





Struktur Data

Saniati, S.ST., M.T.

EPISODE 8P

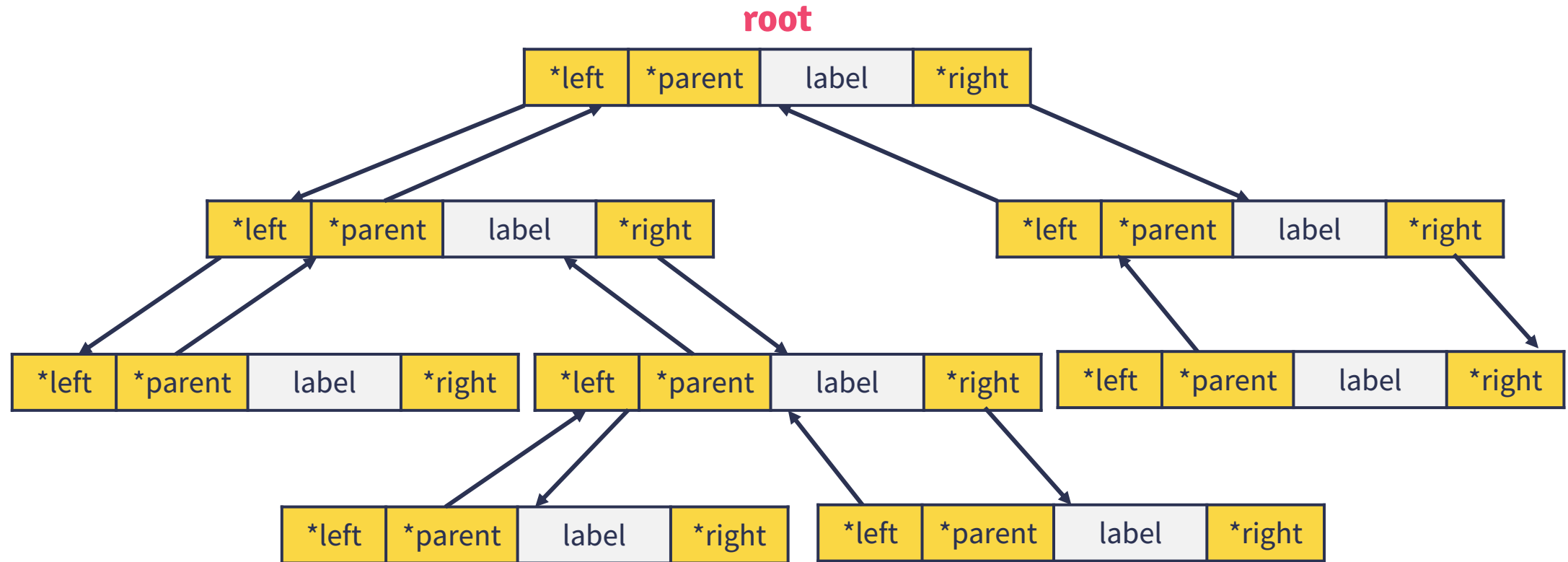
Implementasi Binary Tree C++

Pra-Syarat & Materi yang Dipakai ?

- **Dasar Pemrograman C++**
- **Fungsi pada C++**
- **Pointer**
- **Struct (struktur)**
- **Linked List**



Konsep Implementasi Binary Tree ?



- Masing-masing node dibuat menggunakan linked list.
- Terdapat pointer left (menunjuk ke anak kiri), right (menunjuk ke anak kanan) dan parent (menunjuk ke node orang tua).
- Pointer parent pada node root NULL karena root tidak memiliki orang tua.
- Pointer parent dapat membantu proses saat mengetahui parent dan sibling.

Operasi pada Tree ?

- **Create** : digunakan untuk membentuk binary tree baru yang masih kosong.
- **Clear** : digunakan untuk mengosongkan binary tree yang sudah ada atau menghapus semua node pada binary tree.
- **Empty** : digunakan untuk memeriksa apakah binary tree masih kosong atau tidak.
- **Insert** : digunakan untuk memasukkan sebuah node kedalam tree.
- **Find** : digunakan untuk mencari root, parent, left child, atau right child dari suatu node dengan syarat tree tidak boleh kosong.
- **Update** : digunakan untuk mengubah isi dari node yang ditunjuk oleh pointer current dengan syarat tree tidak boleh kosong.
- **Retrieve** : digunakan untuk mengetahui isi dari node yang ditunjuk pointer current dengan syarat tree tidak boleh kosong.
- **Delete Sub** : digunakan untuk menghapus sebuah subtree (node beserta seluruh descendant-nya) yang ditunjuk pointer current dengan syarat tree tidak boleh kosong.
- **Charateristic** : digunakan untuk mengetahui karakteristik dari suatu tree. Yakni size, height, serta average lenght-nya.
- **Tranverse** : digunakan untuk mengunjungi seluruh node-node pada tree dengan cara tranversal.



Cara Tranversal ?

preOrder

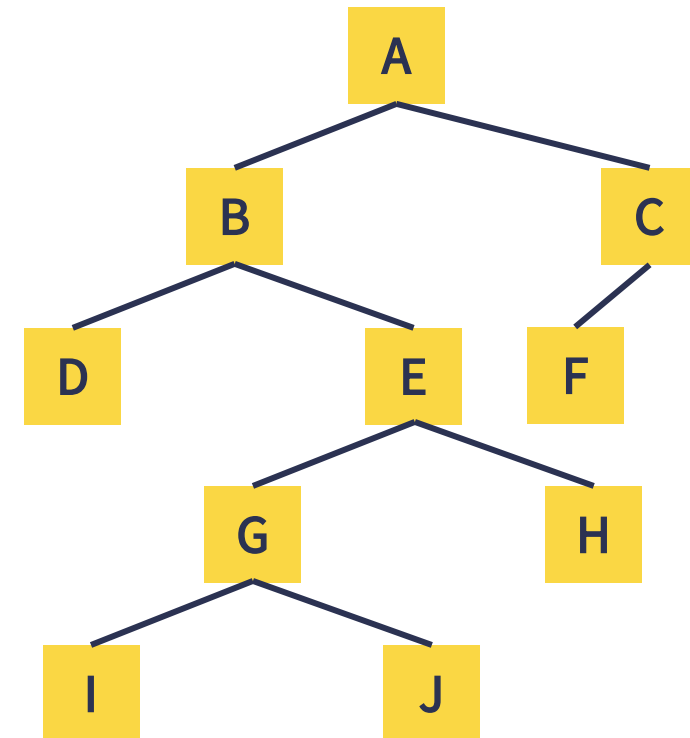
1. Kunjungi root nya.
2. Telusuri subtree kiri.
3. Telusuri subtree kanan.

inOrder

1. Telusuri subtree kiri.
2. Kunjungi root nya.
3. Telusuri subtree kanan.

postOrder

1. Telusuri subtree kiri.
2. Telusuri subtree kanan.
3. Kunjungi root nya.



preOrder = **A,B,D,E,G,I,J,H,C,F**

inOrder = **D,B,I,G,J,E,H,A,F,C**

postOrder = **D,I,J,G,H,E,B,F,C,A**



Video Selanjutnya

Konsep Graph



Thank you

**#KEEPLARNING
#KEEPSPIRITS**



