

## MODUL VII

### SORTING

#### 7.1 TUJUAN

1. Memahami algoritma sorting yang dapat digunakan,
2. Menerapkan algoritma sorting pada studi kasus.

#### 7.2 DASAR TEORI

Algoritma Sorting adalah algoritma untuk meletakkan kumpulan elemen data ke dalam urutan tertentu, berdasarkan satu atau beberapa kunci ke dalam tiap-tiap elemen. Berdasarkan data terurutnya, algoritma sorting dibagi menjadi dua jenis, yaitu:

- Ascending, pengurutan dari nilai terkecil hingga terbesar. Contoh : a, b, c, d, e.
- Descending, pengurutan dari nilai terbesar hingga terkecil. Contoh : e, d, c, b, a.

#### 7.3 PEMBAHASAN

##### A. Insertion sort

Insertion sort adalah sebuah algoritma pengurutan yang mengurutkannya dicek satu persatu mulai dari yang kedua sampai dengan yang terakhir. Apabila ditemukan data yang lebih kecil dari data sebelumnya, maka data tersebut disisipkan pada posisi yang sesuai. Contoh Insertion Sort:

- **Ascending insertion sort**

Buatlah fungsi dengan nama "insertion\_sort"

```
def insertion_sort(array):  
    for i in range(1, len(array)):  
        item = array[i]  
        j = i - 1  
        while j >= 0 and array[j] > item:  
            array[j + 1] = array[j]  
            j -= 1  
        array[j + 1] = item  
    return array
```

Lalu, jalankan program dengan menuliskan perintah berikut:

```
data_array = [32, 53, 11, 43, 91, 100]
print(insertion_sort(data_array))
```

Output dari sort ascending di atas adalah:

```
[11, 32, 43, 53, 91, 100]
```

#### - **Descending insertion sort**

Pada descending sort kita hanya perlu mengubah pembanding “>” menjadi “<” pada `while`.

```
while j >= 0 and array[j] > item:
```

diubah menjadi:

```
while j >= 0 and array[j] < item:
```

Output dari sort descending ketika dijalankan adalah :

```
[100, 91, 53, 43, 32, 11]
```

## **B. Bubble sort**

Bubble Sort adalah algoritma sorting dengan cara membandingkan elemen yang sekarang dengan elemen yang berikutnya. Jika elemen sekarang > elemen berikutnya, maka elemen akan tukar. Jika perbandingan sudah sampai index akhir, maka akan diulang dari index awal. Jika di perbandingan selanjutnya tidak ada perubahan maka sorting selesai.

Proses pengurutan pada algoritma ini dengan membandingkan masing - masing elemen secara berpasangan lalu menukarnya dalam kondisi tertentu. Proses ini akan terus diulang sampai elemen terakhir atau sampai tidak ada lagi elemen yang dapat ditukar. Inilah kenapa algoritma ini diberi nama “Bubble”, dimana gelembung yang terbesar akan naik ke atas.

Contoh Bubble Sort:

### - Ascending Bubble Sort

Buatlah fuction dengan nama bubble\_sort:

```
def bubble_sort(array):  
    n = len(array)  
    for i in range(n):  
        for j in range(n - i - 1):  
            if array[j] > array[j + 1]:  
                array[j], array[j + 1] = array[j + 1], array[j]  
    return array
```

jalankan program dengan menuliskan perintah berikut:

```
data_array = [5, 8, 40, 3, 100, 45]  
print(bubble_sort(data_array))
```

output dari sort ascending di atas adalah:

```
[3, 5, 8, 40, 45, 100]
```

### - Descending Bubble Sort

Pada descending short kita hanya perlu mengubah pembanding ">" menjadi "<" pada if.

```
if array[j] > array[j + 1]:
```

diubah menjadi:

```
if array[j] < array[j + 1]:
```

output dari sort descending ketika dijalankan adalah :

```
[100, 45, 40, 8, 5, 3]
```

## C. Selection Sort

Selection Sort adalah algoritma sorting yang mengurutkan data dengan cara mencari elemen paling kecil dari list, lalu menukar elemen tersebut ke urutan paling awal. Selection Sort adalah algoritma sorting dengan cara mengurutkan dengan membandingkan elemen sekarang (ditandai) dengan elemen yang berikutnya sampai terakhir. Jika ditemukan elemen paling kecil, kemudian ditukar dengan elemen sekarang.

Contoh selection sort:

### - Ascending Selection Sort

Buatlah function dengan nama selection\_sort:

```
def selection_sort(arr):  
    n = len(arr)  
    for i in range(n):  
        min_idx = i  
        for j in range(i+1, n):  
            if arr[min_idx] > arr[j]:  
                min_idx = j  
        arr[i], arr[min_idx] = arr[min_idx], arr[i]  
    return arr
```

jalankan program dengan menuliskan perintah berikut:

```
data_array = [90, 34, 57, 32, 4, 1]  
print(selection_sort(data_array))
```

output dari sort ascending di atas adalah:

```
[1, 4, 32, 57, 34, 90]
```

### - Descending Selection Sort

Pada descending short kita hanya perlu mengubah pembanding ">" menjadi "<" pada if.

```
if arr[min_idx] > arr[j]:
```

diubah menjadi:

```
if arr[min_idx] < arr[j]:
```

output dari sort descending ketika dijalankan adalah :

```
[90, 57, 34, 32, 4, 1]
```

## 7.4 LATIHAN

- Buatlah sebuah program penerimaan mahasiswa. Fitur dari program tersebut dapat menambah data mahasiswa, menghapus data mahasiswa, urutkan data berdasarkan nim, dan cetak seluruh data.

Contoh Output :

### 1. Tambah data Mahasiswa

```
<===== Menu Data Mahasiswa =====>
1. Tambah Data Mahasiswa
2. Hapus Data Mahasiswa
3. Urutkan Data Mahasiswa
4. Lihat Data Mahasiswa
5. Tutup
Pilih: 1
Jumlah mahasiswa: 3
Nama mahasiswa: Kurnia
Nama mahasiswa: Tika
Nama mahasiswa: Jati
```

### 2. Hapus data Mahasiswa

```
<===== Menu Data Mahasiswa =====>
1. Tambah Data Mahasiswa
2. Hapus Data Mahasiswa
3. Urutkan Data Mahasiswa
4. Lihat Data Mahasiswa
5. Tutup
Pilih: 2
Data mahasiswa ['Kurnia', 'Tika', 'Jati']
Hapus mahasiswa: Tika
Data mahasiswa ['Kurnia', 'Jati']
```

### 3. Urutkan data Mahasiswa

```
<===== Menu Data Mahasiswa =====>
1. Tambah Data Mahasiswa
2. Hapus Data Mahasiswa
3. Urutkan Data Mahasiswa
4. Lihat Data Mahasiswa
5. Tutup
Pilih: 3
['Jati', 'Kurnia']
```

#### 4. Cetak data Mahasiswa

```
<===== Menu Data Mahasiswa =====>
1. Tambah Data Mahasiswa
2. Hapus Data Mahasiswa
3. Urutkan Data Mahasiswa
4. Lihat Data Mahasiswa
5. Tutup
Pilih: 4
Nama Mahasiswa: Jati
Nama Mahasiswa: Kurnia
```

#### 5. Tutup

```
<===== Menu Data Mahasiswa =====>
1. Tambah Data Mahasiswa
2. Hapus Data Mahasiswa
3. Urutkan Data Mahasiswa
4. Lihat Data Mahasiswa
5. Tutup
Pilih: 5
Selesai
```

### Code Program:

```
#Tambah Data Mhs
def addMahasiswa():
    jumlah = int(input("Jumlah mahasiswa: "))
    mahasiswa = []
    while(jumlah > 0):
        nama = input("Nama mahasiswa: ")
        mahasiswa.append(nama)
        jumlah = jumlah - 1

    while(True):
        panggil(mahasiswa)
        jumlah = jumlah - 1
        if(jumlah<0):
            break

#Hapus Data Mhs
def removeMahasiswa(arrayMahasiswa):
    mahasiswa = arrayMahasiswa
    print("Data mahasiswa %s" %arrayMahasiswa)
    mahasiswa.remove(input("Hapus mahasiswa: "))
    print("Data mahasiswa %s" %mahasiswa)
    panggil(mahasiswa)

#Urutkan Data Mhs
def ascMahasiswa(arrayMahasiswa):
    mahasiswa = arrayMahasiswa
    mahasiswa.sort()
    print(mahasiswa)
    panggil(mahasiswa)

# Lihat Data Mhs
def viewMahasiswa(arrayMahasiswa):
    mahasiswa = arrayMahasiswa
    for x in mahasiswa:
        print("Nama Mahasiswa: %s" %x)
    panggil(arrayMahasiswa)
```



```

#Menu
def panggil(arrayMahasiswa):
    print("\n<===== Menu Data Mahasiswa =====>")
    print("1. Tambah Data Mahasiswa ")
    print("2. Hapus Data Mahasiswa ")
    print("3. Urutkan Data Mahasiswa ")
    print("4. Lihat Data Mahasiswa ")
    print("5. Tutup ")

    pilih = int(input("Pilih: "))
    if(pilih==1):
        addMahasiswa()
    elif(pilih==2):
        removeMahasiswa(arrayMahasiswa)
    elif(pilih==3):
        ascMahasiswa(arrayMahasiswa)
    elif(pilih==4):
        viewMahasiswa(arrayMahasiswa)
    else:
        print("Selesai")

addMahasiswa()

```

Note : Refrensi bisa dilihat pada link berikut  
<https://www.anbidev.com/pythonsorting>



## 7.5 TUGAS

1. Pada suatu kelas terdapat 5 mahasiswa. Pada akhir semester mereka menerima lembar Indeks Prestasi Semester (IPS), masing-masing mahasiswa tersebut memiliki IPS sebagai berikut: {3.8, 2.9, 3.3, 4.0, 2.7}. Buatlah program untuk mengurutkan IPS mahasiswa tersebut dari yang terbesar hingga terkecil dengan menggunakan algoritma **Bubble sort** !

Contoh output program:

```
Indeks Prestasi Semester (IPS)
List sebelum diurutkan : [3.8, 2.9, 3.3, 4.0, 2.7]
List setelah diurutkan : [4.0, 3.8, 3.3, 2.9, 2.7]
```

2. Pada suatu organisasi memiliki 10 anggota dengan nama masing-masing: *Zhafira, Nirmala, Aksara, Nalendra, Cakra, Sastra, Agni, Bagus, Jerome, Kiara*. Supaya mudah dalam melakukan pencarian, Ketua organisasi akan mengurutkan nama-nama tersebut sesuai dengan alfabet. Buatlah program untuk membantu Pain dengan menggunakan algoritma **Selection Sort**!

Contoh output program:

```
Nama 10 Anggota Organisasi
Before : ['Zhafira', 'Nirmala', 'Aksara', 'Nalendra', 'Cakra', 'Sastra', 'Agni', 'Bagus', 'Jerome', 'Kiara']
After  : ['Agni', 'Aksara', 'Bagus', 'Cakra', 'Jerome', 'Kiara', 'Nalendra', 'Nirmala', 'Sastra', 'Zhafira']
```

3. Buatlah program untuk menginput nama buku lalu muncul pilihan jenis sorting (dengan **Insertion Sort**).

Contoh output program :

```
Masukkan Total Buku : 5
Masukkan Judul Buku ke-1 : Laut Bercerita
Masukkan Judul Buku ke-2 : Bumi Manusia
Masukkan Judul Buku ke-3 : Geez & Ann
Masukkan Judul Buku ke-4 : Hujan
Masukkan Judul Buku ke-5 : Garis Waktu

<===== Urutkan ? =====>
1. Insertion Ascending
2. Insertion Descending
Pilih: 1

Sorting Buku Secara Ascending

Judul Buku ke-1 : Bumi Manusia
Judul Buku ke-2 : Garis Waktu
Judul Buku ke-3 : Geez & Ann
Judul Buku ke-4 : Hujan
Judul Buku ke-5 : Laut Bercerita
```

