

# **Penjelasan algoritma dari metode Bubble sort, Selection Sort dan Insertion Sort**

## **Praktikum 5**

**G.211.21.0004**

### **Bubble sort**

Langkah-langkah:

1. Tentukan panjang daftar  $n$ .
2. Mulai iterasi dari indeks pertama hingga indeks  $n-1$ .
3. Dalam setiap iterasi pertama, atur variabel swapped menjadi False. Ini digunakan untuk memeriksa apakah ada pertukaran elemen dalam satu iterasi.
4. Mulai iterasi kedua di dalam iterasi pertama dari indeks 0 hingga  $n-i-1$ .
5. Bandingkan elemen ke- $j$  dengan elemen ke- $(j+1)$ . Jika elemen ke- $j$  lebih besar dari elemen ke- $(j+1)$ , tukar posisi keduanya dan atur swapped menjadi True.
6. Ulangi langkah 4-5 hingga seluruh daftar terurut satu kali.

Jika tidak ada pertukaran elemen dalam satu iterasi penuh, berarti daftar sudah terurut, dan dapat keluar dari loop.

```
def bubbleSort(arr):
    n = len(arr)
    swapped = False # Untuk mengoptimalkan, menandai apakah ada pertukaran pada suatu iterasi.

    for i in range(n-1):
        for j in range(0, n-i-1):
            if arr[j] > arr[j + 1]:
                # Jika elemen saat ini lebih besar dari elemen berikutnya, lakukan pertukaran.
                swapped = True
                arr[j], arr[j + 1] = arr[j + 1], arr[j]

            # Jika tidak ada pertukaran pada iterasi ini, daftar sudah diurutkan.
            if not swapped:
                break

# Driver code to test above
arr = [64, 34, 25, 12, 22, 11, 90]
bubbleSort(arr)
print("Sorted array is:")
for i in range(len(arr)):
    print("% d" % arr[i], end=" ")

Sorted array is:
11 12 22 25 34 64 90
```

## Selection Sort

Penjelasan algoritma Selection Sort:

1. **n** adalah panjang array.
2. Dengan menggunakan dua loop bersarang, algoritma mencari nilai minimum di dalam sisa array yang belum diurutkan.
3. Setelah menemukan nilai minimum, nilai tersebut ditukar dengan elemen pertama di sisa array yang belum diurutkan.
4. Proses ini diulang untuk setiap elemen hingga seluruh array terurut.

Algoritma Selection Sort memilih elemen terkecil dari array dan menukarnya dengan elemen pertama. Kemudian, memilih elemen terkecil dari sisa array (array tanpa elemen pertama) dan menukarnya dengan elemen kedua, begitu seterusnya. Algoritma ini mengulang proses ini hingga seluruh array terurut.

```
def selectionSort(arr):
    n = len(arr)

    for i in range(n):
        # Cari elemen terkecil dalam sisa daftar yang belum diurutkan.
        min_idx = i
        for j in range(i+1, n):
            if arr[j] < arr[min_idx]:
                min_idx = j

        # Tukar elemen terkecil dengan elemen pertama dalam sisa daftar.
        arr[i], arr[min_idx] = arr[min_idx], arr[i]

# Driver code to test above
arr = [64, 34, 25, 12, 22, 11, 90]
selectionSort(arr)
print("Sorted array is:")
for i in range(len(arr)):
    print("% d" % arr[i], end=" ")

Sorted array is:
11 12 22 25 34 64 90
```

## Insertion sort

Penjelasan algoritma Insertion Sort:

1. **n** adalah panjang array.
2. Loop dimulai dari indeks kedua (indeks pertama dianggap sebagai array yang sudah terurut).
3. Pilih elemen pertama di luar array yang sudah terurut (key).
4. Bandingkan key dengan elemen-elemen di dalam array yang sudah terurut dan pindahkan elemen-elemen yang lebih besar ke posisi yang lebih tinggi.
5. Tempatkan key pada posisi yang benar dalam array yang sudah terurut.

Proses ini diulang hingga seluruh array terurut. Algoritma Insertion Sort mirip dengan cara orang mengurutkan kartu saat bermain kartu; kita

memasukkan kartu baru ke dalam tangan kita dan menyusunnya sedemikian rupa sehingga tangan kita tetap teratur.

```
✓ 0s ▶ def insertionSort(arr):  
    n = len(arr)  
  
    for i in range(1, n):  
        key = arr[i]  
        j = i - 1  
  
        # Pindahkan elemen-elemen yang lebih besar dari key ke posisi yang lebih tinggi.  
        while j >= 0 and key < arr[j]:  
            arr[j + 1] = arr[j]  
            j -= 1  
  
        # Tempatkan key pada posisi yang sesuai.  
        arr[j + 1] = key  
  
    # Driver code to test above  
    arr = [64, 34, 25, 12, 22, 11, 90]  
    insertionSort(arr)  
    print("Sorted array is:")  
    for i in range(len(arr)):  
        print("% d" % arr[i], end=" ")  
  
⇒ Sorted array is:  
11 12 22 25 34 64 90
```