	LABORATORIUM SISTEM INFORMASI **UNIVERSITAS TANJUNGPURA PONTIANAK** Gedung FMIPA Jl. Prof. Dr. Hadari Nawawi Pontianak		
	Hari/Tanggal: Kamis/27 Februari 2025		
Nama Mahasiswa	Rafli Pratama	**Mata Kuliah Praktikum**	Algoritma dan Struktur Data
NIM	H1101241008	**Dosen Pengampu**	Ilhamsyah S.Si, M.Cs
Semester	2 Ganjil	**Paraf Dosen Pengampu**	
Kelas	Sistem Informasi A	**Asisten Praktikum**	
Nilai		**Paraf Asisten Praktikum**	

LEMBAR KERJA PRAKTIKUM

MATERI PRAKTIKUM : BubbleSort

Latihan 1.

```
#bubblesort
def bubblesort(S):
    n = len(S)
    for i in range(n):
        print(S)
        for j in range (n-1):
            if S[j] > S[j+1]:
                S[j],S[j+1] = S[j+1], S[j]
    return S

S = [50,30,40,10,20]
print(f'Sebelum di sortir {S}')
```

```
bubblesort(S)
print(f'Setelah di sortir {S}')
```

Output Program :

```
Sebelum di sortir [50, 30, 40, 10, 20]
[50, 30, 40, 10, 20]
[30, 40, 10, 20, 50]
[30, 10, 20, 40, 50]
[10, 20, 30, 40, 50]
[10, 20, 30, 40, 50]
Setelah di sortir [10, 20, 30, 40, 50]
```

Penjelasan :

Program ini digunakan untuk mengurutkan sebuah list menggunakan **Bubble Sort**, yaitu metode pengurutan dengan cara menukar dua elemen yang berdekatan jika urutannya salah. Proses ini terus diulang hingga seluruh elemen dalam list berada dalam urutan yang benar.

Di dalam program, pertama-tama ada list S = [50,30,40,10,20], yang merupakan data awal yang akan diurutkan. Sebelum masuk ke proses pengurutan, program mencetak list sebelum disortir.

Fungsi bubblesort(S) pertama-tama menentukan panjang list dengan n = len(S). Lalu, ada **dua perulangan** untuk melakukan proses sorting.

Perulangan pertama (for i in range(n)) berfungsi untuk melakukan iterasi sebanyak jumlah elemen dalam list. Di dalam perulangan ini, program juga mencetak kondisi S di setiap iterasi untuk melihat proses perubahan data selama sorting berlangsung.

Perulangan kedua (for j in range(n-1)) bertugas membandingkan setiap elemen yang berdekatan. Jika elemen saat ini (S[j]) lebih besar dari elemen setelahnya (S[j+1]), maka program akan menukarnya (S[j], S[j+1] = S[j+1], S[j]). Proses ini akan membuat elemen terbesar "menggelembung" ke posisi yang seharusnya dalam setiap iterasi.

Setelah fungsi selesai dijalankan, program mencetak list setelah disortir untuk melihat hasil akhirnya.

Latihan 2.

```
def bubblesort(S):
    n = len(S)
    for i in range(n):
        # print(S)
```

```
        for j in range (n-1):
            if S[j] > S[j+1]:
                S[j],S[j+1] = S[j+1], S[j]
    return S

S = list('Sistem informasi kelas pak ilham')
print(f'Sebelum di sortir {S}')
bubblesort(S)
print(f'Sesudah di sortir {S}')
```

Output Program :

```
Sebelum di sortir ['S', 'i', 's', 't', 'e', 'm', ' ', 'i',
'n', 'f', 'o', 'r', 'm', 'a', 's', 'i', ' ', 'k', 'e',
'l', 'a', 's', ' ', 'p', 'a', 'k', ' ', 'i', 'l', 'h',
'a', 'm']
Sesudah di sortir [' ', ' ', ' ', ' ', 'S', 'a', 'a', 'a',
'a', 'e', 'e', 'f', 'h', 'i', 'i', 'i', 'i', 'k', 'k',
'l', 'l', 'm', 'm', 'm', 'n', 'o', 'p', 'r', 's', 's',
's', 't']
```

Penjelasan :

Program ini digunakan untuk mengurutkan karakter dalam sebuah string menggunakan **Bubble Sort**. String yang digunakan adalah "Sistem informasi kelas pak ilham", yang diubah menjadi list karakter sebelum dilakukan proses sorting.

Pertama, string "Sistem informasi kelas pak ilham" diubah menjadi list menggunakan list(), sehingga setiap huruf dan spasi dalam string akan menjadi elemen dalam list S. List ini kemudian dicetak sebelum sorting dilakukan agar kita bisa melihat kondisi awalnya.

Fungsi bubblesort(S) bekerja dengan cara menghitung panjang list S terlebih dahulu (n = len(S)). Kemudian, ada **dua perulangan** untuk melakukan proses sorting.

- **Perulangan pertama (for i in range(n))** digunakan untuk mengontrol jumlah iterasi yang dilakukan dalam proses sorting.
- **Perulangan kedua (for j in range(n-1))** bertugas untuk membandingkan setiap elemen yang berdekatan. Jika elemen saat ini (S[j]) lebih besar dari elemen setelahnya (S[j+1]), maka program akan menukarnya (S[j], S[j+1] = S[j+1], S[j]).

Setelah proses sorting selesai, program mencetak kembali list yang sudah diurutkan. Namun, karena program mengurutkan list berdasarkan **kode ASCII**, maka hasil sorting akan membuat **spasi** berada di

awal list karena memiliki nilai ASCII lebih kecil dibandingkan huruf. Selanjutnya, huruf kapital akan muncul sebelum huruf kecil karena urutan ASCII-nya.

Latihan 3.

```
#Latihan 2
L=[]
for i in range(5):
    nm = input('Masukkan nama')
    L.append(nm)
print(f'List Belum urut adalah {L}')
n = len(L)
for i in range(n):
    for j in range(n -1):
        if L[j] > L[j+1]:
            L[j],L[j+1] = L[j+1], L[j]
print(f'List terurutnya adalah {L}')
```

Output Program :
List Belum urut adalah ['Rafli', 'Nabila', 'Pratama', 'Juwita', 'Rivaldi']
List terurutnya adalah ['Juwita', 'Nabila', 'Pratama', 'Rafli', 'Rivaldi']

Penjelasan :
Program ini digunakan untuk menginput 5 nama, menyimpannya dalam list L, lalu mengurutkannya menggunakan **Bubble Sort** secara **ascending (A-Z)**.

Pertama, program membuat list kosong L = []. Kemudian, ada **perulangan (for i in range(5))** yang meminta user untuk memasukkan nama sebanyak 5 kali. Setiap nama yang dimasukkan akan langsung ditambahkan ke dalam list menggunakan L.append(nm). Setelah semua nama dimasukkan, program mencetak list sebelum diurutkan agar kita bisa melihat urutannya sebelum sorting.

Setelah itu, program mulai melakukan proses **Bubble Sort**. Panjang list dihitung dengan n = len(L), lalu ada **dua perulangan bersarang** untuk mengurutkan elemen dalam list.

- **Perulangan pertama (for i in range(n))** digunakan untuk mengontrol jumlah iterasi sorting.
- **Perulangan kedua (for j in range(n-1))** digunakan untuk membandingkan setiap dua elemen yang berdekatan. Jika elemen di posisi j lebih besar dari elemen di posisi j+1, maka keduanya akan ditukar (L[j], L[j+1] = L[j+1], L[j]).

Setelah sorting selesai, program mencetak list yang sudah terurut. Hasil akhirnya adalah list yang berisi nama-nama dalam **urutan alfabetis** dari A-Z.

Latihan 4.

```
#Latihan 3

L = []
while True:
    nm = int(input('Masukkan angka'))
    if nm != 0:
        L.append(nm)
    else:
        break

print(f'List belum terurut adalah {L}')
n = len(L)
for i in range(n):
    for j in range(n -1):
        if L[j] > L[j+1]:
            L[j],L[j+1] = L[j+1], L[j]

print(f'list terurutnya adalah {L}')
```

Output Program :
List belum terurut adalah [4, 6, 5, 4, 3, 4, 5, 6, 7, 6]
list terurutnya adalah [3, 4, 4, 4, 5, 5, 6, 6, 6, 7]

Penjelasan :

Program ini digunakan untuk menginput angka dari user, menyimpannya dalam list L, lalu mengurutkannya secara ascending (dari kecil ke besar) menggunakan Bubble Sort.

