TUGAS PERTEMUAN 9

STRUKTUR DATA

TREE & BINARY TREE

Disusun oleh:

Rama Pramudya Wibisana 2022320019

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS INFORMATIKA

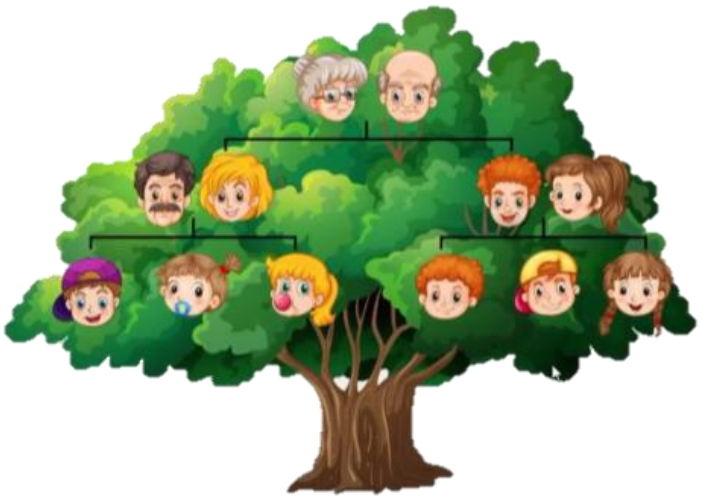
UNIVERSITAS BINA INSANI

BEKASI

2022

A. Apa itu Tree?

Berbicara soal Tree, kita mungkin tanpa sadar sudah mengenal konsep ini dalam kehidupan sehari hari, salah satunya adalah hirarki dalam keluarga, perhatikan contoh gambar di bawah :



Di level pertama ada Kakek dan Nenek, lalu selanjutnya ada Orang Tua, dan yang paling bawah adalah anak-anak, ini merupakan contoh simple dari Hirarki.

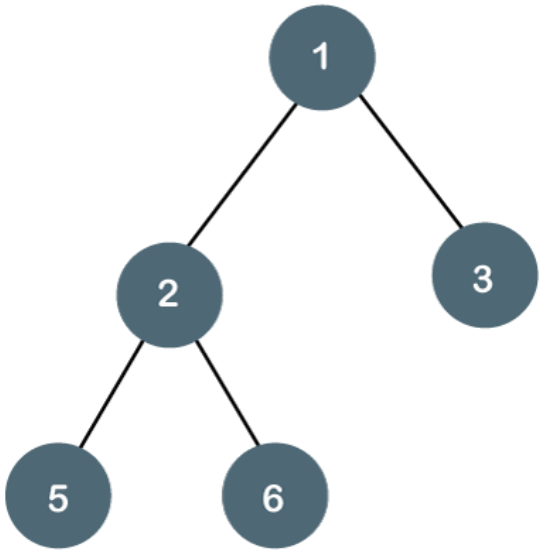
Tidak jauh berbeda dengan ini, Tree dalam Bahasa pemrograman (khususnya c++) juga memiliki konsep yang sama, namun hanya saja tentu memiliki istilah-istilah sendiri dalam penyebutannya.

Dalam c++, istilah itu disebut Node, yap, Kakek Nenek, Orang Tua dan seterusnya itu disebut dengan Node. Nah langsung saja ke penjelasan mengenai Tree dalam konteks pemrograman (C++).

B. Penerapan Tree & Binary Tree Dalam Pemrograman (C++)

Ada istilah baru di sini, Binary Tree, yaitu Tree yang tidak boleh memiliki lebih dari 2 node dalam 1 parent. Simplenya adalah, orang tua tidak boleh memiliki lebih dari 2 anak (Program KB).

Untuk mengetahui lebih jelas, perhatikan contoh simple pada Binary Tree di bawah :



Sekarang mari kita bandingkan dengan hirarki keluarga pada gambar pertama, sama persis bukan strukturnya? Nah dalam pemrograman, setiap lingkaran di atas disebut node. Node memiliki nilai, angka yang ada di dalam node merupakan nilai dari node tersebut.

a. Level 1

   Node pertama (1) yang ada di paling atas disebut root. Root itu artinya sebuah Node yang tidak memiliki parent. Bisa dilihat dengan tidak ada lagi node di atasnya.

b. Level 2

   Node yang ada di level ini disebut Parent (2) & (3). Parent itu ciri-cirinya, dia pasti memiliki node lagi di atasnya (root) dan memiliki anak yang maksimal jumlahnya adalah 2.

c. Level 3

Node di level terakhir sebutannya bisa ada dua, ada leaf Node, atau children (5) & (6). Tapi saya lebih suka menyebutnya leaf node, node ini mempunyai ciri-ciri yaitu tidak ada lagi node setelahnya, artinya hirarki berhenti di node tersebut.

Penjelasan di atas merupakan bagaimana Tree dalam bahasa pemrograman, lalu selanjutnya adalah penerapannya dalam bahasa pemrograman (C++).

C. Syntax

strukturdata_p8 > C++ binarytree.cpp > Node

```cpp
#include <iostream>

using namespace std;

struct Node
{
    int data;
    Node *left;
    Node *right;
};

Node *createNode(int data)
{
    Node *newNode = new Node;
    newNode->data = data;
    newNode->left = newNode->right = nullptr;
    return newNode;
}

int main()
{
    Node *root = createNode(1);
    root->left = createNode(2);
    root->right = createNode(3);
    root->left->left = createNode(5);
    root->left->right = createNode(6);
}
```

Membuat node di C++ memerlukan struct, maka hal pertama yang dilakukan adalah membuat struct dengan nama **Node**.

Selanjutnya membuat variable data dengan tipe integer, untuk menyimpan node nantinya. Dilanjut membuat struct dari **Node** dengan nama left & right.

Kemudian kita buat fungsi **createNode** untuk membuat node cabang nantinya, sekaligus memberi nilai dengan cara menambahkan parameter data pada fungsi tersebut. Isi dari fungsi ini adalah sebagai berikut :

- Buat Node baru dengan nama newNode, ini untuk membuat root, parent dan child nantinya.
- Kemudian inisialisasi data sebagai root, dan data left dan right sebagai nullptr karena untuk pertama nilainya akan kosong.

Di ***int main() { }*** kita eksekusi node node tersebut, mulai dari :

Membuat root dengan membuat struct baru dari Node dengan nama ***root**, dan memberi nilai dengan mengambil fungsi **createNode(1)**, dengan parameter data yang berbentuk int.

Selanjutnya membuat cabang node, tentukan dulu ingin ke arah mana tree-nya, di atas saya akan membuat cabang ke kiri terlebih dahulu dengan memanggil root yang tadi dibuat, dan gunakan fungsi **-> left**. Ini akan otomatis membuat cabang baru ke sebelah kiri, begitu juga jika ingin membuat ke arah kanan **-> right.** Begitu seterusnya.