

Nama : Maulana Alhif Ikhsan
NIM : L200180120
Kelas : E

Modul 9

1) Tentukan jumlah level maximum dan minimum dari ukuran pohon biner berikut.

❖ $n = 10$

- jumlah level minimum = $\text{INT}[\log_2 10] + 1 = 4$
- jumlah level maximum = (level 0 sampai level 9) = 10

❖ $n = 35$

- jumlah level minimum = $\text{INT}[\log_2 35] + 1 = 6$
- jumlah level maximum = (level 0 sampai level 34) = 35

❖ $n = 76$

- jumlah level minimum = $\text{INT}[\log_2 76] + 1 = 7$
- jumlah level maximum = (level 0 sampai level 75) = 76

❖ $n = 345$

- jumlah level minimum = $\text{INT}[\log_2 345] + 1 = 9$
- jumlah level maximum = (level 0 sampai level 344) = 345

2) Ada berapa kemungkinan gambar yang dapat dibentuk dari pohon biner berukuran 5.

$$\begin{aligned} C_n &= \frac{(2n)!}{(n+1)! \cdot n!} \\ &= \frac{(2 \cdot 5)!}{(5+1)! \cdot 5!} \\ &= \frac{10!}{6! \cdot 5!} \\ &= \frac{3628800}{86400} \\ &= 42 \text{ kemungkinan} \end{aligned}$$

3) Tentukan jumlah simpul maksimum suatu pohon biner dengan jumlah level h

a. $h = 3$

$$\text{Jumlah max simpul} = \text{level 0} + \text{level 1} + \text{level 2} = 2^0 + 2^1 + 2^2 = 7$$

b. $h = 4$

$$\text{Jumlah max simpul} = \text{level 0} + \text{level 1} + \text{level 2} + \text{level 3} = 2^0 + 2^1 + 2^2 + 2^3 = 15$$

c. $h = 5$

$$\text{Jumlah max simpul} = \text{level 0} + \text{level 1} + \text{level 2} + \text{level 3} + \text{level 4} = 2^0 + 2^1 + 2^2 + 2^3 + 2^4 = 31$$

d. $h = 6$

$$\text{Jumlah max simpul} = \text{level 0} + \text{level 1} + \text{level 2} + \text{level 3} + \text{level 4} + \text{level 5} = 2^0 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5 = 63$$

4) Diberikan pohon-pohon biner seperti di bawah

- a. Tunjukkan semua properti struktural yang berlaku pada tiap-tiap pohon di atas : *penuh* , *sempurna*, *komplet*. Ingat bahwa sebuah pohon biner bisa saja bersifat penuh sekaligus sempurna dan sebagainya.

a = penuh

b = sempurna

c = komplet dan penuh

d = komplet

e = komplet

- b. Tentukan ukuran tiap pohon.

a = 7

b = 15

c = 14

d = 7

e = 11

- c. Tentukan ketinggian tiap pohon.

a = 4

b = 4

c = 8

d = 4

e = 4

- d. Tentukan lebar tiap pohon.

a = 2

b = 8

c = 2

d = 3

e = 5

5) Perhatikan pohon biner berikut.

- a. Tunjukkan urutan pengunjungan simpul untuk :
 - i. Preorder traversal = 14-78-39-52-83-17-9-41-2-60-23-4-19
 - ii. Inorder traversal = 39-78-17-83-9-52-41-14-60-2-4-23-19
 - iii. Postorder traversal = 39-17-9-83-41-52-78-60-4-19-23-2-14

- b. Simpul mana saja yang merupakan simpul daun ?

→ 39, 17, 9, 41, 60, 4, 19

- c. Simpul mana saja yang merupakan simpul dalam ?

→ 14, 78, 52, 83, 2, 23

- d. Simpul mana saja yang berada di level 4?