Pertama, program membuat list kosong L = []. Lalu, ada perulangan while True yang meminta user untuk memasukkan angka secara terus-menerus. Jika angka yang dimasukkan bukan 0, angka tersebut akan langsung ditambahkan ke dalam list menggunakan L.append(nm). Namun, jika user memasukkan angka

0, program akan keluar dari perulangan dengan break. Setelah semua angka dimasukkan, program mencetak list sebelum diurutkan agar bisa melihat urutan awalnya.

Setelah itu, program mulai melakukan Bubble Sort untuk mengurutkan angka di dalam list L.

- Perulangan pertama (for i in range(n)) digunakan untuk mengontrol jumlah iterasi sorting.
- Perulangan kedua (for j in range(n-1)) digunakan untuk membandingkan dua angka yang berdekatan. Jika angka di posisi j lebih besar dari angka di posisi j+1, maka program akan menukarnya (L[j], L[j+1] = L[j+1], L[j]).

Setelah sorting selesai, program mencetak list yang sudah terurut dari angka terkecil ke terbesar.

Latihan 5.

```
def bubblesort(S):
    n = len(S)
    for i in range(n):
        # print(S)
        for j in range (n-1):
            if S[j] > S[j+1]:
                S[j],S[j+1] = S[j+1], S[j]
    return S

data1 = []
import random as r
for i in range(10):
    data = [r.randint(0,100)]
    data1.append(data)

print(f'Sebelum di urutkan {data1}')

urut = bubblesort(data1)

print(f'Data setelah di urutkan {urut}')
```

Output Program :
Sebelum di urutkan [[84], [27], [82], [2], [11], [65], [1], [42], [87], [12]]
Data setelah di urutkan [[1], [2], [11], [12], [27], [42], [65], [82], [84], [87]]

Penjelasan :

Program ini digunakan untuk **membuat 10 angka acak**, menyimpannya dalam list data1, lalu mengurutkannya menggunakan **Bubble Sort** secara **ascending (dari kecil ke besar)**. Pertama, ada fungsi bubblesort(S) yang bertugas untuk mengurutkan list S menggunakan **Bubble Sort**. Fungsi ini akan melakukan perulangan sebanyak panjang list S, lalu dalam setiap iterasi akan membandingkan dua elemen yang berdekatan. Jika elemen pertama lebih besar dari elemen kedua, maka keduanya akan ditukar. Setelah semua iterasi selesai, list akan terurut dari angka terkecil ke terbesar.

Selanjutnya, program membuat list data1 yang berisi 10 angka acak. Ini dilakukan dengan menggunakan **modul random** dan fungsi r.randint(0,100) yang akan menghasilkan angka acak antara **0 hingga 100**. Angka-angka ini kemudian ditambahkan ke dalam list data1. Namun, ada kesalahan di kode ini, yaitu angka acak disimpan dalam bentuk **list satu elemen** (data = [r.randint(0,100)]), sehingga data1 akan menjadi list dalam list, seperti [[10], [25], [89]], yang menyebabkan Bubble Sort tidak berjalan dengan benar.

Terakhir, program mencetak list sebelum diurutkan, lalu memanggil fungsi bubblesort(data1) untuk mengurutkan angka-angka tersebut. Setelah sorting selesai, program mencetak hasil akhirnya. Namun, karena ada kesalahan dalam cara menyimpan angka acak, sorting tidak akan berjalan dengan benar. Solusi untuk memperbaikinya adalah mengganti data = [r.randint(0,100)] menjadi data = r.randint(0,100), atau langsung menggunakan list comprehension seperti data1 = [r.randint(0,100) for _ in range(10)]. Dengan cara ini, angka-angka dalam data1 akan benar-benar berupa integer, sehingga Bubble Sort bisa berjalan dengan baik dan mengurutkan angka dari yang **terkecil ke terbesar**.

Latihan 6.

