

Rabu, 3 April 2019

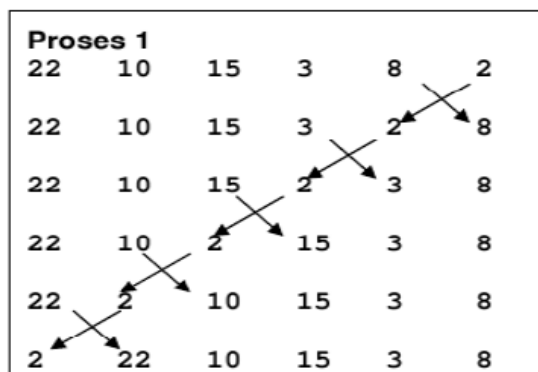
LAPORAN KEGIATAN (BUBBLE SORT)

A. DASAR TEORI

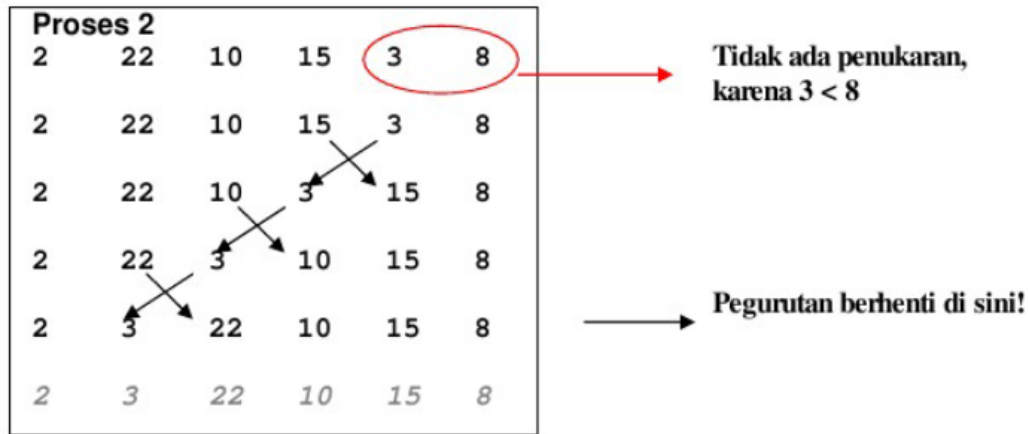
1. Bubble Sort adalah algoritma pengurutan data dengan cara menukar data dengan data disebelahnya secara terus menerus sampai dalam satu iterasi tertentu tidak ada lagi perubahan. Algoritma ini akan mengurutkan data dari yang terbesar ke terkecil (ascending) atau sebaliknya (descending). Diberi nama "**Bubble**" karena proses pengurutan secara berangsur-angsur bergerak/berpindah ke posisinya yang tepat, seperti gelembung yang keluar dari sebuah gelas bersoda. Bubble Sort mengurutkan data dengan cara membandingkan elemen sekarang dengan elemen berikutnya. Untuk pengurutan secara ascending dengan cara membandingkan elemen sekarang lebih besar dari elemen berikutnya maka kedua elemen tersebut ditukar. Sedangkan untuk pengurutan secara descending dengan cara membandingkan elemen sekarang lebih kecil dari elemen berikutnya maka kedua elemen tersebut ditukar.

Contoh array A = [22,10,15,3,8,2].

