LAPORAN KEGIATAN

Analisis Kompleksitas Algoritma dari Algoritma Bubble Sort (Itterative dan Recursive)

Disusun oleh :
Afrizal Syahruluddin Yusuf (1301194288)
Nanda Ihwani Saputri (1301194107)
Rizki Tri Setiawan (1301194012)

Program Studi S1 Informatika – Fakultas Informatika

Universitas Telkom

Jalan Telekomunikasi Terusan Buah Batu, Bandung

Indonesia

1. Judul

Analisis Kompleksitas Algooritma dari Algoritma Bubble Sort (Itterative dan Recursive)

2. Tanggal Pelaksanaan

1 Januari 2021 - 8 Januari 2021

3. Tujuan

Laporan ini dibuat guna membahas efektifitas waktu kompleksitas algoritma dari algoritma bubble sort yang merupakan salah satu bentuk algoritma pengurutan.

4. Landasan teori

4.1. Kompleksitas Waktu

Menurut Traju (2010:1) Kompleksitas waktu, T(n), adalah jumlah operasi yang dilakukan untuk melaksanakan algoritma sebagai fungsi dari ukuran masukan n.

4.2. Notasi Asimtotik

Notasi asimtotik merupakan himpunan fungsi yang dibatasi oleh sutau fungsi n elemen N ($n \epsilon N$) yang cukup besar.

4.3. Basic Operation

Basic operation merupakan suatu instruksi yang jika ditotal sedemikian rupa total pekerjaan yang dilakukan algoritma secara kasar sebanding dengan jumlahnya berkali-kali instruksi atau sekelompok instruksi dilakukan.

5. Hasil pengamatan

Berdasarkan hasil running program, didapatkan data waktu eksekusi yang dilakukan oleh algoritma bubble sort. Sebagai berikut :

```
D:\TEKNIK INFORMATIKA\ANALISIS KOMPLEKSITAS ALGORITMA\Tubes-AKA-master>python main.py
Banyak data : 10
Iterative bubble sort executed in 0.000039
Recursive bubble sort executed in 0.000018
```

```
D:\TEKNIK INFORMATIKA\ANALISIS KOMPLEKSITAS ALGORITMA\Tubes-AKA-master>python main.py
Banyak data : 1000
Iterative bubble sort executed in 0.103479
Recursive bubble sort executed in 0.099835
```

```
D:\TEKNIK INFORMATIKA\ANALISIS KOMPLEKSITAS ALGORITMA\Tubes-AKA-master>python main.py
Banyak data : 100
Iterative bubble sort executed in 0.001003
Recursive bubble sort executed in 0.001526
```

6. Pembahasan

6.1. Pseudocode

```
# Utility function to swap values at two indices in the list
def swap(A, i, j):

    temp = A[i]
    A[i] = A[j]
    A[j] = temp


# Function to perform bubble sort on list
def bubbleSortIterative(A):

    for k in range(len(A) - 1):

        # last k items are already sorted, so inner loop can
        # avoid looking at the last k items
        for i in range(len(A) - 1 - k):
            if A[i] > A[i + 1]:
                swap(A, i, i + 1)
def bubbleSortRecursive(A, n):

    for i in range(n - 1):
        if A[i] > A[i + 1]:
```

```
        swap(A, i, i + 1)

if n - 1 > 1:
    bubbleSortRecursive(A, n - 1)
# the algorithm can be stopped if the inner loop
# didn't do any swap
```

6.2. Analisis Kompleksitas Algoritma

6.2.1. Basic Operation

Algoritma bubble sort itterative = Perbandingan (A[i] > A[i + 1])
Algoritma bubble sort recursive = Perbandingan (A[i] > A[i + 1])

6.2.2. Kompleksitas Waktu (T(n) ≈ C(n))

Algoritma Itterative :

$T(n) \approx C(n)$

$$= \sum_{k=0}^{n-1} \sum_{i=0}^{n-1-k} 1$$

$$= \sum_{k=0}^{n-1} n - k$$

$$= n - k \sum_{k=0}^{n-1} 1$$
$$= (n - k)\,(n)$$
$$\approx n^2 - nk \in O(n^2)$$

Algoritma Recursive :

$T(n) \approx C(n)$

$T(n) = 0 \quad$, untuk $n \leq 1$
$\quad = T(n - 1) + n - 1 \quad$, untuk $n > 1$
$T(n) = T(n - 1) + n - 1$
$\quad = (T(n - 2) + n - 1) + n - 1$

$= T(n - 2) + 2n - 2$
$= (T(n - 3) + n - 1) + 2n - 2$
$= T(n - 3) + 3n - 3$
…
$= T(n - i) + 3i - i$

Karena : $n - i = 1$, sehingga $i = n - 1$

Maka, $T(n - i) + in - i = T(n - (n - 1)) + (n - 1)\, n - (n - 1)$
$= T(1) + n^2 - n - 2n + 1$
$= 0 + n^2 - 2n + 1$
$\approx n^2 - 2n + 1 \in O(n^2)$

