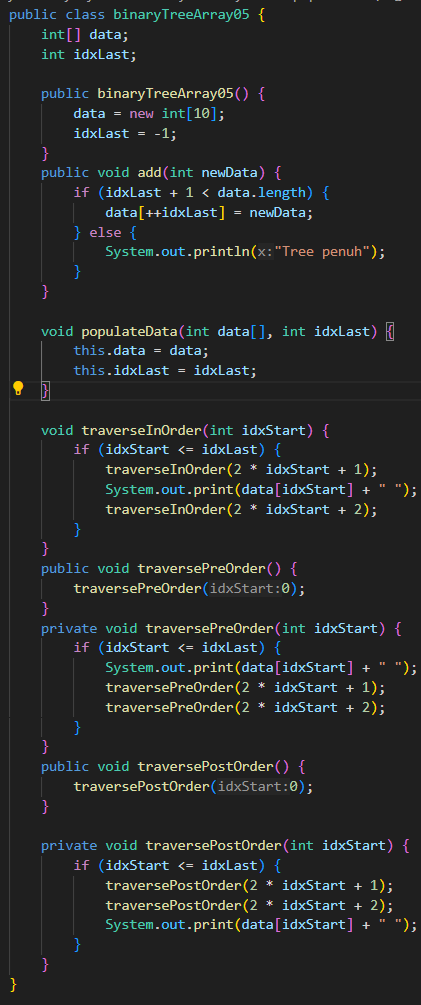




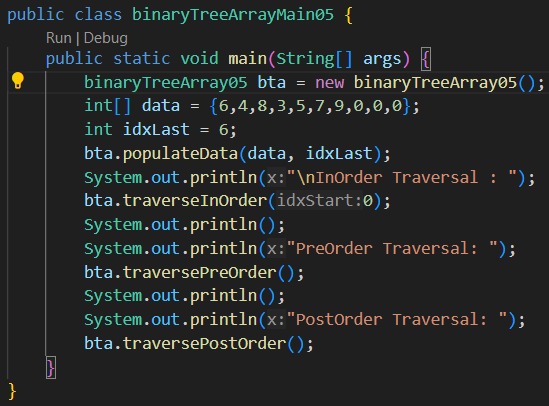
* BinaryTreeMain



* BinaryTreeArray



* BinaryTreeArrayMain



* Hasil:

