

Nama : Ahyana Ilham W

NIM : L200170170

Kelas : D

Modul 9

Nomor 6 dan 7

```
modul9.py - D:\modul9.py (3.6.3)
File Edit Format Run Options Window Help

G.kanan=[]

detailpr=[A.data, B.data, C.data, D.data, E.data, F.data, G.data, H.data, I.data]
level=[]

def preorder(sub):
    if sub is not None:
        print(sub.data)
        preorder(sub.kiri)
        preorder(sub.kanan)
    else:
        return

def inorder(sub):
    if sub is not None:
        inorder(sub.kiri)
        print(sub.data)
        inorder(sub.kanan)
    else:
        return

def postorder(sub):
    if sub is not None:
        postorder(sub.kiri)
        postorder(sub.kanan)
        print(sub.data)
    else:
        return

def size(node):
    if node is None:
        return 0
    else:
        return (size(node.kiri) + 1 + size(node.kanan))

def maxDepth(node):
    if node is None:
        return 0
    else:
        lDepth = maxDepth(node.kiri)
        rDepth = maxDepth(node.kanan)
        if (lDepth > rDepth):
            return lDepth+1
        else:
            return rDepth+1

Python 3.6.3
File Edit Shell Debug Options Window Help

Python 3.6.3 (v2.6.3:20fcd80, Oct 3 2017, 17:28:49) [AMD64 v1.900 32 bit (Intel)]
> on win32
Type "copyright", "credits" or "license()" for more information.
>>>

===== RESTART: D:\modul9.py =====
Urutan dari Binary Tree adalah :
Tinggi maksimal dari Binary Tree adalah : 4
A. data : Level 0
B. data : Level 1
C. data : Level 1
D. data : Level 2
E. data : Level 2
F. data : Level 2
G. data : Level 2
H. data : Level 2
I. data : Level 2
>>>
```

Nomor 8

```
modul9.py - D:\modul9.py (3.6.3)
File Edit Format Run Options Window Help

if (lDepth > rDepth):
    return lDepth+1
else:
    return rDepth+1

def traverse(root):
    llist=[]
    current_level = [root]
    lv=0
    while current_level:
        print(' ', join(str(node) for node in current_level))
        next_level = []
        for n in current_level:
            if n.kiri:
                next_level.append(n.kiri)
            if n.kanan:
                next_level.append(n.kanan)
        current_level = next_level
        lv+=1
    llist.append(lv)
    return llist

def cetakDataLevel(root):
    traverse(A)
    print(' ', level[0])
    for i in range(len(level)):
        print(' ', level[i])

print('Urutan dari Binary Tree adalah : ', size(A))
print(' ')
print('Tinggi maksimal dari Binary Tree adalah : ', maxDepth(A))
print(' ')
cetakDataLevel(A)
```