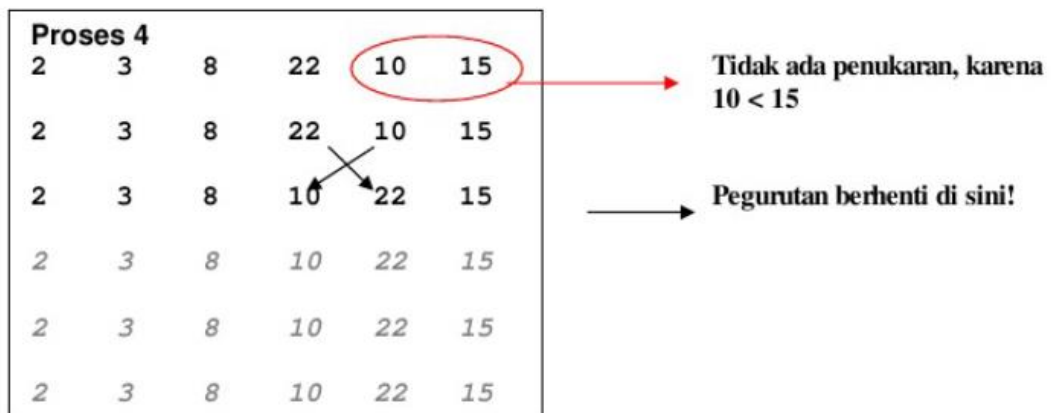
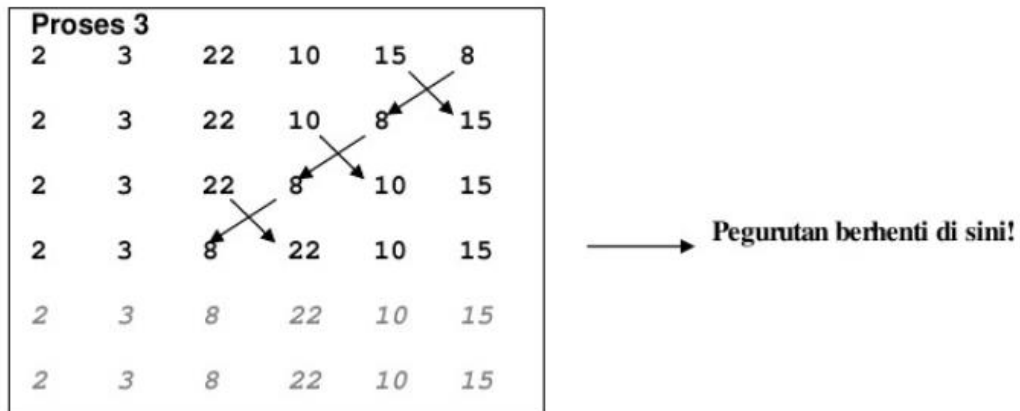
Proses Bubble Sort



Pada program di atas, pengecekan dimulai dari data yang paling akhir, kemudian dibandingkan dengan data di depannya, jika data di depannya lebih besar maka akan ditukar.



Pada proses kedua, pengecekan dilakukan sampai dengan data ke-2 karena data pertama pasti sudah paling kecil.



Proses 5

2	3	8	10	22	15
2	3	8	10	15	22
2	3	8	10	15	22
2	3	8	10	15	22
2	3	8	10	15	22
2	3	8	10	15	22

→ Pegurutan berhenti di sini!

B. PEMBAHASAN

1. Pada praktikum [pertama](#) ini akan mencoba membuat program menampilkan nilai maksimum, jumlah nilai maksimum, dan indeks nilai maksimum pada Array.

Sudah ada Array **A** satu dimensi. Inisialisasi isinya dengan ilustrasi sebagai berikut:

0	1	2	3	4	5	6	7	8	9
12	15	7	10	25	2	17	25	5	20

Susun program untuk mencetak:

- a. Isi array yang nilainya terbesar. Outputnya seperti berikut ini: **max = 25.**
- b. Isi array yang terbesar, dan ada berapa buah nilai yang terbesar tersebut. Outputnya seperti berikut ini: **max = 25**

jum = 2

- c. Isi array yang terbesar, ada berapa buah nilai yang terbesar tersebut, dan sebutkan lokasi(indeks) dimana nilai tersebut berada. Outputnya seperti berikut ini: **max = 25**

jum = 2

indeks = 4, 7

Algoritma

Pertama membuat List angka untuk menyimpan angka-angkanya. Kemudian membuat Array kosong dengan struktur data List untuk menyimpan angka

