

NAMA : Moch Afif Abdillah

NIM : L200170004

KELAS : A

Soal-soal untuk mahasiswa

1. Diberikan pohon biner dengan ukuran n , berapakah jumlah level minimum yang bisa dimuatnya ? Berapakah jumlah level maksimum nya? Tentukan untuk nilai n berikut.

a) $n = 10$

Level minimum = 3

Level maximum = 9

b) $n = 35$

Level minimum = 6

Level maximum = 34

c) $n = 76$

Level minimum = 7

Level maximum = 75

d) $n = 345$

Level minimum = 8

Level maximum = 344

2. Gambarlah semua bentuk pohon biner berukuran 5 yang mungkin. Ada berapa kemungkinan ?

Jawab :

$$C_n = (2n)! / (n+1)! * n!$$

$$= (2*5)! / (5+1)! * 5!$$

$$= 10! / 6! * 5!$$

$$= 3628800 / 86400$$

$$= 42 \text{ kemungkinan}$$

3. Berapakah jumlah simpul maksimum suatu pohon biner dengan jumlah level h ? Tentukan untuk nilai h berikut

a. $h = 3$

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

$$= 1 + 2 + 4$$

$$= 7$$

b. $h = 4$

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

$$= 1 + 2 + 4 + 8$$

$$= 15$$

c. $h = 5$

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

$$= 1 + 2 + 4 + 8 + 16$$

$$= 31$$

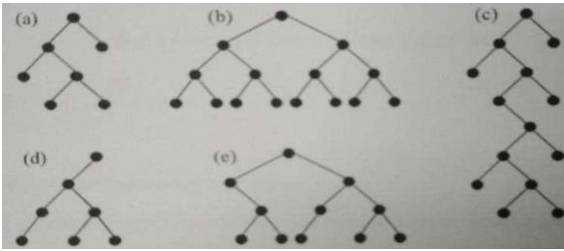
d. $h = 6$

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

$$= 1 + 2 + 4 + 8 + 16 + 32$$

$$= 63$$

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

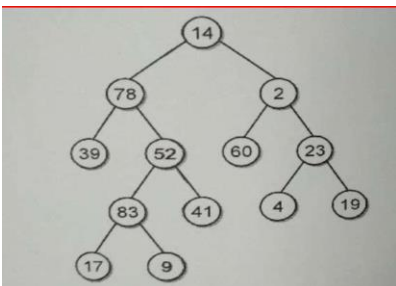
c. Tentukan ketinggian tiap pohon.

- 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

Soal-soal pemograman

6. Buatlah fungsi ukuranPohon (akar) yang akan mendapatkan ukuran sebuah pohon biner.

```
#nomor6
class simpulpohonbiner(object):
    def __init__(self, data):
        self.data = data
        self.kiri = None
        self.kanan = None

    def ukuranpohon (akar, count = 0 ):
        if akar is None :
            return count

        return ukuranpohon(akar.kiri, ukuranpohon(akar.kanan, count +1))

a = simpulpohonbiner ('Ambarawa')
b = simpulpohonbiner ('Bantul')
c = simpulpohonbiner ('Cimahi')
d = simpulpohonbiner ('Denpasar')
e = simpulpohonbiner ('Enrekang')
f = simpulpohonbiner ('Flores')
g = simpulpohonbiner ('Garut')
h = simpulpohonbiner ('Halmahera Timur')
i = simpulpohonbiner ('Indramayu')

a.kiri = b; a.kanan = c
b.kiri = d; b.kanan = e
c.kiri = f; c.kanan = g
e.kiri = h
g.kanan = i
```

```
----- RESTART: P.708
>>> ukuranpohon(a)
9
>>>
===== DESTART: P.710
```

7. Buatlah sebuah fungsi tinggiPohon (akar) yang akan mendapatkan ketinggian sebuah pohon biner.

```
#nomor7
class tinggipohonbiner (object):
    def __init__ (self, data) :
        self.data = data
        self.kiri = None
        self.kanan = None
    def tinggipohon(akar) :
        if akar is None:
            return 0
        else :
            return max(tinggipohon(akar.kiri), tinggipohon(akar.kanan))+1

a = simpulpohonbiner ('Ambarawa')
b = simpulpohonbiner ('Bantul')
c = simpulpohonbiner ('Cimahi')
d = simpulpohonbiner ('Denpasar')
e = simpulpohonbiner ('Enrekang')
f = simpulpohonbiner ('Flores')
g = simpulpohonbiner ('Garut')
h = simpulpohonbiner ('Halmahera Timur')
i = simpulpohonbiner ('Indramayu')

a.kiri = b; a.kanan = c
b.kiri = d; b.kanan = e
c.kiri = f; c.kanan = g
e.kiri = h
g.kanan = i
```

```
===== RESTART: D:/UMS/
>>> tinggipohon(a)
4
>>>
----- RESTART: D:/UMS/
```

8. Buatlah sebuah fungsi yang mencetak data tiap simpul sekaligus level dimana simpul itu berada. Silahkan memilih akan memakai preorder traversal, inorder traversal, atau postorder traversal. Contoh sepotong hasilnya adalah seperti di bawah ini (jika kamu memakai preorder traversal).

```
>>> cetakDataDanLevel(A)
```

```
Ambarawa, level 0
```

```
Bantul, level 1
```

```
Denpasar, level 2
```

```
Enrekang, level 2
```

```
Halmahera Timur, level 3
```

```
Cimahi, level 1
```

```
#nomor8
class simpulpohonbiner (object):
    def __init__ (self, data) :
        self.data = data
        self.kiri = None
        self.kanan = None

    def cetak(subpohon, count = 0):
        if subpohon is not None:
            print (subpohon.data + ', level ' + str (count))
            (cetak(subpohon.kiri, count + 1), cetak(subpohon.kanan, count + 1))

a = simpulpohonbiner ('Ambarawa')
b = simpulpohonbiner ('Bantul')
c = simpulpohonbiner ('Cimahi')
d = simpulpohonbiner ('Denpasar')
e = simpulpohonbiner ('Enrekang')
f = simpulpohonbiner ('Flores')
g = simpulpohonbiner ('Garut')
h = simpulpohonbiner ('Halmahera Timur')
i = simpulpohonbiner ('Indramayu')

a.kiri = b; a.kanan = c
b.kiri = d; b.kanan = e
c.kiri = f; c.kanan = g
e.kiri = h
g.kanan = i
```

```
>>> cetak(a)
Ambarawa, level 0
Bantul, level 1
Denpasar, level 2
Enrekang, level 2
Halmahera Timur, level 3
Cimahi, level 1
Flores, level 2
Garut, level 2
Indramayu, level 3
```