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

S = [45,43,47,40,20]
print(f'Sebelum di sortir {S}')
bubblesort(S)
```

```
print(f'Setelah di sortir {S}')
```

Output Program :
Sebelum di sortir [45, 43, 47, 40, 20]
[45, 43, 47, 40, 20]
[43, 45, 40, 20, 47]
[43, 40, 20, 45, 47]
[40, 20, 43, 45, 47]
[20, 40, 43, 45, 47]
Setelah di sortir [20, 40, 43, 45, 47]

Penjelasan :

Program ini dibuat untuk **mengurutkan angka dalam list S menggunakan Bubble Sort** secara **ascending (dari kecil ke besar)**. Pertama, program mendefinisikan fungsi bubblesort(S) yang akan mengambil list S sebagai input dan mengurutkannya. Di dalam fungsi ini, panjang list dihitung menggunakan len(S), lalu dilakukan dua perulangan bersarang.

Perulangan pertama (for i in range(n)) digunakan untuk mengontrol **jumlah iterasi sorting**, sedangkan perulangan kedua (for j in range(n-1)) digunakan untuk membandingkan dua elemen yang berdekatan dalam list. Jika elemen pertama lebih besar dari elemen kedua (S[j] > S[j+1]), maka keduanya ditukar (S[j], S[j+1] = S[j+1], S[j]). Proses ini akan terus berulang hingga seluruh angka dalam list terurut dari yang terkecil ke terbesar.

Setelah fungsi bubblesort(S) dibuat, program mendeklarasikan list S dengan nilai [45, 43, 47, 40, 20]. Lalu, program mencetak list sebelum diurutkan dengan print(f'Sebelum di sortir {S}'). Selanjutnya, fungsi bubblesort(S) dipanggil untuk melakukan sorting, dan dalam setiap iterasi, list akan ditampilkan menggunakan print(S) agar kita bisa melihat perubahan urutan setiap kali angka ditukar. Setelah sorting selesai, program mencetak hasil akhirnya dengan print(f'Setelah di sortir {S}').

Namun, ada satu kesalahan dalam kode ini, yaitu fungsi bubblesort(S) tidak mengembalikan list yang sudah terurut ke variabel S. Sehingga, meskipun sorting dilakukan di dalam fungsi, S di luar fungsi tetap tidak berubah. **Solusi untuk memperbaikinya** adalah mengubah pemanggilan fungsi menjadi S = bubblesort(S), agar hasil sorting tersimpan kembali dalam S.

Percobaan 1.

```
#Percobaan 1
def bubblesort(S):
    n = len(S)
    for i in range(n):
        # print(S)
        for j in range (n-1):
```

```
            if S[j][1] > S[j+1][1]:
                S[j],S[j+1] = S[j+1], S[j]
    return S

S = [['Alan', 72], ['Budi', 55], ['Kusuma', 82], ['Ardi',
61], ['Taufik', 95], ['Desi',63]]
print(f'Sebelum di urutkan {S}')
bubblesort(S)
print(f'Setelah di urutkan {S}')
```

Output Program :
Sebelum di urutkan [['Alan', 72], ['Budi', 55], ['Kusuma', 82], ['Ardi', 61], ['Taufik', 95], ['Desi', 63]]
Setelah di urutkan [['Budi', 55], ['Ardi', 61], ['Desi', 63], ['Alan', 72], ['Kusuma', 82], ['Taufik', 95]]

Penjelasan :

Program ini dibuat untuk **mengurutkan data berdasarkan nilai angka pada indeks ke-1 dalam list S menggunakan Bubble Sort** secara **ascending (dari kecil ke besar)**. Dalam program ini, setiap elemen list S adalah **sub-list berisi nama dan angka**, seperti ['Alan', 72], di mana **nama berada di indeks ke-0 dan angka berada di indeks ke-1**.

Fungsi bubblesort(S) bekerja dengan menghitung panjang list menggunakan len(S), lalu melakukan **dua perulangan bersarang** untuk menerapkan algoritma Bubble Sort. Perulangan pertama (for i in range(n)) mengontrol jumlah iterasi sorting, sementara perulangan kedua (for j in range(n-1)) membandingkan dua elemen yang berdekatan berdasarkan **nilai angka di indeks ke-1** (S[j][1] > S[j+1][1]). Jika angka pada elemen pertama lebih besar dari angka pada elemen kedua, maka elemen-elemen tersebut ditukar (S[j], S[j+1] = S[j+1], S[j]). Proses ini terus berulang hingga semua elemen dalam list terurut dari angka **terkecil ke terbesar**.

Setelah fungsi bubblesort(S) selesai dibuat, program mendeklarasikan list S yang berisi beberapa nama beserta angkanya. Program mencetak **data sebelum diurutkan**, lalu memanggil fungsi bubblesort(S) untuk melakukan sorting. Terakhir, program mencetak **data setelah diurutkan**.

Namun, ada kesalahan dalam kode ini, yaitu **fungsi bubblesort(S) tidak mengembalikan hasil sorting ke variabel S di luar fungsi**. Sehingga, meskipun sorting dilakukan di dalam fungsi, data yang dicetak setelahnya tetap tidak berubah. **Solusi untuk memperbaikinya** adalah dengan menyimpan hasil sorting ke dalam variabel S saat memanggil fungsi, seperti S = bubblesort(S), agar hasil sorting tersimpan dan bisa ditampilkan dengan benar.

Percobaan 2.

