

## MINGGU 6

### Tree

#### DESKRIPSI TEMA

(Apa yang akan dipelajari mahasiswa pada minggu ini)

#### CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

1. Mahasiswa mampu memahami dan mengaplikasikan konsep Binary Tree menggunakan Bahasa C.
2. Mahasiswa mampu memahami dan mengaplikasikan konsep Huffman Tree menggunakan Bahasa C.

#### PENUNJANG PRAKTIKUM

1. Aplikasi CodeBlock
2. Aplikasi Dev-C++ (alternatif)

#### LANGKAH-LANGKAH PRAKTIKUM

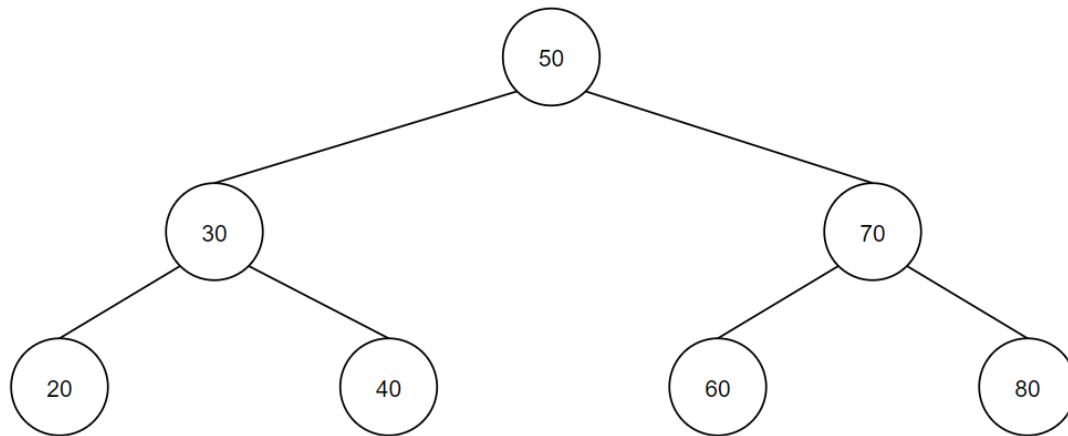
##### A. Binary Tree

- Tutorial 1.1 – Inserting Data to Tree
  - a. Buatlah sebuah file dengan nama Wo3\_NIM\_BT1.c
  - b. Ketik ulang code di bawah ini, jalankan, dan perhatikan hasilnya.

```
1  #include <stdio.h>
2  #include <malloc.h>
3  #include <stdlib.h>
4
5  // Deklarasi Struct untuk Linked List dalam Tree
6  typedef struct Node{
7      int key;
8      struct Node *left, *right;
9  }Node;
10
11 // Function yang digunakan untuk membuat node baru
12 Node *newNode(int item){
13     Node *temp = (Node*) malloc(sizeof(Node));
14     temp->key = item;
15     temp->left = temp->right = NULL;
16     return temp;
17 }
```

```
19 // Function yang digunakan untuk memasukkan node baru ke dalam Tree
20 v Node *insert(Node *node, int key){
21 v // Jika Tree kosong
22 // Atau ketika berada pada node kosong (left / right)
23 if (node == NULL)
24     return newNode(key);
25
26 // Jika key kurang dari key pada node sementara
27 if (key < node->key)
28     // Turun ke left dari node sementara
29     node->left = insert(node->left, key);
30
31 // Jika key lebih dari key pada node sementara
32 else if (key > node->key)
33     // Turun ke right dari node sementara
34     node->right = insert(node->right, key);
35
36 // return node jika sudah berhasil insert
37 return node;
38 }
39
40 v int main(){
41 // Buat root sebagai NULL agar tree dianggap kosong
42 Node *root = NULL;
43
44 v // Implementasi sederhana untuk tree
45 // Masukkan terlebih dahulu data ke dalam root
46 root = insert(root, 50);
47 // Data sisanya tinggal dimasukkan ke dalam root
48 insert(root, 50);
49 insert(root, 30);
50 insert(root, 20);
51 insert(root, 40);
52 insert(root, 70);
53 insert(root, 60);
54 insert(root, 80);
55
56 return 0;
57 }
```

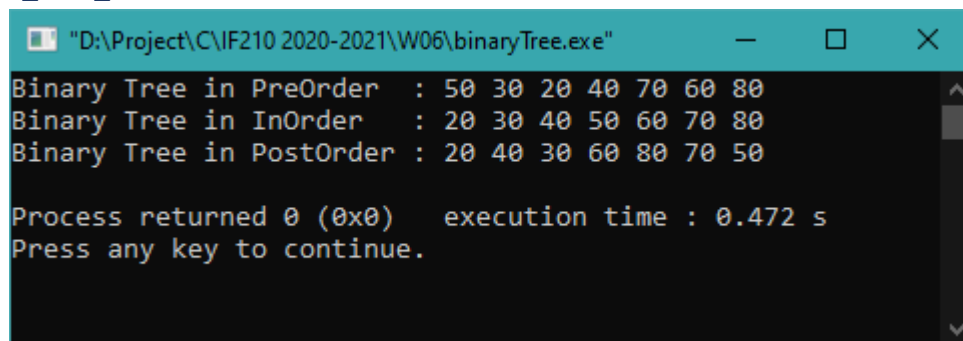
c. Dengan code tersebut, Tree yang dibuat akan berbentuk sebagai berikut:



## B. Tugas

### - Tugas 1

Buatlah output untuk Tutorial 1.1. Buatlah dalam bentuk PreOrder, InOrder, dan PostOrder. Contoh hasil jadinya adalah sebagai berikut. Simpan program dengan nama Wo6\_NIM\_T1.c



```

D:\Project\C\IF210 2020-2021\W06\binaryTree.exe
Binary Tree in PreOrder : 50 30 20 40 70 60 80
Binary Tree in InOrder : 20 30 40 50 60 70 80
Binary Tree in PostOrder : 20 40 30 60 80 70 50

Process returned 0 (0x0)   execution time : 0.472 s
Press any key to continue.
  
```

### - Tugas 2

Buatlah sebuah program menggunakan Binary Tree untuk menyusun ekspresi matematika berikut agar menjadi ekspresi matematika yang benar. Simpan program dengan nama Wo6\_NIM\_T2.c

- \* + a b c d
- \* + a b + c d
- + / a b / c d

## REFERENSI

