

Topik: ADT Pohon Biner

Tujuan Praktikum:

- Memahami konsep Pohon Biner.
- Memahami implementasinya dalam bahasa C.

PETUNJUK PRAKTIKUM:

1. Untuk setiap file yang Anda buat, cantumkan header sebagai berikut:

<pre>/* NIM>Nama : Nama file : Topik : Tanggal : Deskripsi : */</pre>
--


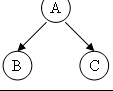

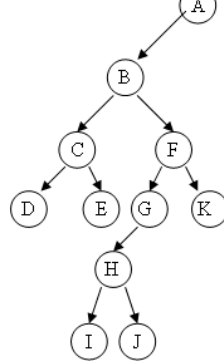
2. Seluruh file di-upload setelah dikompres menjadi 1 file dengan nama: P11<nim>.zip.
3. Softcopy materi kuliah/diktat, termasuk yang terkait dengan pemrograman dengan Bahasa C dapat dilihat pada situs <http://kuliah.itb.ac.id/app245/course/> pada link **IF2110 Algorithm & Data Structure**.
4. HANYA ADT YANG DAPAT DI-COMPILE YANG AKAN DIPERIKSA. File yang tidak dapat di-*compile* akan otomatis mendapatkan nilai 0.
5. Tugas ini bersifat INDIVIDUAL. Anda dipersilakan membaca dan menggunakan material yang Anda punya, tetapi tidak ada toleransi bagi pencontek. Jika terbukti, baik yang dicontek maupun yang mencontek akan mendapatkan nilai 0.
6. Ikuti petunjuk asisten untuk pengumpulan tugas ini.

SELAMAT BEKERJA.

Soal 1. Pohon Biner (Bobot : 100%)

- Ambillah **ADT Pohon Biner** yang telah Anda kerjakan sebagai tugas pra-praktikum.
- Pastikan type infotype menjadi **character**. Asumsikan bahwa pohon biner adalah pohon dengan elemen-elemen yang **unik**.
- Buatlah sebuah **program utama** yang memanfaatkan ADT Pohon Biner of character tersebut yang melakukan hal-hal sebagai berikut:
 - Membaca isi sebuah file eksternal yang berisi pohon biner dengan representasi linier dengan format yang diajarkan di kuliah dengan menggunakan procedure **BuildTree (modifikasilah bila berbeda)**. Gunakan algoritma untuk **membangun pohon biner secara rekursif** dengan memanfaatkan mesin karakter. *Input* akan di-*supply* melalui stdin.

Contoh isi file dan pohon biner yang direpresentasikannya:

Input	Merepresentasikan pohon biner sbb:
() .	Pohon kosong
(A()) .	
(A(B())(C())) .	
(A(B())()) .	
(A(B(C(D())(E()))(F(G(H(I()))(J())))))(K()) .	

Format Input

- Sebuah ekspresi untuk membangun pohon P (diakhiri titik).
- Sebuah masukan X (bertype infotype).

Format Ouput

- Baris pertama berisi banyaknya elemen dalam pohon P
- Baris kedua berisi semua elemen dari P secara **preorder** (jika P tidak kosong).
- Baris ketiga berisi list semua elemen daun dari P dari kiri ke kanan (jika P tidak kosong). Jika P kosong, tuliskan newline.
- Baris keempat berisi apakah P adalah pohon seimbang atau bukan, keluarkan "true" bila merupakan pohon seimbang dan "false" bila bukan. Pohon kosong dianggap seimbang. Definisi: pohon seimbang adalah pohon yang memiliki beda jumlah seluruh node pada anak kiri dan kanan maksimum satu. Masing-masing subtree dari pohon seimbang merupakan pohon seimbang.
- Baris kelima berisi apakah X adalah daun atau bukan, keluarkan "true" bila daun dan "false" bila bukan.
- Baris keenam berisi tinggi/level X di pohon P. Outputkan -1 bila X tidak ada pada P.

- Baris ketujuh berisi saudara (*sibling*) dari X, yaitu elemen yang memiliki *parent* yang sama dengan X (jika ada). Sebuah elemen maksimum hanya memiliki 1 saudara. Bila tidak ada, keluarkan newline
- Baris kedelapan berisi list semua elemen yang memiliki level yang sama dengan X dari kiri ke kanan. X termasuk yang ditulis. Bila X tidak ada pada pohon, maka keluarkan newline.
- Baris kesembilan berisi elemen pohon biner P terbaru secara **preorder**. Definisi pohon biner terbaru: jika X tidak ada di P, maka tambahkan X sebagai elemen daun terkiri pohon P, jika X sudah ada maka outputkan pohon biner apa adanya.

Contoh Input-Output

Input	Output
() . A	0 () true false -1 (A () ())
(A(B () ()) (C () ())) . B	3 (A (B () ()) (C () ())) B, C true true 2 C B, C (A (B () ()) (C () ()))
(A(B () ()) ()) . C	2 (A (B () ()) ()) B true false -1 (A (B (C () ()) ()) ())
(A(B(D () ()) (E () ())) (C(F () ()) (G () ()))) . F	7 (A (B (D () ()) (E () ())) (C (F () ()) (G () ()))) D, E, F, G true true 3 G D, E, F, G (A (B (D () ()) (E () ())) (C (F () ()) (G () ())))

Kumpulkanlah file:

listrekursif.h, mesinkar.h, pohon.h,

listrekursih.c, mesinkar.c, pohon.c,

serta mpohon.c dalam 1 file berekstensi *.zip sesuai standar penamaan praktikum