



sumber referensi:

<https://www.brainacademy.id/blog/bilangan-fibonacci>

PENJELASAN STUDI KASUS

Kita ingin mengetahui berapa banyak pasangan kelinci yang akan ada setelah beberapa bulan jika setiap pasangan kelinci menghasilkan satu pasangan baru setiap bulan.

Fibonacci Rekursif

Fibonacci rekursif adalah sebuah cara untuk menghitung bilangan Fibonacci dengan menggunakan fungsi yang memanggil dirinya sendiri. Bilangan Fibonacci adalah sebuah barisan bilangan yang dimulai dari 0 dan 1, dan setiap angkanya adalah hasil penjumlahan dari dua angka sebelumnya. Misalnya, bilangan Fibonacci pertama adalah 0, kedua adalah 1, ketiga adalah $0 + 1 = 1$, keempat adalah $1 + 0 = 1$, dan seterusnya

```
main.py > ...
1 v def fibonacci_recursive(n):
2 v   if n <= 1:
3   |   return n
4 v   else:
5     return fibonacci_recursive(n-1) + fibonacci_recursive(n-2)
6
7 # Contoh penggunaan:
8 result_recursive = fibonacci_recursive(10)
9 print("Rekursif:", result_recursive)
```

Input Size = n

Basic operation =

- Pengecekan kondisi: $n \leq 1$
- Pengembalian nilai: $\text{return } n$ dan $\text{return } \text{fibonacci_recursive}(n-1) + \text{fibonacci_recursive}(n-2)$

$$C(n) = C(n-1) + C(n-2) + 1$$

Fibonacci Iteratif

Fibonacci iteratif adalah sebuah cara untuk menghitung bilangan Fibonacci dengan menggunakan algoritma yang berulang-ulang. Algoritma ini akan mengulangi proses penjumlahan dua angka sebelumnya sampai mendapatkan nilai yang diinginkan.

```
main.py > ...
1 v def fibonacci_iterative(n):
2 v   if n <= 1:
3   |   return n
4
5   fib = [0] * (n + 1)
6   fib[1] = 1
7
8 v   for i in range(2, n+1):
9     fib[i] = fib[i-1] + fib[i-2]
10
11  return fib[n]
12
13 # Contoh penggunaan:
14 result_iterative = fibonacci_iterative(10)
15 print("Iteratif:", result_iterative)
```

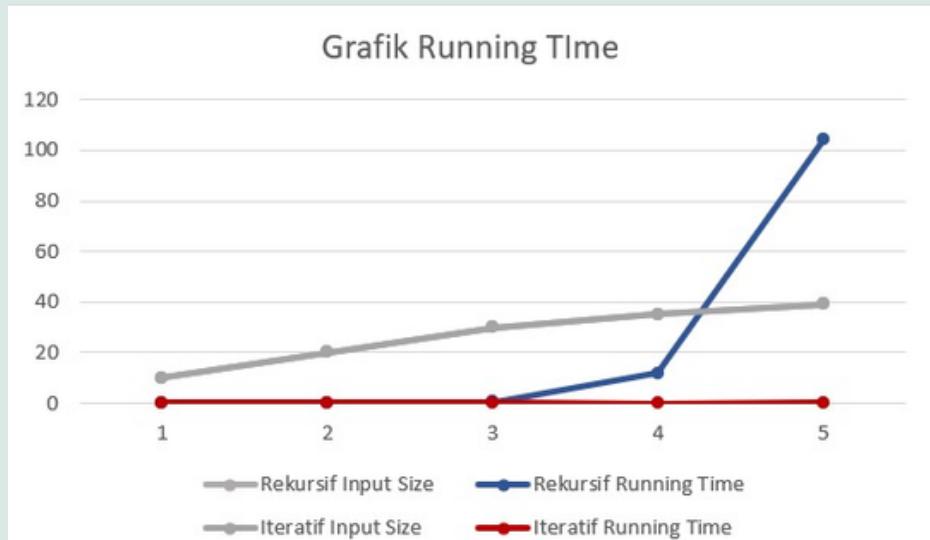
Input Size = n

Basic operation =

- Pengujian kondisi: $\text{if } n \leq 1: \text{return } n$
- Inisialisasi array: $\text{fib} = [0] * (n + 1)$ dan $\text{fib}[1] = 1$
- Iterasi: $\text{for } i \text{ in range}(2, n+1)$
- Penjumlahan: $\text{fib}[i] = \text{fib}[i-1] + \text{fib}[i-2]$

$$C(n) = 2n$$

Gambar Grafiknya



Analisis

Berdasarkan hasil grafik disamping dapat disimpulkan bahwa algoritma Fibonacci iteratif jauh lebih efisien pada saat *running time*. Selain itu, meskipun tidak terlihat pada grafik disamping, algoritma Fibonacci iteratif juga dapat meng-input size lebih besar daripada Fibonacci Rekursif.