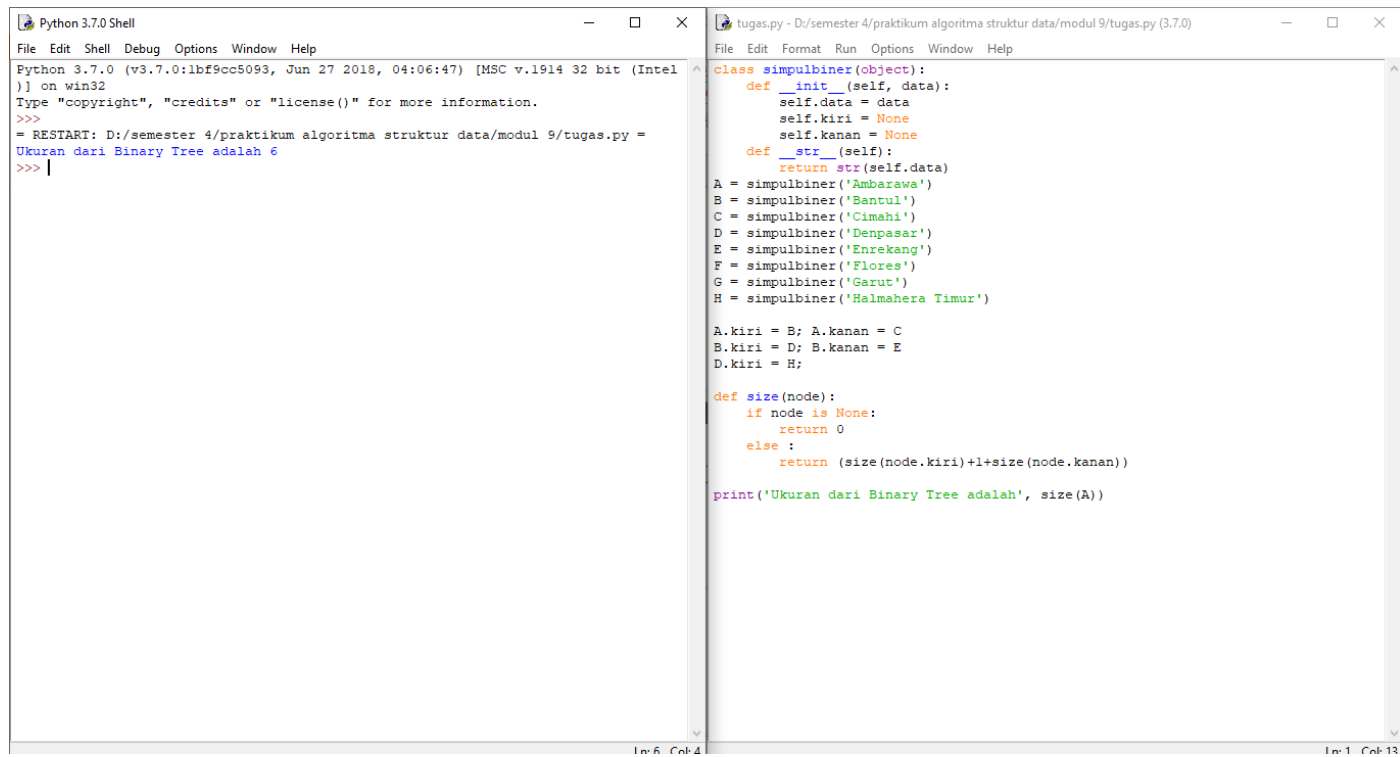
→ 17, 9

- e. Tulis semua simpul yang berada di dalam jalur dari simpul akar menuju simpul
 - i. $83 = 15 - 78 - 52 - 83$
 - ii. $39 = 14 - 78 - 39$
 - iii. $4 = 14 - 2 - 23 - 4$
 - iv. $9 = 14 - 78 - 52 - 83 - 9$

- f. Perhatikan simpul 52 Tentukan !
 - i. Keturunannya = 83, 41
 - ii. Leluhurnya = 78, 14
 - iii. Saudaranya = 39

- g. Tentukan kedalaman dari tiap-tiap simpul ini :
 - i. 78 = level 1
 - ii. 41 = level 2
 - iii. 60 = level 2
 - iv. 19 = level 3

- 6) Buatlah fungsi ukuran pohon(akar) yang akan mendapatkan ukuran sebuah pohon binner



The image shows two windows from a Python 3.7.0 environment. The left window is the Python Shell, and the right window is a text editor showing a Python script named `tugas.py`.

