**ISD**

Journal 13


Disusun oleh :
Yehezkiel Theysa Fredy (607062300091)

**Program Studi D3 Rekayasa Perangkat Lunak Aplikasi**

**Fakultas Ilmu Terapan**

**Universitas Telkom**

**Bandung**

**2024**

```java
public class BinarySearchTree {
    Node root;

    // Konstruktor untuk inisialisasi pohon biner dengan root sebagai null
    public BinarySearchTree() {
        root = null;
    }

    // Metode untuk memasukkan sebuah elemen ke dalam pohon
    public void insert(char key) {
        root = insertRec(root, key);
    }

    // Metode rekursif untuk memasukkan elemen baru ke dalam pohon
    private Node insertRec(Node root, char key) {
        // Jika root adalah null, buat Node baru dan kembalikan sebagai root
        if (root == null) {
            root = new Node(key);
            return root;
        }
        // Jika kunci lebih kecil dari data root, masukkan ke sub-pohon kiri
        if (key < root.data)
            root.left = insertRec(root.left, key);
        // Jika kunci lebih besar dari data root, masukkan ke sub-pohon kanan
        else if (key > root.data)
            root.right = insertRec(root.right, key);

        // Kembalikan root yang telah dimodifikasi
        return root;
    }

    // Metode untuk mencari sebuah elemen di dalam pohon
    public boolean search(char key) {
        return searchRec(root, key);
    }

    // Metode rekursif untuk mencari elemen di dalam pohon
    private boolean searchRec(Node root, char key) {
        // Jika root adalah null, elemen tidak ditemukan
        if (root == null)
            return false;
        // Jika data root sama dengan kunci, elemen ditemukan
        if (root.data == key)
            return true;
        // Jika kunci lebih kecil dari data root, cari di sub-pohon kiri
        return key < root.data ? searchRec(root.left, key) : searchRec(root.right, key);
    }

    // Metode untuk menampilkan elemen pohon dalam urutan inorder
    public void inorder() {
        inorderRec(root);
    }

    // Metode rekursif untuk menampilkan elemen dalam urutan inorder
    private void inorderRec(Node root) {
        if (root != null) {
            inorderRec(root.left);
            System.out.print(root.data + " ");
```

```java
    // Metode untuk menampilkan elemen pohon dalam urutan preorder
    public void preorder() {
        preorderRec(root);
    }

    // Metode rekursif untuk menampilkan elemen dalam urutan preorder
    private void preorderRec(Node root) {
        if (root != null) {
            System.out.print(root.data + " ");
            preorderRec(root.left);
            preorderRec(root.right);
        }
    }

    // Metode untuk menampilkan elemen pohon dalam urutan postorder
    public void postorder() {
        postorderRec(root);
    }

    // Metode rekursif untuk menampilkan elemen dalam urutan postorder
    private void postorderRec(Node root) {
        if (root != null) {
            postorderRec(root.left);
            postorderRec(root.right);
            System.out.print(root.data + " ");
        }
    }
}
```

Java: Ready  Ln 91, Col 1  Spaces: 4  UTF-8  CRLF  {} Java

Gambar BST