maksimum dan indeks angka maksimum. Setelah itu membuat suatu perulangan sebanyak jumlah angka yang ada di dalam List, lalu menambahkan percabangan di dalam perulangan tersebut. Jika angka yang ada pada List lebih besar dari sama dengan angka maksimum pada List, maka angka tersebut di masukkan ke dalam List angka maksimum, setelah itu memasukkan indeks angka tersebut ke dalam List indeks angka maksimum.

```
< > 1_soal.py x
1 data = [12,15,7,10,25,2,17,25,5,20]
2 maks = []
3 indeks = []
4 for i in range(0, len(data) - 1):
5     if data[i] >= max(data):
6         maks.append(data[i])
7         indeks.append(i)
8 print("Max : ", end="")
9 for m in maks:
10     print(m, end=" ")
11 print()
12 print("Jum :",len(maks))
13 print("Indeks: ", end="")
14 for i in indeks:
15     print(i, end=" ")
```

Source Code Program

<Penjelasan dari Source Code di atas>

- Pada baris nomor 1, berfungsi untuk menyimpan angka-angka ke dalam List.
- Pada baris nomor 2, berfungsi untuk menampung angka maksimum.
- Pada baris nomor 3, berfungsi untuk menampung indeks angka maksimum.

- Pada baris nomor 4, berfungsi untuk melakukan perulangan sebanyak jumlah angka pada List.
- Pada baris nomor 5 sampai 7, berfungsi untuk melakukan suatu percabangan dimana jika angka-angka pada List lebih besar dari angka maksimum yaitu angka 25. Maka angka tersebut akan di masukkan ke dalam List maks dan Indeks angkanya akan dimasukan ke dalam List indeks.
- Pada baris nomor 8 sampai 15, berfungsi untuk menampilkan angka maksimum, jumlah angka maksimum, dan indeks angka maksimum.

<Hasil Program di jalankan menggunakan IDLE Python>

```
Max : 25 25
Jum : 2
Indeks: 4 7
>>> |
```

2. Pada praktikum kedua ini akan mengembangkan program pada praktikum pertama, program kali ini adalah menukar(swap) isi Array **A** pada indeks 0 dengan indeks 1, ilustrasi: (ingat algoritma tukar isi gelas)

Array A sebelum di tukar isi indeks 0 dengan indeks 1

0	1	2	3	4	5	6	7	8	9
12	15	7	10	25	2	17	25	5	20

Array A setelah di tukar isi indeks 0 dengan indeks 1

0	1	2	3	4	5	6	7	8	9
15	12	7	10	25	2	17	25	5	20

Algoritma

Pertama membuat List angka untuk menyimpan angka-angkanya. Kemudian membuat variabel **tempat** yang berisikan angka pada List dengan indeks ke-0 yang fungsinya untuk menyimpan angka sementara. Kemudian angka pada List dengan indeks ke-0 di ubah isinya menjadi angka pada List indeks ke-1. Setelah itu angka pada List dengan indeks ke-1 di ubah isinya menjadi variabel **tempat**.

```

< > 2_soal.py x
1 data = [12,15,7,10,25,2,17,25,5,20]
2 tempat = data[0]
3 data[0] = data[1]
4 data[1] = tempat
5 print(data)

```

Source Code Program

<Penjelasan dari Source Code di atas>

- Pada baris nomor 1, berfungsi untuk menyimpan angka-angka ke dalam List.
- Pada baris nomor 2, berfungsi untuk menyimpan angka indeks ke-0 pada List data ke variabel tempat.
- Pada baris nomor 3, berfungsi untuk mengubah angka indeks ke-0 pada List data dengan angka indeks ke-1 pada List data.
- Pada baris nomor 4, berfungsi untuk mengubah angka indeks ke-1 pada List data dengan isi variabel tempat.
- Pada baris nomor 5, berfungsi untuk menampilkan angka pada List data yang telah di perbarui.

