

Jurnal 13: Binary Search Tree

Kerjakanlah soal berikut. Jawaban dinilai berdasarkan dua kriteria: a. Kode sesuai dengan permintaan soal, dan b. Sesi tanya-jawab dengan Asprak. Kerjakan sesuai dengan prinsip Java Coding Style.

Diberikan sebuah BST yang berisi NIM Mahasiswa sebagai Berikut

111, 90, 150, 80, 100, 130, 170, 95, 140

Modifikasi kode Tree dan TreeNode dengan ketentuan berikut:

1. **Tambahkan method findMin()** pada kelas Tree yang mengembalikan nilai terkecil dari BST.
2. **Tambahkan method findMax()** pada kelas Tree yang mengembalikan nilai terbesar dari BST.
3. Buat method height() yang mengembalikan tinggi dari pohon.
4. Modifikasi method searchBST() agar mengembalikan TreeNode jika ditemukan, atau null jika tidak.
5. Urutkan memakai Tranversal InOrder

Tips Implementasi

- Hanya membutuhkan 3 buah class saja, yaitu Main, Tree dan TreeNode.
- **Cari Min/Max:** Telusuri node paling kiri untuk min, paling kanan untuk max.
- **Tranversal Inorder:** Kunjungi kiri → node → kanan secara rekursif.

Contoh Output

```
Inorder traversal:
80 90 95 100 111 130 140 150 170

Nilai terkecil: 80
Nilai terbesar: 170
Tinggi pohon: 4

Mencari data 80:
Data ditemukan: 80

Mencari data 100:
Data ditemukan: 100
```

Java Coding Style

Ikuti gaya penulisan berikut di kode program agar mendapat nilai maksimal

1. Setiap nama kelas, variabel dan konstanta harus dapat menggambarkan isinya.

Misal, membuat variabel untuk menyimpan jumlah penumpang di bus. Deklarasi:

```
int penumpang = 0;           // Bagus, OK
int p = 0;                   // Tidak OK
```

2. Penulisan nama harus menggunakan huruf besar/kecil yang sesuai.

Nama kelas: UpperCamelCase. Misal: PenumpangBus

Nama variabel: lowerCamelCase. Misal: penumpangBus

Nama konstanta: CAPITAL_CASE. Misal: PENUMPANG_BUS

Nama method: lowerCamelCase. Misal: hitungPenumpangBus

3. Penggunaan tab/spasi ketika berbeda blok program, WAJIB masuk 1 tab atau 4 spasi ke dalam.

```
public class PenumpangBus {
    public static void main(String[] args) {
        int penumpang = 0;
        penumpang = penumpang + 5;
        penumpang = penumpang - 4 + 2;
        penumpang = penumpang - 1;
        penumpang = penumpang - 2 + 3;
        penumpang = penumpang - 2 + 5;
        penumpang = penumpang - 1 + 3;
        System.out.println(penumpang);
    }
}
```

4. Gunakan komentar seperlunya.

Javadoc comment (yang warna biru) hanya untuk mengomentari kelas, method dan variable

```
/** Jumlah penumpang di bus */
int penumpang = 0;

/** Ada 5 orang penumpang naik ke bus */
penumpang = penumpang + 5;

/** Penumpang turun 4 orang, naik 2
penumpang = penumpang - 4 + 2;
```