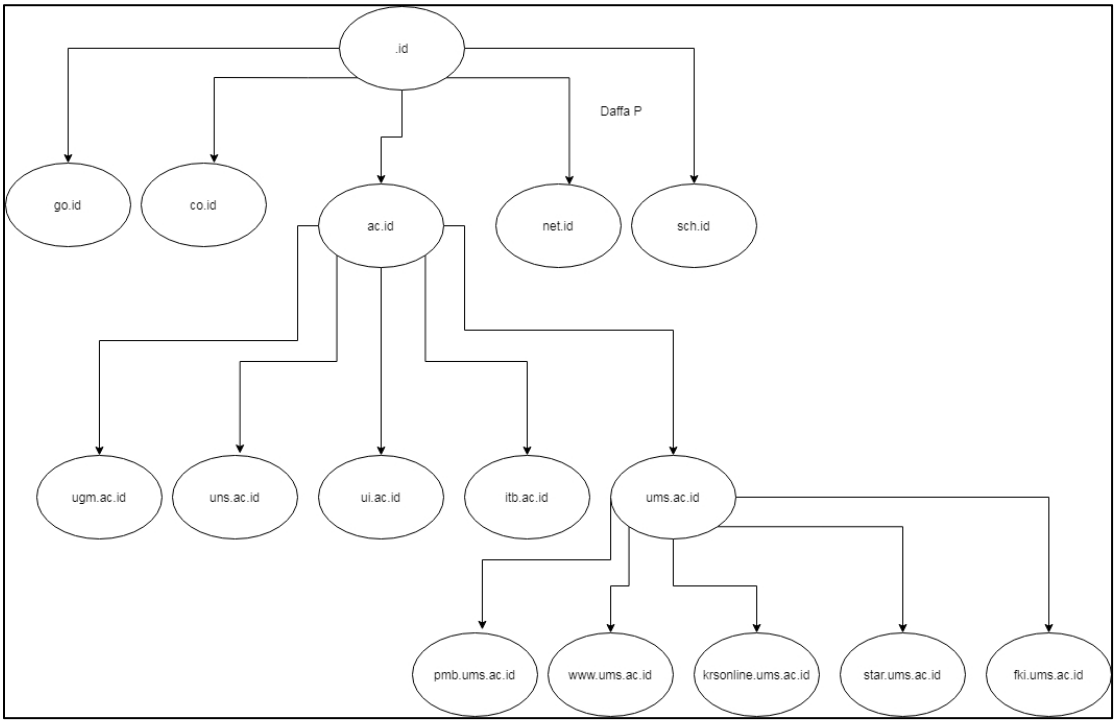
LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA
MODUL 9
”POHON BINER”

Oleh:

NAMA : Daffa Putra Alwansyah
NIM : L200190031
KELAS : B
PRODI : Informatika

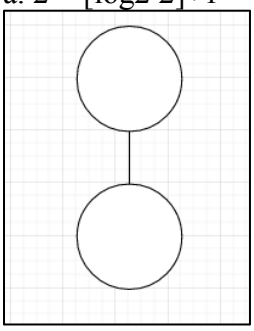
Fakultas Komunikasi dan Informatika
Universitas Muhammadiyah Surakarta