```
def bubblesort(S):
    n = len(S)
    for i in range(n):
        # print(S)
        for j in range (n-1):
            if S[j][1] > S[j+1][1]:
                S[j],S[j+1] = S[j+1], S[j]
    return S

def input_data ():
    data = []
    while True:
        nama = input('Masukkan nama : ("stop") untuk
berhenti')
        if nama in ['stop','Stop', 'STOP']:
            break
        nilai = int(input('Masukkan Nilai'))
        data.append([nama,nilai])
    return data

data = input_data()
print(f'Sebelum di urutkan {data}')
bubblesort(data)
print(f'Setelah di urutkan {data}')
```

Output Program :
Sebelum di urutkan [['Rafli', 90], ['Nabila', 100], ['Elga', 75], ['Tedrik', 80], ['Nabila Imoed', 10000]]
Setelah di urutkan [['Elga', 75], ['Tedrik', 80], ['Rafli', 90], ['Nabila', 100], ['Nabila Imoed', 10000]]

Penjelasan :

Program ini dibuat untuk **mengurutkan data berdasarkan nilai angka menggunakan Bubble Sort**, di mana setiap data berisi **nama dan nilai** dalam bentuk list bersarang ([[nama, nilai], [nama, nilai], ...]).

Pertama, ada fungsi bubblesort(S), yang bertugas melakukan **proses pengurutan** menggunakan algoritma Bubble Sort. Panjang list dihitung menggunakan len(S), lalu program melakukan dua

perulangan bersarang. Perulangan pertama (for i in range(n)) digunakan untuk mengontrol jumlah iterasi sorting, sedangkan perulangan kedua (for j in range(n-1)) digunakan untuk **membandingkan nilai pada indeks ke-1** dalam setiap sub-list (S[j][1] > S[j+1][1]). Jika nilai dari elemen pertama lebih besar dari nilai elemen kedua, maka kedua elemen tersebut **ditukar** (S[j], S[j+1] = S[j+1], S[j]). Proses ini berulang hingga seluruh elemen terurut dari **nilai terkecil ke terbesar**.

Selanjutnya, ada fungsi input_data(), yang digunakan untuk **memasukkan data nama dan nilai secara manual**. Fungsi ini akan meminta user untuk memasukkan nama, lalu memasukkan nilai dalam bentuk angka (int). Data yang dimasukkan kemudian disimpan dalam list data dalam bentuk **list bersarang** ([nama, nilai]). Jika user mengetikkan "stop", maka proses input akan berhenti.

Setelah itu, program memanggil fungsi input_data() dan menyimpannya dalam variabel data. List sebelum diurutkan dicetak dengan print(f'Sebelum di urutkan {data}'), kemudian fungsi bubblesort(data) dijalankan untuk mengurutkan data berdasarkan nilai. Terakhir, hasil setelah diurutkan ditampilkan dengan print(f'Setelah di urutkan {data}').

Namun, ada **kesalahan dalam kode ini**, yaitu **hasil sorting tidak disimpan kembali ke variabel data**. Sehingga, meskipun sorting dilakukan, list data di luar fungsi tidak berubah. **Solusinya**, tambahkan data = bubblesort(data) agar hasil sorting tersimpan dengan benar dan dapat ditampilkan.