

**LAPORAN PRAKTIKUM
PEMROGRAMAN 1
MODUL 7**



Oleh:

Gideon Toranawa Ladiyo

2211104022

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.

1. 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” Lalu, jalankan program dengan menuliskan perintah berikut:

```
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 pembandingan “>” 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 sort kita hanya perlu mengubah pembandingan ">" 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]
```

II. GUIDED

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 :

Tambah data Mahasiswa

```
===== Menu Data Mahasiswa =====
1. Tambah Data Mahasiswa
2. Hapus Data Mahasiswa
3. Urutkan Data Mahasiswa
4. Lihat Data Mahasiswa
5. Keluar
Pilih: 1
Jumlah Mahasiswa: 3
Nama Mahasiswa: gideon
Nama Mahasiswa: toranawa
Nama Mahasiswa: ladiyo
```

Hapus data Mahasiswa

```
===== Menu Data Mahasiswa =====
1. Tambah Data Mahasiswa
2. Hapus Data Mahasiswa
3. Urutkan Data Mahasiswa
4. Lihat Data Mahasiswa
5. Keluar
Pilih: 2
Data Mahasiswa ['gideon', 'toranawa', 'ladiyo']
Hapus Mahasiswa: toranawa
Data Mahasiswa: ['gideon', 'ladiyo']
```

Urutkan data Mahasiswa

```
===== Menu Data Mahasiswa =====
1. Tambah Data Mahasiswa
2. Hapus Data Mahasiswa
3. Urutkan Data Mahasiswa
4. Lihat Data Mahasiswa
5. Keluar
Pilih: 3
['gideon', 'ladiyo', 'toranawa']
```

Cetak data Mahasiswa

```
===== Menu Data Mahasiswa =====
1. Tambah Data Mahasiswa
2. Hapus Data Mahasiswa
3. Urutkan Data Mahasiswa
4. Lihat Data Mahasiswa
5. Keluar
Pilih: 4
Nama Mahasiswa: gideon
Nama Mahasiswa: ladiyo
Nama Mahasiswa: toranawa
```

Tutup

```
===== Menu Data Mahasiswa =====
1. Tambah Data Mahasiswa
2. Hapus Data Mahasiswa
3. Urutkan Data Mahasiswa
4. Lihat Data Mahasiswa
5. Keluar
Pilih: 5
Anda Keluar!
PS D:\kuliah\pemrograman\praktikum\laporan\7>
```

Code Program:

```
1  #Menu
2  def panggil(mhs):
3      print("\n===== Menu Data Mahasiswa =====")
4      print("1. Tambah Data Mahasiswa ")
5      print("2. Hapus Data Mahasiswa")
6      print("3. Urutkan Data Mahasiswa ")
7      print("4. Lihat Data Mahasiswa ")
8      print("5. Keluar ")
9
10     while True:
11         pilih = input("Pilih: ")
12         if pilih == '1':
13             addmhs(mhs)
14             break
15         elif pilih == '2':
16             removemhs(mhs)
17             break
18         elif pilih == '3':
19             ascmhs(mhs)
20             break
21         elif pilih == '4':
22             viewmhs(mhs)
23             break
24         elif pilih == '5':
25             print("Anda Keluar!")
26             break
27         else:
28             print("Masukkan pilihan yang valid")
29
```

```

30 #Tambah
31 def addmhs(mhs):
32     while True:
33         jumlah = input("Jumlah Mahasiswa: ")
34         if not jumlah.isdigit():
35             print("Jumlah mahasiswa harus berupa angka")
36             continue
37
38         jumlah = int(jumlah)
39
40         if jumlah <= 0:
41             print("Jumlah mahasiswa harus lebih besar dari 0")
42             continue
43
44         while(jumlah > 0):
45             nama = input("Nama Mahasiswa: ")
46             mhs.append(nama)
47             jumlah = jumlah - 1
48         panggil(mhs)
49         break
50
51 #Hapus
52 def removemhs(mhs):
53     print("Data Mahasiswa %s" %mhs)
54     remove_mhs = input("Hapus Mahasiswa: ")
55     while True:
56         if remove_mhs in mhs:
57             mhs.remove(remove_mhs)
58             print("Data Mahasiswa: %s"%mhs)
59             panggil(mhs)
60             break
61         else:
62             print("Nama tersebut tidak ada dalam data")
63             removemhs(mhs)
64             break
65
66 #urut
67 def ascmhs(mhs):
68     mhs.sort()
69     print(mhs)
70     panggil(mhs)
71
72 #lihat
73 def viewmhs(mhs):
74     if not mhs :
75         print("Belum ada mahasiswa")
76     else:
77         for x in mhs:
78             print("Nama Mahasiswa: %s" %x)
79     panggil(mhs)
80
81 mhs = []
82 panggil(mhs)

```


III. UNGUIDED

Tugas 1

Program untuk mengurutkan IPS mahasiswa dari yang terbesar hingga terkecil dengan menggunakan algoritma Bubble sort

Source Code :

```
1  def bubble_sort(arr):
2      n = len(arr)
3      for i in range(n-1):
4          for j in range(n-1-i):
5              if arr[j] < arr[j+1]:
6                  arr[j], arr[j+1] = arr[j+1], arr[j]
7      return arr
8
9  ips = [3.8, 2.9, 3.3, 4.0, 2.7]
10 print("List sebelum diurutkan :", ips)
11 ips_sort = bubble_sort(ips)
12 print("List setelah diurutkan :", ips_sort)
```

Output :

```
Hom011.py
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]
```

Tugas 2

Program mengurutkan anggota organisasi dengan Selection sort

Source Code :

```
1 def selection_sort(arr):
2     n = len(arr)
3     for i in range(n-1):
4         smallest = i
5         for j in range(i+1, n):
6             if arr[j] < arr[smallest]:
7                 smallest = j
8         arr[i], arr[smallest] = arr[smallest], arr[i]
9     return arr
10
11 anggota = ['Zhafira', 'Nirmala', 'Aksara', 'Nalendra', 'Cakra', 'Sastra', 'Agni', 'Bagas', 'Jerome', 'Kiara']
12 print("Before :", anggota)
13 anggota_sort = selection_sort(anggota)
14 print("After :", anggota_sort)
```

Output :

```
nomor2.py
Before : ['Zhafira', 'Nirmala', 'Aksara', 'Nalendra', 'Cakra', 'Sastra', 'Agni', 'Bagas', 'Jerome', 'Kiara']
After : ['Agni', 'Aksara', 'Bagas', 'Cakra', 'Jerome', 'Kiara', 'Nalendra', 'Nirmala', 'Sastra', 'Zhafira']
PS D:\kuliah\pemrograman\praktikum\laporan\7>
```

Tugas 3

Program untuk menginput nama buku lalu muncul pilihan jenis sorting
(dengan Insertion Sort).

```
1 def insertion_asc(array):
2     for i in range(1, len(array)):
3         item = array[i]
4         j = i - 1
5         while j >= 0 and array[j] > item:
6             array[j + 1] = array[j]
7             j -= 1
8         array[j + 1] = item
9
10 def insertion_desc(array):
11     for i in range(1, len(array)):
12         item = array[i]
13         j = i - 1
14         while j >= 0 and array[j] < item:
15             array[j + 1] = array[j]
16             j -= 1
17         array[j + 1] = item
```

Pertama, buat fungsi untuk ascending sort dan descending sort. Kemudian buat fungsi untuk menambah buku, fungsi untuk memilih sort, dan fungsi untuk mencetak hasil.

```

19 def add_buku(arr):
20     total_buku = int(input("Masukkan Total Buku : "))
21     for i in range(total_buku):
22         nama_buku = input(f"Masukkan judul buku ke-{"
23             arr.append(nama_buku)
24
25 def pilih_sort(arr):
26     print("\n<===== Urutkan? =====>")
27     print("1. Insertion Ascending")
28     print("2. Insertion Descending")
29     while True:
30         pilihan = input("Pilih(1/2) : ")
31         if pilihan == '1' or pilihan == '2':
32             break
33         else:
34             print("Masukkan pilihan 1 atau 2")
35     while True:
36         if pilihan == '1':
37             insertion_asc(arr)
38             print("\nSorting buku secara Ascending")
39             break
40         elif pilihan == '2':
41             insertion_desc(arr)
42             print("\nSorting buku secara Descending")
43             break
44         else:
45             print("Masukkan pilihan yang valid!")
46
47 def cetak_hasil(arr):
48     print(" ")
49     for i, n in enumerate(arr):
50         print(f"Judul buku ke-{"
51
52 data_buku = []
53 add_buku(data_buku)
54 pilih_sort(data_buku)
55 cetak_hasil(data_buku)

```

Output Ketika memilih ascending sort:

```

Masukkan Total Buku : 3
Masukkan judul buku ke-1 : matematika
Masukkan judul buku ke-2 : sains
Masukkan judul buku ke-3 : bahasa

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

Sorting buku secara Ascending

Judul buku ke-1 : bahasa
Judul buku ke-2 : matematika
Judul buku ke-3 : sains

```

Output Ketika memilih descending sort:

```
nama13.py
Masukkan Total Buku : 3
Masukkan judul buku ke-1 : matematika
Masukkan judul buku ke-2 : sains
Masukkan judul buku ke-3 : bahasa

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

Sorting buku secara Descending

Judul buku ke-1 : sains
Judul buku ke-2 : matematika
Judul buku ke-3 : bahasa
```