Latihan 9.1

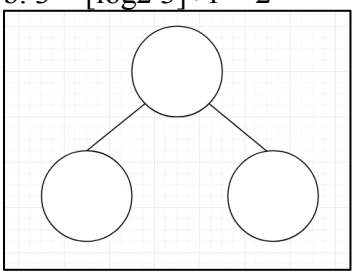


Latihan 9.2

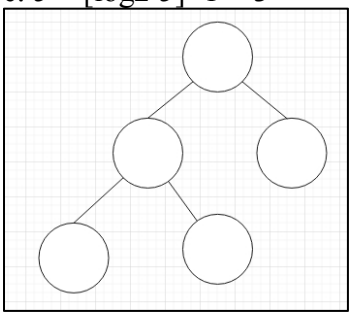
a. 2 = [log2 2]+1 = 2



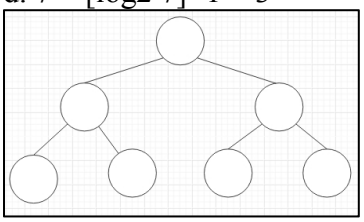
b. 3 = [log2 3]+1 = 2



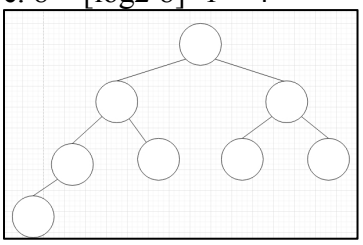
c. 5 = [log2 5]+1 = 3



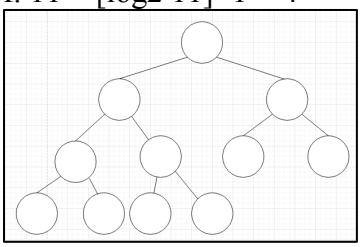
d. 7 = [log2 7]+1 = 3



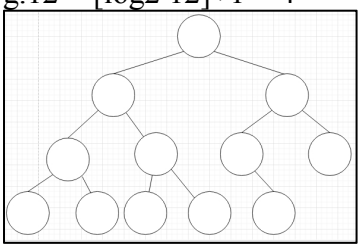
e. 8 = [log2 8]+1 = 4



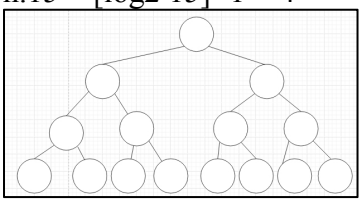
f. 11 = [log2 11]+1 = 4



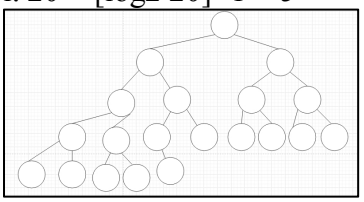
g.12 = [log2 12]+1 = 4



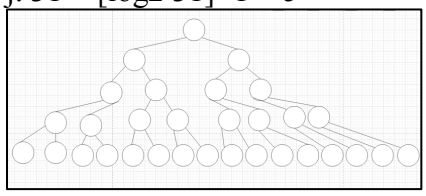
h.15 = [log2 15]+1 = 4



i. 20 = [log2 20]+1 = 5



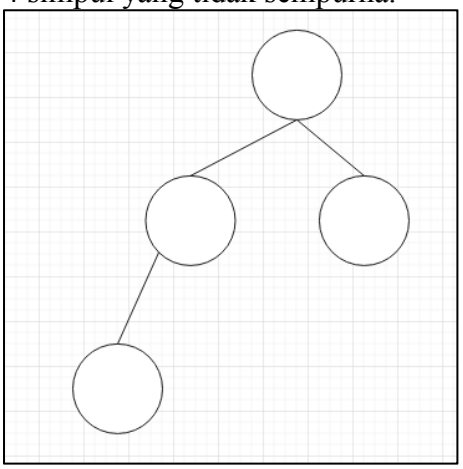
j. 31 = [log2 31]+1 = 5



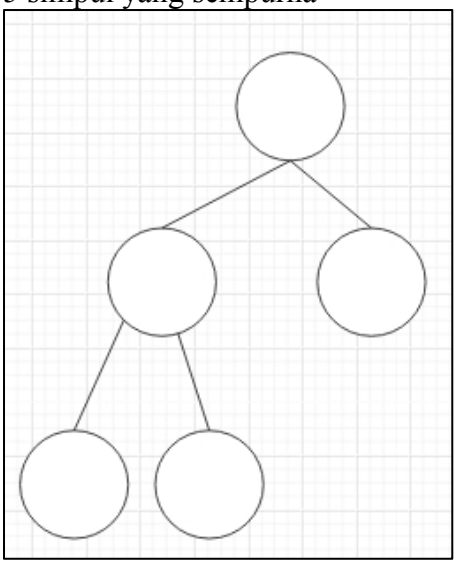
Latihan 9.3

a) Pohon biner sempurna minimal memiliki 2 anak, untuk 4 simpul belum menjadi pohon biner sempurna dan ukurannya tidak melebihi 2^n-1 (rumus pohon biner sempurna) dari ukurannya.

