**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 = INT[log2 10]+1 = 4
        + jumlah level maximum = (level 0 samapi level 9) = 10

❖ n = 35

* + - * + jumlah level minimum = INT[log2 35]+1 = 6
        + jumlah level maximum = (level 0 sampai level 34) = 35

❖ n = 76

* + - * + jumlah level minimum = INT[log2 76]+1 = 7
        + jumlah level maximum = (level 0 sampai level 75) = 76

❖ n = 345

* + - * + jumlah level minimum = INT[log2 345]+1 = 9
        + jumlah level maximum = (level 0 sampai level 344) = 345

1. Ada berapa kemungkinan gambar yang dapat dibentuk dari pohon binner berukuran 5. Cn = (2n)! / (n+1)! \* n!

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

= 10! / 6! \* 5!

= 3628800 / 86400

= 42 kemungkinan

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

Jumlah max simpul = level 0 + level 1 + level2 = 20 + 21 + 22 = 7 b. h = 4

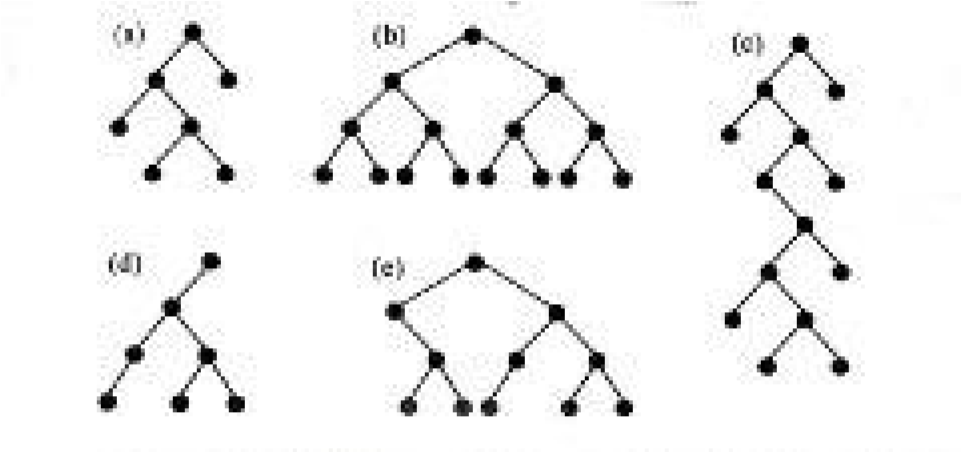
Jumlah max simpul = level 0 + level 1 +level2 + level3 = 20 + 21 + 22 + 23= 15 c. h = 5

Jumlah max simpul = level 0 + level 1 +level2 + level 3 + level 4 = 20 + 21 + 22 + 23 + 24 = 31 d. h = 6

Jumlah max simpul = level 0 + level 1 +level2 + level 3 + level 4 + level 5 = 20 +

21 + 22 + 23 + 24 + 25 = 63

1. Diberikan pohon-pohon biner seperti di bawah



* 1. 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 = komplit dan penuh

d = komplit

e = komplit

* 1. Tentukan ukuran tiap pohon.

a = 7

b = 15

c = 14

d = 7

e = 11

* 1. Tentukan ketinggian tiap pohon.

a =4

b= 4

c =8

d= 4

e=4

* 1. Tentukan lebar tiap pohon.

a=2

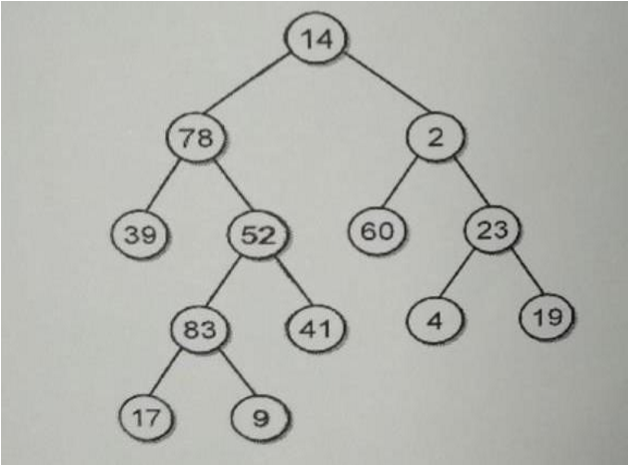
b=8

c=2

d=3

e=5

1. Perhatikan pohon biner berikut.



a. Tunjukan 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

1. 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

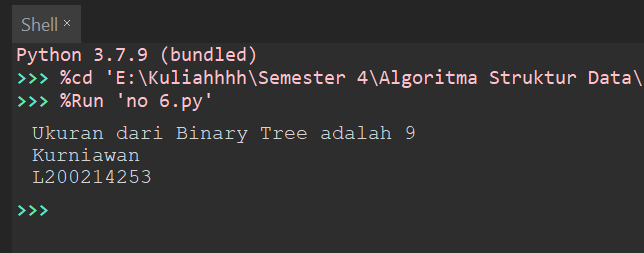
e.kiri = h;

g.kanan = i

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

print("Kurniawan")

print("L200214253")



1. 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')

1. 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

1. b.kiri = d; b.kanan = e

c.kiri = f; c.kanan = g

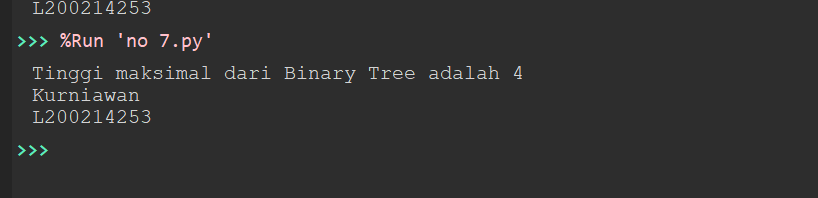
e.kiri = h;

g.kanan = i

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

print("Kurniawan")

print("L200214253")



1. 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")

