TUGAS DAA-1

Nama : Kintan Yulia Rahayu

Nim :1116101378

Cobalah untuk membuat atau menggunakan program di atas untuk mendapatkan bilangan Fibonacci ke-0, 1, 2, 10, 50, 100, dan seterusnya. Apakah waktu yang dibutuhkan untuk mendapatkan masing-masing bilangan Fibonacci tersebut tetap sama?

# Tugas

Sederhanakan algoritma untuk mendapatkan bilangan Fibonacci ke-*n* sehingga waktu yang dibutuhkan untuk mendapatkan masing-masing bilangan relatif sama!

ALGORITMA AWAL :

class Fibonacci

**{**

**public** double get**(**int n**)**

**{**

**if** **(**n **<=** 1**)** **return** n**;**

**else** **return** get**(**n **-** 1**)** **+** get**(**n **-** 2**);**

**}**

**}**

class Program

**{**

static void Main**(**string**[]** args**)**

**{**

int n**;**

double result**;**

Fibonacci fibo **=** **new** Fibonacci**();**

Console**.**Write**(**"Request bilangan Fibonacci ke : "**);**

n **=** Int32**.**Parse**(**Console**.**ReadLine**());**

result **=** fibo**.**get**(**n**);**

Console**.**WriteLine**(**"Bilangan Fibonacci ke-{0} adalah {1}"**,**n**,**result**);**

Console**.**Read**();**

**}**

**}**

ALGORITMA yang saya temukan :

**public** static void **Main**(string[] args)  
        {  
              
            Console.**Write**("Masukkan bilangan Anda : ");  
            **int** batas = Convert.**ToInt32**(Console.**ReadLine**());  
              
            **double** a = 0;  
            **double** b = 1;  
            **int** i ;  
                  
            **for**(i = 0;i < batas;i++)  
            {  
                a = a + b ;  
                b = a - b ;  
                  
            }  
            Console.**WriteLine**("Bilangan Fibonacci ke {0} adalah {1}",i,b);  
            Console.**ReadKey**(**true**);  
        }

PENJABARAN :

Dengan menggunakan algoritma yang paling sederhana, dan menggunakan beberapa referensi dari beberapa sumber. Penggunaan sistem looping atau perulangan dapat menyederhanakan pengerjaan dan mempersingkat atau lebih ke efisien (pendapapt/opini). Di perulangan ini, penggunaan for bisa menjadi yang sederhana, bisa menemukan jumlah deret ke-50 atau bahkan ke-100 dengan cepat.

Dengan for bila batas yang dicari belum ditemukan, maka program tersebut masih terus berjalan. Namun apabila program tersebut sudah menemukan batas yang dicari maka progra tersebut berhenti dan memunculkan deret yang dicari.