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.
  4. Deadline pengumpulan TP Modul 2 adalah Senin, 30 September 2024 pukul 07.30 WIB.
  5. TIDAK ADA TOLERANSI KETERLAMBATAN, TERLAMBAT ATAU

TIDAK MENGUMPULKAN TP MAKA DIANGGAP TIDAK MENGERJAKAN.

* 1. DILARANG PLAGIAT (PLAGIAT = E).
  2. Kerjakan TP dengan jelas agar dapat dimengerti.
  3. 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

# STRUKTUR DATA



## Nama :

Zulfa Mustafa Akhyar Iswahyudi (2311104010)

## Dosen :

Yudha Islami Sulistya

# PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

# FAKULTAS INFORMATIKA

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

1. Tujuan

Untuk melatih kompetensi Mahasiswa untuk memperdalam skill pemrograman C++

1. Tools

Codeblocks, VSCode, Github

**TUGAS PENDAHULUAN - UNGUIDED**

-

**LATIHAN – UNGUIDED**

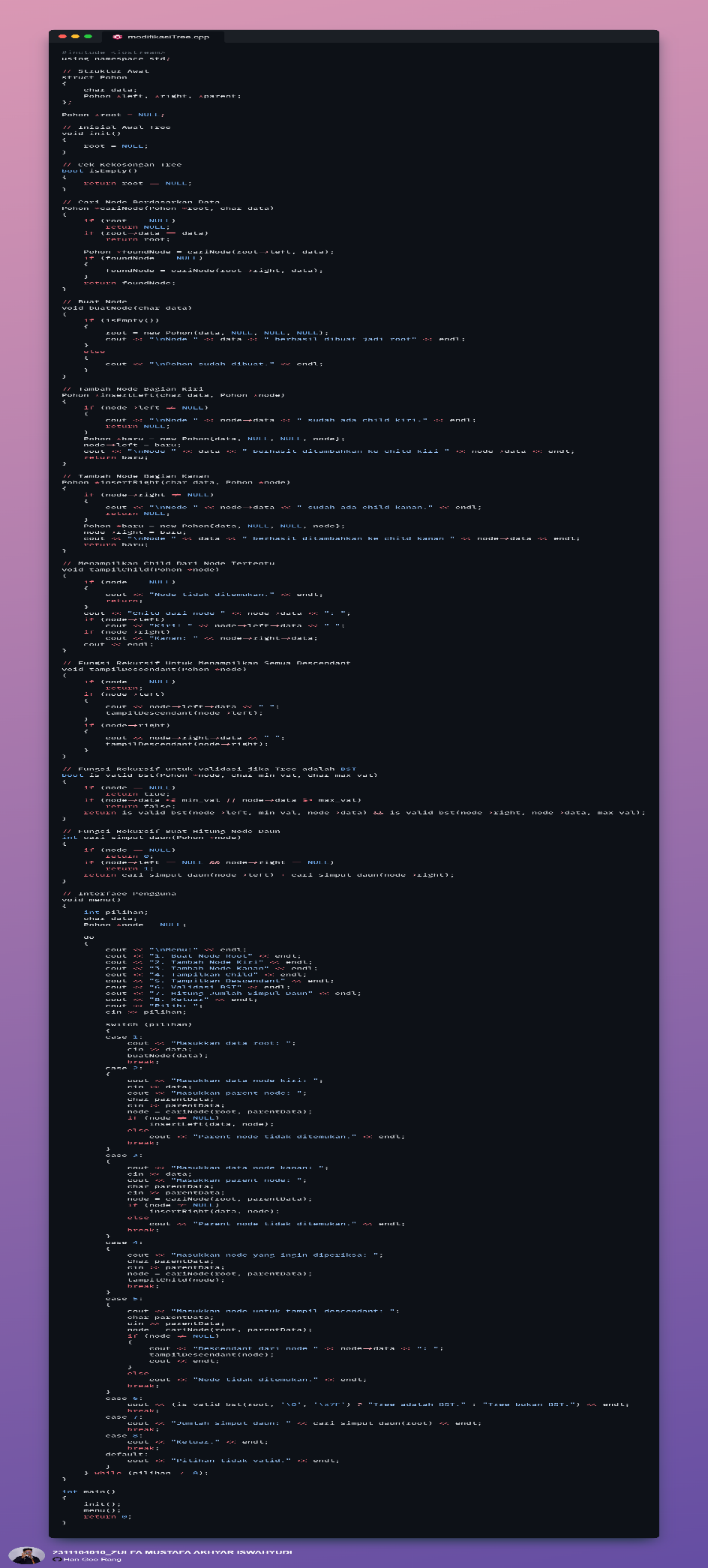
Tiga Soal unguided ini :

1. Modifikasi guided tree diatas dengan program menu menggunakan input data tree dari user dan berikan fungsi tambahan untuk menampilkan node child dan descendant dari node yang diinputkan!

