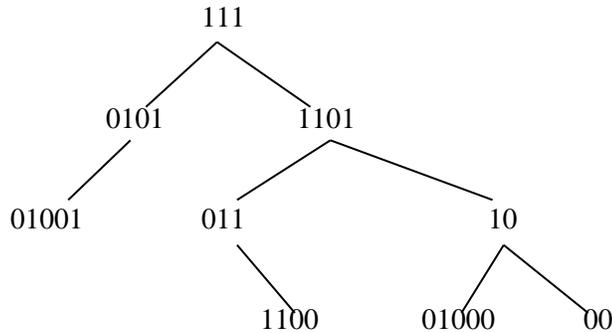


Jawaban Kuis ke-4 IF2091 Struktur Diskrit (3 SKS) : Pohon dan Kompleksitas Algoritma
Dosen: Rinaldi Munir & Harlili
Rabu, 1 Desember 2010
Waktu: 60 menit

1. Diketahui sebuah pohon dengan lintasan *preorder* dan *inorder*:
Preorder: { 111, 0101, 01001, 1101, 011, 1100, 10, 01000, 00 }
Inorder: {01001, 0101, 111, 011, 1100, 1101, 01000, 10, 00}
Buatlah *postorder* dari pohon tersebut dan gambarkan pohonnya.

Jawaban:

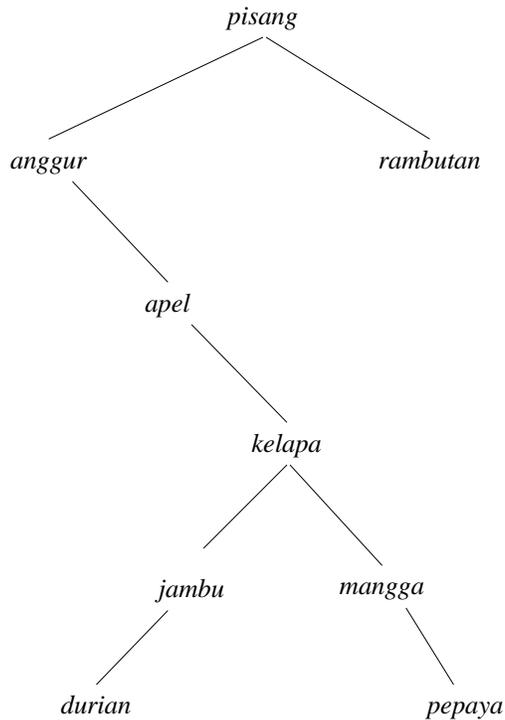
Postorder: {01001, 0101, 1100, 011, 01000, 0010, 1101, 111}



2. Bangunlah pohon pencarian (*search tree*) dengan urutan pembacaan data adalah *pisang*, *anggur*, *apel*, *kelapa*, *mangga*, *pepaya*, *jambu*, *rambutan*, dan *durian*. Berapa banyak perbandingan yang dilakukan untuk mencari kata *jeruk* di dalam pohon tersebut? (pengurutan data di dalam pohon berdasarkan huruf abjad).

Jawaban:

Pohon pencarian biner yang terbentuk:

