Soal Esay Dynamic Programming

Koefisien binomial diberikan oleh $\begin{bmatrix} n \\ k \end{bmatrix} = \frac{n!}{k!(n-k)!}$ untuk $0 \leq k \leq n$, dan dapat diformulasikan secara rekursif sebagai berikut:

$$\begin{bmatrix} n \\ k \end{bmatrix} = \begin{cases} \begin{bmatrix} n-1 \\ k-1 \end{bmatrix} + \begin{bmatrix} n-1 \\ k \end{bmatrix} & , 0 < k < n \\ 1 & , k = 0 \text{ atau } k = n \end{cases}$$

sehingga koefisien binomial ini bisa diselesaikan dengan algoritma Dynanic Programming sebagai berikut:

```
function bin2(n, k: integer): integer;
var
  i,j: index;
  B: array[0..n,0..k] of integer;
begin
  for i:= 0 to n do
    for j:= 0 to minimum(i,k) do
      if j = 0 or j = i then
        B[i,j]:= 1
      else
        B[i,j]:= B[i−1,j−1] + B[i−1,j]
      end
    end
  end;
  bin2:= B[n,k]
end;
```

Contoh:
B[7, 6] = B[6, 5] + B[6, 6] = ?

a) Dengan algoritma di atas, hitung B[3, 2] = bin2 (3, 2) = $\begin{bmatrix} 3 \\ 2 \end{bmatrix}$ dengan menuliskan setiap B[i, j] untuk $0 \leq i \leq n$ dan $0 \leq j \leq k$ seperti pada contoh (memperlihatkan dekomposisi jika ada dan nilai akhirnya).

   Karena harus menelusuri algoritma dynamic programming, jawaban harus berurutan sebagai berikut:

   B[0,0] = 1

   Baris 1:

   B[1,0] = 1
   B[1,1] = 1

   Baris 2:

   B[2,0] = 1
   B[2,1] = B[1,0] + B[1,1] = 1 + 1 = 2
   B[2,2] = 1

   Baris 3:

   B[3,0] = 1
   B[3,1] = B[2,0] + B[2,1] = 1 + 2 = 3
   B[3,2] = B[2,1] + B[2,2] = 2 + 1 = 3

b) Karena memiliki bentuk rekursif, koefisien dapat juga diselesaikan dengan pendekatan Divide & Conquer. Mana algoritma perhitungan koefisien binomial yang lebih lebih efisien, apakah dengan pendekatan Dynamic Programming atau Divide & Conquer, dan jelaskan mengapa bisa lebih efisien?

   Dynamic Programming lebih efisien karena elemen B[x, y] yang paling sering dipakai dihitung terlebih dahulu dan disimpan nilainya untuk dipergunakan pada perhitungan elemen lainnya yang membutuhkan. Sedangkan pada Divide & Conquer terjadi pengulangan perhitungan B pada nilai x dan y yang lebih rendah.

min { C_qm + min {C_mn + f(n, null)},   C_qn + min {C_nm + f(m, null)} }

$$min\{C_{qm} + \min\{C_{mn} + f(n, \emptyset)\} \quad , \quad C_{qn} + \min\{C_{nm} + f(m, \emptyset)\}\}$$