

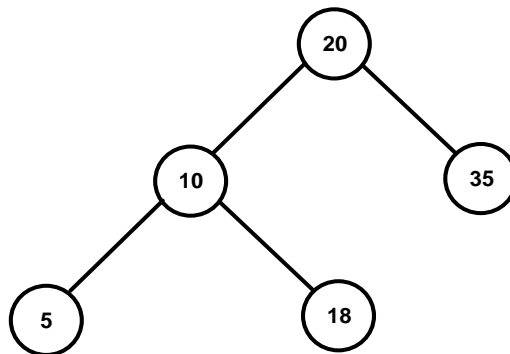
Modul 11 TREE (BAGIAN KEDUA)

TUJUAN PRAKTIKUM

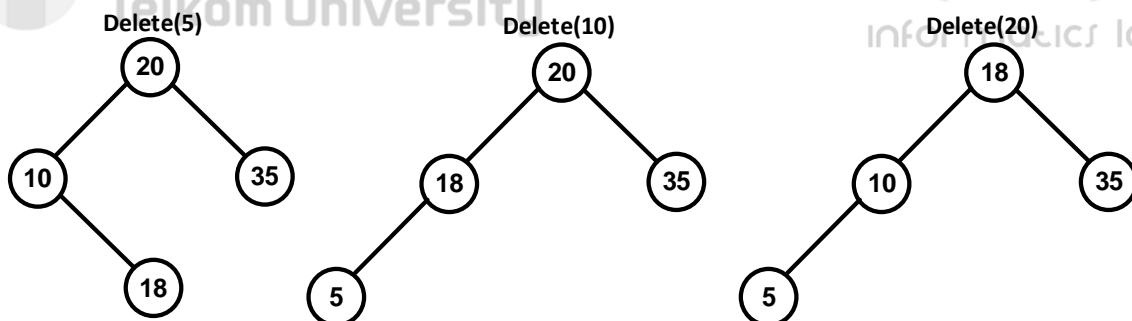
1. Mengimplementasikan struktur data *tree*, khususnya *Binary Tree*.

A. Delete

1. *LEAF*, tidak perlu dilakukan modifikasi.
2. *Node* dengan 1 *Child*, maka *child* langsung menggantikan posisi *Parent*.
3. *Node* dengan 2 *Child*:
 - *Left Subtree*, yang diambil adalah *node* yang paling kiri (nilai terbesar).
 - *Right Subtree*, yang diambil adalah *node* yang paling kanan (nilai terkecil).



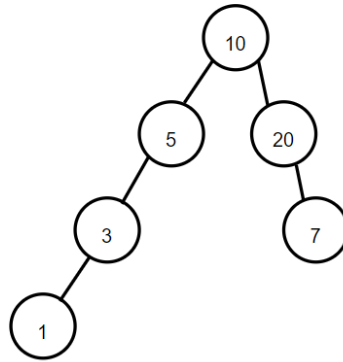
Gambar 11-1 Binary Search Tree sebelum di-Delete



Gambar 11-2 Binary Search Tree setelah di-Delete

B. Most-Left

Most-left node adalah node yang berada paling kiri dalam *tree*. Dalam konteks *binary search tree* (BST), *most-left node* adalah *node* dengan nilai terkecil, yang dapat ditemukan dengan mengikuti anak kiri (*left child*) dari root hingga mencapai node yang tidak memiliki anak kiri lagi.

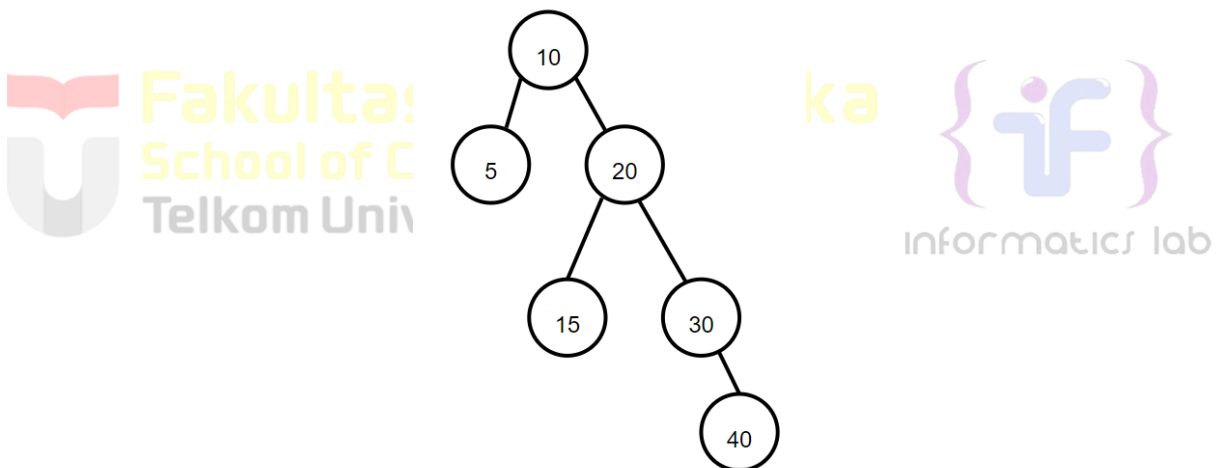


Gambar 11-3 *most-left tree*

Pada *tree* diatas, *most-left tree* adalah = 1

C. **Most-Right**

Most-right node adalah node yang berada paling kanan dalam *tree*. Dalam konteks *binary search tree* (BST), *most-right node* adalah *node* dengan nilai terbesar, yang dapat ditemukan dengan mengikuti anak kanan (*right child*) dari root hingga mencapai node yang tidak memiliki anak kanan lagi.



Gambar 11-4 *most-right tree*

Pada *tree* diatas, *most-right tree* adalah = 40