6.2.3. Notasi Asimtotik

Best case = $O(n)$
Average case = $O(n^2)$
Worst case = $O(n^2)$

6.2.4. Class Eficiency

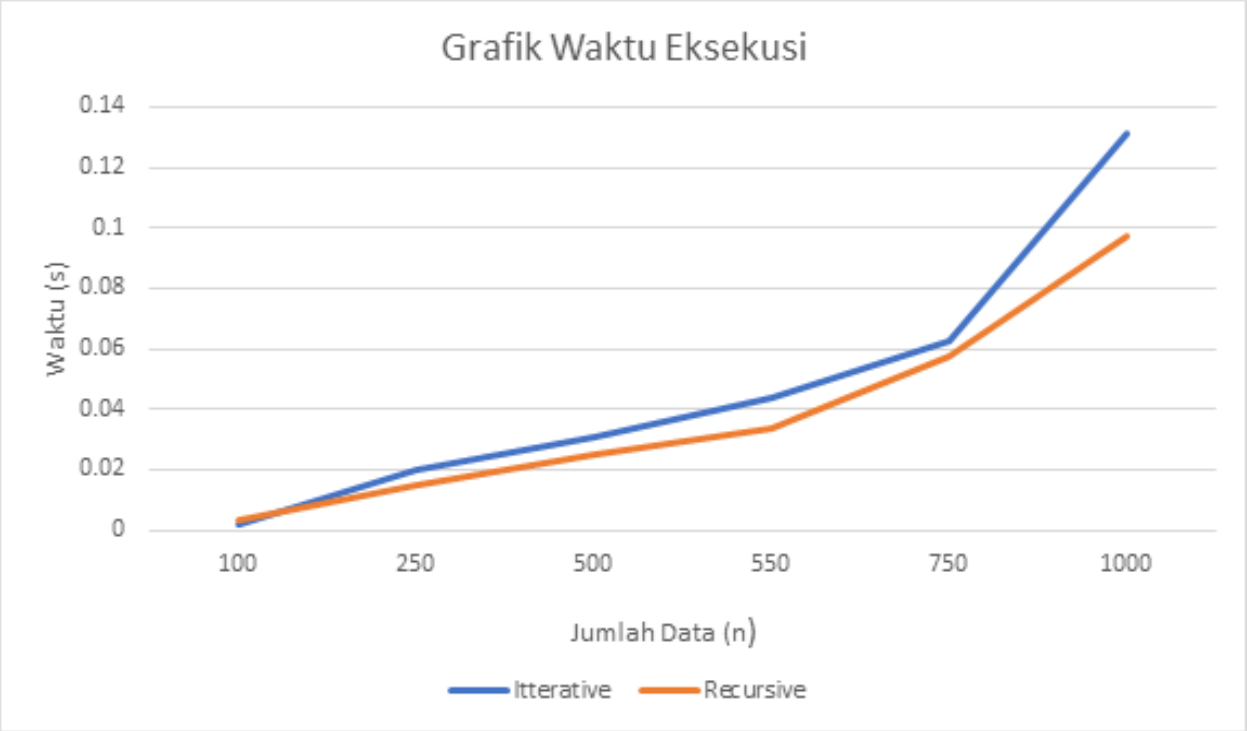
Best case = linier
Average case = Kuadratik
Worst case = Kuadratik

6.3. Analisis Tambahan

6.3.1. Tabel Waktu Eksekusi(s)

Banyak Data (n)	Itterative	Recursive
100	0,001645	0,002962
250	0,020173	0,014655
500	0,030693	0,024932
550	0,043944	0,033422
750	0,062679	0,057164
1000	0,131299	0,097635

6.3.2. Grafik



7. Kesimpulan

Algoritma bubble sort merupakan algoritma sederhana yang sangat mudah untuk dipelajari. Selain itu, algoritma bubble sort juga memiliki definisi yang terurut serta jelas dalam algoritmanya. Dalam kasus terbaik (best case), kompleksitas algoritma Bubble Sort adalah O(n). Namun, untuk kasus umum serta worst case, kompleksitas algoritma pengurutan gelembung adalah $O(n^2)$. Dengan interval data antara 100 sampai dengan 1.000 elemen. Waktu eksekusi diukur dengan satuan Second (s). Diketahui pula, bahwa waktu eksekusi algoritma bubble sort recursive lebih singkat daripada waktu eksekusi algoritma bubble sort itterative.

8. Daftar Pustaka

[1] Tanggal akses 4 Januari 2021
https://dosen.perbanas.id/kompleksitas-algoritma/#:~:text=%E2%80%BASebuah%20masalah%20dapat%20mempunyai%20banyak%20algoritma%20penyelesaian.&text=Algoritma%20yang%20efisien%20adalah%20algoritma,jumlah%20data%20yang%20dip%20roses.

[2] Tanggal akses 6 januari 2021
https://www.studytonight.com/data-structures/bubble-sort#:~:text=Complexity%20Analysis%20of%20Bubble%20Sort&text=Hence%20the%20time%20complexity%20of,required%20i.e.%20for%20temp%20variable.

[3] Tanggal akses 6 Januari 2021
https://www.geeksforgeeks.org/bubble-sort/