

MODUL PRAKTIKUM

STRUKTUR DATA

C/C++

Oleh:

Hermawan ST, M.Kom.

Informatika 2023

Uniersitas Trunojoyo Madura

I. ADT TREE

1. Pengertian

Tree dalam pemrograman adalah struktur data yang terdiri dari simpul atau node yang terhubung satu sama lain melalui garis atau sisi. Setiap simpul dalam pohon memiliki anak-anak (sub-pohon) yang terhubung melalui sisi. Setiap simpul, kecuali simpul teratas, memiliki satu simpul di atasnya yang disebut sebagai induk.

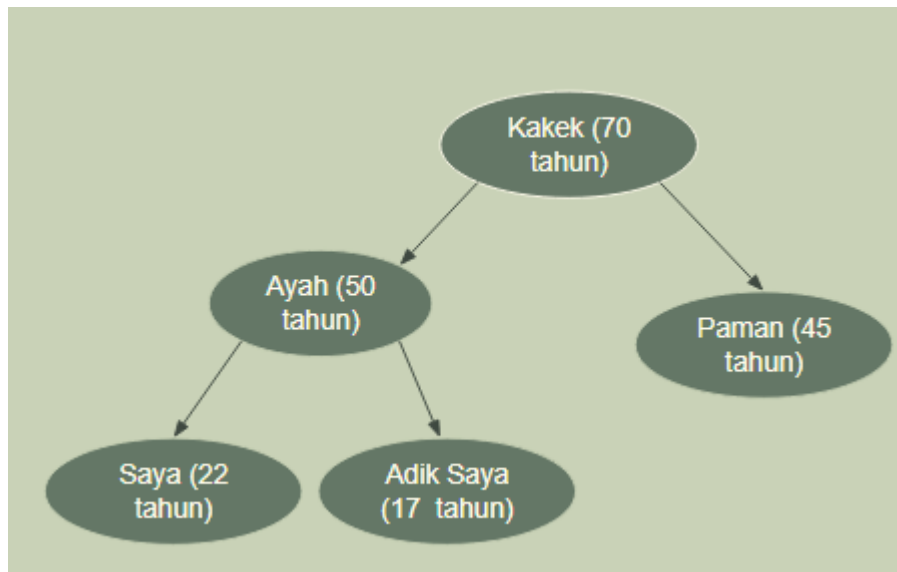
Tree dapat digunakan untuk merepresentasikan hierarki atau struktur data dengan relasi parent-anak antar elemen. Beberapa contoh penggunaan tree dalam pemrograman adalah sebagai berikut: pengorganisasian file dan direktori dalam sistem operasi, representasi struktur keluarga, implementasi struktur data seperti Binary Search Tree (BST), dan banyak lagi.

2. Istilah & Hubungan Komponen Tree

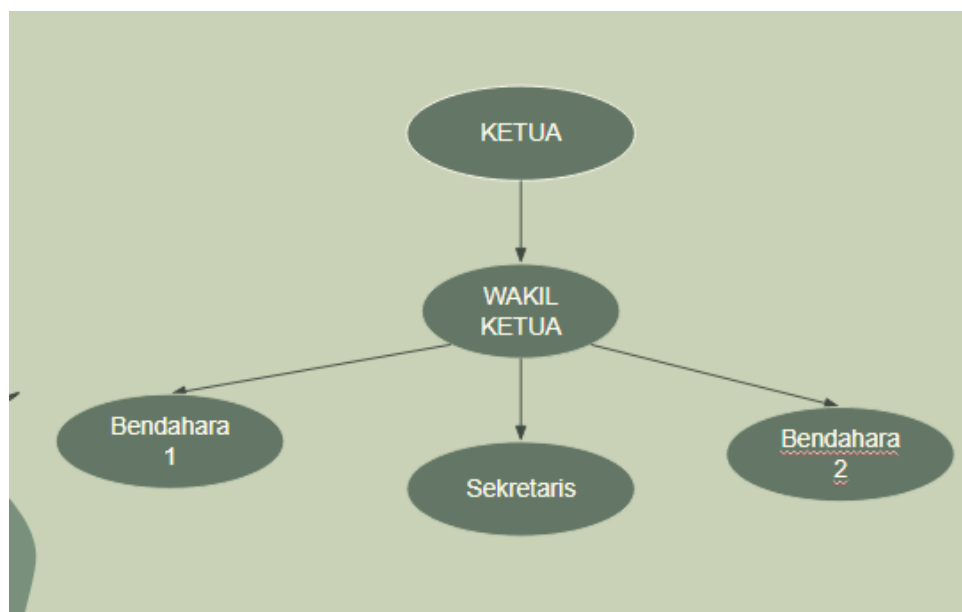
- Node(simpul) : Merupakan simpul dari masing masing data dari suatu tree
- Predecessor(Pendahulu) : Node yang berada diatas node tertentu
- Successor(Penerus): Node yang berada dibawah node tertentu
- Ancestor(Leluhur): Seluruh node yang berada diatas node tertentu dan terletak pada jalur yang sama
- Descendant(Keturunan) : Seluruh node yang terletak setelah node tertentu dan terletak pada jalur yang sama
- Parent(Orangtua): Pendahulu satu level di atas suatu node
- Child(Anak) : Penerus satu level di bawah suatu node
- Sibling(Saudara) :Node yang memiliki parent yang sama
- Height : Banyaknya tingkatan dalam suatu tree
- Root: Node awal yang tidak memiliki pendahulu
- Leaf : Node yang tidak memiliki penerus
- Degree : Banyaknya child dalam suatu node
- Depth(Kedalaman): Level dari suatu node

