

**PRAKTIKUM ALGORITMA DAN
STRUKTUR DATA**

Modul 9

Pohon Biner



Disusun oleh:

DONI WAHYU SAPUTRO

L200200169

G

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS MUHAMMADIYAH SURAKARTA

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.

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

3) Tentukan jumlah simpul maksimum suatu pohon biner dengan jumlah level h a. $h = 3$

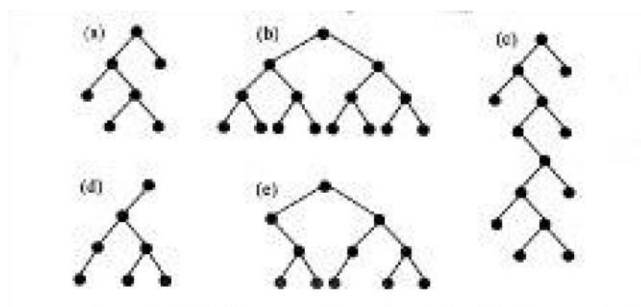
$$\begin{aligned} \text{Jumlah max simpul} &= \text{level 0} + \text{level 1} + \text{level 2} = 2^0 + 2^1 \\ &+ 2^2 = 7 \end{aligned} \quad \text{b. } h = 4$$

$$\begin{aligned} \text{Jumlah max simpul} &= \text{level 0} + \text{level 1} + \text{level 2} + \text{level 3} = 2^0 + 2^1 + 2^2 \\ &+ 2^3 = 15 \end{aligned} \quad \text{c. } h = 5$$

$$\begin{aligned} \text{Jumlah max simpul} &= \text{level 0} + \text{level 1} + \text{level 2} + \text{level 3} + \text{level 4} = 2^0 + 2^1 + 2^2 + \\ &2^3 + 2^4 = 31 \end{aligned} \quad \text{d. } h = 6$$

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

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

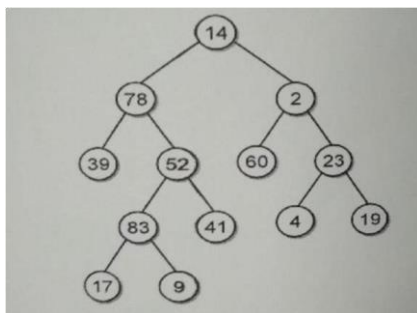
b = 8

c = 2

d = 3

e = 5

- 5) Perhatikan pohon biner berikut.



- a. Tunjukkan urutan pengunjungan simpul untuk :

- Preorder traversal = 14-78-39-52-83-17-9-41-2-60-23-4-19
- Inorder traversal = 39-78-17-83-9-52-41-14-60-2-4-23-19
- 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

- $83 = 15 - 78 - 52 - 83$
- $39 = 14 - 78 - 39$
- $4 = 14 - 2 - 23 - 4$
- $9 = 14 - 78 - 52 - 83 - 9$

