

# **LAPORAN PRAKTIKUM PEMROGRAMAN 1**



Disusun Oleh:

Nama : Septiandi Nugraha

NIM : 21104060

Kelas : SE05 B

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK  
FAKULTAS INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

## **TAHAP 7**

### **SORTING**

#### **7.1 TUJUAN**

1. Mampu memahami pembuatan algoritma sorting dalam bahasa Python
2. Mampu menerapkan konsep dan pembuatan algoritma sorting dalam studi kasus nyata

#### **7.2 DASAR TEORI**

Algoritma sorting merupakan algoritma yang digunakan untuk meletakkan kumpulan elemen data ke dalam sebuah urutan. Berdasarkan data berurut, algoritma sorting dapat dibagi menjadi dua jenis, sebagai berikut.

1. *Ascending*

Ascending merupakan pengurutan dari nilai terkecil sampai dengan nilai terbesar. Contohnya : 10, 15, 25, 30

2. *Descending*

Descending merupakan pengurutan dari nilai terbesar sampai dengan nilai terkecil. Contohnya : 30, 25, 15, 10

##### **A. Insertion Sort**

Insertion Sort yaitu algoritma pengurutan dimana ketika mengurutkan data akan di periksa satu persatu dimulai dari data kedua hingga data terakhir. Ketika data paling kecil ditemukan maka data tersebut akan disisipkan di posisi yang sesuai.

### Contoh Insertion Sort:

#### a. Ascending Insertion Sort

Dalam contoh ini, akan dibuat sebuah 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
```

Berikutnya, kita akan menjalankan program tersebut dengan menuliskan perintah seperti berikut:

```
list_a = [4, 1, 2, 5, 3, 7, 8]  
print(insertion_sort(list_a))
```

Kemudian kita jalankan, maka output yang dihasilkan dari sorting ascending diatas adalah seperti berikut:

**[1, 2, 3, 4, 5, 7, 8]**

#### b. Descending Insertion Sort

Dalam contoh ini, akan dibuat sebuah 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
```

```
array[j + 1] = item

return array
```

Berikutnya, kita akan menjalankan program tersebut dengan menuliskan perintah seperti berikut:

```
list_a = [4, 1, 2, 5, 3, 7, 8]
print(insertion_sort(list_a))
```

Kemudian kita jalankan, maka output yang dihasilkan dari sorting ascending diatas adalah seperti berikut:

**[8, 7, 5, 4, 3, 2, 1]**

## B. Bubble Sort

Bubble sort merupakan algoritma sorting yaitu dengan cara membandingkan suatu elemen yang sekarang dengan elemen yang berikutnya. Jika *elemen sekarang > elemen berikutnya*, maka elemen akan bertukar. Jika perbandingan ini sudah sampai ke index akhir. Maka akan diulang lagi ke index awal dan jika perbandingan selanjutnya tidak ada perubahan maka sorting selesai.

Proses pengurutan pada algoritma ini yaitu dengan membandingkan masing-masing suatu 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 gelombang yang terbesar akan naik keatas.

Contoh bubble sort yaitu:

```
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
```

a. Ascending Bubble sort

Dalam contoh ini, akan dijalankan program dengan menuliskan perintah berikut.

```
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
```

Berikutnya, kita akan menjalankan program tersebut dengan menuliskan perintah seperti berikut:

```
list_b = [5, 6, 2, 1, 9, 8, 7]  
print(bubble_sort(list_b))
```

Kemudian kita jalankan, maka output yang dihasilkan dari sorting ascending diatas adalah seperti berikut:

**[1, 2, 5, 6, 7, 8, 9]**

b. Descending Bubble sort

Dalam contoh ini, akan dijalankan program dengan menuliskan perintah berikut.

```
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
```

Berikutnya, kita akan menjalankan program tersebut dengan menuliskan perintah seperti berikut:

```
list_b = [5, 6, 2, 1, 9, 8, 7]  
print(bubble_sort(list_b))
```

Kemudian kita jalankan, maka output yang dihasilkan dari sorting descending diatas adalah seperti berikut:

**[9, 8, 7, 6, 5, 2, 1]**

### 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 sebagai berikut.

```
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  
  
        array[i], array[min_index] = array[min_index], array[i]  
  
    return array
```

#### a. Ascending Selection Sort

Dalam contoh ini, akan dijalankan program dengan menuliskan perintah berikut.

```
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  
  
        array[i], array[min_index] = array[min_index], array[i]  
  
    return array
```

Berikutnya, kita akan menjalankan program tersebut dengan menuliskan perintah seperti berikut:

```
list_c = [4, 6, 9, 3, 1, 2, 4]
print(selection_sort(list_c))
```

Kemudian kita jalankan, maka output yang dihasilkan dari sorting ascending diatas adalah seperti berikut:

**[1, 2, 3, 4, 4, 6, 9]**

#### b. Ascending Selection Sort

Dalam contoh ini, akan dijalankan program dengan menuliskan perintah berikut.

```
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

        array[i], array[min_index] = array[min_index], array[i]

    return array
```

Berikutnya, kita akan menjalankan program tersebut dengan menuliskan perintah seperti berikut:

```
list_c = [4, 6, 9, 3, 1, 2, 4]
print(selection_sort(list_c))
```

Kemudian kita jalankan, maka output yang dihasilkan dari sorting ascending diatas adalah seperti berikut:

**[9, 6, 4, 4, 3, 2, 1]**

### 7.3 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 :

- Tambah data mahasiswa

```
<===== Data Mahasiswa =====
1. Tambah Data Mahasiswa
2. Hapus Data Mahasiswa
3. Urutkan Data Mahasiswa
4. Lihat Data Mahasiswa
5. Tutup
HARAP MEMASUKKAN PILIHAN DENGAN NOMOR 1 SAMPAI 5
Pilih: 1
Jumlah Mahasiswa: 5
Nama Mahasiswa: Haji
Nama Mahasiswa: Mema
Nama Mahasiswa: Shite
Nama Mahasiswa: Orngi
Nama Mahasiswa: Iyaszyi
```

- Hapus data mahasiswa

```
<===== Data Mahasiswa =====
1. Tambah Data Mahasiswa
2. Hapus Data Mahasiswa
3. Urutkan Data Mahasiswa
4. Lihat Data Mahasiswa
5. Tutup
HARAP MEMASUKKAN PILIHAN DENGAN NOMOR 1 SAMPAI 5
Pilih: 2
Data Mahasiswa ['Haji', 'Mema', 'Shite', 'Orngi', 'Iyaszyi']
Hapus Mahasiswa: Haji
Data Mahasiswa ['Mema', 'Shite', 'Orngi', 'Iyaszyi']
```

- Urutkan data mahasiswa

```
<===== Data Mahasiswa =====
1. Tambah Data Mahasiswa
2. Hapus Data Mahasiswa
3. Urutkan Data Mahasiswa
4. Lihat Data Mahasiswa
5. Tutup
HARAP MEMASUKKAN PILIHAN DENGAN NOMOR 1 SAMPAI 5
Pilih: 3
['Iyaszyi', 'Mema', 'Orngi', 'Shite']
```



- Cetak data mahasiswa

```
[ Iyaszyi ] [ Mema ] [ Orngi ] [ Shite ]  
<===== Data Mahasiswa =====  
1. Tambah Data Mahasiswa  
2. Hapus Data Mahasiswa  
3. Urutkan Data Mahasiswa  
4. Lihat Data Mahasiswa  
5. Tutup  
HARAP MEMASUKKAN PILIHAN DENGAN NOMOR 1 SAMPAI 5  
Pilih: 4  
Nama Mahasiswa: Iyaszyi  
Nama Mahasiswa: Mema  
Nama Mahasiswa: Orngi  
Nama Mahasiswa: Shite
```

- Maka code program sebagai berikut:

```
array_mahasiswa = []  
  
def add_mahasiswa():  
    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  
  
def remove_mahasiswa(array_mahasiswa):  
    mahasiswa = array_mahasiswa  
    print("Data Mahasiswa %s" %array_mahasiswa)  
    mahasiswa.remove(input("Hapus Mahasiswa: "))  
    print("Data Mahasiswa %s" %mahasiswa)  
    panggil(array_mahasiswa)  
  
def asc_mahasiswa(array_mahasiswa):  
    mahasiswa = array_mahasiswa  
    mahasiswa.sort()  
    print(mahasiswa)  
    panggil(array_mahasiswa)
```

```

def view_mahasiswa(array_mahasiswa):
    mahasiswa = array_mahasiswa
    for x in mahasiswa:
        print("Nama Mahasiswa: %s" %x)
    panggil(array_mahasiswa)

def panggil(array_mahasiswa):
    print("<===== 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" )
    print("HARAP MEMASUKKAN PILIHAN DENGAN NOMOR 1 SAMPAI 5" )

    pilih = int(input("Pilih: "))
    if(pilih == 1):
        add_mahasiswa()
    elif(pilih == 2):
        remove_mahasiswa(array_mahasiswa)
    elif(pilih == 3):
        asc_mahasiswa(array_mahasiswa)
    elif(pilih == 4):
        view_mahasiswa(array_mahasiswa)
    else:
        print("Selesai")

panggil(array_mahasiswa = array_mahasiswa)

```

## 7.4 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, 2.4}. Buatlah program untuk mengurutkan IPS mahasiswa tersebut dari yang terbesar hingga terkecil dengan menggunakan algoritma Bubble sort !