Python Shell (Left Window):

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/semester 4/praktikum algoritma struktur data/modul 9/tugas.py =
Ukuran dari Binary Tree adalah 6
>>> |
```

tugas.py (Right Window):

```
class simpulbinner(object):
    def __init__(self, data):
        self.data = data
        self.kiri = None
        self.kanan = None
    def __str__(self):
        return str(self.data)

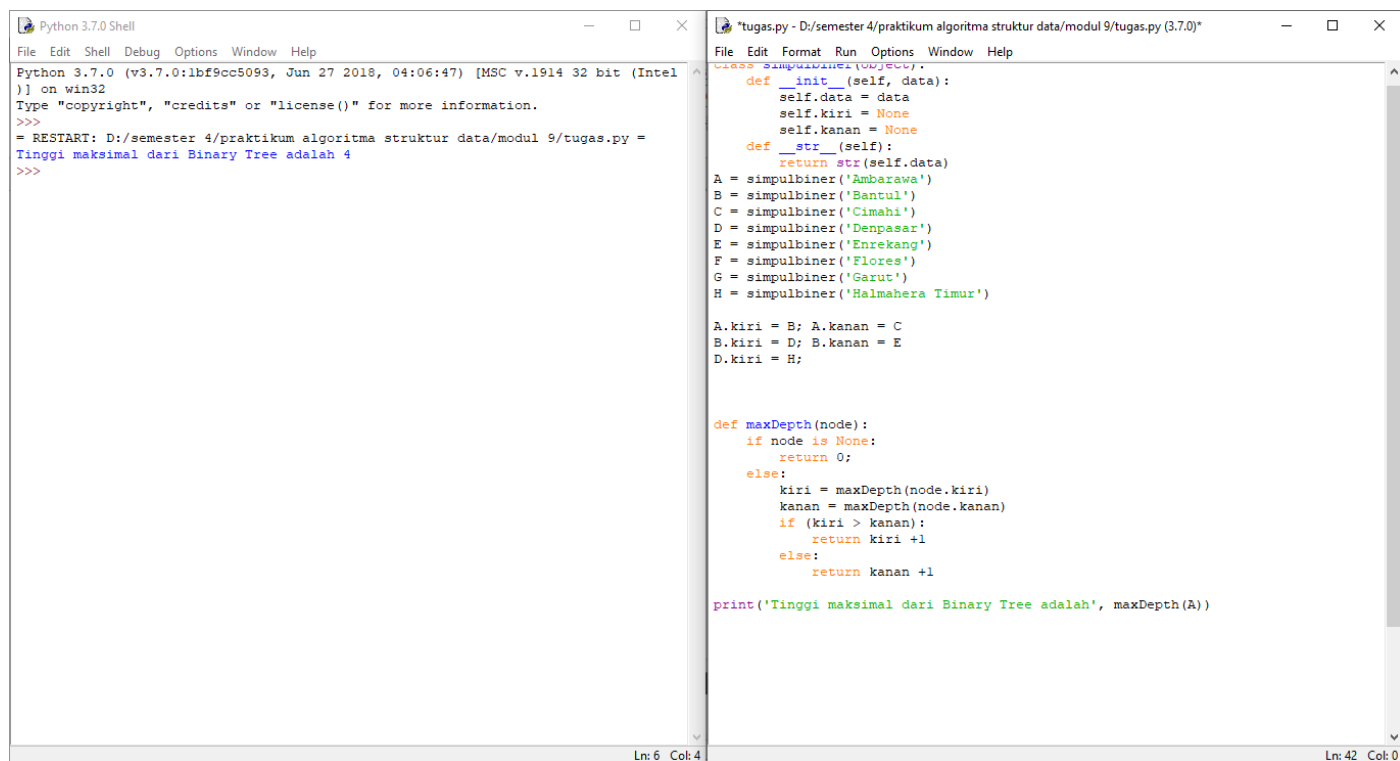
A = simpulbinner('Ambarawa')
B = simpulbinner('Bantul')
C = simpulbinner('Cimahi')
D = simpulbinner('Denpasar')
E = simpulbinner('Enrekang')
F = simpulbinner('Flores')
G = simpulbinner('Garut')
H = simpulbinner('Halmahera Timur')

A.kiri = B; A.kanan = C
B.kiri = D; B.kanan = E
D.kiri = H;

