

**LAPORAN PRAKTIKUM**  
**PERTEMUAN 10-11**



**Nama :**

Razhendriya Vania Ramadhan Suganjarsarwat (2311104048)

**Dosen :**

**WAHYU ANDI SAPUTRA**

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2024**

## **I. TUJUAN**

Tujuan dari percobaan ini adalah untuk mempelajari struktur data pohon biner (Binary Tree), memahami cara membuat, memanipulasi, dan mengelola node dalam pohon. Selain itu, laporan ini bertujuan untuk:

1. Mengimplementasikan pohon biner dalam bahasa pemrograman C++.
2. Mempelajari teknik traversal pohon biner seperti pre-order, in-order, dan post-order.
3. Memahami cara memeriksa validitas sebuah pohon sebagai Binary Search Tree (BST).
4. Menghitung jumlah simpul daun dalam pohon.

## **II. TOOL**

Vscode

## **III. DASAR TEORI**

**Pohon biner adalah struktur data yang terdiri dari node, di mana setiap node memiliki paling banyak dua anak, yaitu anak kiri (left child) dan anak kanan (right child). Pohon biner sering digunakan dalam berbagai aplikasi, seperti pencarian data, struktur ekspresi matematika, atau organisasi hierarkis.**

### **Jenis Traversal Pohon**

1. **Pre-order Traversal:** Dimulai dari node root, kemudian anak kiri, lalu anak kanan.
2. **In-order Traversal:** Dimulai dari anak kiri, kemudian node root, lalu anak kanan.
3. **Post-order Traversal:** Dimulai dari anak kiri, kemudian anak kanan, lalu node root.

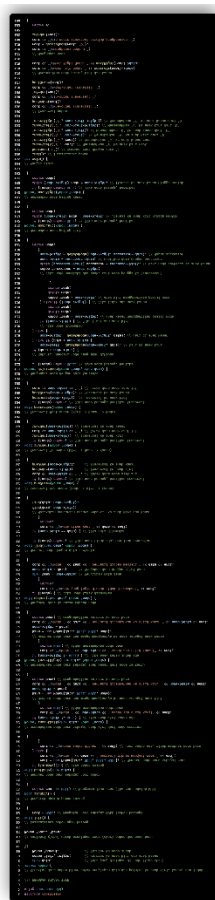
### **Binary Search Tree (BST)**

**BST adalah jenis pohon biner dengan aturan bahwa nilai pada anak kiri lebih kecil daripada nilai root, dan nilai pada anak kanan lebih besar. Hal ini memudahkan pencarian data secara efisien.**

### **Simpul Daun**

**Simpul daun adalah node dalam pohon yang tidak memiliki anak kiri maupun kanan.**  
**Menghitung jumlah simpul daun penting untuk menganalisis struktur pohon.**

## **IV. GUIDED**



### **Penjelasan Program**

- **Fungsi buatNode:** Membuat node baru sebagai root pohon.
- **Fungsi insertLeft dan insertRight:** Menambahkan node baru sebagai anak kiri dan kanan.
- **Traversal:** Menampilkan isi pohon dalam berbagai urutan (pre-order, in-order, post-order).

## V. UNGUIDED



### Penjelasan Program

- **Validasi BST:** Memeriksa properti BST menggunakan batas minimal dan maksimal.
- **Jumlah Simpul Daun:** Menghitung node tanpa anak.

## VI. KESIMPULAN

Dari percobaan ini, dapat disimpulkan bahwa:

1. Pohon biner adalah struktur data yang memungkinkan penyimpanan dan manipulasi data secara hierarkis.
2. Traversal pohon penting untuk menampilkan data sesuai urutan yang diinginkan.
3. Properti BST harus dipenuhi agar pohon biner dapat mendukung operasi pencarian secara efisien.
4. Fungsi tambahan seperti menghitung jumlah simpul daun membantu analisis struktur pohon.