

**ALGORITHM AND DATA STRUCTURE PRACTICUM**

**MODULE 9**

**THREE BINER**



**CREATED BY :**

**KURNIAWAN BAGASKARA**

**L200214253**

**INFORMATICS STUDY PROGRAM**

**FACULTY OF COMMUNICATION AND INFORMATION SCIENCE**

**MUHAMMADIYAH SURAKARTA UNIVERSITY**

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.  $C_n = \frac{(2n)!}{(n+1)! \cdot n!}$

$$= \frac{(2 \cdot 5)!}{(5+1)! \cdot 5!}$$

$$= \frac{10!}{6! \cdot 5!}$$

$$= \frac{3628800}{86400}$$

$$= 42 \text{ kemungkinan}$$

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

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

$$2^0 + 2^1 + 2^2 = 7 \quad \text{b. } h = 4$$

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

$$2^1 + 2^2 + 2^3 = 15 \quad \text{c. } h = 5$$

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

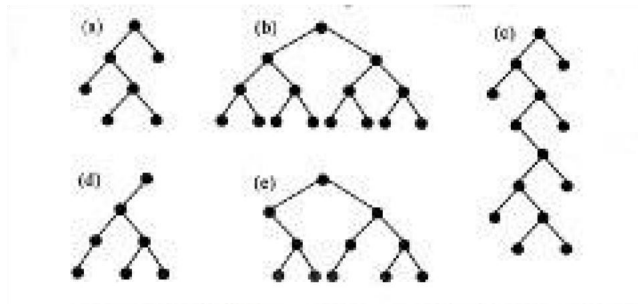
$$+ 2^2 + 2^3 + 2^4 = 31 \quad \text{d. } h = 6$$

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

$$= 2^0 +$$

$$2^1 + 2^2 + 2^3 + 2^4 + 2^5 = 63$$

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

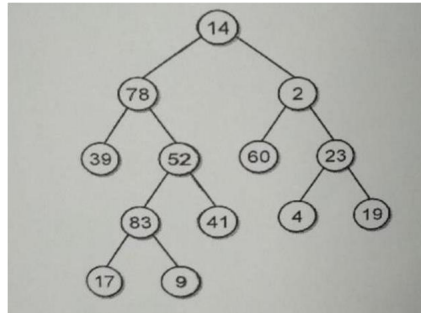
b = 8

c = 2

d = 3

e = 5

- 5) Perhatikan pohon biner berikut.



a. Tunjukkan urutan pengunjungan simpul untuk :

- 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 = 14 - 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. 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))
```

```
print("Kurniawan")
```

```
print("L200214253")
```

```
Shell ×
Python 3.7.9 (bundled)
>>> %cd 'E:\Kuliahhhh\Semester 4\Algoritma Struktur Data\
>>> %Run 'no 6.py'

Ukuran dari Binary Tree adalah 9
Kurniawan
L200214253
>>>
```

```
7. 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')
```

```
8. 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
```

```
9. 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))
```

```
print("Kurniawan")
```

```
print("L200214253")
```

```
L200214253
>>> %Run 'no 7.py'
Tinggi maksimal dari Binary Tree adalah 4
Kurniawan
L200214253
>>>
```

8. 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):

    if sub is not None:

        postorder(sub.kiri)

        postorder(sub.kanan)

        print(sub.data)

```

def traverse(root):

    lvlist=[]

    current_level = [root]

    lv=0

    while 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 cetak(rot):

    traverse(A)

    print(root.data, ', Level 0')    for i in range(len(level)):

        print(datalist[i+1], ', Level', level[i])

cetak(A)

```

```
print("=====preorder=====")
```

```
preorder(A)
```

```
print("=====inorder=====")
```

```
inorder(A)
```

```
print("=====postorder=====")
```

```
postorder(A)
```

```
print("Kurniawan")
```

```
print("L200214253")
```

```
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
Enrekang
Bantul
Cimahi
Ambarawa
Kurniawan
L200214253
```