LEMBAR PRAKTIKUM 10 : TREE

I. Identitas Praktikan

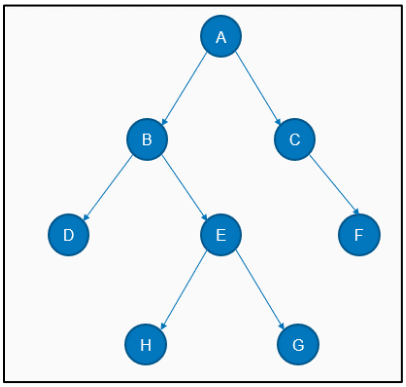
NIM : 2210131210001

NAMA : Muhammad Fikri Ramadhan

II. Tujuan PRAKTIKUM:

- Peserta memahami Konsep Tree
- Peserta mampu memahami proses algoritma Tree
- Peserta mampu mengimplementasikan algoritma Tree
- Peserta mampu mengembangkan dan membuat program python menggunakan konsep algoritma Tree

III. Aktivitas



1. Buatlah implementasi dari tree diatas kedalam Python menggunakan list of list lalu:
    a. Tampilkan root
    b. Tampilkan subtree kiri
    c. Tampilkan subtree kanan
2. Dari tree pada soal nomor 1, buatlah **Fungsi** untuk menentukan:
    a. Jumlah simpul
    b. Jumlah daun
    c. Tinggi Tree

3. Dari soal nomor 1 dan 2 jelaskan secara ringkas kode dan outputnya

IV. Hasil Praktik

(kode python)

```python
# Pohon yang diberikan
newTree = ['a', #root
           ['b', #subtree kiri
            ['d'],
            ['e',
             ['h'],
             ['g']] ],
           ['c', #subtree kanan
            ['f']]
          ]

def get_total_nodes(tree):
    if not tree:
        return 0

    count = 1  # Menginisialisasi dengan root node
    for child in tree[1:]:
        count += get_total_nodes(child)  # Menambahkan jumlah simpul di setiap anak
    return count

def get_leaf_count(tree):
    if not tree:
        return 0

    count = 0
    if len(tree) == 1:
        return 1  # Jika node tidak memiliki anak, maka itu adalah daun

    for child in tree[1:]:
        count += get_leaf_count(child)  # Menambahkan jumlah daun di setiap anak
    return count
```

```python

def get_tree_height(tree):
    if not tree:
        return 0

    heights = []
    if len(tree) == 1:
        return 1  # Jika node tidak memiliki anak, maka tingginya adalah 1

    for child in tree[1:]:
        heights.append(get_tree_height(child))  # Mengumpulkan tinggi dari setiap anak
    return max(heights) + 1  # Tinggi tree adalah maksimum dari tinggi anak ditambah 1

# Menggunakan fungsi-fungsi untuk menghitung jumlah simpul, jumlah daun, dan tinggi pohon
total_nodes = get_total_nodes(newTree)
leaf_count = get_leaf_count(newTree)
tree_height = get_tree_height(newTree)

print (newTree)
print()
print ('1.a. Root = ', newTree[0])
print ('1.b. Subtree kiri = ', newTree[1])
print ('1.c. Subtree kanan = ', newTree[2])
print()
print("2.a. Jumlah simpul:", total_nodes)  # Output: 8
print("2.b. Jumlah daun:", leaf_count)  # Output: 4
print("2.c. Tinggi tree:", tree_height)  # Output: 4

```

(Hasil program)

```
['a', ['b', ['d'], ['e', ['h'], ['g']]], ['c', ['f']]]

1.a. Root =  a
1.b. Subtree kiri =  ['b', ['d'], ['e', ['h'], ['g']]]
1.c. Subtree kanan =  ['c', ['f']]

2.a. Jumlah simpul: 8
2.b. Jumlah daun: 4
2.c. Tinggi tree: 4
```

(Penjelasan)

1. 1. a. Root merupakan akar dari pohon atau puncak yang tertinggi, jadi root disana adalah ‘a’ dan tree adalah list of list
2. 1.b. 'b' adalah subtree kiri dari root, yang merupakan anak dari induk ‘a’
3. 1.c. 'c' adalah subtree kanan dari root, yang merupakan ana dari ‘a’
4. 2.a. jumlah simpul, untuk menghitung jumlah nodes, digunakan sebuah fungsi untuk menghitung nodes yang menggunakan pendekatan rekursif, jadi tree yang kosongndi hitung 0 dan yang berisi diitung 1, jdi di totalkan
5. 2.b. jumlah daun, untuk menghitung jumlah daun dalam pohon, mengunakan fungsi, apabila pohon kosong, maka jumlah daun adalah 0 dan apabila jika pohon hanya memiliki satu elemen tanpa anak, maka itu adalah daun, jika pohon memiliki anak anak, maka akan menghitung jumlah daun pada setiap anak
6. 2.c. menghitung tinggi dari pohon, untuk menghitung nya, dibuatlah fungsi. Jika pohon memiliki anak-anak, fungsi akan menghitung tinggi dari setiap anak dan mengambil maksimum tinggi anak, kemudian menambahkannya dengan 1