**ALGORITMA DAN STRUKTUR DATA**

**MODUL 9**

**POHON BINER**



**Disusun oleh:**

Muhammad Ferizal Fadhli

L200210119

D

**TEKNIK INFORMATIKA**

**FAKULTAS KOMUNIKASI DAN INFORMATIKA**

**UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**2022/2023**

# Soal-soal untuk mahasiswa

1. Diberikan pohon biner dengan ukuran *n*, berapakah jumlah level minimum yang bisa dimuatnya ? Berapakah jumlah level maksimum nya? Tentukan untuk nilai *n* berikut.
   1. n = 10
      * Level minimum = 3
      * Level maximum = 9
   2. n = 35
      * Level minimum = 6
      * Level maximum = 34
   3. n = 76
      * Level minimum = 7
      * Level maximum = 75 d) n = 345
      * Level minimum = 8
      * Level maximum = 344
2. Gambarlah semua bentuk pohon biner berukuran 5 yang mungkin. Ada berapa kemungkinan ?

Jawab :

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

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

= 10! / 6! \* 5!

= 3628800 / 86400

= 42 kemungkinan

1. Berapakah jumlah simpul maksimum suatu pohon biner dengan jumlah level *h*?

Tentukan untuk nilai *h* berikut

1. h = 3

Jumlah max simpul = level 0 + level 1 + level2

= 1 + 2 + 4

= 7

1. h = 4

Jumlah max simpul = level 0 + level 1 +level2 + level3

= 1 + 2 + 4+8

= 15

1. h = 5

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

= 1 + 2 + 4 + 8 + 16

= 31

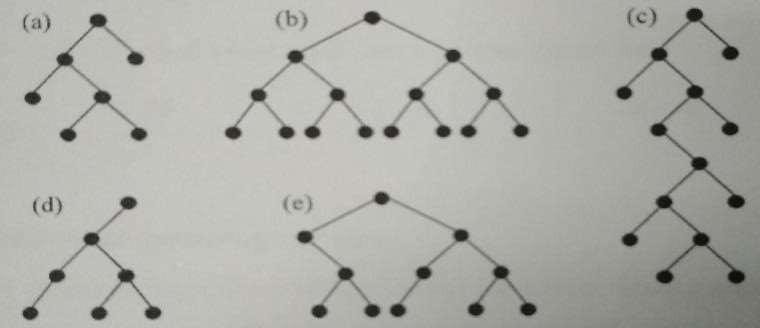
1. h = 6

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

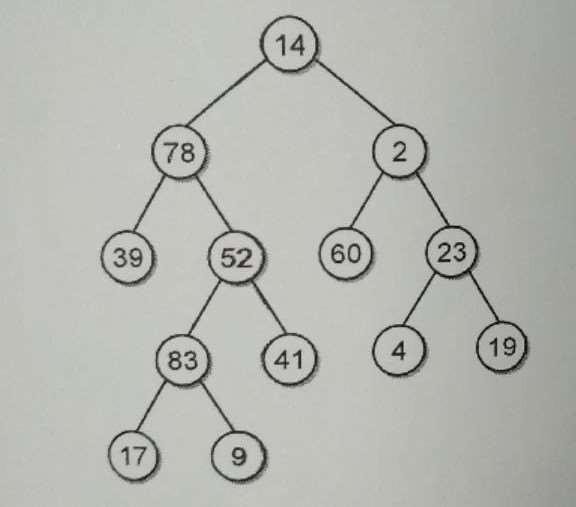
= 1 + 2 + 4 + 8 + 16 + 32

= 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
2. Tentukan ukuran tiap pohon.
   * a = 7
   * b = 15
   * c = 14
   * d = 7
   * e = 11
3. Tentukan ketinggian tiap pohon.
   * a = 4
   * b = 4
   * c = 8
   * d = 4
   * e = 4
4. Tentukan lebar tiap pohon.
   * a = 2
   * b = 8
   * c = 2
   * d = 3
   * e = 5
5. Perhatikan pohon biner berikut.



1. 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

1. Simpul mana saja yang merupakan simpul daun ? 39, 17, 9, 41, 60, 4, 19
2. Simpul mana saja yang merupakan simpul dalam ? 14,78, 52, 83, 2, 23
3. Simpul mana saja yang berada di level 4? 17, 9
4. 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

1. Perhatikan simpul 52. Tentukan
   1. Keturunannya = 83, 41
   2. Leluhurnya = 78, 14
   3. Saudaranya = 39
2. Tentukan kedalaman dari tiap-tiap simpul ini :
   1. 78 = level 1
   2. 41 = level 2
   3. 60 = level 2
   4. 19 = level 3

# Soal-soal pemograman

1. Buatlah fungsi **ukuranPohon** (akar) yang akan mendapatkan ukuran sebuah pohon biner.

Berikut Kode :

from simpul import \*

def ukuraanPohon(akar):

    if akar is None:

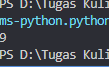
        return 0

    else:

        return 1 + ukuraanPohon(akar.kiri) + ukuraanPohon(akar.kanan)

print(ukuraanPohon(A))

Output:



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

Berikut Kode:

from simpul import \*

def tinggiPohon(akar):

    if akar is None:

        return 0

    else:

        return 1 + max(tinggiPohon(akar.kiri), tinggiPohon(akar.kanan))

print(tinggiPohon(A))

Output:



1. Buatlah sebuah fungsi yang mencetak data tiap simpul sekaligus level dimana simpul itu berada. Silahkan memilih akan memakai *preorder traversal, inorder traversal,* atau *postorder traversal.* Contoh sepotong hasilnya adalah seperti di bawah ini (jika kamu memakasi *preorder traversal*).

>>> cetakDataDanLevel(A) Ambarawa, level 0

Bantul, level 1

Denpasar, level 2

Enrekang, level 2 Halmahera Timur, level 3 Cimahi, level 1

Berikut Kode :

from simpul import \*

def cetakDataDanLevel(subPohon, level = 0):

    if subPohon is not None:

        print(subPohon.data, ",Level", level)

        cetakDataDanLevel(subPohon.kiri, level + 1)

        cetakDataDanLevel(subPohon.kanan, level + 1)

cetakDataDanLevel(A)

Output:

