Tugas Pendahuluan Modul 9

STRUKTUR DATA - Ganjil 2024/2025

Tree

Ketentuan Tugas Pendahuluan

1. Tugas Pendahuluan dikerjakan secara Individu.
2. TP ini bersifat WAJIB, tidak mengerjakan = PENGURANGAN POIN JURNAL / TES ASESMEN.
3. Hanya MENGUMPULKAN tetapi TIDAK MENGERJAKAN = PENGURANGAN POIN JURNAL

/ TES ASESMEN.

1. Deadline pengumpulan TP Modul 2 adalah Senin, 30 September 2024 pukul 07.30 WIB.
2. TIDAK ADA TOLERANSI KETERLAMBATAN, TERLAMBAT ATAU TIDAK MENGUMPULKAN TP MAKA DIANGGAP TIDAK MENGERJAKAN.
3. DILARANG PLAGIAT (PLAGIAT = E).
4. Kerjakan TP dengan jelas agar dapat dimengerti.
5. Codingan diupload di Github dan upload Laporan di Lab menggunakan format PDF dengan ketentuan: TP\_MOD\_[XX]\_NIM\_NAMA.pdf

CP (WA):

* Andini (082243700965)
* Imelda (082135374187)

**SELAMAT MENGERJAKAN^^**

**LAPORAN PRAKTIKUM**

**PERTEMUAN 9**

**TREE**

A black and red logo

Description automatically generated

**Nama :**

Tsaqif Hisyam Saputra (2311104024)

**Dosen :**

Yudha Islami Sulistya

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

1. Tujuan

* Mahasiswa mampu menjelaskan konsep dari struktur data Tree (Binary Tree)
* Mahasiswa mampu membuat program dengan menerapkan struktur data Tree (Binary Tree)
* Mahasiswa mampu membuat fungsi tambah simpul dan harus simpul pada struktur data Tree (Binary Tree)

1. Tools   
   Visual Studio Code dengan C++ Extensions Pack & Codeblocks
2. Latihan – Unguided

A screen shot of a computer

Description automatically generatedA screen shot of a computer screen

Description automatically generated

Program diatas adalam implementasi dari pohon biner, program dimulai dengan mendeklarasikan struktur Pohon untuk merepresentasikan simpul pada pohon biner. Struktur ini memiliki empat anggota **data** (nilai simpul), **left** (child kiri), **right** (child kanan), dan **parent** (simpul induk).

Dalam program tersebut memiliki inisialisasi Pohon dengan fungsi **init()** yang menginisialisasi pohon dengan mengatur **root** menjadi **NULL**, menandaakan pohon kosong.

A black background with white text

Description automatically generated

Fungsi **buatNode** membuat simpul root jika pohon kosong. Fungsi **insertLeft** dan **insertRight** memungkinkan penambahan child kiri dan kanan pada simpul tertentu, dengan melakukan pengecekan agar simpul tersebut belum memiliki child di posisi yang di maksud.

A computer screen with text

Description automatically generated

A screen shot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated

Untuk memanipulasi data simpul memakai fungsi **update** untuk mengubah nilai data dari simpul yang ditunjuk, fungsi **retrieve** yang menampilkan data simpul trtentu, dan fungsi **find** yang menampilkan data simpul beserta informasi terkait seperti parent dan sibling, serta menangani simpul yang mungkin tidak memiliki parent atau sibling.

A screen shot of a computer program

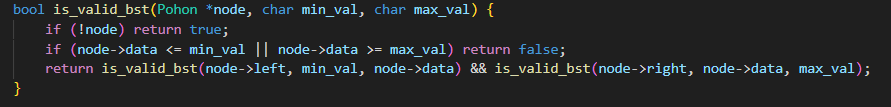
Description automatically generated

Lalu untuk menmpilkan descendant memakai fungsi **showDescendants** menampilkan semua turunan dari simpul yang dipilih menggunakan rekursi.