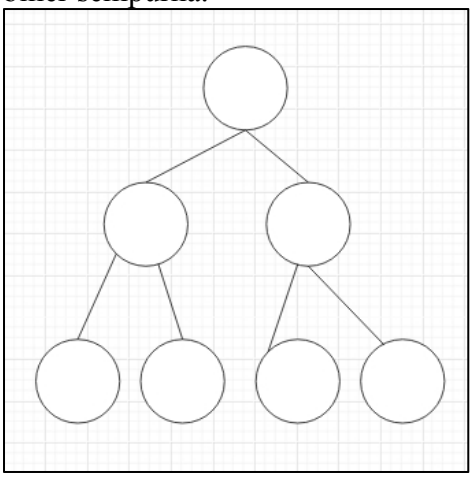
4 simpul yang tidak sempurna.



5 simpul yang sempurna

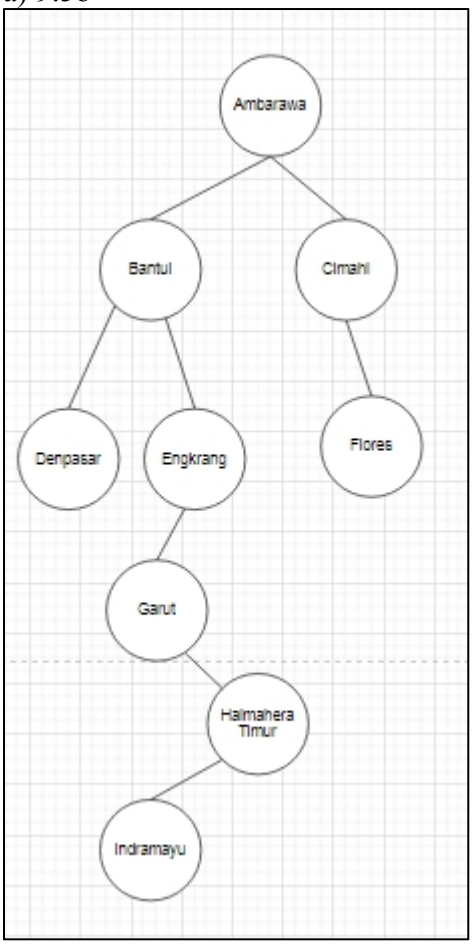


b) 2^n-1 untuk rumus pohon biner sempurna, misal angka 3 = 2^3-1 = 7. bisa dilihat gambarnya, dengan memakari rumus tersebut, berapapun angkanya akan menjadi pohon biner sempurna.

