

LAPORAN PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
MODUL 9.2
POHON BINER



Disusun oleh:
Bima Triadmaja
L200210137
E

TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
2022/2023

9.6 Soal-soal untuk Mahasiswa

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

a) $n = 10$

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

b) $n = 35$

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

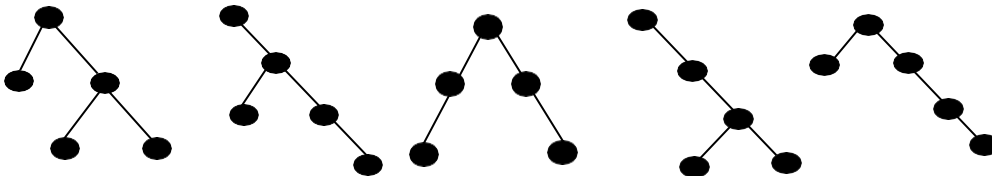
c) $n = 76$

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

d) $n = 345$

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

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



$$\begin{aligned}
 C_n &= (2n)! / (n+1)! * n! \\
 C_5 &= (2*5)! / (5+1)! * 5! \\
 &= 10! / 6! * 5! \\
 &= 3628800 / 86400 \\
 &= 42 \text{ kemungkinan}
 \end{aligned}$$

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

a) $h = 3$

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

b) $h = 4$

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

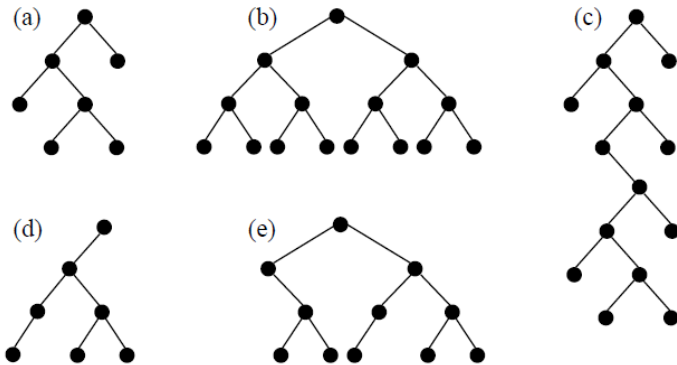
c) $h = 5$

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

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 : *penuhi* , *sempurna*, *komplit*. Ingat bahwa sebuah pohon biner bisa saja bersifat penuh sekaligus sempurna dan sebagainya.

- a = penuh
- b = sempurna
- c = komplit dan penuh
- d = komplit
- e = komplit

b. Tentukan ukuran tiap pohon.

- a = 7
- b = 15
- c = 14
- d = 7
- e = 11

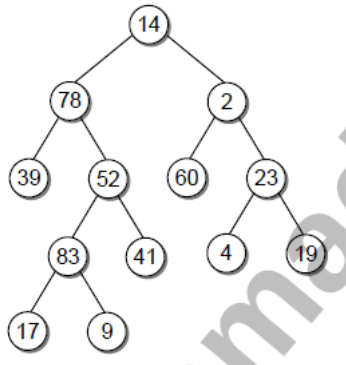
c. Tentukan ketinggian tiap pohon.

- 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. $83 = 14 - 78 - 52 - 83$

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 pemrograman

Nomor 6 :

#NO.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))

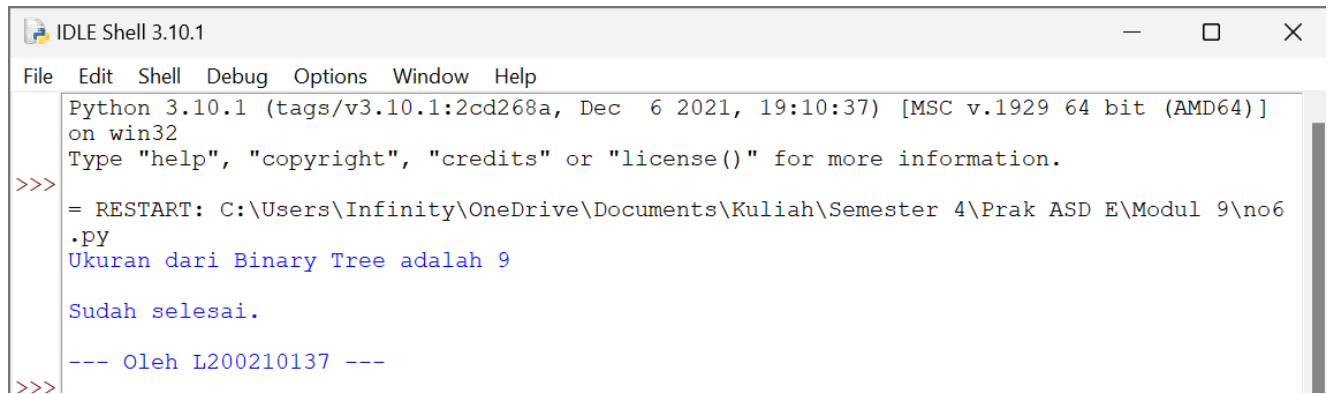
a = SimpulPohonBiner('Klaten')
b = SimpulPohonBiner('Bekasi')
c = SimpulPohonBiner('Jakarta')
d = SimpulPohonBiner('Depok')
e = SimpulPohonBiner('Tangerang')
f = SimpulPohonBiner('Solo')
g = SimpulPohonBiner('Surakarta')
h = SimpulPohonBiner('Cilacap')
i = SimpulPohonBiner('Trenggalek')
j = SimpulPohonBiner('Yogyakarta')

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('')
```

```
print('Sudah selesai.')

print('\n--- Oleh L200210137 ---')
```



```
IDLE Shell 3.10.1
File Edit Shell Debug Options Window Help
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Infinity\OneDrive\Documents\Kuliah\Semester 4\Prak ASD E\Modul 9\n6
.py
Ukuran dari Binary Tree adalah 9

Sudah selesai.

--- Oleh L200210137 ---
>>>
```

Nomor 7 :

#NO.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('Klaten')
b = SimpulPohonBiner('Bekasi')
c = SimpulPohonBiner('Jakarta')
d = SimpulPohonBiner('Depok')
e = SimpulPohonBiner('Tangerang')
f = SimpulPohonBiner('Solo')
g = SimpulPohonBiner('Surakarta')
h = SimpulPohonBiner('Cilacap')
i = SimpulPohonBiner('Trenggalek')
j = SimpulPohonBiner('Yogyakarta')

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))
print('')
print('Sudah selesai.')

print('\n--- Oleh L200210137 ---')

```

```

>>> = RESTART: C:\Users\Infinity\OneDrive\Documents\Kuliah\Semester 4\Prak ASD E\Modul 9\n07
.py
Tinggi maksimal dari Binary Tree adalah 4
Sudah selesai.
|
--- Oleh L200210137 ---
>>>

```

Nomor 8 :

#NO.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('Klaten')
B=simpulbiner('Sukoharjo')
C=simpulbiner('Surakarta')
D=simpulbiner('Jakarta')
E=simpulbiner('Solo')
H=simpulbiner('Yogyakarta')

```

```

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

cetakDataDanLevel(A)
print('')

```



```

print("-"*10,"preorder traversal","-"*10)
preorder(A)
print("-"*10,"inorder traversal","-"*10)
inorder(A)
print("-"*10,"postorder traversal","-"*10)
postorder(A)
print('')
print('Sudah selesai.')

print('\n--- Oleh L200210137 ---')

```

```

>>> = RESTART: C:\Users\Infinity\OneDrive\Documents\Kuliah\Semester 4\Prak ASD E\Modul 9\n08
.py
Klaten , Level 0
Sukoharjo , Level 1
Surakarta , Level 1
Jakarta , Level 2
Solo , Level 2
Yogyakarta , Level 3

----- preorder traversal -----
Klaten
Sukoharjo
Jakarta
Yogyakarta
Solo
Surakarta
----- inorder traversal -----
Yogyakarta
Jakarta
Sukoharjo
Solo
Klaten
Surakarta
----- postorder traversal -----
Yogyakarta
Jakarta
Solo
Sukoharjo
Surakarta
Klaten

Sudah selesai.

--- Oleh L200210137 ---
>>>

```