

PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
(Algorithm and Data Structure)

LAPORAN TUGAS
MODUL 9



Nama : Shafa Bani Saputra

NIM : L200190151

Kelas : G

PROGRAM STUDI INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH
SURAKARTA

1. Berapakah jumlah level minimum dan maksimum yang bisa dimuat dengan ukuran (n)

a) $n=10$ tinggi max=10

$$\text{tinggi min} = \lceil \log_2 10 \rceil + 1 = 4$$

b) $n=35$ tinggi max=35

$$\text{tinggi min} = \lceil \log_2 35 \rceil + 1 = 6$$

c) $n=76$ tinggi max=76

$$\text{tinggi min} = \lceil \log_2 76 \rceil + 1 = 7$$

d) $n=345$ tinggi max=345

$$\text{tinggi min} = \lceil \log_2 345 \rceil + 1 = 9$$

2. Gambar semua bentuk kemungkinan pohon biner yang berukuran 5 $C_n =$

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

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

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

$$= 3628800 / 86400$$

$$= 42 \text{ kemungkinan}$$

3. Berapakah jumlah simpul maksimum dengan jumlah level (h)

a) $h=3$ jumlah simpul max = $2^0 + 2^1 + 2^2 = 7$

b) $h=4$ jumlah simpul max = $2^0 + 2^1 + 2^2 + 2^3 = 15$

c) $h=5$ jumlah simpul max = $2^0 + 2^1 + 2^2 + 2^3 + 2^4 = 31$

d) $h=6$ jumlah simpul max = $2^0 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5 = 63$

4. Dierikan pohon-pohon berikut

- a) Tunjukkan semua property Struktural yang berlaku pada tiap tiap pohon diatas penuh, sempurna, komplit. Ingat bahwa sebuah pohon biner bisa saja bersifat penuh sekaligus sempurna dan sebaliknya.

- i. Struktur pohon = penuh, komplit

- ii. Struktur pohon = sempurna, penuh, komplit
- iii. Struktur pohon = penuh, komplit
- iv. Struktur pohon = -
- v. Struktur pohon = komplit

b) Tentukan Ukuran Pohon

- i. Ukuran pohon = 7
- ii. Ukuran pohon = 15
- iii. Ukuran pohon = 14
- iv. Ukuran pohon = 7
- v. Ukuran pohon = 11

c) Tentukan ketinggian tiap pohon

- i. Ketinggian pohon = 4
- ii. Ketinggian pohon = 4
- iii. Ketinggian pohon = 8
- iv. Ketinggian pohon = 4
- v. Ketinggian pohon = 4

d) Tentukan lebar pohon

- i. Lebar pohon = 2
- ii. Lebar pohon = 8
- iii. Lebar pohon = 2
- iv. Lebar pohon = 3
- v. Lebar pohon = 4

5. Perhatikan pohon biner berikut

a) Tunjukkan urutan pengujian simpul untuk :

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

iii. Postorder traversal = 39 17 83 78 52 9 41 14 60 4 2 23 19

b) Simpul daun = 39 41 17 9 60 4 19

c) Simpul dalam = 14 78 52 83 2 23

d) Simpul yang berada di level 4 = 17 9

e) Tulis semua simpul yang berada didalam jalur dari simpul akar menuju simpul :

i. Akar-83= 14 78 52 83

ii. Akar-39= 14 78 39

iii. Akar-4= 14 2 23 4

iv. Akar-9= 14 78 52 83 9

f) Simpul 52

i. Keturunannya= 78

ii. Leluhurnya= 78 14

iii. Saudaranya= 39

g) Menentukan kedalaman simpul

i. 78 dengan kedalaman 1

ii. 41 dengan kedalaman 3

iii. 60 dengan kedalaman 2

iv. 19 dengan kedalaman 3

6.

```

File Edit Format Run Options Window Help
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('Cinahi')
D=simpulbiner('Denpasar')
E=simpulbiner('Enrekang')
H=simpulbiner('Halimahera Timur')

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

##6
def size(node):
    if node is None:
        return 0
    else:
        return (size(node.kiri)+ 1 + size(node.kanan))

print('Ukuran dari Binary Tree ini adalah', size(A))

IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
===== RESTART: E:\KULIAH\SEMESTER 4\Prak.Algostruk\Modul 9\ASDModul9.py =====
>>>
===== RESTART: E:\KULIAH\SEMESTER 4\Prak.Algostruk\Modul 9\ASDModul9.py =====
=
Ukuran dari Binary Tree ini adalah 6
>>>

```

7.

```

'''
===== RESTART: E:\KULIAH\SEMESTER 4\Prak.Algostruk\Modul 9\ASDModul9.py =====
=
Tinggi max Binnary Tree ini adalah 4
>>>
'''
*ASDModul9.py - E:\KULIAH\SEMESTER 4\Prak.Algostruk\Modul 9\ASDModul9.py (3.9.2)*
File Edit Format Run Options Window Help
###7
def tinggipohon(a):
    if a is None:
        return 0 ;

    else :
        kirtinggi = tinggipohon(a.kiri)
        kantinggi = tinggipohon(a.kanan)

        if (kirtinggi > kantinggi):
            return kirtinggi+1
        else:
            return kantinggi+1

print("Tinggi max Binnary Tree ini adalah",tinggipohon(A))

```

8.

```

####
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

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(root):
    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)

```

```

===== RESTART: E:\KULIAH\SEMESTER 4\Prak.Algostruk\Modul 9\ASIModul9.py =====
>>>
===== RESTART: E:\KULIAH\SEMESTER 4\Prak.Algostruk\Modul 9\ASIModul9.py =====
=
Urutan dari Binary Tree ini adalah 4
>>>
===== RESTART: E:\KULIAH\SEMESTER 4\Prak.Algostruk\Modul 9\ASIModul9.py =====
=
Tinggi max Binary Tree ini adalah 4
>>>
===== RESTART: E:\KULIAH\SEMESTER 4\Prak.Algostruk\Modul 9\ASIModul9.py =====
=
Adasawa , Level 0
Bantul , Level 1
Cinahi , Level 1
Dengasar , Level 2
Enrekang , Level 2
Halmahera Timur , Level 3
=====*preorder*=====
Adasawa
Bantul
Dengasar
Halmahera Timur
Enrekang
Cinahi
=====*inorder*=====
Halmahera Timur
Dengasar
Bantul
Enrekang
Adasawa
Cinahi
=====*postorder*=====
Halmahera Timur
Dengasar
Enrekang
Bantul
Cinahi
Adasawa
>>>

```

```

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(root):
    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)

```