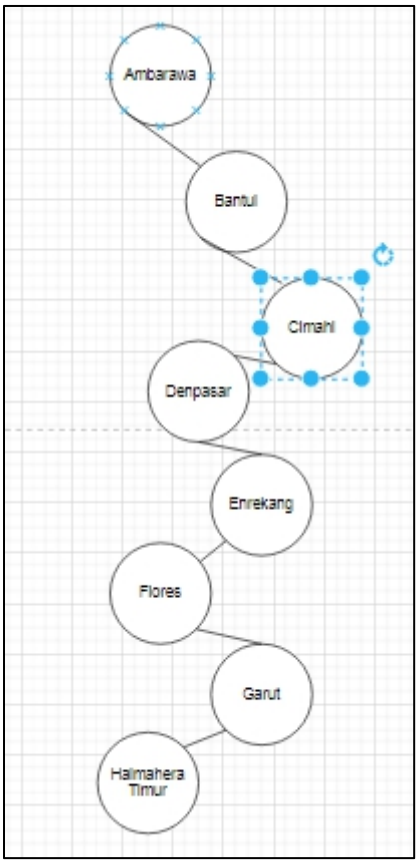


Latihan 9.4

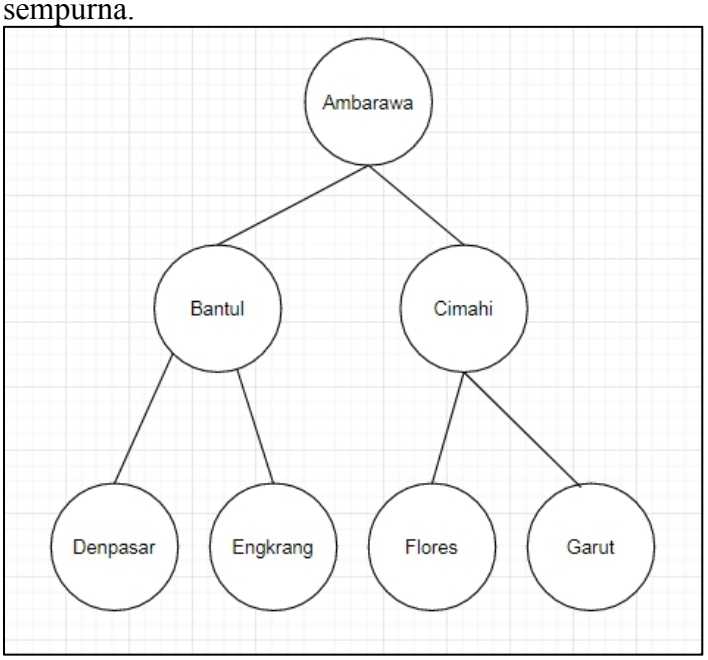
a) 9.5b



9.5b



b) Membuat pohon biner sempurna dari contoh diatas dengan rumus 2^n-1
contoh 3, 3 = 2^3-1 = 7. kenapa sebagian? Karena hanya jika tidak sebagian hasilnya tidak sempurna.



Latihan 9.5

latian.py - D:/Kuliah/Semester 4/Praktikum Algoritma dan Struktur Data/Modul9/latian.py (3.7.5)

File Edit Format Run Options Window Help

```
E = _SimpulPohonBiner('Enrekang')
F = _SimpulPohonBiner('Flores')
G = _SimpulPohonBiner('Garut')
H = _SimpulPohonBiner('Halmahera Timur')
I = _SimpulPohonBiner('Indramayu')
J = _SimpulPohonBiner('Jakarta')

# Menghubungkan simpul ortu-anak
A.kiri = B; A.kanan = C
B.kiri = D; B.kanan = E
C.kiri = F; C.kanan = G
E.kiri = H
G.kanan = I


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

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

# postorder traversal
def postorderTrav(subpohon):
    if subpohon is not None :
        postorderTrav(subpohon.kiri)
        postorderTrav(subpohon.kanan)
        print(subpohon.data)

print(preorderTrav(A))
print(inorderTrav(A))
print(postorderTrav(A))
```

```
= RESTART: D:/Kuliah/Semester 4/Praktikum Algoritma
ian.py
Ambarawa
Bantul
Denpasar
Enrekang
Halmahera Timur
Cilacap
Flores
Garut
Indramayu
None
Denpasar
Bantul
Halmahera Timur
Enrekang
Ambarawa
Flores
Cilacap
Garut
Indramayu
None
Denpasar
Halmahera Timur
Enrekang
Bantul
Flores
Indramayu
Garut
Cilacap
Ambarawa
None
>>>
```

Soal 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

- jumlah level minimum = INT[Log_2 10] + 1 = 4
- jumlah level maksimum = 10 = (level 0 samapi level 9)

b) n = 35

- jumlah level minimum = INT[Log_2 35] + 1 = 6
- jumlah level maksimum = 35 = (level 0 sampai level 34)

c) n = 76

- jumlah level minimum = INT[Log_2 76] + 1 = 7
- jumlah level maksimum = 76 = (level 0 sampai level 75)

d) n = 345

- jumlah level minimum = INT[Log_2 345] + 1 = 9
- jumlah level maksimum = 345 = (level 0 sampai level 344)

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

Cn = (2n)! / (n+1)! * n!

