NAMA : ARTHA NUGRAHA FAJAR SIDIQ

NIM : 1117000439

1. PENJABARAN FIBONACCI

Bilangan Fibonacci adalah suatu deret bilangan bulat (*integer*) yang tak berhingga dimana bilangan tersebut secara berurutan dapat didefinisikan sebagai berikut ini :

1 1 2 3 5 8 13 21 34 55 89.....n

Deret bilangan Fibonacci di atas, maka dapat dipahami bahwa untuk mencari nilai suku ke-n pada deret bilangan Fibonacci dengan cara menjumlahkan nilai deret yang saling berdekatan.

2. ALGORITMA ITERASI

Saya menggunakan algoritma iterasi dimana cara pencariannya adalah dengan menjumlahkan nilai data terbaru (n) dengan nilai sebelum data terbaru (n-1).

Berikut algoritma yang saya gunakan

1. Start
2. Inisialisasi/menentukan nilai min (batas bawah) = 0, max (batas atas) = 1, dan value (nilai data awal) = 0.
3. Masukkan suku ke-n bilangan fibonacci yang ingin dicari.
4. Jika kondisi nilai variabel i tidak sama dengan nilai n (batas), maka proses yang dijalankan sebagai berikut :
    a. Nilai variabel i dinaikkan sebesar 1.
    b. Nilai variabel value adalah hasil penambahan nilai dari variabel min dan max.
    c. Nilai variabel min sama dengan nilai variabel max.
    d. Nilai variabel max sama dengan nilai variabel value.
5. Jika kondisi nilai variabel i sama dengan nilai n, maka proses dihentikan.
6. Hasil pencarian nilai bilangan Fibonacci pada suku ke-n ditampilkan dalam bentuk deret
7. Selesai

3. SOURCE CODE

```
package tugasdaa;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.text.DecimalFormat;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class TugasDAA {
    public static void main(String[] args) {
        int jum = 6;
        int[] n = {0, 1, 2, 10, 50, 100};
        int[] result = new int[n.length];
        String[] waktu = new String[n.length];
        BufferedReader numberinput, suminput;
        long timeStart = System.currentTimeMillis();
        if (jum <= 0) {
            System.out.print("Close Program");
        } else {
            System.out.println("(Search Fibonacci Numbers)\n");
            for (int j = 0; j < jum; j++) {
                System.out.print("Fibonacci Number " + n[j] + "\n");
                int min = 0;
                int max = 1;
                int value = 0;
                for (int i = 0; i <= n[j]; i++) {
                    result[j] = min;
                    System.out.print(min + " ");
                    value = min + max;
                    min = max;
                    max = value;
                }
                long timeFinish = System.currentTimeMillis();
                double newRunTime = (double) (timeFinish - timeStart) /
1000;
                DecimalFormat runtimeDF = new DecimalFormat("##0.000000");
                System.out.println("\nDuration of execution : " +
runtimeDF.format(newRunTime) + " seconds\n");
                waktu[j] = runtimeDF.format(newRunTime) + " seconds";
            }
            System.out.println("(Result Fibonacci Numbers)\n");
            for (int j = 0; j < jum; j++) {
                System.out.println("Fibonacci Number " + n[j] + " = " +
result[j] + "( " + waktu[j] + ")");
            }
        }
    }
}
```

4. OUTPUT PROGRAM ALGORITMA ITERASI

```
(Search Fibonacci Numbers)

Fibonacci Number 0
0
Duration of execution : 0.001000 seconds

Fibonacci Number 1
0 1
Duration of execution : 0.024000 seconds

Fibonacci Number 2
0 1 1
Duration of execution : 0.025000 seconds

Fibonacci Number 10
0 1 1 2 3 5 8 13 21 34 55
Duration of execution : 0.025000 seconds

Fibonacci Number 50
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025 121393 196418 317811 514229 832040 1346269 21783
Duration of execution : 0.026000 seconds

Fibonacci Number 100
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025 121393 196418 317811 514229 832040 1346269 21783
Duration of execution : 0.028000 seconds

(Result Fibonacci Numbers)

Fibonacci Number 0 = 0( 0.001000 seconds)
Fibonacci Number 1 = 1( 0.024000 seconds)
Fibonacci Number 2 = 1( 0.025000 seconds)
Fibonacci Number 10 = 55( 0.025000 seconds)
Fibonacci Number 50 = -298632863( 0.026000 seconds)
Fibonacci Number 100 = -980107325( 0.028000 seconds)
BUILD SUCCESSFUL (total time: 0 seconds)
```

5. OUTPUT PROGRAM ALGORITMA REKURSIF

```
(Search Fibonacci Numbers)

Fibonacci Number 0
0
Duration of execution : 0.001000 seconds

Fibonacci Number 1
0 1
Duration of execution : 0.036000 seconds

Fibonacci Number 2
0 1 1
Duration of execution : 0.036000 seconds

Fibonacci Number 10
0 1 1 2 3 5 8 13 21 34 55
Duration of execution : 0.036000 seconds

Fibonacci Number 50
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025 121393 196418 317811 514229 832040 1346269 21783
Duration of execution : 0.038000 seconds

Fibonacci Number 100
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025 121393 196418 317811 514229 832040 1346269 21783
Duration of execution : 0.041000 seconds

(Result Fibonacci Numbers)

Fibonacci Number 0 = 0( 0.001000 seconds)
Fibonacci Number 1 = 1( 0.036000 seconds)
Fibonacci Number 2 = 1( 0.036000 seconds)
Fibonacci Number 10 = 55( 0.036000 seconds)
Fibonacci Number 50 = -298632863( 0.038000 seconds)
Fibonacci Number 100 = -980107325( 0.041000 seconds)
BUILD SUCCESSFUL (total time: 0 seconds)
```

6. HASIL AKHIR

Berdasarkan dari percobaan diatas, dapat saya simpulkan bahwa pencarian nilai suku ke-n pada bilangan Fibonacci lebih cepat dan lebih efisien menggunakan fungsi iterasi daripada menggunakan fungsi rekursif.