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

print('Ukuran dari Binary Tree adalah', size(A))
```

- 7) Buatlah sebuah fungsi tinggi pohon (akar) yang akan mendapatkan ketinggian pohon binner



The image shows two windows from a Python 3.7.0 environment. The left window is the Python Shell, and the right window is a text editor showing a Python script named `tugas.py`.

Python Shell (Left Window):

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/semester 4/praktikum algoritma struktur data/modul 9/tugas.py =
Tinggi maksimal dari Binary Tree adalah 4
>>>
```

tugas.py (Right Window):

```
class simpulbinner(object):
    def __init__(self, data):
        self.data = data
        self.kiri = None
        self.kanan = None
    def __str__(self):
        return str(self.data)

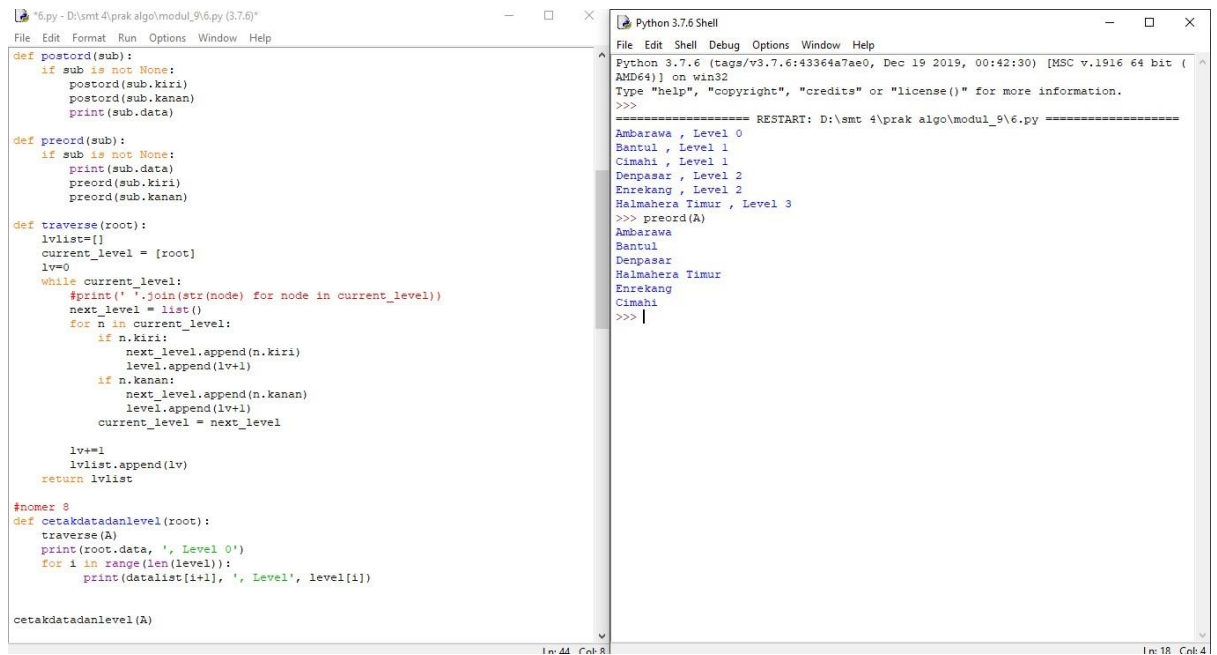
A = simpulbinner('Ambarawa')
B = simpulbinner('Bantul')
C = simpulbinner('Cimahi')
D = simpulbinner('Denpasar')
E = simpulbinner('Enrekang')
F = simpulbinner('Flores')
G = simpulbinner('Garut')
H = simpulbinner('Halmahera Timur')

A.kiri = B; A.kanan = C
B.kiri = D; B.kanan = E
D.kiri = H;

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

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

- 8) Buatlah sebuah fungsi yang mencetak data tiap simpul sekaligus level dimana simpul itu berada.



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

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

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

#nomer 8
def cetakdatadanlevel(root):
    traverse(A)
    print(root.data, ', Level 0')
    for i in range(len(level)):
        print(data[i+1], ', Level', level[i])

cetakdatadanlevel(A)
```

```
Python 3.7.6 Shell
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\smt 4\prak algo\modul_9\6.py =====
Ambarawa , Level 0
Bantul , Level 1
Cimahi , Level 1
Denpasar , Level 2
Enrekang , Level 2
Halmahera Timur , Level 3
>>> preord(A)
Ambarawa
Bantul
Denpasar
Halmahera Timur
Enrekang
Cimahi
>>> |
```