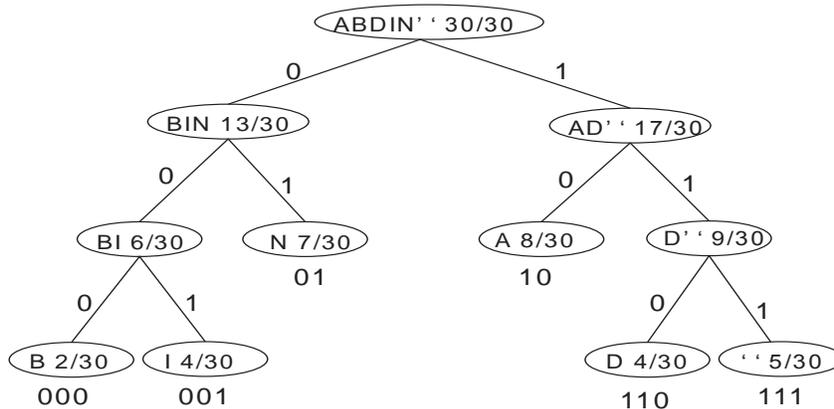


Untuk mencari kata *jeruk* di dalam pohon di atas, pencarian dimulai dengan melakukan perbandingan mulai dari akar pohon. Kata yang dibandingkan dengan *jeruk* adalah *pisang*, *anggur*, *apel*, *kelapa*, dan *jambu*. Karena anak kanan dari simpul *jambu* adalah kosong, maka pencarian berhenti sampai di sini. Jadi hanya ada 5 kali perbandingan dengan tidak menemukan kata *jeruk*.

3. Terdapat sebuah string "BIDAN ANIDA DAN NANI BINA DANA"
 - a. Hitung frekuensi kemunculan tiap karakter (termasuk spasi).
 - b. Buatlah pohon huffmannya
 - c. Hitung panjang rangkaian bit yang dihasilkan jika string di atas di ubah menjadi kode Huffman yang telah dibuat.

Jawaban:

- a. A = 8/30, B = 2/30, D = 4/30, I = 4/30, N = 7/30, ' ' = 5/30
- b. Salah satu solusi:



c. $8*A + 2*B + 4*D + 4*I + 7*N + 5*' ' = 8*2 + 2*3 + 4*3 + 4*3 + 7*2 + 5*3 = 45$

4. Apakah $T(n) = 3 + 9 + 27 + \dots + 3^n$ merupakan $O(n \cdot 3^n)$? Buktikan jawaban Anda.

Jawaban:

$T(n) = 3 + 9 + 27 + \dots + 3^n$ merupakan $O(n \cdot 3^n)$ karena dapat dipilih C dan n_0 sehingga:

$$T(n) = 3 + 9 + 27 + \dots + 3^n = \frac{3(3^{n+1}-1)}{3-1} = \frac{3}{2}(3^{n+1}-1) \leq \frac{3}{2} \cdot 3^{n+1} \leq \frac{3}{2} \cdot n \cdot 3^n \text{ untuk } n \geq 1 \text{ (} C=3/2, n_0=1\text{)}$$

Solusi Alternatif

$$3 + 9 + 27 + \dots + 3^n \leq 3^n + 3^n + 3^n + \dots + 3^n \text{ (sebanyak } n\text{)} = n \cdot 3^n \text{ untuk } n \geq 1 \text{ (} C=1, n_0=1\text{)}$$

5. Tentukan kompleksitas dalam notasi Big-O dari potongan program dalam Bahasa C dibawah ini jika melihat banyaknya jumlah proses dummy = dummy+k :

```

int i, j, k, n, dummy;
dummy=0;
for (i=1; i<=n; ++i)
{ for (j=1; j<=i; ++j)
  { for (k=1; k<=j^3; ++k)
    { dummy = dummy+k; }
  }
}

```

Jawaban:

$$T(n) = 1^3 + (1^3 + 2^3) + (1^3 + 2^3 + 3^3) + \dots + (1^3 + 2^3 + 3^3 + \dots + n^3)$$

$$T(n) = 1^3 n + 2^3 (n-1) + 3^3 (n-2) + \dots + (n-1)^3 2 + n^3$$

$$T(n) \leq n^3 n + n^3 (n-1) + n^3 (n-2) + \dots + n^3 2 + n^3$$

$$T(n) \leq n^3(n + n - 1 + n - 2 + \dots + 2 + 1)$$

$$T(n) \leq n^3 \left(\frac{n(n+1)}{2} \right)$$

$$T(n) \leq n^3 \left(\frac{n^2 + n}{2} \right)$$

$$T(n) \leq \frac{n^5 + n^4}{2}$$

Sehingga kompleksitas algoritma tersebut yaitu $O(n^5)$