2. Buatlah fungsi rekursif is\_valid\_bst(node, min\_val, max\_val) untuk memeriksa apakah suatu pohon memenuhi properti Binary Search Tree. Uji fungsi ini pada berbagai pohon, baik yang valid maupun tidak valid sebagai BST.

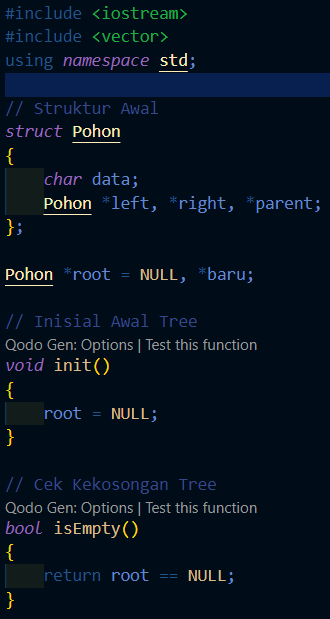
3. Buatlah fungsi rekursif cari\_simpul\_daun(node) untuk menghitung jumlah simpul daun dalam Binary Tree. Simpul daun adalah node yang tidak memiliki anak kiri maupun kanan.

Kita jadikan jadi satu program yang sudah dimodifikasi:



Kita bedah satu per satu dalam penjelasan kali ini :

1. Struktur Awal



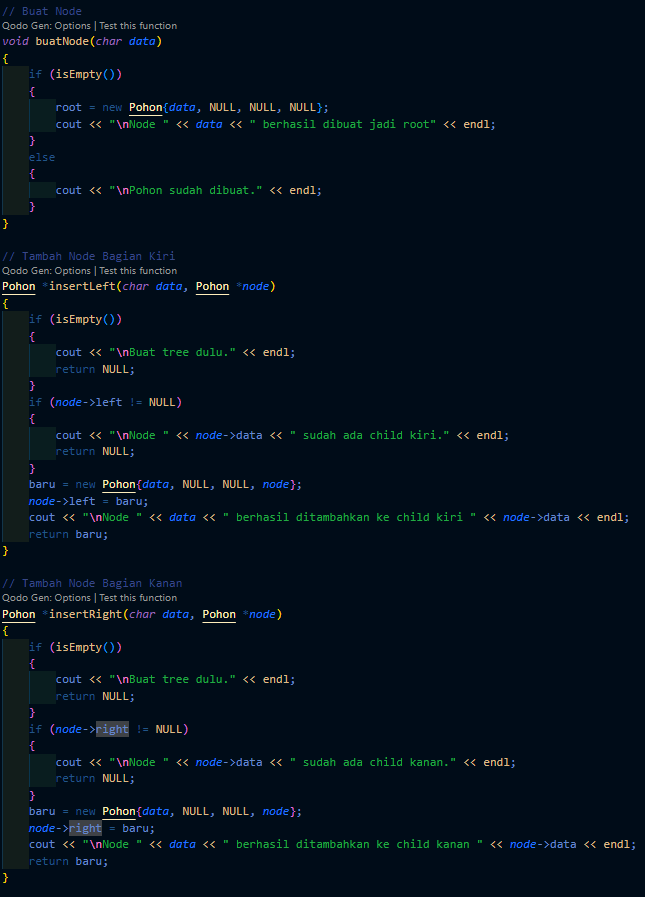
Mulai dari Struct sampai init adalah proses untuk membuat struktural mentah dari program Tree dengan deklarasi ‚data‘ dengan tipe data char dan deklarasi Tree beserta cabang-cabang seperti \*Rigth, \*Left, dan \*Parent.

Atur juga bahwa \*root sebagai NULL yang menandakan bahwa Tree masih kosong. Buat method untuk menentukan True/False untuk mengecek apakah Tree kosong atau tidak.



Jangan lupa juga untuk menambahkan method cariNode dengan inisial Tree ‘Pohon‘ untuk mendeteksi seluruh node pada Tree untuk ditemukan nilai yang sesuai dengan inputan yang dicari User.

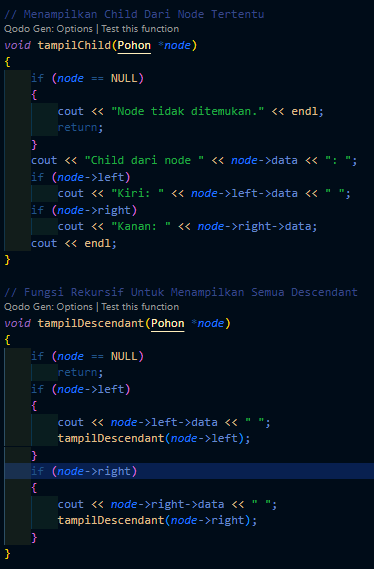
1. Node



Selanjutnya adalah method-method dengan deklarasi ‘Pohon‘ sebagai konstanta Tree yang sudah dibuat. Satu untuk create, satu untuk membuat Node pada bagian kiri, satu untuk membuat Node pada bagian kanan. Untuk masing-masing method diisikan fungsi isEmpty agar Tree di cek setiap saat ingin melakukan manipulasi Node pada Tree.

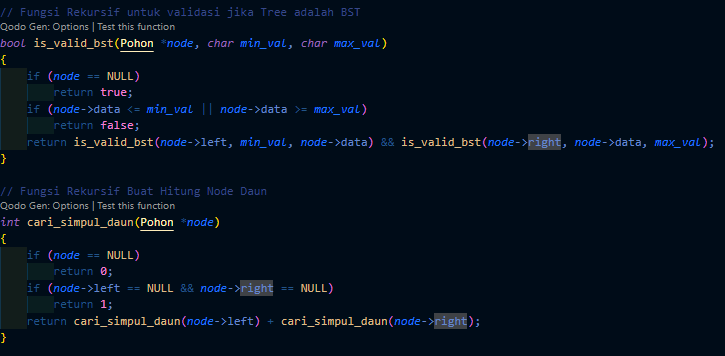
Pengkondisian juga dilakukan untuk menambahkan, mengurangi, atau membuat node baru agar sistem dapat memproses pembaruan Tree.

1. Display Output



Selanjutnya adalah method-method untuk menampilkan data node dari Tree. Disini terbagi menjadi dua yang dimana kedua method untuk mencari child maupun descendant memiliki struktur sintaks kode yang mirip yaitu pengkondisian untuk mengecek seluruh node dari bagian kiri maupun kanan dan menampilkan data-data node yang tertangkap oleh pengkondisian.

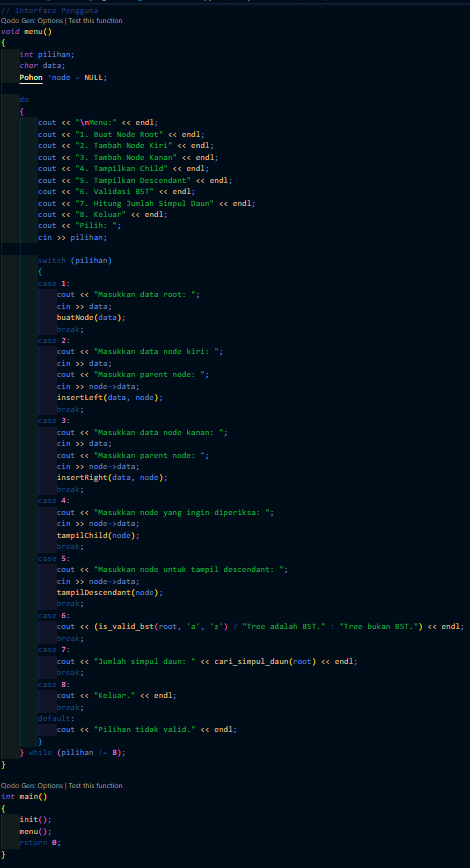
1. Recursive Function



Selanjutnya adalah method boolean dan method integer yang punya fungsi berbeda. Method **is\_valid\_bst** bertujuan untuk melakukan validasi seluruh node pada Tree. Didalamnya kita menggunakan pengkondisian dengan pipeline agar node paling kecil dan node paling besar dapat divalidasi untuk setiap node bagian kiri dan node bagian kanan.

Sedangkan untuk method **cari\_simpul\_daun** bertujuan untuk menjumlahkan seluruh node pada Tree. Disini kita mengecek seluruh node dari kiri maupun node dari kanan agar dapat dijumlahkan. Jangan lupa untuk membuat pengkondisian yang mengecek kekosongan node pada Tree.

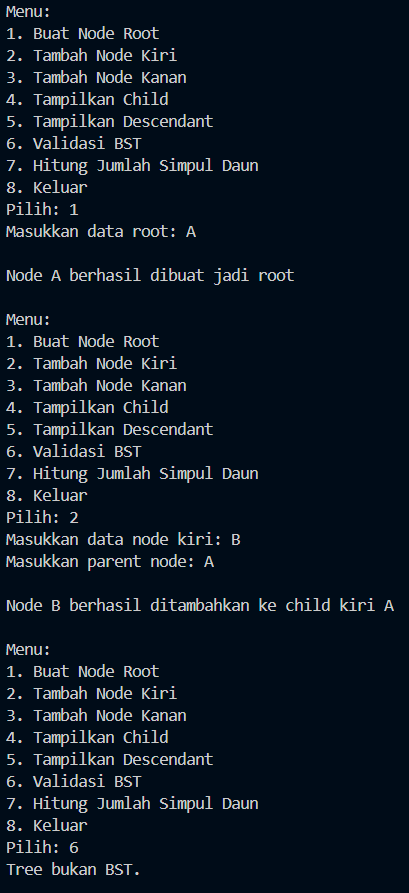
1. Interface Menu

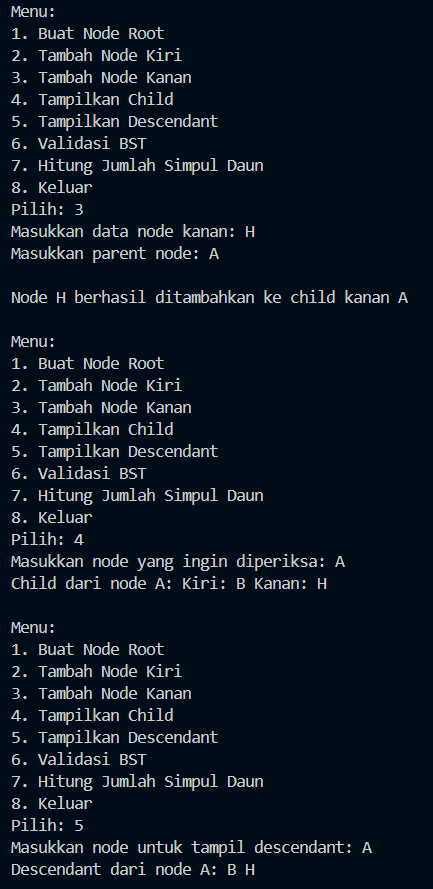


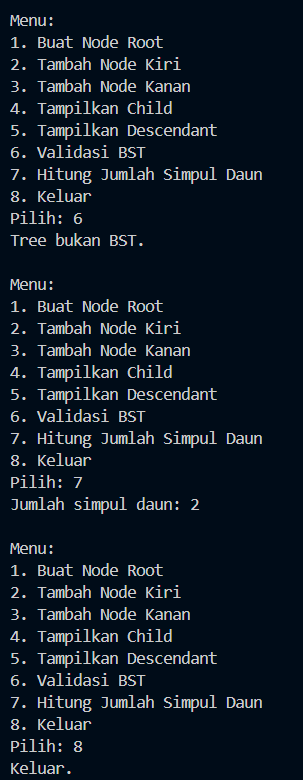
Kita masuk ke Main Program. Main Program ini berisi menu output untuk melakukan manipulasi Tree beserta inputan dari User pada Terminal. Kita deklarasikan dulu agar inputan dari User berupa angka (integer) dengan variabel ‘pilihan‘, lalu data pada node adalah tipe data ‘char‘, dan deklarasi Tree ‘Pohon‘ dan \*node agar di set default kosong.

Kemudian baru kita memasuki menu-menu untuk membuat dan memanipulasi Tree. Percabangannya kita pakai ‘Switch-case’. Jadi untuk setiap menu akan diberikan operasi program yang berbeda sehingga program Tree dapat terimplementasi secara sempurna.

Output :







**Semoga Selalu diberi kemudahan^^**