LAPORAN PRAKTIKUM STRUKTUR DATA



OLEH :

ALIFFIA HUMAIRAH

NIM : 2311531004

DOSEN PENGAMPU :

DR. WAHYUDI,S.T,MT.

DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

PADANG

1. TUJUAN
2. Mahasiswa dapat memahami penggunaan Node dan Tree.
3. Mahasiswa dapat memahami cara kerja Node dan Tree pada Java.
4. Mahasiswa dapat membuat program menggunaan Node dan Tree.
5. KAJIAN TEORI
6. **Node**

**Node (Simpul)** adalah elemen dasar pembentuk struktur data pohon. Ia berfungsi seperti wadah yang menyimpan informasi dan petunjuk ke bagian lain dalam pohon.

**Komponen Penting Node:**

* **Data:** Informasi aktual yang disimpan dalam node. Ini bisa berupa angka, karakter, teks, atau objek lain tergantung pada kebutuhan program.
* **Referensi (Pointer):** Petunjuk ke node lain. Bergantung pada jenis pohon, sebuah node mungkin memiliki:
  + **Tidak ada referensi:** Jika node merupakan leaf node (node daun) yang tidak memiliki anak.
  + **Satu referensi:** Misalnya pada node anak tunggal di pohon linear.
  + **Dua referensi:** Seperti pada pohon biner yang memiliki referensi ke anak kiri dan anak kanan.
  + **Lebih dari dua referensi:** Pada pohon N-ary, di mana N bisa bernilai 3, 4, dan seterusnya.

Node atau simpul adalah elemen dasar dalam struktur data pohon. Node merepresentasikan item data individual di dalam pohon. Biasanya, node memiliki dua komponen:

* **Nilai data:** Informasi aktual yang disimpan dalam node.
* **Referensi (pointer):** Tautan ke node lain, membangun hubungan hierarki dalam pohon.
  + Referensi ini dapat bervariasi tergantung jenis pohon (misalnya, binary tree memiliki pointer ke child node kiri dan kanan).

1. **Pohon**

**Pohon** merupakan struktur data fundamental dalam ilmu komputer yang digunakan untuk merepresentasikan hubungan hierarki dan mengorganisir data secara efisien. Pohon sering digunakan untuk mengimplementasikan algoritma pencarian, pengurutan, dan operasi lainnya.

**Struktur Pohon**

Sebuah pohon terdiri dari **node** (simpul) yang terhubung oleh **hubungan** (edge). Setiap node dapat memiliki nol atau lebih **node anak** (child node). Node teratas disebut **node akar** (root node). Node yang tidak memiliki node anak disebut **node daun** (leaf node).

**Jenis-jenis Pohon**

Ada banyak jenis pohon, tetapi beberapa yang paling umum meliputi:

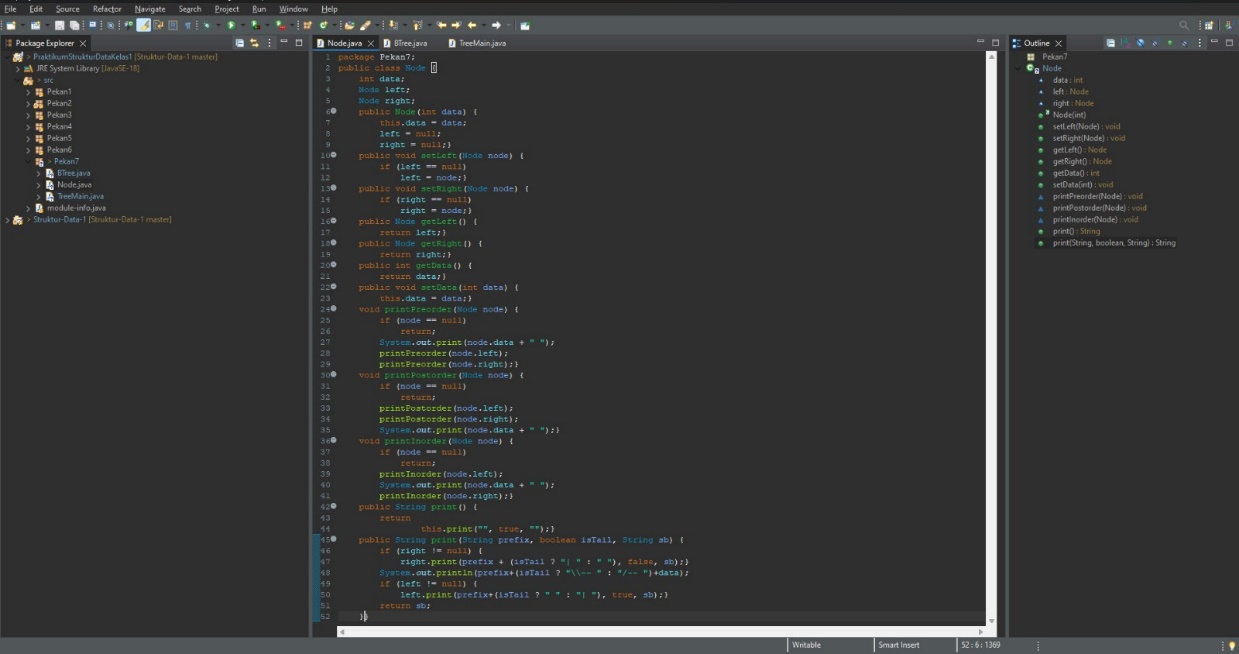
* **Pohon Biner** (Binary Tree): Setiap node maksimal memiliki dua node anak, yaitu **anak kiri** (left child) dan **anak kanan** (right child).
* **Pohon Pencarian Biner** (Binary Search Tree, BST): Jenis khusus pohon biner di mana data pada setiap node lebih besar dari semua data pada sub-pohon sebelah kirinya dan lebih kecil dari semua data pada sub-pohon sebelah kanannya. Ini membuatnya efisien untuk pencarian.
* **Pohon N-ary** (N-ary Tree): Setiap node dapat memiliki nol atau lebih node anak.

Struktur data hierarki yang terinspirasi dari pohon di dunia nyata. Terdiri dari node yang terhubung dengan edge (referensi). Ciri-ciri utama:

* **Node akar:** Node teratas, tanpa node induk. Merupakan titik awal untuk menjelajahi pohon.
* **Node induk:** Node yang memiliki satu atau lebih child node di bawahnya.
* **Child node:** Node yang memiliki node induk di atasnya. Dapat memiliki zero atau lebih child node itu sendiri.
* **Leaf node (terminal node):** Node tanpa child node. Mewakili akhir cabang dalam pohon.

