

Laporan Praktikum Algostruk  
Modul 9  
Pohon Biner

---

6 dan 7. Berikut screenshot programnya :

```
class simpulbiner(object):
    def __init__(self, data):
        self.data=data
        self.kiri=None
        self.kanan=None

    def __str__(self):
        return str(self.data)

A=simpulbiner('Ambarawa')
B=simpulbiner('Bantul')
C=simpulbiner('Cimahi')
D=simpulbiner('Denpasar')
E=simpulbiner('Enrekang')
F=simpulbiner('Flores')
G=simpulbiner('Garut')
H=simpulbiner('Halmahera Timur')
I=simpulbiner('Indramayu')
J=simpulbiner('Jakarta')

A.kiri=B; A.kanan=C
B.kiri=D; B.kanan=E
C.kiri=F; C.kanan=G
E.kiri=H
G.kanan=I

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

def preord(sub):
    if sub is not None:
        print(sub.data)
        preord(sub.kiri)
        preord(sub.kanan)
def inord(sub):
    if sub is not None:
        inord(sub.kiri)
        print(sub.data)
        inord(sub.kanan)
```

```
def postord(sub):
    if sub is not None:
        postord(sub.kiri)
        postord(sub.kanan)
        print(sub.data)

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

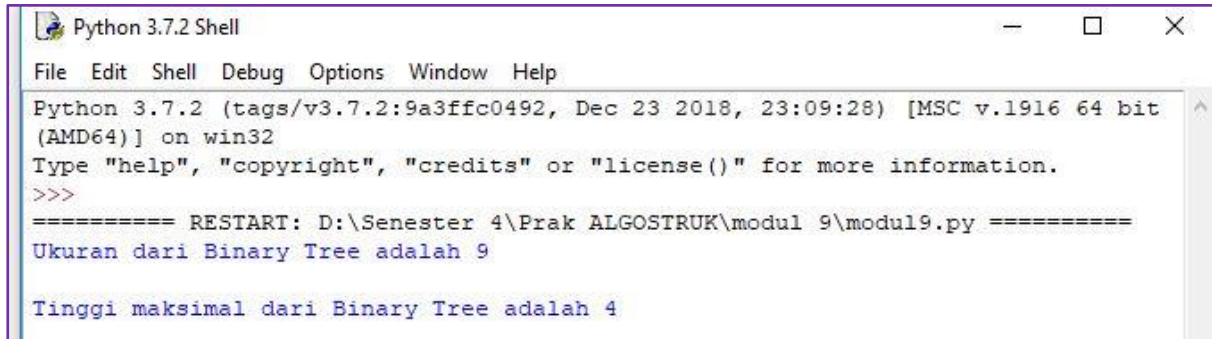
def maxDepth(node):
    if node is None:
        return 0 ;

    else :
        lDepth = maxDepth(node.kiri)
        rDepth = maxDepth(node.kanan)

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

**Muhibah Fata Tika**  
**L200170156**  
**D**  
**Prak Algostruk**

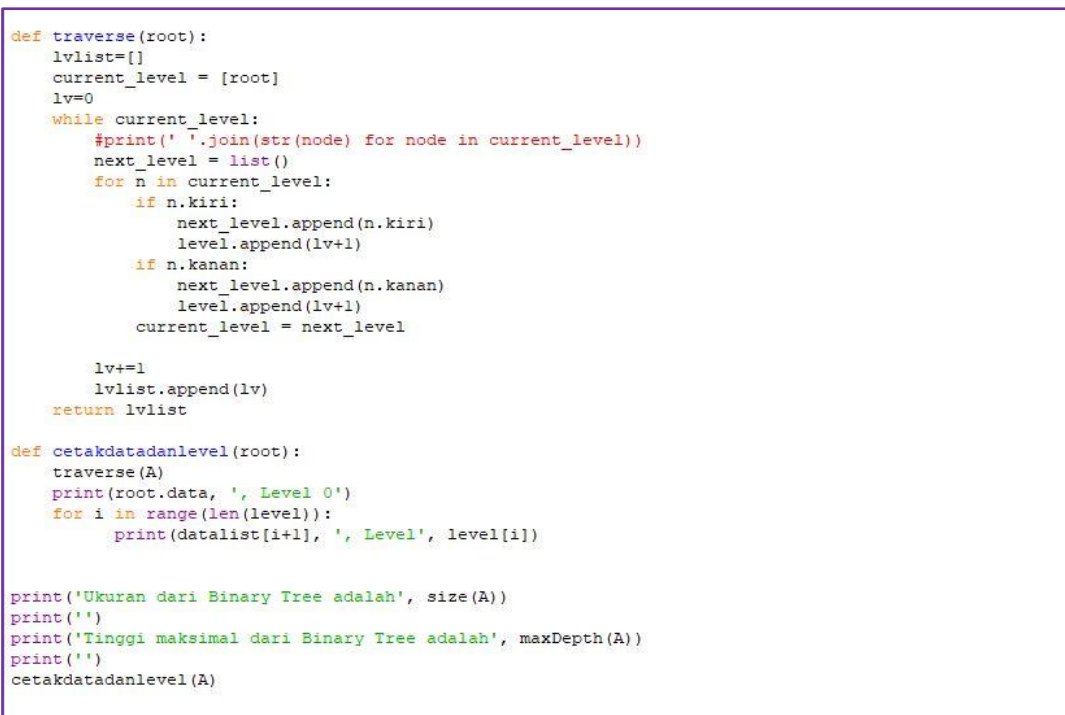
Berikut outputnya :



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\Senester 4\Prak ALGOSTRUK\modul 9\modul9.py =====
Ukuran dari Binary Tree adalah 9

Tinggi maksimal dari Binary Tree adalah 4
```

8. Berikut screenshot programnya :



```
def traverse(root):
    lvlist=[]
    current_level = [root]
    lv=0
    while current_level:
        #print(' '.join(str(node) for node in current_level))
        next_level = list()
        for n in current_level:
            if n.kiri:
                next_level.append(n.kiri)
                level.append(lv+1)
            if n.kanan:
                next_level.append(n.kanan)
                level.append(lv+1)
            current_level = next_level

        lv+=1
        lvlist.append(lv)
    return lvlist

def cetakdatadanlevel(root):
    traverse(A)
    print(root.data, ', Level 0')
    for i in range(len(level)):
        print(datalist[i+1], ', Level', level[i])

print('Ukuran dari Binary Tree adalah', size(A))
print('')
print('Tinggi maksimal dari Binary Tree adalah', maxDepth(A))
print('')
cetakdatadanlevel(A)
```

Berikut outputnya :

**Muhibah Fata Tika**  
**L200170156**  
**D**  
**Prak Algostruk**

```
Ambarawa , Level 0  
Bantul , Level 1  
Cimahi , Level 1  
Denpasar , Level 2  
Enrekang , Level 2  
Flores , Level 2  
Garut , Level 2  
Halmahera Timur , Level 3  
Indramayu , Level 3  
>>>
```