

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN STRUKTUR DATA**  
**MODUL 9**



**DISUSUN OLEH :**

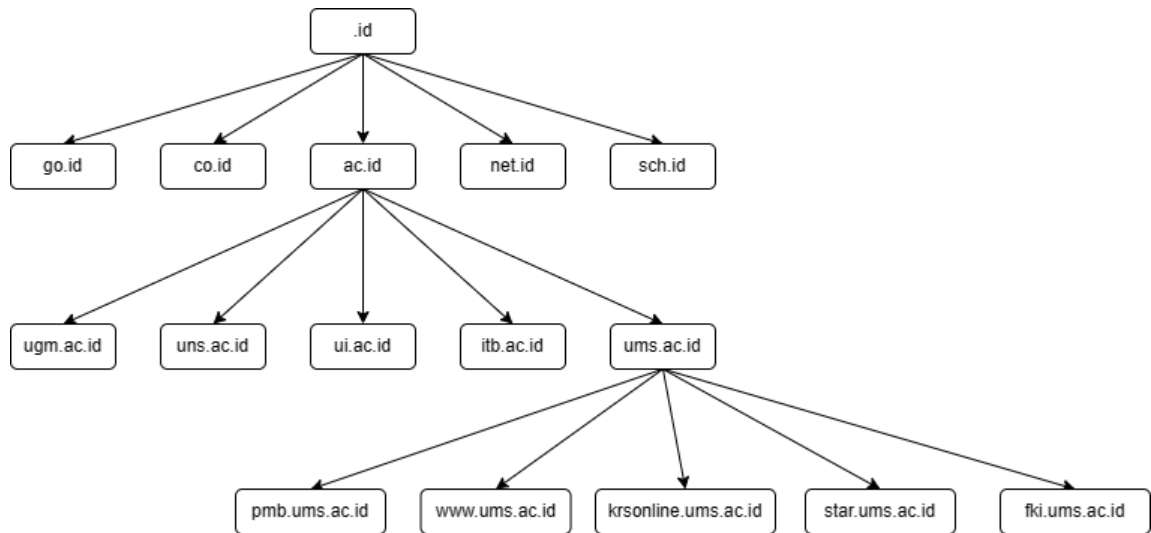
<b>NIM</b>	<b>:</b>	<b>L200210021</b>
<b>NAMA</b>	<b>:</b>	<b>Muhammad Irfan Abidin</b>
<b>KELAS</b>	<b>:</b>	<b>B</b>

**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS KOMUNIKASI DAN INFORMATIKA**  
**UNIVERSITAS MUHAMMADIYAH SURAKARTA**

## Pohon Biner

### Latihan

**9.1** Sebuah contoh pohon yang lain adalah struktur pengalamatan domain di web. Misal domain .id berperan sebagai akarnya. Maka di bawahnya ada sub-domain seperti go.id, co.id, ac.id, net.id, sch.id. Dan, contoh berikutnya, di bawah ac.id ada domain seperti ugm.ac.id, uns.ac.id, ui.ac.id, itb.ac.id, ums.ac.id. Di bawah ums.ac.id ada (di antaranya) pmb.ums.ac.id, www.ums.ac.id, krsonline.ums.ac.id, star.ums.ac.id, fki.ums.ac.id. Sekarang gambarlah struktur pohon domain-domain ini seingat kamu.



**9.2** Hitunglah ketinggian minimum sebuah pohon biner dengan ukuran berikut dan buatlah sketsa pohon biner untuk masing-masing ukuran.