Code Program:

```
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_nilai = [3.8, 2.9, 3.3, 4.0, 2.4]  
  
print("==== Mengurutkan Nilai Indeks Prestasi Siswa====")  
print(f"Nilai sebelum pengurutan : {list_nilai}")  
bubble_sort(list_nilai)  
print(f"Nilai sesudah pengurutan : {list_nilai}")
```

Output yang dihasilkan:

```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Try the new cross-platform PowerShell https://aka.ms/pscore6  
  
PS E:\Kuliah\Praktikum Pemrograman 1> & C:/Users/ASUS/AppData/Local/Programs/Python/Python310/python.exe "e:/Kuliah/Praktikum Pemrograman 1/Pertemuan 7 Sesudah UTS/Code/tugasno1.py"  
==== Mengurutkan Nilai Indeks Prestasi Siswa====  
Nilai sebelum pengurutan : [3.8, 2.9, 3.3, 4.0, 2.4]  
Nilai sesudah pengurutan : [4.0, 3.8, 3.3, 2.9, 2.4]  
PS E:\Kuliah\Praktikum Pemrograman 1>
```

2. Pada suatu organisasi memiliki 10 anggota dengan nama masing-masing : Pain, Konan, Tobi, Zetsu, Sasori, Hidan, Deidara, Kisame, Kakuzu, dan Itachi . 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!

Code program:

```
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  
  
        array[i], array[min_index] = array[min_index], array[i]  
  
    return array  
  
anggota = ["Pain", "Konan", "Tobi", "Zetsu", "Sasori", "Hidan",  
"Deidara", "Kisame", "Kakuzu", "Itachi"]  
  
print("==== Program Mengurutkan Anggota Organisasi ====")  
print(f"Anggota Terdiri Dari: {anggota}")  
selection_sort(anggota)  
print(f"Urutan Anggota Yaitu: {anggota}")
```

Output yang dihasilkan:

```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Try the new cross-platform PowerShell https://aka.ms/pscore6  
  
PS E:\Kuliah\Praktikum Pemrograman 1> & C:/Users/ASUS/AppData/Local/Programs/Python/Python310/python.exe "e:/Kuliah/Praktikum Pemrograman 1/Pertemuan 7 Sesudah UTS/Code/tugasno2.py"  
==== Program Mengurutkan Anggota Organisasi ====  
Anggota Terdiri Dari: ['Pain', 'Konan', 'Tobi', 'Zetsu', 'Sasori', 'Hidan', 'Deidara', 'Kisame', 'Kakuzu', 'Itachi']  
Urutan Anggota Yaitu: ['Deidara', 'Hidan', 'Itachi', 'Kakuzu', 'Kisame', 'Konan', 'Pain', 'Sasori', 'Tobi', 'Zetsu']  
PS E:\Kuliah\Praktikum Pemrograman 1>
```

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

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\Kuliah\Praktikum Pemrograman 1> & C:/Users/ASUS/AppData/Local/Programs/Python/Python310/python.exe "e:/Kuliah/Praktikum Pemrograman 1/Pertemuan 7 Sesudah UTS/Code/tugasno3.py"
Masukkan Total Buku : 3
Masukkan Judul Buku ke1 : Pada Suatu Hari
Masukkan Judul Buku ke2 : Kancil di Jalan Tol PaliKanci
Masukkan Judul Buku ke3 : Ada Musang di Gunung Ciremai

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

Sorting Buku Secara Ascending
indeks Buku ke-1 : Ada Musang di Gunung Ciremai
indeks Buku ke-2 : Kancil di Jalan Tol PaliKanci
indeks Buku ke-3 : Pada Suatu Hari
PS E:\Kuliah\Praktikum Pemrograman 1>

```

Jawaban:

Code Program:

```

book = []
def add_book():
    jumlah = int(input("Masukkan Total Buku : "))
    for i in range(jumlah):
        jdl = input(f"Masukkan Judul Buku ke{i + 1} : ")
        book.append(jdl)
        jumlah = jumlah - 1
    while(True):
        jumlah = jumlah - 1
        if(jumlah<0):
            break

def ascending_buku(Nbook):
    book = Nbook
    for i in range(1, len(book)):
        item = book[i]
        j = i - 1
        while j >= 0 and item < book[j]:
            book[j + 1] = book[j]
            j -= 1
        book[j + 1] = item

```

```

    print("Sorting Buku Secara Ascending")
    for x in range(len(book)):
        print(f"indeks Buku ke-{x+1} : %s" %book[x])

def descending_book(Nbook):
    book = Nbook
    for i in range(1, len(book)):
        item = book[i]
        j = i - 1
        while j >= 0 and item > book[j]:
            book[j + 1] = book[j]
            j -= 1
        book[j + 1] = item
    print("Sorting Buku Secara Descending")
    for x in range(len(book)):
        print(f"Judul Buku ke-{x+1} : %s" %book[x])

add_book()

print (" ")
print("<=====Urutkan?=====>")
print("1. Insertion Ascending")
print("2. Insertion Descending")
pilih = int(input("Pilih: "))
print(" ")

if(pilih==1):
    ascending_buku(book)
elif(pilih==2):
    descending_book(book)
else:
    print("Pilihan tidak ada dalam program")

```