

1. Class `TreeNode.java`
 - a. Variabel `leftNode` dan `rightNode`: Menyimpan referensi ke anak kiri dan kanan.
 - b. Konstruktor: Menginisialisasi nilai node dan menyetel anak kiri dan kanan ke null.
 - c. Metode `insert`: Menyisipkan nilai baru ke dalam pohon. Jika nilai lebih kecil dari data node saat ini, masuk ke subtree kiri, jika lebih besar, masuk ke subtree kanan. Duplikat nilai diabaikan.
2. Class `Tree.java`
 - a. `Tree` adalah kelas yang merepresentasikan pohon biner.
 - b. `root` adalah referensi ke simpul akar dari pohon.
 - c. `insertNode` adalah metode untuk menyisipkan node baru ke dalam pohon.
 - d. `preorderTraversal`, `inorderTraversal`, dan `postorderTraversal` adalah metode untuk melakukan traversal pohon sesuai urutan yang diinginkan.
 - e. `searchBST` adalah metode untuk mencari nilai tertentu dalam pohon biner.
3. Class `Main.java`
 - a. `Main` merupakan kelas utama yang digunakan untuk menjalankan program.
 - b. Dalam `main` method, kita membuat objek `Tree` dan menyisipkan beberapa node ke dalamnya.
 - c. Kemudian, kita melakukan traversals (`preorder`, `inorder`, `postorder`) pada pohon yang telah dibuat.
 - d. Terakhir, kita mencari nilai tertentu dalam pohon dan mencetak hasilnya.
4. Gambar Tree