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

= 10! / 6! * 5!

= 3628800 / 86400

= 42 kemungkinan

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

a) $h = 3$
Jumlah maksimum simpul = level 0 + level 1 + level2
$= 2^0 + 2^1 + 2^2$
$= 7$

b) $h = 4$
Jumlah maksimum simpul = level 0 + level 1 +level2 + level3
$= 2^0 + 2^1 + 2^2 + 2^3$
$= 15$

c) $h = 5$
Jumlah maksimum simpul = level 0 + level 1 +level2 + level 3 + level 4
$= 2^0 + 2^1 + 2^2 + 2^3 + 2^4$
$= 31$

d) $h = 6$
Jumlah maksimum simpul = level 0 + level 1 +level2 + level 3 + level 4 + level 5
$= 2^0 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5$
$= 63$

4. Diberikan pohon-pohon biner seperti dibawah.

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 = 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.

a. Tunjukan 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 = 15 – 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

##========Nomor 6========

Modul 9.py - D:\Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul9\Modul 9.py (3.7.5)

File Edit Format Run Options Window Help

```
##========Nomor 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('Banyuwangi')
b = SimpulPohonBiner('Jakarta')
c = SimpulPohonBiner('Cilacap')
d = SimpulPohonBiner('Denpasar')
e = SimpulPohonBiner('Sukoharjo')
f = SimpulPohonBiner('Solo')
g = SimpulPohonBiner('Surakarta')
h = SimpulPohonBiner('Malang')

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 ini adalah', ukuranPohon(a))
```

Output:

```
= RESTART: D:/Kuliah/Semester 4/Praktiku
ul 9.py
Ukuran dari Binary Tree ini adalah 8
>>>
```

##========Nomor 7========

Modul 9.py - D:\Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul9\Modul 9.py (3.7.5)

File Edit Format Run Options Window Help

```
##========Nomor 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('Banyuwangi')
b = SimpulPohonBiner('Jakarta')
c = SimpulPohonBiner('Cilacap')
d = SimpulPohonBiner('Denpasar')
e = SimpulPohonBiner('Sukoharjo')
f = SimpulPohonBiner('Solo')
g = SimpulPohonBiner('Surakarta')
h = SimpulPohonBiner('Malang')

a.kiri = b; a.kanan = c
b.kiri = d; b.kanan = e
c.kiri = f; c.kanan = g
e.kiri = h;

print('Tinggi Maksimum dari Binary Tree ini adalah', tinggiPohon(a))
```

Output:

```
= RESTART: D:/Kuliah/Semester 4/Praktikum Algori
ul 9.py
Tinggi Maksimum dari Binary Tree ini adalah 4
>>>
```

##========Nomor 8========

Modul 9.py - D:\Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul9\Modul 9.py (3.7.5)

File Edit Format Run Options Window Help

```
##========Nomor 8========

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

def cetakDataDanLevel(akar, count = 0):
    if akar is not None:
        print (akar.data + ', level '+ str(count))
        (cetakDatadanLevel(akar.kiri, count+1),
         cetakDatadanLevel(akar.kanan, count+1))

a = SimpulPohonBiner('Banyuwangi')
b = SimpulPohonBiner('Jakarta')
c = SimpulPohonBiner('Cilacap')
d = SimpulPohonBiner('Denpasar')
e = SimpulPohonBiner('Sukoharjo')
f = SimpulPohonBiner('Solo')
g = SimpulPohonBiner('Surakarta')
h = SimpulPohonBiner('Malang')

a.kiri = b; a.kanan = c
b.kiri = d; b.kanan = e
c.kiri = f; c.kanan = g
e.kiri = h;

datalist=[a.data, b.data, c.data, f.data, e.data, f.data, g.data, h.data]
level=[]

def preorder(sub):
    if sub is not None:
        print(sub.data)
        preorder(sub.kiri)
        preorder(sub.kanan)
```

Modul 9.py - D:\Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul9\Modul 9.py (3.7.5)

File Edit Format Run Options Window Help

```
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(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)
```

Output: