LAPORAN TUGAS BESAR ANALISIS KOMPLEKSITAS ALGORITMA BUBBLE SORT DAN SELECTION SORT

Disusun oleh:

Muhamad Rafli Susanto - 1301204052
Royadi Kayantho - 1301204064

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

2021

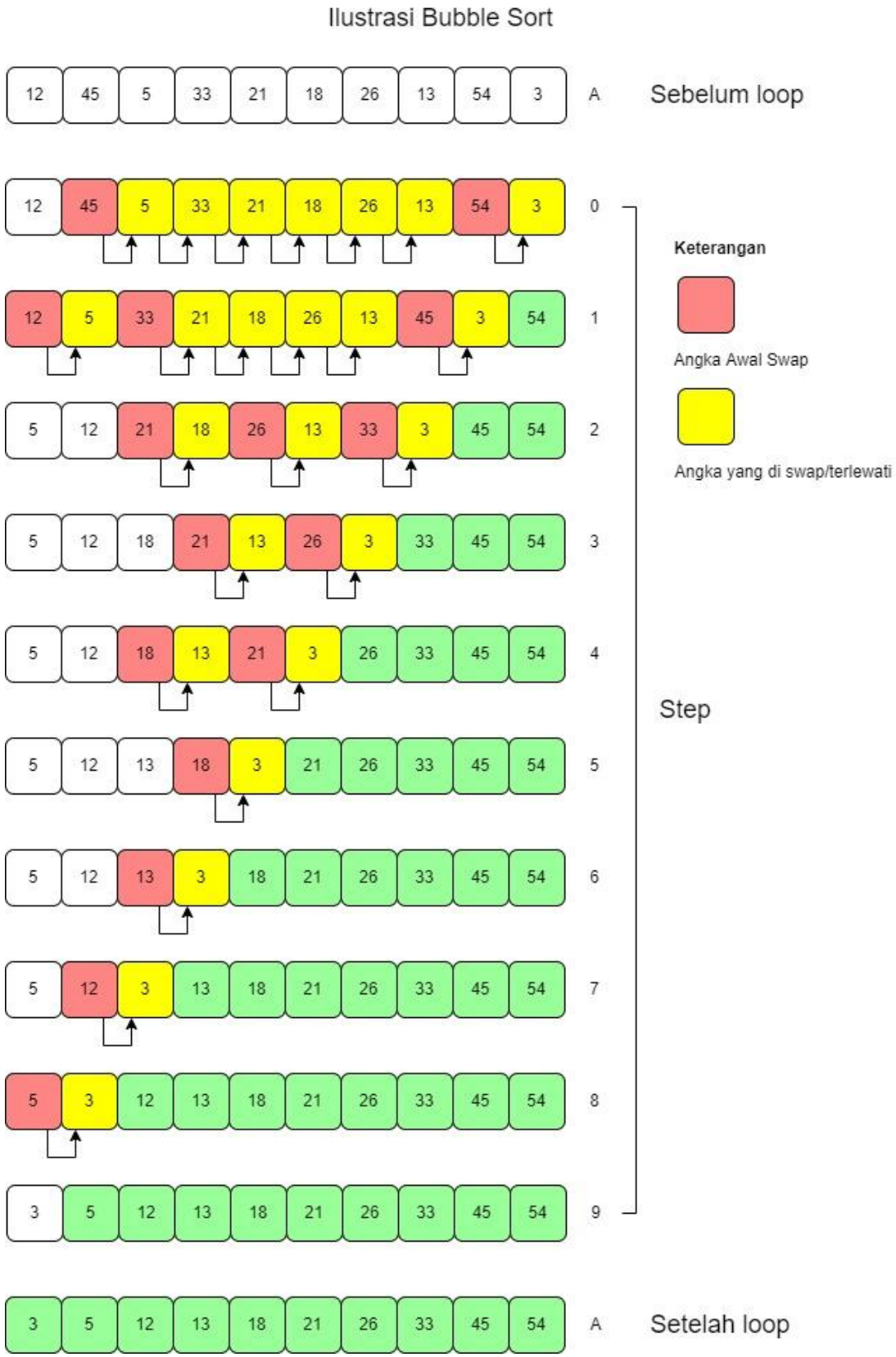
1. Algoritma Bubble Sort

Bubble sort merupakan sebuah teknik pengurutan data dengan cara menukar dua data yang bersebelahan jika urutan dari data tersebut salah (lebih besar/kecil). Proses akan dimulai dari membandingkan data pertama dan kedua, kedua dan ketiga, begitu seterusnya sampai data elemen ke terakhir. Proses akan berhenti jika semua data sudah terurut (step = n-1). Algoritma ini mempunyai kelas efisiensi O(n^2).

