Nama : Putri Nadia Ayu Rahmawati

Nim : 1116101344

Pada minggu lalu saya mengirimkan suatu source code yang berisi progran perhitungan fibbonaci, yang melalukan teknik iterasi dengan memanfaatkan looping *for*, berikut adalah source code nya

//

**int** a=0;  
            **int** b=1, c=1;  
              
            Console.**Write**("masukkan jumlah deret fibonacci: ");  
            **int** jumlah = Convert.**ToInt32**(Console.**ReadLine**());  
              
              
            **for** ( **int** i=0; i <= jumlah; i++)  
            {  
                  
                Console.**Write**(a+" ");  
                  
                c = a + b;  
                a = b;  
                b = c;

//

Setelah mencari referensi di berbagai sumber saya menemukan bahwa rumus kompleksitas waktu sebuah algoritma adalah sebagai berikut (jumlah saya ganti N)



T(n)

N 🡨 = 🡪1

A 🡨 0 = 🡪 1

B 🡨 1 = 🡪 1

C 🡨 a+b = 🡪 N

A 🡨 b = 🡪 N

B 🡨 c = 🡪 N

Total 🡪 3+3N

Operasi Aritmatika

a+b 🡪 n

Output

A 🡪 n

|  |  |
| --- | --- |
| Variabel | Jumlah |
| D | 3+3n |
| I | N |
| A | N |
| TOTAL | 3+5n |

T(n) = (3+3n)+(n)+(n)

T(n) = 3+5n

Untuk uji coba n

|  |  |  |
| --- | --- | --- |
| N | Rumus | Hasil |
| 10 | 3+5(10) | 53 |
| 100 | 3+5(100) | 503 |
| 1000 | 3+5(1000) | 5003 |

Dari hasil ini bisa disimpulkan bahwa pertumbuhan hasil dari ujicoba tersebut nilainya konstan sesuai dengan jumlah masukkan dan menurut saya bisa di sebut 0(n)

waktu

O(n)

fibonacci

**Kekurangan**

Untuk yang ini saya hanya bisa mencari kompleksitas waktu t(n) karena keterbatasan pemahaman saya untuk menjabarkan telescoping maupun induksi matematika