**NAMA : HASAN TOLABI**

**NIM : 1116101347**

1. PENJABARAN FIBONACCI ( Pengertian)

Bilangan FIBONACCI (Kalau dalam Matematika) sering disebut dengan barisan adalah barisan yang berawal dari 0 dan 1, kemudian angka berikutnya didapat dengan cara menambahkan kedua bilangan yang berurutan sebelumnya. Dengan aturan ini, maka barisan bilangan Fibonaccci yang pertama adalah:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ……. dst

Jika diamati deret bilangan FIBONACCI di atas, maka dapat dipahami bahwa nilai bilangan FIBONACCI suku ke-n dalam deret tersebut dapat dihitung dengan menjumlahkan dua bilangan terdekat pada urutan sebelumnya.

1. ALGORITMA ITERASI

Saya menggunakan algoritma iterasi yaitu mencari perulangan dan saya menanggap ini lebih efisien daripada rekursif karena pencarian lebih cepat dan lebih efektif

Berikut algoritma yang saya gunakan adalah :

1. Jika n=0, maka F(0)=0
2. Jika n=1, maka F(1)=1
3. Jika n>1 berlaku rumus F(n-1) + F(n-2)
4. Jika n=2, maka F(2-1) + F(2-2) = F(1) + F(0) = 1 + 0 = 1
5. Jika n=3 maka F(3-1) + F(3-2) = F(2) + F(1)  = 1 + 1 = 2
6. Jika n =4 maka F(4-1) + F(4-2) = F(3) + F(2) = 2 + 1 = 3
7. Dst

Hasil deret bilangan Fibonacci adalah :  0,1,1,2,3, dst

1. Mulai
2. Menentukan nilai a=0 , b=1 , c=1;
3. Masukkan suku ke-n (jumlah)
4. Jika nilai i memenuhi nilai n maka ke langkah 7
5. Jika nilai i belum memenuhi nilai n maka di lakukan perulangan
6. I = 0 , i <=jumlah , i ++
7. Setelah mendapatkan hasil muncul deret fibonacci dengan batas atas c
8. Selesai

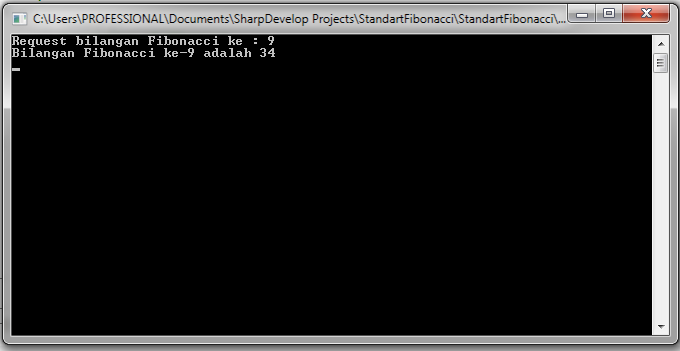
Dan ini contoh kode program dalam bentuk C#

## Class Febonacci

/\*  
 \* Created by SharpDevelop.  
 \* User: PROFESSIONAL  
 \* Date: 2017-08-13  
 \* Time: 1:07 PM  
 \*   
 \* To change this template use Tools | Options | Coding | Edit Standard Headers.  
 \*/  
**using** System;  
**using** System.Data;  
**using** MySql.Data.MySqlClient;  
  
**namespace** StandartFibonacci  
{  
    /// <**summary**>  
    /// Description of Fibonacci.  
    /// <**/summary**>  
    ///   
      
      
    **public** class Fibonacci  
    {  
        **public** **double** **get**(**int** n)  
        {  
             **if**(n==0)  
                     return 0;  
           **else** **if**(n==1)  
                     return 1;  
           **else**  
     //fungsi rekursif  
                     return **get**(n-1)+**get**(n-2);  
        }  
  
    }  
}

## Main Program

} /\*  
 \* Created by SharpDevelop.  
 \* User: PROFESSIONAL  
 \* Date: 2017-08-13  
 \* Time: 1:04 PM  
 \*   
 \* To change this template use Tools | Options | Coding | Edit Standard Headers.  
 \*/  
**using** System;  
  
**namespace** StandartFibonacci{  
    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**();  
        }  
    }  
}

Hasil percobaan : 

KESIMPULAN :

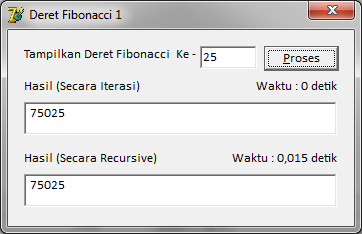
Untuk n lebih besar dari 1, maka nilainya adalah merupakan hasil penjumlahan dari 2 bilangan yang ada di depannya. Dengan aturan ini, maka barisan bilangan Fibonaccci adalah sebagai berikut:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946...

Untuk n >1, maka rumus F(n-1) + F(n-2) selalu digunakan ketika nilai n berubah. Karena itu, F(n-1) + F(n-2) dibuat menjadi sebuah fungsi. Ketika nilai n bertambah di dalam fungsi, maka nilai tersebut dikirim ke dalam fungsinya  dengan cara passing by value ketika memanggil dirinya sendiri.

1. Hasil Akhir

Berdasarkan berbagai sumber yang saya kunjungi cara iterasi tersebut dapat mempersingkat waktu pencarian bilangan fibonacci ke n , ini salah satu contoh hasil perhitungan bilangan fibonacci dengan menggunakan teknik iterasi dan teknik rekursif



1. Kekurangan karena di tugas di jabarkan bahwa harus mencari hasil ke n, dan saya hanya masih bisa mencari deret untuk hasil akhirnya, untuk ke depan bisa di perbaiki agar bisa mencari hasil akhir dengan cara iterasi