A screen shot of a computer code

Description automatically generated

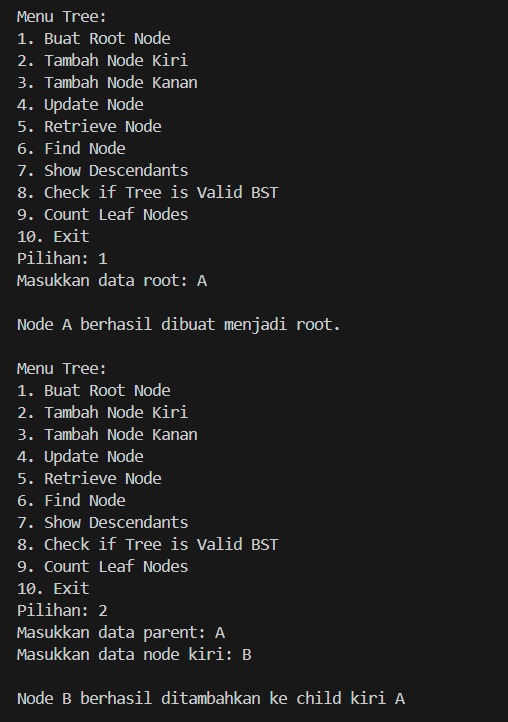
Untuk mengvalidasi BST memakai **is\_valid\_bst** yang merupakan fungsi rekursif yang memeriksa apakah pohon memenuhii properti BST, di mana setiap simpul kiri harus memiliki nilai lebih kecil dan setiap simpul kanan harus lebih besar dari simpul induk. Fungsi ini mengembalikan true jika pohon valid sebagai BST, atau false jika tidak.

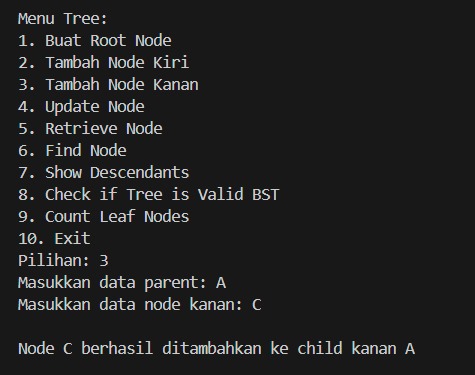


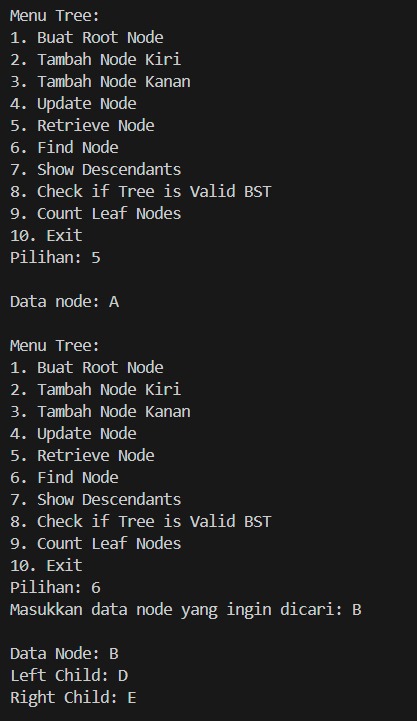
Menhitung simpul daun memerlukan fungsi **cari\_simpul\_daun**  menghitung jumlah simpul daun dalam pohon. Smpul daun adalah simpul yang tidak memiliki anak kiri maupun kanan.

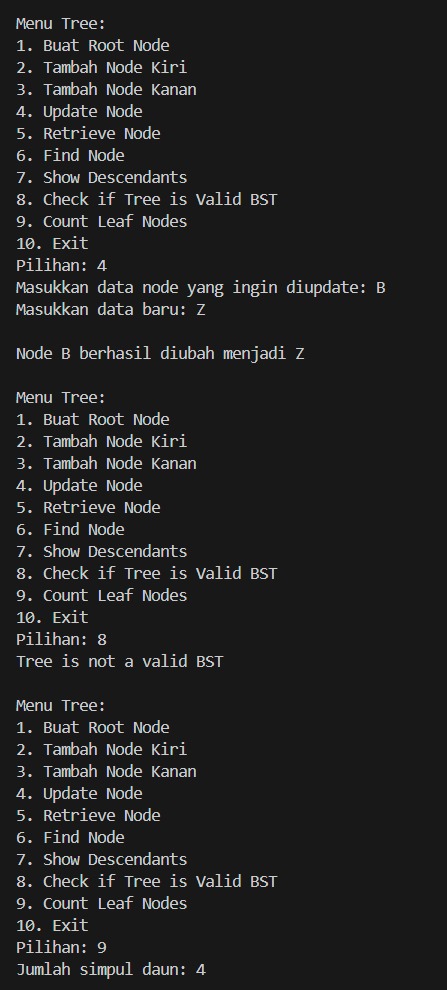
A computer code on a black background

Description automatically generated

Output:  








**Semoga Selalu diberi kemudahan^^**