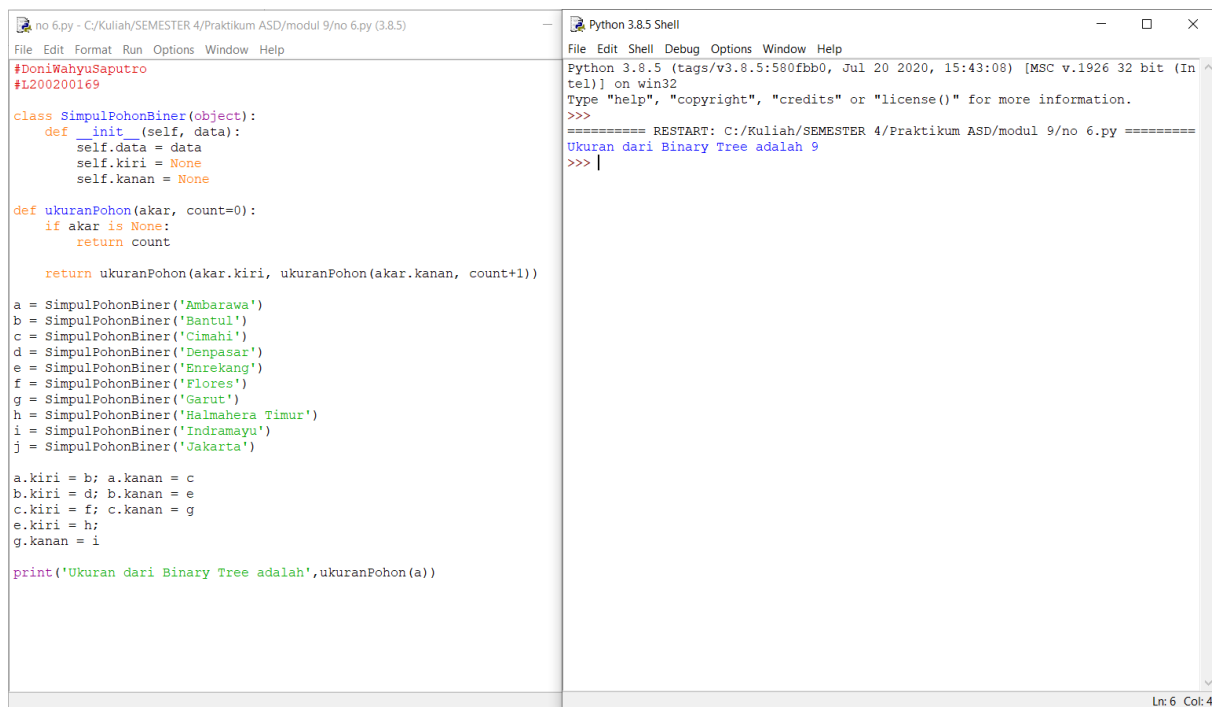
f. Perhatikan simpul 52 Tentukan!

- Keturunannya = 83, 41
- Leluhurnya = 78, 14
- Saudaranya = 39

g. Tentukan kedalaman dari tiap-tiap simpul ini :

- 78 = level 1
- 41 = level 2
- 60 = level 2
- 19 = level 3

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



```
no 6.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 9/no 6.py (3.8.5)
File Edit Format Run Options Window Help

#DoniWahyuSaputro
#L200200169

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')
j = SimpulPohonBiner('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

print('Ukuran dari Binary Tree adalah', ukuranPohon(a))

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 9/no 6.py =====
Ukuran dari Binary Tree adalah 9
>>> |
```

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

```

no 7.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 9/no 7.py (3.8.5)
File Edit Format Run Options Window Help
#DoniWahyuSaputro
#L200200169

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

def tinggiPohon(akar, count=0):
    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')
j = SimpulPohonBiner('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

print ('Tinggi maksimal dari Binary Tree adalah', tinggiPohon(a))

```

```

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 9/no 7.py =====
Tinggi maksimal dari Binary Tree adalah 4
>>> |

```

8) Buatlah sebuah fungsi yang mencetak data tiap simpul sekaligus level di mana simpul itu berada. Silakan memilih akan memakai preorder traversal, inorder traversal, atau postorder traversal.

```

no 8.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 9/no 8.py (3.8.5)
File Edit Format Run Options Window Help
#DoniWahyuSaputro
#L200200169

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')
H=simpulbiner('Halmahera Timur')

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

datalist=[A.data, B.data, C.data, D.data, E.data, H.data]
level=[]

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

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

def postorder(sub):

```

```

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 9/no 8.py =====
Ambarawa , Level 0
Bantul , Level 1
Cimahi , Level 1
Denpasar , Level 2
Enrekang , Level 2
Halmahera Timur , Level 3
=====preorder=====
Ambarawa
Bantul
Denpasar
Halmahera Timur
Enrekang
Cimahi
=====inorder=====
Halmahera Timur
Denpasar
Bantul
Enrekang
Ambarawa
Cimahi
=====postorder=====
Halmahera Timur
Denpasar
Bantul
Enrekang
Cimahi
Ambarawa
>>> |

```