- $2 = \log_2(2 + 1) = \log_2(3) \approx 1.585 = 2$
- $3 = \log_2(3 + 1) = \log_2(4) = 2$
- $5 = \log_2(5 + 1) = \log_2(6) \approx 2.585 = 3$
- $7 = \log_2(7 + 1) = \log_2(8) = 3$
- $8 = \log_2(8 + 1) = \log_2(9) \approx 3.169 = 4$
- $11 = \log_2(11 + 1) = \log_2(12) \approx 3.585 = 4$
- $12 = \log_2(12 + 1) = \log_2(13) \approx 3.700 = 4$
- $15 = \log_2(15 + 1) = \log_2(16) = 4$
- $20 = \log_2(20 + 1) = \log_2(21) \approx 4.397 = 5$
- $31 = \log_2(31 + 1) = \log_2(32) = 5$

### 9.3 Kerjakan yang berikut ini

- Dengan mengacu pada kode di atas, buatlah pohon biner yang diilustrasikan pada Gambar 9.5b dan Gambar 9.5c.

```
modul09 - class__simpulPohonBiner.py

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

    def preorderTrav(subpohon):
        if subpohon is not None:
            print(subpohon.data)
            preorderTrav(subpohon.kiri)
            preorderTrav(subpohon.kanan)

    def inorderTrav(subpohon):
        if subpohon is not None:
            inorderTrav(subpohon.kiri)
            print(subpohon.data)
            inorderTrav(subpohon.kanan)

    def postorderTrav(subpohon):
        if subpohon is not None:
            postorderTrav(subpohon.kiri)
            postorderTrav(subpohon.kanan)
            print(subpohon.data)
```

```
modul09 - Lat_9_4.py

from class__simpulPohonBiner import _SimpulPohonBiner, preorderTrav, inorderTrav, postorderTrav

# Membuat simpul-simpul dan mengisi data
a = _SimpulPohonBiner('Ambarawa')
b = _SimpulPohonBiner('Bantul')
c = _SimpulPohonBiner('Cimahi')
d = _SimpulPohonBiner('Denpasar')
e = _SimpulPohonBiner('Enrekang')
f = _SimpulPohonBiner('Flores')
g = _SimpulPohonBiner('Garut')
h = _SimpulPohonBiner('Halimahera Timur')
i = _SimpulPohonBiner('Indramayu')
j = _SimpulPohonBiner('Jakarta')

# a
print("pohon biner a")
a.kiri = b; a.kanan = c
b.kiri = d; b.kanan = e
c.kiri = f; c.kanan = g
e.kiri = h
g.kiri = i

# b
print("\npohon biner b")
a.kiri = b; a.kanan = c
b.kiri = d; b.kanan = e
c.kanan = f
e.kiri = g
g.kanan = h
h.kiri = i

# c
print("\npohon biner c")
a.kanan = b
b.kanan = c
c.kiri = d
d.kanan = e
e.kiri = f
f.kanan = g
g.kiri = h
```

9.4 Perhatikan kembali kode pada halaman 93 (contoh membangun sebuah pohon biner). Cetaklah semua data di simpul-simpulnya dengan cara preorder traversal, inorder traversal, dan postorder traversal. Verifikasilah bahwa yang tercetak adalah sesuai dengan tiga gambar keterangan yang sudah disampaikan.

#### Code program

```
modul09 - Lat_9_4.py

from class__simpulPohonBiner import _SimpulPohonBiner, preorderTrav, inorderTrav, postorderTrav

# Membuat simpul-simpul dan mengisi data
a = _SimpulPohonBiner('Ambarawa')
b = _SimpulPohonBiner('Bantul')
c = _SimpulPohonBiner('Limahi')
d = _SimpulPohonBiner('Denpasar')
e = _SimpulPohonBiner('Enrekang')
f = _SimpulPohonBiner('Flores')
g = _SimpulPohonBiner('Garut')
h = _SimpulPohonBiner('Halmahera Timur')
i = _SimpulPohonBiner('Indramayu')
j = _SimpulPohonBiner('Jakarta')

# a
print("pohon biner a")
a.kiri = b; a.kanan = c
b.kiri = d; b.kanan = e
c.kiri = f; c.kanan = g
e.kiri = h
g.kiri = i; g.kanan = j

print("preorderTrav : ")
preorderTrav(a)
print("\ninorderTrav : ")
inorderTrav(a)
print("\npostorderTrav : ")
postorderTrav(a)

# b
print("\npohon biner b")
a.kiri = b; a.kanan = c
b.kiri = d; b.kanan = e
c.kanan = f
e.kiri = g
g.kanan = h
h.kiri = i

print("preorderTrav : ")
preorderTrav(a)
print("\ninorderTrav : ")
inorderTrav(a)
print("\npostorderTrav : ")
postorderTrav(a)

# c
print("\npohon biner c")
a.kanan = b
b.kanan = c
c.kiri = d
d.kanan = e
e.kiri = f
f.kanan = g
g.kiri = h

print("preorderTrav : ")
preorderTrav(a)
print("\ninorderTrav : ")
inorderTrav(a)
print("\npostorderTrav : ")
postorderTrav(a)
```

## Output

### Pohon biner a:

- preorderTrav

```
pohon biner a
preorderTrav :
Ambarawa
Bantul
Denpasar
Enrekang
Halmahera Timur
Cimahi
Flores
Garut
Indramayu
```

- inorderTrav

```
inorderTrav :
Denpasar
Bantul
Halmahera Timur
Enrekang
Ambarawa
Flores
Cimahi
Indramayu
Garut
```

- postorderTrav

```
postorderTrav :
Denpasar
Halmahera Timur
Enrekang
Bantul
Flores
Indramayu
Garut
Cimahi
Ambarawa
```

### Pohon biner b:

- preorderTrav

```
pohon biner b
preorderTrav :
Ambarawa
Bantul
Denpasar
Enrekang
Garut
Indramayu
Halmahera Timur
Indramayu
Cimahi
Flores
Flores
```

- inorderTrav

```
inorderTrav :
Denpasar
Bantul
Indramayu
Garut
Indramayu
Halmahera Timur
Enrekang
Ambarawa
Flores
Cimahi
Flores
```

- postorderTrav

```
postorderTrav :
Denpasar
Indramayu
Indramayu
Halmahera Timur
Garut
Enrekang
Bantul
Flores
Flores
Cimahi
Ambarawa
```

**Pohon biner c:**

- preorderTrav

```
pohon biner c
preorderTrav :
Ambarawa
Bantul
Denpasar
Enrekang
Flores
Garut
Halmahera Timur
Indramayu
Halmahera Timur
Indramayu
Cimahi
Denpasar
Enrekang
Flores
Garut
Halmahera Timur
Indramayu
Halmahera Timur
Indramayu
Flores
Garut
Halmahera Timur
Indramayu
Halmahera Timur
Indramayu
Bantul
Denpasar
Enrekang
Flores
Garut
Halmahera Timur
Indramayu
Halmahera Timur
Indramayu
Cimahi
Denpasar
Enrekang
Flores
Garut
Halmahera Timur
Indramayu
Halmahera Timur
Indramayu
Flores
Garut
Halmahera Timur
```

- inorderTrav

```
inorderTrav :
Denpasar
Flores
Indramayu
Halmahera Timur
Garut
Indramayu
Halmahera Timur
Enrekang
Bantul
Denpasar
Flores
Indramayu
Halmahera Timur
Garut
Indramayu
Halmahera Timur
Enrekang
Cimahi
Flores
Indramayu
Halmahera Timur
Garut
Indramayu
Halmahera Timur
Ambarawa
Denpasar
Flores
Indramayu
Halmahera Timur
Garut
Indramayu
Halmahera Timur
Enrekang
Bantul
Denpasar
Flores
Indramayu
Halmahera Timur
Garut
Indramayu
Halmahera Timur
Enrekang
Cimahi
Flores
Indramayu
Halmahera Timur
Garut
```

- postorderTrav

```
postorderTrav :
Indramayu
Halmahera Timur
Halmahera Timur
Indramayu
Halmahera Timur
Garut
Flores
Enrekang
Denpasar
Indramayu
Halmahera Timur
Indramayu
Halmahera Timur
Garut
Flores
Cimahi
Bantul
Indramayu
Halmahera Timur
Garut
Flores
Enrekang
Denpasar
Indramayu
Halmahera Timur
Indramayu
Halmahera Timur
Garut
Flores
Cimahi
Bantul
Ambarawa
```

## Tugas

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

- $n = 10$ 
  - Jumlah level minimum =  $\log_2(10+1) = 4$
  - Jumlah level maksimum = 10
- $n = 35$ 
  - Jumlah level minimum =  $\log_2(35+1) \approx 5.129 = 6$
  - Jumlah level maksimum = 35
- $n = 76$ 
  - Jumlah level minimum =  $\log_2(76+1) \approx 6.248 = 7$
  - Jumlah level maksimum = 76
- $n = 345$ 
  - Jumlah level minimum =  $\log_2(345+1) \approx 8.438 = 9$
  - Jumlah level maksimum = 345

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

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

$$C_5 = \frac{(2*5)!}{(5+1)! * 5!}$$

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

$$= (10*9*8*7*6!) / (6! * 5!)$$

$$= (10*9*8*7) / 5!$$

$$= 5040 / 120$$

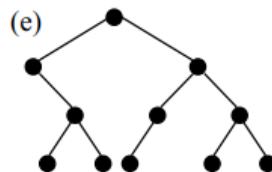
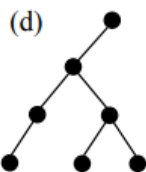
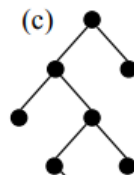
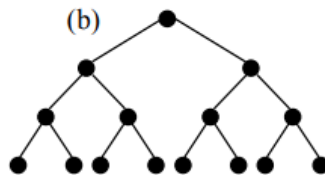
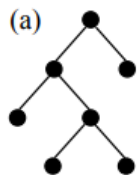
$$= 42$$

Jadi banyak kemungkinan bentuk biner berukuran 5 yaitu 42 kemungkinan.

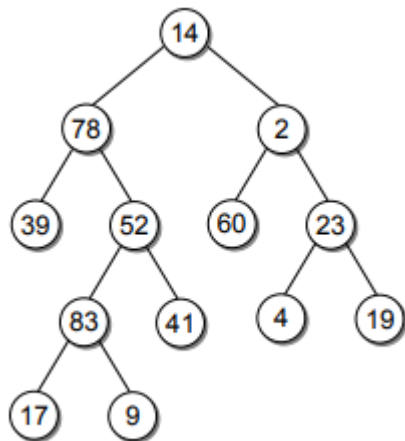
- Berapakah jumlah simpul maksimum suatu pohon biner dengan jumlah level  $h$ ? Tentukan untuk nilai  $h$  berikut.

- $h = 3$   
Jumlah simpul maks. =  $2^3 - 1 = 8 - 1 = 7$
- $h = 4$   
Jumlah simpul maks. =  $2^4 - 1 = 16 - 1 = 15$
- $h = 5$   
Jumlah simpul maks. =  $2^5 - 1 = 32 - 1 = 31$
- $h = 6$   
Jumlah simpul maks. =  $2^6 - 1 = 64 - 1 = 63$

- 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. penuh, sempurna, komplet
    - c. -
    - d. komplit
    - e. -
  - b. Tentukan ukuran tiap pohon
    - a. 7
    - b. 15
    - c. 14
    - d. 7
    - e. 11
  - c. Tentukan ketinggian tiap pohon
    - a. 3
    - b. 3
    - c. 7
    - d. 3
    - e. 3
  - 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, dan 19



