

**LAPORAN PRAKTIKUM**  
**PEMROGRAMAN 1**  
**MODUL 7**



Oleh:

Nama : Alfian Mutakim

Nim : 2211104024

Kelas : SE06A

**PRODI S1 REKAYASA PERANGKAT LUNAK**  
**FAKULTAS INFORMATIKA**  
**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**  
**2023**

## I. 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.

### A. PEMBAHASAN

#### 1. 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):
    start = timeit.default_timer()
    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
    stop = timeit.default_timer()
    print(f"insertion sort successful! Elapsed
time: + {stop - start} ")
    return array
```

Lalu, jalankan program dengan menuliskan perintah berikut:

```
list_1 = [2,9,5,99,65,4,22,90]
print(f"Bifore : {list_1}")
insertion_sort(list_1)
print(f"After : {list_1}")
```

Output dari sort ascending di atas adalah:

```
Bifore : [2, 9, 5, 99, 65, 4, 22, 90]
insertion sort successfull! Elapsed time: + 3.0399998649954796e-05
After : [2, 4, 5, 9, 22, 65, 90, 99]
PS C:\Users\User\Documents\Praktikum Pemograman1\LapModul7_SE06A_2211104024_Al
```

- Descending insertion sort

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

Diubah menjadi:

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

Output dari descending ketika kita jalankan:

```
After : [99, 90, 65, 22, 9, 5, 4, 2]
```

## 2. 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:

```
# bubble sort
def bubble_sort(array):
    start = timeit.default_timer()
    for i in range (len(array)):
        for j in range (len(array)-i-1):
            if array [j] > array[j+1]:
                array[j], array [j+1] = array
                [j+1], array[j]

    stop = timeit.default_timer()
    print(f"bubble sort successful! Elapsed time:
+ {stop - start} ")
    return array
```

jalankan program dengan menuliskan perintah berikut:

```
list_1 = [2,9,5,99,65,4,22,90]
print(f"Bifore : {list_1}")
bubble_sort(list_1)
print(f"After : {list_1}")
```

output dari sort ascending di atas adalah:

```
Bifore : [2, 9, 5, 99, 65, 4, 22, 90]
bubble sort successful! Elapsed time: + 4.910002462565899e-05
After : [2, 4, 5, 9, 22, 65, 90, 99]
PS C:\Users\User\Documents\Praktikum Pemograman1\LapModul7_SE06A_2211104024
```

- Descending Bubble Sort

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

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

Output dari sort descending ketika dijalankan:

```
Bifore : [2, 9, 5, 99, 65, 4, 22, 90]
bubble sort successful! Elapsed time: + 4.9800146371126175e-05
After : [99, 90, 65, 22, 9, 5, 4, 2]
PS C:\Users\User\Documents\Praktikum Pemograman1\LapModul7_SE06A_2211104024
```

### 3. 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`:

```
# selection sort
def selection_sort(arr):
    start = timeit.default_timer()
    n = len(arr)
    for i in range(n):
        min_index = i
        for j in range(i+1, n):
            if arr[j] < arr[min_index]:
                min_index = j
        arr[i], arr[min_index] = arr[min_index],
arr[i]

    stop = timeit.default_timer()
    print(f"Selection sort successful! Elapsed
time: {stop - start} seconds")
    return arr
```

jalankan program dengan menuliskan perintah berikut:

```
list_1 = [90, 34, 57, 32, 4, 1]
print(f"Before: {list_1}")
sorted_list = selection_sort(list_1.copy())
print(f"After: {sorted_list}")
```

output dari sort ascending di atas adalah:

```
Before: [90, 34, 57, 32, 4, 1]
Selection sort successful! Elapsed time: 3.8400059565901756e-05 seconds
After: [1, 4, 32, 34, 57, 90]
PS C:\Users\User\Documents\Praktikum Pemrograman1\LapModul7_SE06A_2211104024
```

- Descending Selection Sort Pada descending sort kita hanya perlu mengubah pembandingan “>” menjadi “<” pada if.  
diubah menjadi:

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

output dari sort descending ketika dijalankan adalah :

```
Selection sort successful! Elapsed time: 4.309997893869877e-05
After: [90, 57, 34, 32, 4, 1]
```

## II. GUIDED

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

Kode 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)
1

#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")

arrayMahasiswa = []
panggil (arrayMahasiswa)

```



Output yang dihasilkan:

```
<===== 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: alfian
Nama mahasiswa: david
Nama mahasiswa: aulia

<===== 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 ['alfian', 'david', 'aulia']
Hapus mahasiswa: aulia
Data mahasiswa ['alfian', 'david']

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

<===== 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: alfian
Nama Mahasiswa: david
```

Ketika pengguna ingin menyelesaikan program:

```
<===== Menu Data Mahasiswa =====>
1. Tambah Data Mahasiswa
2. Hapus Data Mahasiswa
3. Urutkan Data Mahasiswa
4. Lihat Data Mahasiswa
5. Tutup
Pilih: 5
Selesai
PS C:\Users\User\Documents\Praktikum Pemograman1\Lap
```

### III. UNGUIDED

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 !

Kode program:

```
print("indeks prestasi Semester (IPS)")
# bubble sort
def bubble_sort(array):
    for i in range (len(array)):
        for j in range (len(array)-i-1):
            if array [j] < array[j+1]:
                array[j], array [j+1] = array [j+1],
array[j]

    return array

list_1 = [3.8,2.9,3.3,4.0,2.7]
print(f"list sebelum diurutkan : {list_1}")
bubble_sort(list_1)
print(f"list setelah diurutkan : {list_1}")
```

Output yan dihasilkan:

```
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]
PS C:\Users\User\Documents\Praktikum Pemograman1\LapModul7_SE06A
```

Penjelasan:

Kode program di atas menggunakan algoritma Bubble Sort untuk mengurutkan elemen-elemen dalam sebuah array. Fungsi `bubble\_sort` digunakan untuk melakukan pengurutan dengan membandingkan dan menukar elemen-elemen secara berulang hingga seluruh array terurut.

Setelah diurutkan, array tersebut ditampilkan dalam urutan descending. Dalam contoh kode di atas, list `[3.8, 2.9, 3.3, 4.0, 2.7]` diurutkan menjadi `[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 namanama tersebut sesuai dengan alfabet. Buatlah program untuk membantu Pain dengan menggunakan algoritma Selection Sort!

Kode Program:

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

list_1 = ["Zhafira", "Nirmala", "Aksara", "Nalendra",
          "Cakra", "Sastra", "Agni", "Bagas", "Jerome", "Kiara"]
print(f"Before: {list_1}")
sorted_list = selection_sort(list_1.copy())
print(f"After: {sorted_list}")
```

Output yang dihasilkan:

```
> & C:/Users/User/A
on3.11.exe "c:/Users/User/Documents/Praktikum Pemograman1/LapModul7_SE06A_2211104024_Alfian Mutakim/Tugas2.py
Before: ['Zhafira', 'Nirmala', 'Aksara', 'Nalendra', 'Cakra', 'Sastra', 'Agni', 'Bagas', 'Jerome', 'Kiara']
After: ['Agni', 'Aksara', 'Bagas', 'Cakra', 'Jerome', 'Kiara', 'Nalendra', 'Nirmala', 'Sastra', 'Zhafira']
PS C:\Users\User\Documents\Praktikum Pemograman1\LapModul7_SE06A_2211104024_Alfian Mutakim> ]
```

Penjelasan:

Kode program di atas merupakan implementasi algoritma Selection Sort dalam bahasa Python. Algoritma ini digunakan untuk mengurutkan elemen-elemen dalam sebuah array dengan cara mencari elemen terkecil secara berulang dan menukarnya dengan elemen terkecil pada posisi yang sesuai.

Pada contoh diatas mengurutkan nama sesuai dengan abjad huruf yang tadinya “Zhafira, Nirmala, Aksara, Nalendra, Cakra, Sastra, Agni, Bagas, Jerome, Kiara”

menjadi 'Agni', 'Aksara', 'Bagas', 'Cakra', 'Jerome', 'Kiara', 'Nalendra', 'Nirmala', 'Sastra', 'Zhafira'.

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

Kode program:

```
def addbuku():
    jumlah = int(input("Masukkan total buku: "))
    buku = []
    while jumlah > 0:
        nama = input("judul buku ke-{}: ".format(jumlah))
        buku.append(nama)
        jumlah = jumlah - 1

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

# Urutkan Data buku - insertion sort
def insertion_sort(arraybuku):
    buku = arraybuku
    for i in range(1, len(arraybuku)):
        item = arraybuku[i]
        j = i - 1

        while j >= 0 and arraybuku[j] >= item:
            arraybuku[j + 1] = arraybuku[j]
            j -= 1

        arraybuku[j + 1] = item

    return arraybuku

buku = []
```

```

# print("Before: {}".format(buku))
# insertion_sort(buku)
# print("After: {}".format(buku))

# Urutkan Data buku - insertion sort descending
def insertion_sort_Descending(arraybuku):
    buku = arraybuku
    for i in range(1, len(arraybuku)):
        item = arraybuku[i]
        j = i - 1
        while j >= 0 and arraybuku[j] <= item:
            arraybuku[j + 1] = arraybuku[j]
            j -= 1

        arraybuku[j + 1] = item

    return arraybuku

buku = []
# print("Before: {}".format(buku))
# insertion_sort_Descending(buku)
# print("After: {}".format(buku))

# Menu
def panggil(arraybuku):
    print("\n<===== Urutkan? =====>")
    print("1. Metode Ascending")
    print("2. Metode Descending")
    pilih = int(input("Pilih: "))

    if pilih == 1:
        insertion_sort(arraybuku)
    else:
        insertion_sort_Descending(arraybuku)

    print("Sorted array: {}".format(arraybuku))

addbuku()

```

Output dari kode program tersebut:

```
Apps/python3.11.exe C:/Users/User/Documents/Praktikum Pemrograman1/LapModul7_SE06A_2211104024
Masukkan total buku: 3
judul buku ke-3: Bumi manusia
judul buku ke-2: Alangkah lucunya negri ini
judul buku ke-1: The Guardion

<===== Urutkan? =====>
1. Metode Ascending
2. Metode Descending
Pilih: 1
Sorted array: ['Alangkah lucunya negri ini', 'Bumi manusia', 'The Guardion']
PS C:\Users\User\Documents\Praktikum Pemrograman1\LapModul7_SE06A_2211104024_Alfian Mutakim>
```

Penjelasan :

Kode program di atas merupakan sebuah program Python yang digunakan untuk menginputkan daftar buku, mengurutkannya secara ascending atau descending menggunakan metode insertion sort, dan menampilkan daftar buku yang telah diurutkan.

- Fungsi ``addbuku()`` digunakan untuk mengumpulkan daftar buku dari pengguna. Program meminta pengguna untuk memasukkan jumlah total buku, kemudian program akan meminta pengguna untuk memasukkan judul buku satu per satu sesuai dengan jumlah yang ditentukan. Daftar buku kemudian disimpan dalam list ``buku``.
- Fungsi ``insertion_sort(arraybuku)`` digunakan untuk mengurutkan daftar buku secara ascending menggunakan metode insertion sort.
- Fungsi ``insertion_sort_Descending(arraybuku)`` mirip dengan ``insertion_sort()``, namun digunakan untuk mengurutkan daftar buku secara descending.
- Fungsi ``panggil(arraybuku)`` digunakan untuk menampilkan menu dan meminta pengguna untuk memilih metode pengurutan (ascending atau descending). Berdasarkan pilihan pengguna, fungsi ini akan memanggil fungsi ``insertion_sort()`` atau ``insertion_sort_Descending()`` sesuai dengan metode yang dipilih. Setelah pengurutan selesai, daftar buku yang telah diurutkan akan ditampilkan.
- Pada akhir program, fungsi ``addbuku()`` dipanggil untuk memulai proses pengumpulan dan pengurutan daftar buku.