<Hasil Program di jalankan menggunakan IDLE Python>

```

[15, 12, 7, 10, 25, 2, 17, 25, 5, 20]
>>> |

```

3. Pada praktikum [ketiga](#) ini akan mengembangkan program pada praktikum kedua, program kali ini adalah menukar(swap) isi Array **A** pada indeks ke-i dengan indeks ke-i+1 (i dimulai dari 0 sampai **A.length-2**), dengan syarat jika isi Array **A** pada indeks ke-i lebih besar dari isi Array **A** indeks ke-i+1. (Gunakan perulangan)

Ilustrasi:

i i+1									
0	1	2	3	4	5	6	7	8	9
12	15	7	10	25	2	17	25	5	20
12	15	7	10	25	2	17	25	5	20

i i+1									
0	1	2	3	4	5	6	7	8	9
12	15	7	10	25	2	17	25	5	20
12	7	15	10	25	2	17	25	5	20

Hingga sampai di bawah ini

								i	i+1
0	1	2	3	4	5	6	7	8	9
12	7	10	15	2	17	25	5	25	20
12	7	10	15	2	17	25	5	20	25

Algoritma

Pertama membuat List angka untuk menyimpan angka-angkanya. Kemudian membuat suatu perulangan sebanyak jumlah angka yang ada di dalam List di kurang 1, karena Array indeksnya di mulai dari angka 0. Lalu menambahkan suatu percabangan di dalam perulangan tersebut. Jika angka pada List lebih besar dari angka berikutnya, maka kedua angka tersebut bertukar posisi.

```

< > 3_soal.py x
1 data = [12,15,7,10,25,2,17,25,5,20]
2 for i in range(0,len(data) - 1):
3     if data[i] > data[i+1]:
4         data[i],data[i+1] = data[i+1],data[i]
5 print(data)

```

Source Code Program

<Penjelasan dari Source Code di atas>

- Pada baris nomor 1, berfungsi untuk menyimpan angka-angka ke dalam List.
- Pada baris nomor 2, berfungsi untuk melakukan perulangan sebanyak jumlah angka pada List data.
- Pada baris nomor 3, berfungsi untuk melakukan suatu percabangan dimana jika angka sekarang pada List data lebih besar dari angka berikutnya pada List data maka kedua angka tersebut bertukar posisi.
- Pada baris nomor 5, berfungsi untuk menampilkan angka pada List yang telah di perbarui.

<Hasil Program di jalankan menggunakan IDLE Python>

```
[12, 7, 10, 15, 2, 17, 25, 5, 20, 25]  
>>> |
```

4. Pada praktikum [keempat](#) ini akan mengembangkan program pada praktikum ketiga, program kali ini adalah melakukan **bubbling** untuk angka yang belum di bubble-up sehingga isi Array **A** adalah:

0	1	2	3	4	5	6	7	8	9
2	5	7	10	12	15	17	20	25	25

Algoritma

Pertama membuat List angka untuk menyimpan angka-angkanya. Kemudian membuat suatu perulangan sebagai Iterasi, sebanyak jumlah angka yang ada di dalam List. Di dalam perulangan tersebut, menambahkan perulangan kembali sebanyak jumlah angka yang ada di dalam di kurang 1, karena Array indeksinya di mulai dari angka 0. Selanjutnya menambahkan suatu percabangan, jika angka pada List lebih besar dari angka berikutnya, maka kedua angka tersebut bertukar posisi.


```

4_soal.py
1 data = [12,15,7,10,25,2,17,25,5,20]
2 for i in range(0,len(data)):
3     for j in range(0, len(data) - 1):
4         if data[j] > data[j+1]:
5             data[j],data[j+1] = data[j+1],data[j]
6 print(data)

```

<Penjelasan dari Source Code di atas>

- Pada baris nomor 1, berfungsi untuk menyimpan angka-angka ke dalam List.
- Pada baris nomor 2, berfungsi untuk melakukan perulangan sebanyak jumlah angka pada List sebagai iterasi/proses.
- Pada baris nomor 3, berfungsi untuk melakukan perulangan sebanyak jumlah angka pada List.
- Pada baris nomor 4, berfungsi untuk melakukan suatu percabangan dimana jika angka sekarang pada List data lebih besar dari angka berikutnya pada List data maka kedua angka tersebut bertukar posisi.
- Pada baris nomor 5, berfungsi untuk menampilkan angka pada List yang telah di perbarui.