- c) Simpul mana saja yang merupakan simpul dalam?  
14, 78, 2, 52, 23, dan 83
- d) Simpul mana saja yang berada di level 4?  
17 dan 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 dan 41
  - ii. Leluhurnya = 78 dan 14
  - iii. Saudaranya = 39
- g) Tentukan kedalaman dari tiap tiap simpul ini:
  - i. 78 = 1
  - ii. 41 = 3
  - iii. 60 = 2
  - iv. 19 = 3

Soal – soal pemrograman

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

**Code program**

```
modul09 - Tug_9_6.py

class Simpul:
    def __init__(self, data):
        self.data = data
        self.left = None
        self.right = None

def ukuranPohon(akar):
    if akar is None:
        return 0
    else:
        return (1 + ukuranPohon(akar.left) + ukuranPohon(akar.right))

# Membangun pohon biner
akar = Simpul(1)
akar.left = Simpul(2)
akar.right = Simpul(3)
akar.left.right = Simpul(4)
akar.right.left = Simpul(5)
akar.right.right = Simpul(6)
akar.left.right.left = Simpul(7)
akar.left.right.right = Simpul(8)
akar.right.left.left = Simpul(9)
akar.right.right.left = Simpul(10)
akar.right.right.right = Simpul(11)

# Menghitung ukuran pohon
ukuran = ukuranPohon(akar)
print("Ukuran pohon:", ukuran)
```

**Output**

```
Ukuran pohon: 11
PS D:\Kuliah\Prak. ASD>
```

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

#### Code program

```
modul09 - Tug_9_7.py

class Simpul:
    def __init__(self, data):
        self.data = data
        self.left = None
        self.right = None

def tinggiPohon(akar):
    if akar is None:
        return 0
    else:
        tinggi_kiri = tinggiPohon(akar.left)
        tinggi_kanan = tinggiPohon(akar.right)
        return max(tinggi_kiri, tinggi_kanan) + 1

# Membangun pohon biner
akar = Simpul(1)
akar.left = Simpul(2)
akar.right = Simpul(3)
akar.left.right = Simpul(4)
akar.right.left = Simpul(5)
akar.right.right = Simpul(6)
akar.left.right.left = Simpul(7)
akar.left.right.right = Simpul(8)
akar.right.left.left = Simpul(9)
akar.right.right.left = Simpul(10)
akar.right.right.right = Simpul(11)

# Menghitung ketinggian pohon
tinggi = tinggiPohon(akar)
print("Ketinggian pohon:", tinggi)
```

#### Output

```
Ketinggian pohon: 4
PS D:\Kuliah\Prak. ASD> |
```

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. Contoh sepotong hasilnya adalah seperti di bawah ini (jika kamu memakai preorder traversal).

```
>>> cetakDataDanLevel(A)
Ambarawa, level 0
Bantul, level 1
Denpasar, level 2
Enrekang, level 2
Halmahera Timur, level 3
Cimahi, level 1
```

## Code program

```
modul09 - Tug_9_8.py

class _SimpulPohonBiner(object):
    def __init__(self, data):
        self.data = data
        self.left = None
        self.right = None

    def cetakDataDanLevel(simpul, level=0):
        if simpul is None:
            return

        print(simpul.data + ", level " + str(level))

        cetakDataDanLevel(simpul.left, level + 1)
        cetakDataDanLevel(simpul.right, level + 1)

# Membuat simpul-simpul dan mengisi data
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')

# Menghubungkan simpul ortu-anak
a.left = b; a.right = c
b.left = d; b.right = e
c.left = f; c.right = g
e.left = h
g.left = i; g.right = j

cetakDataDanLevel(a)
```

## Output

```
Ambarawa, level 0
Bantul, level 1
Denpasar, level 2
Enrekang, level 2
Halmahera Timur, level 3
Cimahi, level 1
Flores, level 2
Garut, level 2
Indramayu, level 3
Jakarta, level 3
PS D:\Kuliah\Prak. ASD>
```