3. Ordered dan Ordered Tree

- Ordered Tree
 - Antar sibling terdapat urutan usia
 - Node yang paling kiri usianya paling tua
 - Contohnya :Silsilah Keluarga



- Unordered Tree
- Antar sibling tidak terdapat urutan
- Contohnya : Struktur Organisasi



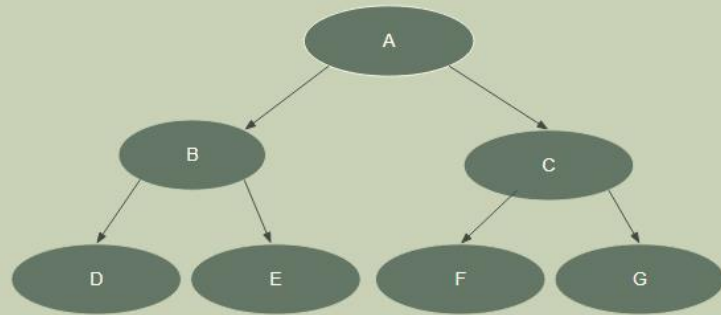
4. Binary Tree

Merupakan tree dengan syarat tiap node memiliki maksimal dua subtree

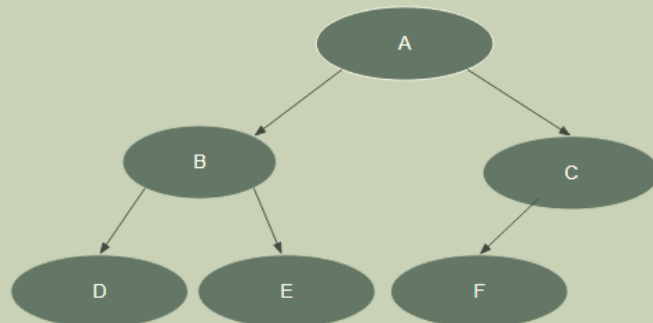
- Binary tree boleh tidak memiliki subtree
- Binary boleh memiliki subtree sebelah kiri
- Binary boleh memiliki subtree sebelah kanan

- Binary boleh memiliki subtree sebelah kanan dan kiri
- Macam-macam binary Tree

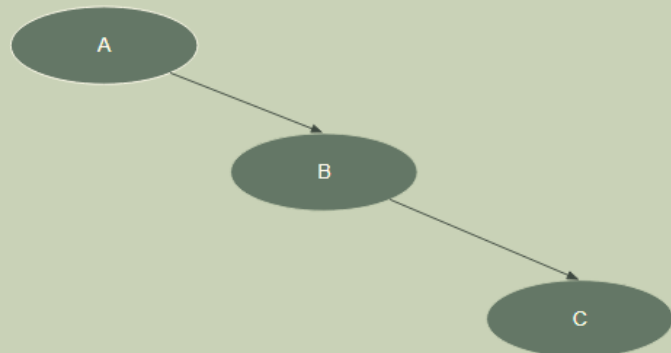
Full Binary Tree



Complete Binary Tree



Skew Binary Tree



- Tree Transversal : Merupakan teknik menyusuri tiap node dalam sebuah tree secara sistematis, sehingga semua node dapat dan hanya satu kali dikunjungi
 - **Preorder:**
 - Kunjungi root
 - Telusuri subtree kiri
 - Telusuri subtree kanan
 - **Inorder**
 - Telusuri subtree kiri
 - Kunjungi root
 - Telusuri subtree kanan
 - **Post Order**
 - Telusuri subtree kiri
 - Telusuri subtree kanan
 - Kunjungi root

Contoh Implementasi

```
#include <iostream>

class Node {

public:

    int value;
```

```

    Node* left;

    Node* right;

    Node(int value) {

        this->value = value;

        this->left = nullptr;

        this->right = nullptr;

    }

};


void inorderTraversal(Node* root) {

    if (root == nullptr)

        return;

    inorderTraversal(root->left);

    std::cout << root->value << " ";

    inorderTraversal(root->right);

}


Node* createBinaryTree() {

    Node* root = new Node(1);

    return root;

}


void addChildren(Node* root) {

    Node* leftChild = new Node(2);

```

```

    Node* rightChild = new Node(3);

    Node* nodebaru1 = new Node(4);

    Node* nodebaru2 = new Node(5);


    root->left = leftChild;

    root->right = rightChild;

    rightChild->right=nodebaru1;

    rightChild->left=nodebaru2;

}

int main() {

    // Langkah 1: Membuat pohon biner

    Node* root = createBinaryTree();


    // Langkah 2: Menambahkan anak-anak

    addChildren(root);


    // Langkah 3: Melakukan traversal inorder

    inorderTraversal(root);

    std::cout << std::endl;


    delete root->left;

    delete root->right;

```

```
delete root;  
  
return 0;  
}
```

TUGAS PRAKTIKUM

1. Buatlah sebuah code yang menampilkan binary tree Pre order dan Post order (**Jika bisa memodif agar dapat menampilkan gambar/struktur tree nya menjadi nilai tambah**)