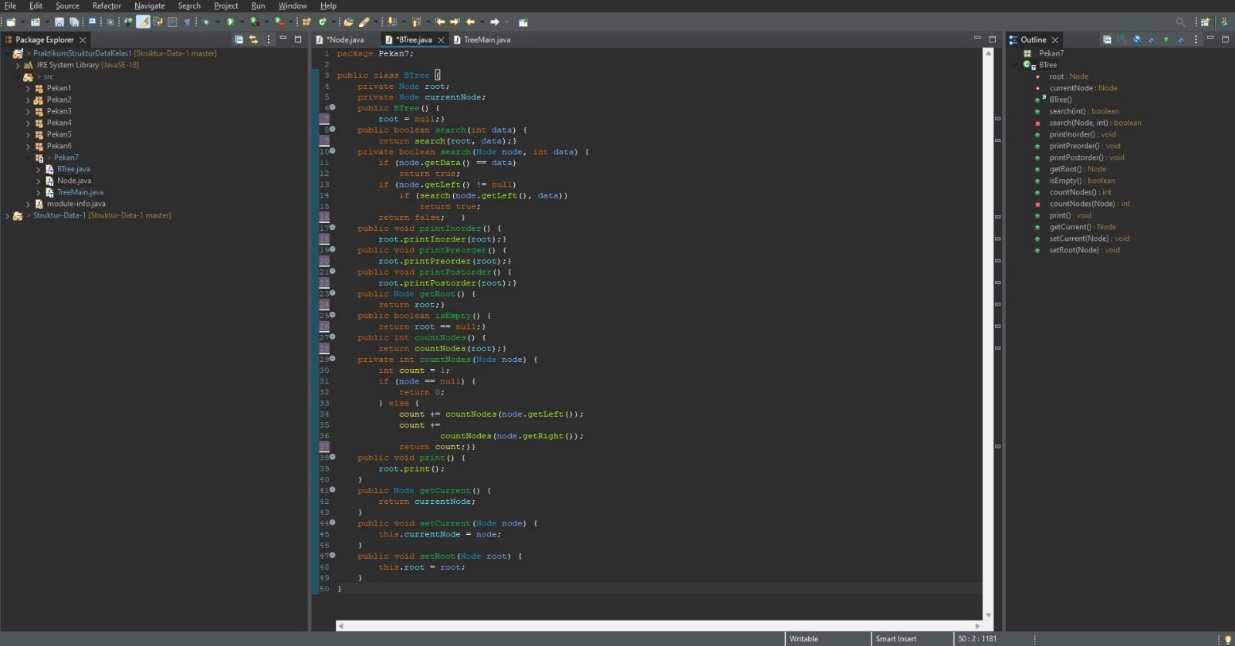
1. HASIL KERJA
   1. Node

Input:



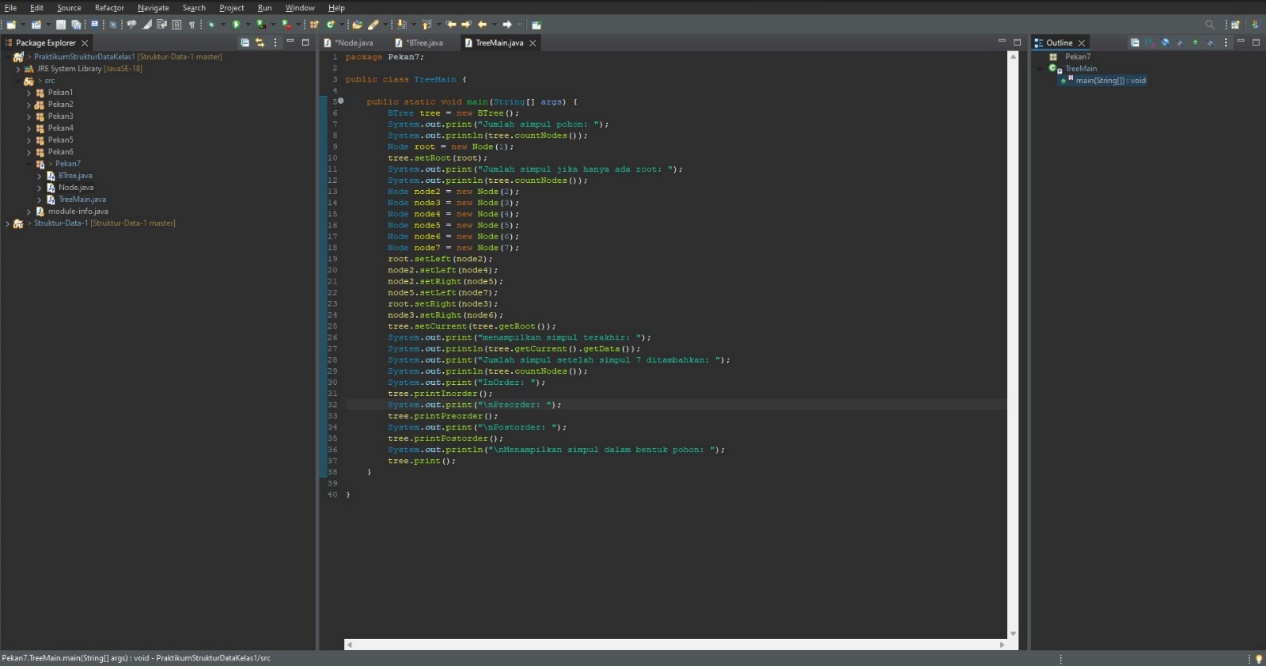
* 1. Btree

Input:



* 1. Tree main

Input:



Output:

