

**LAPORAN PRAKTIKUM
PEMROGRAMAN 1
MODUL VII**



Oleh:

Nita Fitrotul Mar'ah

2211104005

S1SE06A

**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. 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”

```
import timeit
print('Ascending')
print('')
#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 successfull Elapsed time: + {stop}")
    return array
list_1 = [45, 7, 9, 32, 41, 33, 51, 9, 24, 66]
print(f"Before: {list_1}")
insertion_sort(list_1)
print(f"After: {list_1}")
```

Output dari sort ascending di atas adalah:

```
Before: [45, 7, 9, 32, 41, 33, 51, 9, 24, 66]
insertion sort successfull Elapsed time: + 1.680001150816679e-0
5
After: [7, 45, 45, 45, 45, 45, 45, 45, 45, 45]
PS C:\Users\Wita>
```

- Descending insertion sort

Pada descending sort kita hanya perlu mengubah pembandingan “>” menjadi “<” pada

while.

diubah menjadi:

```
import timeit
print('Ascending')
print('')
#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 successfull Elapsed time: + {stop -
    return array
list_1 = [45, 7, 9, 32, 41, 33, 51, 9, 24, 66]
print(f"Before: {list_1}")
insertion_sort(list_1)
print(f"After: {list_1}")
```

Output dari sort descending ketika dijalankan adalah :

```
Before: [45, 7, 9, 32, 41, 33, 51, 9, 24, 66]
insertion sort successfull Elapsed time: + 0.002752600004896521
6
After: [45, 7, 7, 7, 7, 7, 7, 7, 7, 7]
PS C:\Users\Wita>
```

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:

```
import timeit
print('Ascending')
print('')
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 successfull Elapsed time: + {stop - st
    return array

list_1 = [45, 7, 9, 32, 41, 33, 51, 9, 24, 66]
print(f"Before: {list_1}")
bubble_sort(list_1)
print(f"After: {list_1}")
```

output dari sort ascending di atas adalah:

```
Before: [45, 7, 9, 32, 41, 33, 51, 9, 24, 66]
Bubble sort successfull Elapsed time: + 0.0016667000018060207
After: [7, 9, 9, 24, 32, 33, 41, 45, 51, 66]
PS C:\Users\Wita>
```

- Descending Bubble Sort

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

diubah menjadi:

```

import timeit
print('Ascending')
print('')
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 successfull Elapsed time: + {stop - st
    return array

list_1 = [45, 7, 9, 32, 41, 33, 51, 9, 24, 66]
print(f"Before: {list_1}")
bubble_sort(list_1)
print(f"After: {list_1}")

```

output dari sort descending ketika dijalankan adalah :

```

Before: [45, 7, 9, 32, 41, 33, 51, 9, 24, 66]
Bubble sort successfull Elapsed time: + 7.760000880807638e-05
After: [66, 51, 45, 41, 33, 32, 24, 9, 9, 7]
PS C:\Users\Nita>

```

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:

```

import timeit
print('Ascending')
print('')
def selection_sort(array):
    start = timeit.default_timer()
    for i in range(len(array)):
        min_index = i
        for j in range(i+1, len(array)):
            if array[min_index] > array[j]:
                min_index = j
        array[i], array[min_index] = array[min_index], array[i]

    stop = timeit.default_timer()
    print(f"Selection sort successfull Elapsed time: + {stop - start}")
    return array

list_1 = [45, 7, 9, 32, 41, 33, 51, 9, 24, 66]
print(f"Before: {list_1}")
selection_sort(list_1)
print(f"After: {list_1}")

```

output dari sort ascending di atas adalah:

```

Before: [45, 7, 9, 32, 41, 33, 51, 9, 24, 66]
Selection sort successfull Elapsed time: + 4.089999129064381e-05
After: [7, 9, 9, 24, 32, 33, 41, 45, 51, 66]
PS C:\Users\Nita>

```

- Descending Selection Sort

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

diubah menjadi:

```

import timeit
print('Ascending')
print('')
def selection_sort(array):
    start = timeit.default_timer()
    for i in range(len(array)):
        min_index = i
        for j in range(i+1, len(array)):
            if array[min_index] < array[j]:
                min_index = j
        array[i], array[min_index] = array[min_index], array[i]

    stop = timeit.default_timer()
    print(f"Selection sort successfull Elapsed time: + {stop - start}")
    return array

list_1 = [45, 7, 9, 32, 41, 33, 51, 9, 24, 66]
print(f"Before: {list_1}")
selection_sort(list_1)
print(f"After: {list_1}")

```

output dari sort descending ketika dijalankan adalah :

```

Before: [45, 7, 9, 32, 41, 33, 51, 9, 24, 66]
Selection sort successfull Elapsed time: + 4.1399995097890496e-05
After: [66, 51, 45, 41, 33, 32, 24, 9, 9, 7]
PS C:\Users\Nita>

```

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 :

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: 4
Nama mahasiswa: Nita
Nama mahasiswa: Najib
Nama mahasiswa: June
Nama mahasiswa: Marah
```

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 ['Nita', 'Najib', 'June', 'Marah']
Hapus mahasiswa: Marah
Data Mahasiswa ['Nita', 'Najib', 'June']
```

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: 2
Data mahasiswa ['Nita', 'Najib', 'June', 'Marah']
Hapus mahasiswa: Marah
Data Mahasiswa ['Nita', 'Najib', 'June']
```

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 June
Nama Mahasiswa Najib
Nama Mahasiswa Nita
```

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
PS C:\Users\Nita> █
```

Code Program:

```

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

    panggil(mahasiswa)

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

def ascMahasiswa(arrayMahasiswa):
    arrayMahasiswa.sort()
    print(arrayMahasiswa)
    panggil(arrayMahasiswa)

def viewMahasiswa(arrayMahasiswa):
    for x in arrayMahasiswa:
        print("Nama Mahasiswa %s" % x)
    panggil(arrayMahasiswa)

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:

```

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 !

Contoh output program:

```
Indeks Prestasi Semester (IPS)
Before: [3.8, 2.9, 3.3, 4.0, 2.7]
After: [2.7, 2.9, 3.3, 3.8, 4.0]
PS C:\Users\Nita>
```

Code:

```
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

ips_mahasiswa = [3.8, 2.9, 3.3, 4.0, 2.7]
print(f'Indeks Prestasi Semester (IPS)')
print(f"Before: {ips_mahasiswa}")
bubble_sort(ips_mahasiswa)
print(f"After: {ips_mahasiswa}")
```

2. Pada suatu organisasi memiliki 10 anggota dengan nama masing-masing: Zhafira, Nirmala, Aksara, Nalendra, Cakra, Sastra, Agni, Bagas, 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!

Contoh output program:

```
Before : ['Zhafira', 'Nirmala', 'Askara', 'Nalendra', 'Cakra', 'Sastra', 'Agni', 'Bagas', 'Jerome', 'Kiara']
After : ['Agni', 'Askara', 'Bagas', 'Cakra', 'Jerome', 'Kiara', 'Nalendra', 'Nirmala', 'Sastra', 'Zhafira']
PS C:\Users\Nita>
```

Code:

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

anggota = ["Zhafira", "Nirmala", "Askara", "Nalendra", "Cakra", "Sastra", "Agni", "Bagas", "Jerome", "Kiara"]
print("Before : ", anggota)
print("After : ", selection_sort(anggota))
```

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

Contoh output program :

```
Masukkan Total Buku : 10
Masukkan Judul Buku ke-1 : hari ini
Masukkan Judul Buku ke-2 : aku
Masukkan Judul Buku ke-3 : capek
Masukkan Judul Buku ke-4 : tapi
Masukkan Judul Buku ke-5 : aku
Masukkan Judul Buku ke-6 : suka
Masukkan Judul Buku ke-7 : semua
Masukkan Judul Buku ke-8 : berjalan
Masukkan Judul Buku ke-9 : dengan
Masukkan Judul Buku ke-10 : baik
<===== Urutkan ? =====>
1. Insertion Ascending
2. Insertion Descending
Pilih: 2
Sorting Buku Secara Descending
Judul Buku ke-1 : tapi
Judul Buku ke-2 : suka
Judul Buku ke-3 : semua
Judul Buku ke-4 : hari ini
Judul Buku ke-5 : dengan
Judul Buku ke-6 : capek
Judul Buku ke-7 : berjalan
Judul Buku ke-8 : baik
Judul Buku ke-9 : aku
Judul Buku ke-10 : aku
PS C:\Users\Nita>
```

```

Masukkan Total Buku : 10
Masukkan Judul Buku ke-1 : bukan
Masukkan Judul Buku ke-2 : tidak suka
Masukkan Judul Buku ke-3 : tapi
Masukkan Judul Buku ke-4 : kadang
Masukkan Judul Buku ke-5 : banyak hal
Masukkan Judul Buku ke-6 : yang tidak bisa
Masukkan Judul Buku ke-7 : dipaksain
Masukkan Judul Buku ke-8 : buat
Masukkan Judul Buku ke-9 : jadi
Masukkan Judul Buku ke-10 : miliku
<===== Urutkan ? =====>
1. Insertion Ascending
2. Insertion Descending
Pilih: 1
Sorting Buku Secara Ascending
Judul Buku ke-1 : banyak hal
Judul Buku ke-2 : buat
Judul Buku ke-3 : bukan
Judul Buku ke-4 : dipaksain
Judul Buku ke-5 : jadi
Judul Buku ke-6 : kadang
Judul Buku ke-7 : miliku
Judul Buku ke-8 : tapi
Judul Buku ke-9 : tidak suka
Judul Buku ke-10 : yang tidak bisa
PS C:\Users\Nita>

```

Code:

```

buku = []

def asc_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

def dsc_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

def addBuku():
    total = int(input("Masukkan Total Buku : "))
    count = 1
    while(total > 0):
        nama = input(f'Masukkan Judul Buku ke-{count} : ')
        buku.append(nama)
        count+=1
        total = total - 1

```

```

    while(True):
        total = total - 1
        if(total < 0):
            break

def menu():
    addBuku()
    print("<===== Urutkan ? =====>")
    print("1. Insertion Ascending")
    print("2. Insertion Descending"]

    pilih = int(input("Pilih: "))
    if (pilih == 1):
        print("Sorting Buku Secara Ascending")
        asc_insertion_sort(buku)
    elif (pilih == 2):
        print("Sorting Buku Secara Descending")
        dsc_insertion_sort(buku)
    else:
        print("Pilihan tidak tersedia..")

    hitung = 1
    for i in buku:
        print(f'Judul Buku ke-{hitung} : ', i)
        hitung+=1

```