1.1 Pseudocode Bubble Sort

```
procedure bubbleSort(in/out array[] : integer, in n : integer)
kamus
        step, i, temp : integer
algoritma
    for step <- 0 to n do
        for i <- 0 to n-step do
          if (array[i] > array[i+1]) then
              temp <- array[i]
              array[i] <- array[i+1]
              array[i+1] <- temp
          endif
          endfor
    endfor
endprocedure
```

1.2 Ilustrasi Algoritma Bubble Sort



1.3 Implementasi Algoritma Bubble Sort pada Python

```python
def bubbleSort(arr):
    for step in range(0, len(arr)):
        for i in range(0, len(arr)-step-1):
            if(arr[i] > arr[i+1]):
                temp = arr[i]
                arr[i] = arr[i+1]
                arr[i+1] = temp
    return arr

mylist = [5, 4, 10, 8, 23, 42, 1, 22]
mylist = bubbleSort(mylist)
print(mylist)
```

1.4 Analisis Kompleksitas Algoritma Bubble Sort

$a)$ $Basic\ Operation$: $array[i] > array[i+1]$
$Perbandingan$ $(1\ kali\ setiap\ iterasi)$
$b)$ $Kelas\ Efisiensi$:

$$T(n) = \sum_{step=0}^{n} \sum_{i=0}^{n-step} 1$$

$$= \sum_{step=0}^{n} (n - step) - 0 + 1$$

$$= \sum_{step=0}^{n} n - step + 1 = \sum_{step=0}^{n} n - \sum_{step=0}^{n} step + \sum_{step=0}^{n} 1$$

$$= n \sum_{step=0}^{n} 1 - \sum_{step=0}^{n} step + \sum_{step=0}^{n} 1$$

$$= n((n - 0) + 1) - \frac{n-(n+1)}{2} + (n - 0) + 1$$

$$= n(n) + 1 + \frac{1}{2} + n + 1$$

$$= n^2 + n + \frac{5}{2} \in O(n^2)$$

2. Algoritma Selection Sort

Selection sort merupakan sebuah teknik pengurutan data dengan cara mencari nilai tertinggi atau terendah di dalam array kemudian menempatkan nilai tersebut di tempat semestinya. Proses akan berhenti jika semua elemen sudah terurut (Pass = n-1). Algoritma ini mempunyai kelas efisiensi O(n^2) sama seperti Bubble Sort.