<Hasil Program di jalankan menggunakan IDLE Python>

```

[2, 5, 7, 10, 12, 15, 17, 20, 25, 25]
>>> |

```

5. Pada praktikum [kelima](#) ini akan mengembangkan program pada praktikum keempat, program kali ini adalah menambahkan sebuah inputan sebagai banyaknya data dan sebuah inputan sebagai data.

Algoritma

Pertama membuat sebuah input yaitu jumlah data yang di masukkan, setelah itu membuat input lagi untuk memasukkan angka-angka yang akan di simpan di dalam List. Angka-angka tersebut di pisah dengan spasi. Kemudian melakukan

pengecekan jika jumlah seluruh angka pada List lebih dari jumlah data yang di masukkan, maka akan memunculkan sebuah pesan. Jika pernyataan tersebut tidak terpenuhi maka akan lanjut ke proses berikutnya. Kemudian membuat suatu perulangan sebagai Iterasi, sebanyak jumlah angka yang ada di dalam List. Di dalam perulangan tersebut, menambahkan perulangan kembali sebanyak jumlah angka yang ada di dalam List di kurang 1, karena Array indeksnya di mulai dari angka 0. Selanjutnya menambahkan suatu percabangan, jika angka pada List lebih besar dari angka berikutnya, maka kedua angka tersebut bertukar posisi.

```
< > 5_soal.py x
1  jml_data = int(input("Masukkan jumlah data yang ingin di masukkan: "))
2  data = [int(s) for s in input("Masukkan data: ").split()]
3  if len(data) > jml_data:
4      print("Maaf, data yang anda masukkan melebihi %s data" % (jml_data))
5  elif len(data) < jml_data:
6      print("Maaf, data yang anda masukkan kurang dari %s data" % (jml_data))
7  else:
8      for i in range(0,len(data)):
9          for j in range(0, len(data) - 1):
10             if data[j] > data[j+1]:
11                 data[j],data[j+1] = data[j+1],data[j]
12     print(data)
```

<Penjelasan dari Source Code di atas>

- Pada baris nomor 1, berfungsi untuk menginputkan jumlah data angka.
- Pada baris nomor 2, berfungsi untuk menginputkan data sebanyak jumlah data yang telah di inputkan sebelumnya sekaligus data tersebut akan di masukkan ke dalam List.
- Pada baris nomor 3 sampai 7, berfungsi untuk melakukan suatu percabangan dimana jika jumlah angka-angka yang di masukkan lebih besar dari jumlah data angka yang di inputkan maka akan menampilkan pesan "Maaf, data yang anda masukkan melebihi (jumlah data sekian) data". Kemudian program berhenti atau selesai. Selanjutnya akan melakukan pengecekan kembali, jika jumlah angka-angka yang di masukkan kurang dari jumlah data angka yang di inputkan maka akan

menampilkan pesan "Maaf, data yang anda masukkan kurang dari (jumlah data sekian) data". Kemudian program berhenti atau selesai. Apabila dari kedua pernyataan tersebut tidak terpenuhi maka akan lanjut ke proses berikutnya.

- Pada baris nomor 8, berfungsi untuk melakukan perulangan sebanyak jumlah angka pada List sebagai iterasi/proses.
- Pada baris nomor 9, berfungsi untuk melakukan perulangan sebanyak jumlah angka pada List.
- Pada baris nomor 10, berfungsi untuk melakukan suatu percabangan dimana jika angka sekarang pada List data lebih besar dari angka berikutnya pada List data maka kedua angka tersebut bertukar posisi.
- Pada baris nomor 11, berfungsi untuk menampilkan angka pada List yang telah di perbarui.

```
<Hasil Program di jalankan menggunakan IDLE Python>  
Masukkan jumlah data yang ingin di masukkan: 6  
Masukkan data: 4 8 1 2 3 9  
[1, 2, 3, 4, 8, 9]  
>>> |
```

C. KESIMPULAN

Bubble Sort adalah salah satu algoritma pengurutan data. Bubble Sort ini merupakan metode pengurutan yang tidak efisien karena ketika mengurutkan data yang sangat besar akan sangat lambat prosesnya.

D. DAFTAR PUSTAKA

Wibowo,Fitri. __, *Bubble Sort*, [pdf]