

PERANCANGAN ANALISIS ALGORITMA II



Disusun Oleh :

Isra Septia Cahyani

F551 21 060

B

PROGRAM STUDI S1-TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

FAKULTAS TEKNIK

UNIVERSITAS TADULAKO

2023

A. Algoritma Bubble Sort

1. Kode Program

```
1 def bubble_sort(arr):
2     n = len(arr)
3     #perulangan untuk mengiterasi sebanyak n - 1 kali
4     for i in range(n):
5         swapped = False
6         #perulangan untuk membandingkan pasangan elemen yang berdekatan
7         for j in range(0, n - i - 1):
8             if arr[j] > arr[j + 1]:
9                 arr[j], arr[j + 1] = arr[j + 1], arr[j]
10                swapped = True
11
12        # Jika tidak ada pertukaran atau array sudah terurut maka perulangan berhenti
13        if not swapped:
14            break
15        # Menampilkan array pada setiap iterasi
16        print(f"Iterasi {i + 1}: {arr}")
17    return arr
18
19    # contoh nilai
20    arr = [27, 25, 11, 18, 16, 9]
21    print("Array sebelum diurutkan:", arr)
22    sorted_arr = bubble_sort(arr) #memanggil fungsi bubble sort yang telah didefinisikan
23    print("Array setelah diurutkan:", sorted_arr)
```

2. Hasil Run

```
BubbleSort x
C:\Users\ASUS\PycharmProjects\pythonProject\PAA_tugas3\Scripts\python.exe C:/Users/ASUS/PycharmProjects/PAA_tugas3/BubbleSort.py
Array sebelum diurutkan: [27, 25, 11, 18, 16, 9]
Iterasi 1: [25, 11, 18, 16, 9, 27]
Iterasi 2: [11, 18, 16, 9, 25, 27]
Iterasi 3: [11, 16, 9, 18, 25, 27]
Iterasi 4: [11, 9, 16, 18, 25, 27]
Iterasi 5: [9, 11, 16, 18, 25, 27]
Array setelah diurutkan: [9, 11, 16, 18, 25, 27]

Process finished with exit code 0
```

B. Algoritma Insertion Sort

1. Kode Program

```
1 def insertion_sort(arr):
2     n = len(arr)
3     #perulangan untuk memproses setiap elemen pada bagian tidak terurut
4     for i in range(1, n):
5         key = arr[i]
6         j = i - 1
7         while j >= 0 and arr[j] > key:
8             arr[j + 1] = arr[j]
9             j -= 1
10        arr[j + 1] = key
11
12        # Menampilkan array pada setiap iterasi
13        print(f"Iterasi {i}: {arr}")
14    return arr
15
16    arr = [64, 34, 25, 12, 22, 11]
17    print("Array sebelum diurutkan:", arr)
18    sorted_arr = insertion_sort(arr)
19    print("Array setelah diurutkan:", sorted_arr)
20
21
```

2. Hasil Run

```
InsertionSort x
C:\Users\ASUS\PycharmProjects\pythonProject\PAA_tugas3\Scripts\python.exe C:\Users\ASUS\PycharmProjects\PAA_tugas3\InsertionSort.py
Array sebelum diurutkan: [64, 34, 25, 12, 22, 11]
Iterasi 1: [34, 64, 25, 12, 22, 11]
Iterasi 2: [25, 34, 64, 12, 22, 11]
Iterasi 3: [12, 25, 34, 64, 22, 11]
Iterasi 4: [12, 22, 25, 34, 64, 11]
Iterasi 5: [11, 12, 22, 25, 34, 64]
Array setelah diurutkan: [11, 12, 22, 25, 34, 64]

Process finished with exit code 0
```

C. Analisis

Dalam kedua nilai array yang digunakan untuk perbandingan Bubble Sort dan Insertion Sort. Dimana nilai Array yang digunakan yaitu [64, 34, 25, 12, 22, 11]. Ketika array ini diurutkan menggunakan Insertion Sort, proses pengurutan akan dilakukan dengan membandingkan dan menukar elemen-elemen secara berurutan. Setiap iterasi akan menampilkan array pada tahap pengurutan yang berbeda. Hasil akhirnya adalah [11, 12, 22, 25, 34, 64]. Dan ketika array ini diurutkan menggunakan Bubble Sort, perulangan akan membandingkan dan menukar pasangan elemen yang berdekatan. Setiap iterasi akan menampilkan array pada tahap pengurutan yang berbeda. Hasil akhirnya adalah [9, 11, 16, 18, 25, 27].

Dari kedua kasus dan algoritma tersebut, pada contoh array yang digunakan, Insertion Sort lebih optimal dari pada Bubble Sort. Meskipun keduanya memiliki kompleksitas waktu yang sama dalam kasus terburuk, Insertion Sort menunjukkan kinerja yang lebih baik dalam pengurutan array tersebut. Hal ini dikarenakan Insertion Sort hanya memproses bagian array yang belum terurut, sementara Bubble Sort melakukan pertukaran elemen secara berulang, bahkan jika array sudah terurut sebelumnya. Selain itu, Insertion Sort juga memiliki keunggulan dalam mengurutkan array yang hampir terurut atau memiliki jumlah elemen yang sedikit. Oleh karena itu, dalam kasus ini, Insertion Sort adalah pilihan yang lebih optimal daripada Bubble Sort.