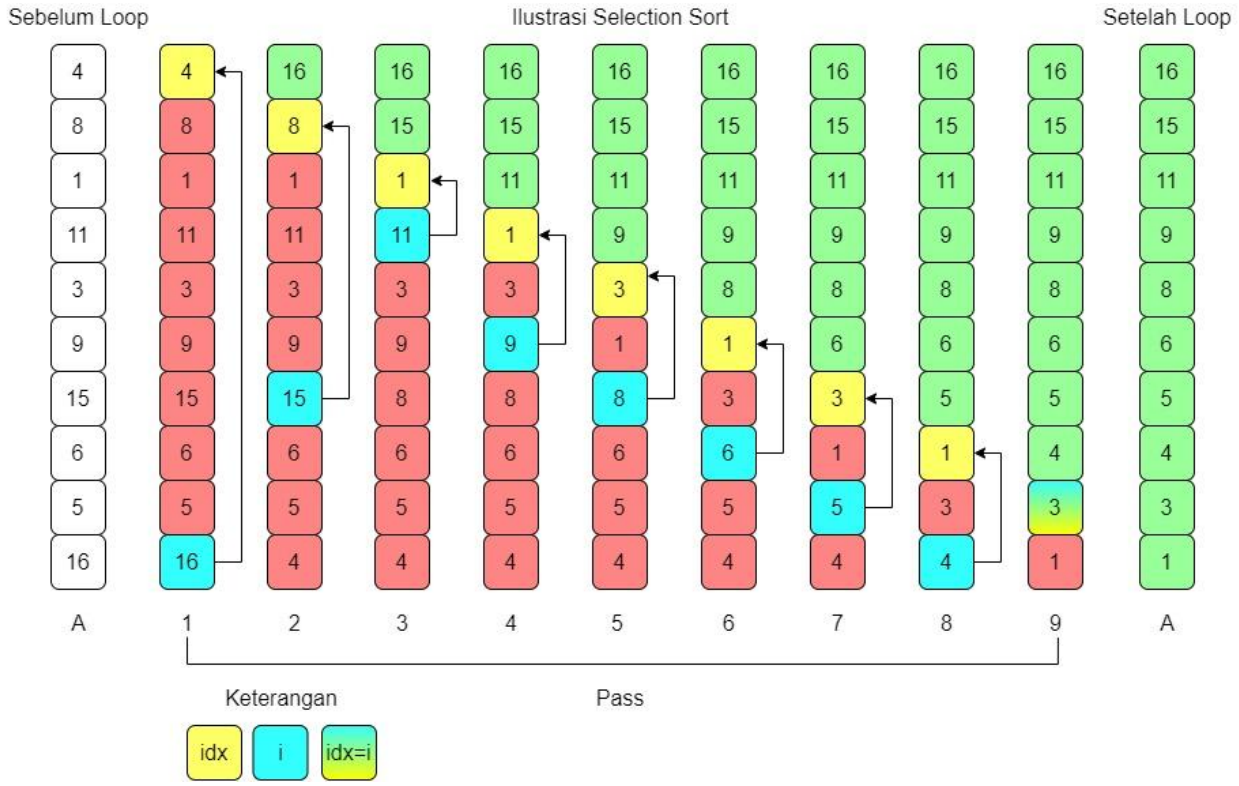
2.1 Pseudocode Selection Sort

```
procedure selectionSort(in/out array[] integer, in n : integer)
```

```
kamus
    pass, idx, i : integer
    temp : integer
```

```
algoritma
    for pass <- 1 to n do
        idx <- pass - 1
        for i <- pass to n do
            if (array[idx] > array[i]) then
                idx <- i
            endif
        endfor
        temp <- array[pass-1]
        array[pass-1] <- array[idx]
        array[idx] <- temp
    endfor
endprocedure
```

2.2 Ilustrasi Algoritma Selection Sort



2.3 Implementasi Algoritma Selection Sort pada Python

```python
def selectionSort(arr):
    for Pass in range(1, len(arr)):
        idx = Pass - 1
        for i in range(Pass, len(arr)):
            if arr[idx] > arr[i]:
                idx = i
        temp = arr[Pass-1]
        arr[Pass-1] = arr[idx]
        arr[idx] = temp
    return arr

mylist = [5, 4, 10, 8, 23, 42, 1, 22]
mylist = selectionSort(mylist)
print(mylist)
```

2.4 Analisis Kompleksitas Algoritma Selection Sort

$a)$ $Basic\ Operation$: $array[idx] > array[i]$

$Perbandingan\ (1\ kali\ setiap\ iterasi)$

$b)$ $Kelas\ Efisiensi$:

$$T(n) = \sum_{pass=1}^{n-1} \sum_{i=pass}^{n} 1$$

$$= \sum_{pass=1}^{n-1} (n - pass) + 1$$

$$= \sum_{pass=1}^{n-1} n - \sum_{pass=1}^{n-1} pass + \sum_{pass=1}^{n-1} 1$$

$$= n \sum_{pass=1}^{n-1} 1 - \sum_{pass=1}^{n-1} pass + \sum_{pass=1}^{n-1} 1$$

$$= n((n-1) - 1 + 1) - \frac{(n-1)-n}{2} + (n-1) - 1 + 1$$

$$= n^2 - n + \frac{1}{2} + n - 1$$

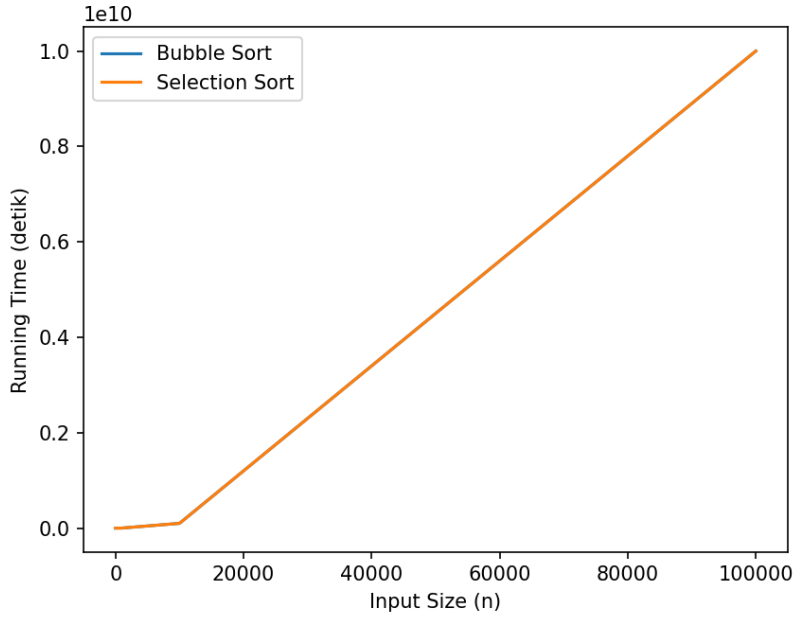
$$= n^2 - \frac{1}{2} \in O(n^2)$$

3. Perbandingan Running Time Algoritma Bubble Sort dengan Selection Sort

3.1 Tabel Perbandingan 1

Input Size (n)	**Bubble Sort** $T(n) = n^2 + n + \frac{5}{2}$	**Selection Sort** $T(n) = n^2 - \frac{1}{2}$
10	112.5 detik ≈ 00:01:52	99.5 detik ≈ 00:01:39
100	10102.5 detik ≈ 02:48:22	9999.5 detik ≈ 02:46:39
1000	1001002.5 detik ≈ 11 hari, 14:03:22	999999.5 detik ≈ 11 hari, 13:46:39
10000	100010002.5 detik ≈ 1157 hari, 12:33:22	99999999.5 detik ≈ 1157 hari, 09:46:39
100000	10000100002.5 detik ≈ 115741 hari, 21:33:22.	9999999999.5 detik ≈ 115740 hari, 17:46:39

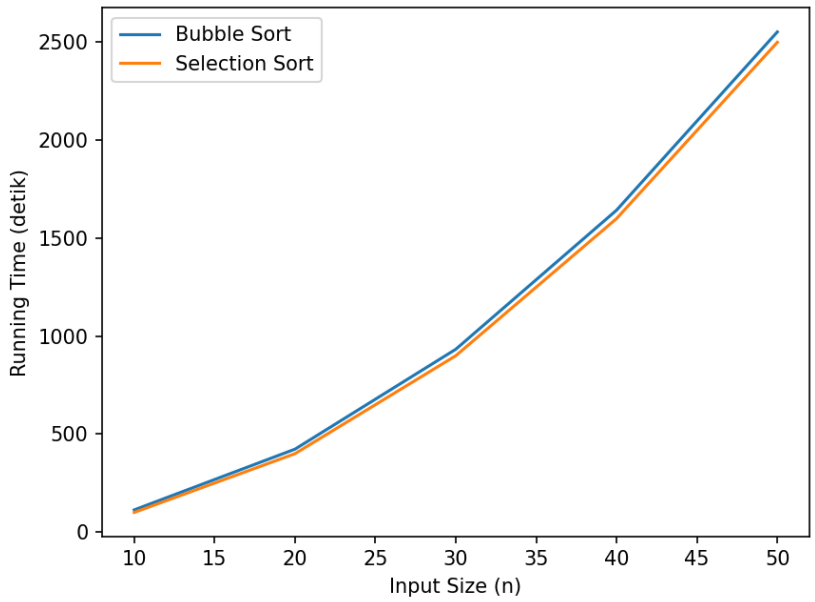
3.2 Grafik Perbandingan 1



3.3 Tabel Perbandingan 2

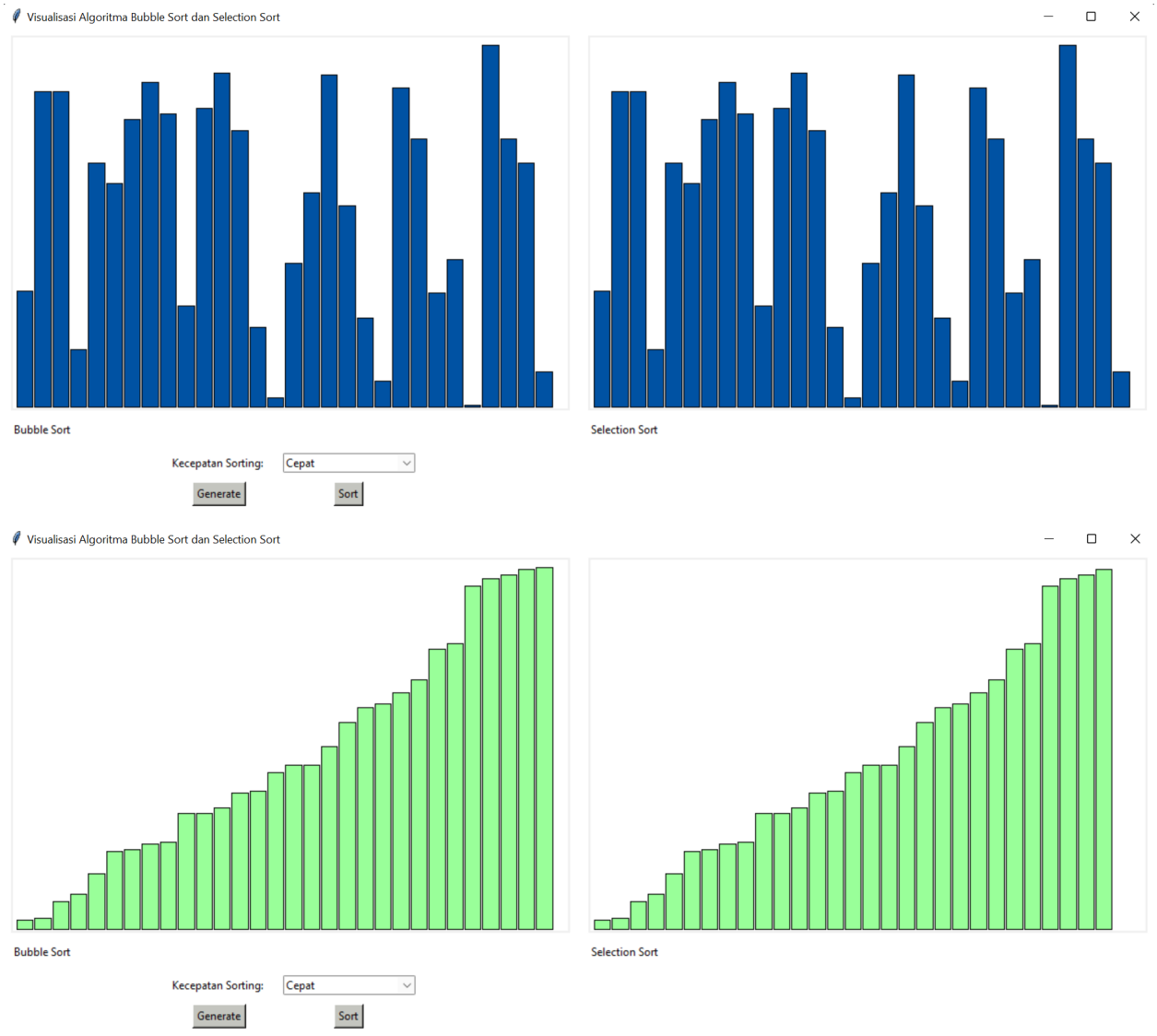
Input Size (n)	**Bubble Sort** $T(n) = n^2 + n + \frac{5}{2}$	**Selection Sort** $T(n) = n^2 - \frac{1}{2}$
10	112.5 detik ≈ 00:01:52	99.5 detik ≈ 00:01:39
20	422.5 detik ≈ 00:07:02	399.5 detik ≈ 00:06:39
30	932.5 detik ≈ 00:15:32	899.5 detik ≈ 00:14:59
40	1642.5 detik ≈ 00:27:22	1599.5 detik ≈ 00:26:39
50	2552.5 detik ≈ 00:42:32	2499.5 detik ≈ 00:41:39

3.4 Grafik Perbandingan 2



4. Visualisasi Bubble Sort dan Selection Sort menggunakan TKINTER Python

4.1 Preview Visualisasi



4.2 Link Source Code Visualisasi (Aplikasi) dan Video Demo

Link GitHub:
https://github.com/raflisusanto/tubes_aka

Link Video (YouTube):
https://youtu.be/rpEjzKkSTZs