

Praktikum Struktur Data

Modul 5 - Sorting

Senin, 9 April 2022

Tujuan dari modul ini agar mahasiswa memahami konsep Sorting (Pengurutan), serta modifikasinya untuk mempercepat proses pengurutan. Kerjakan tugas-tugas yang terdapat dalam modul ini, dengan ketentuan sebagai berikut :

1. Semua jawaban modul dikerjakan dalam format *.ipynb (baik konsep maupun implementasi), jangan lupa diberikan **heading** di setiap cell jawaban, misalkan untuk jawaban konsep Nomor 1, diberikan heading **Konsep_1**. Silahkan dilihat contoh *lecture notes* yang saya telah bagikan melalui website.
2. Penamaan file ipynb adalah : NPM_ModulX_TopikModul.ipynb, misalkan, 200411100077_Modul5_Sorting.ipynb
3. Print menjadi file pdf, dokumen ipynb tersebut dengan nama yang sama, hanya saja berekstensi pdf, misalkan, 200411100077_Modul5_Sorting.pdf
4. Buat video dalam bentuk live code salah satu nomor, sertakan voice over dan camera ON
5. Submit **link collaboratory** yang berisi file ipynb tersebut, dan submit **file pdf**
6. Kejujuran selalu jadi yang utama, kerjakan sendiri, tidak diperkenankan plagiarisme

1 Konsep

Kerjakan berikut dengan ketentuan, modulus satu digit terakhir NPM kalian dengan 2, hasil modulus adalah kode soal berikut ini. Tulis ringkasan atau penjelasan hal-hal berikut, dengan kata-kata kalian sendiri mengenai :

- **Kode Soal = 0**, Bubble Sort, pengertian, dan fungsi Bubble Sort
- **Kode Soal = 1**, Selection Sort, pengertian, dan fungsi Selection Sort

2 Implementasi-1

Kerjakan berikut dengan ketentuan, modulus satu digit terakhir NPM kalian dengan 4, hasil modulus adalah kode soal berikut ini.

2.1 *Kode Soal=0*

Buat fungsi **Modifikasi Algoritma Sorting - Bubble Sort**, untuk pengurutan secara **ascending**, dengan ketentuan :

- Proses pengurutan (perbandingan) dimulai dari indeks terakhir sampai dengan indeks awal
- Data yang memiliki nilai kecil, diletakkan terlebih dahulu di indek-indeks awal, sehingga pada akhir iterasi indeks-indeks awal ditempati data yang memiliki nilai kecil.
- Contoh Proses perbandingan yang dilakukan pada modifikasi algoritma bubble sort ini dan output yang harus dihasilkan pada fungsi yang akan dibuat, dapat dilihat pada Gambar 1

```

a=[10,8,6,5,1]
bubbleSort(a)

Data yang akan diurutkan : [10, 8, 6, 5, 1]
Outer Loop #1
is data[4] < data[3] ? True , Data : [10, 8, 6, 1, 5]
is data[3] < data[2] ? True , Data : [10, 8, 1, 6, 5]
is data[2] < data[1] ? True , Data : [10, 1, 8, 6, 5]
is data[1] < data[0] ? True , Data : [1, 10, 8, 6, 5]
Outer Loop #2
is data[4] < data[3] ? True , Data : [1, 10, 8, 5, 6]
is data[3] < data[2] ? True , Data : [1, 10, 5, 8, 6]
is data[2] < data[1] ? True , Data : [1, 5, 10, 8, 6]
Outer Loop #3
is data[4] < data[3] ? True , Data : [1, 5, 10, 6, 8]
is data[3] < data[2] ? True , Data : [1, 5, 6, 10, 8]
Outer Loop #4
is data[4] < data[3] ? True , Data : [1, 5, 6, 8, 10]
Sorted Data [1, 5, 6, 8, 10]

```

(a)

```

a=[10,9,1,3,8,7,11]
bubbleSort(a)

Data yang akan diurutkan : [10, 9, 1, 3, 8, 7, 11]
Outer Loop #1
is data[6] < data[5] ? False , Data : [10, 9, 1, 3, 8, 7, 11]
is data[5] < data[4] ? True , Data : [10, 9, 1, 3, 7, 8, 11]
is data[4] < data[3] ? False , Data : [10, 9, 1, 3, 7, 8, 11]
is data[3] < data[2] ? False , Data : [10, 9, 1, 3, 7, 8, 11]
is data[2] < data[1] ? True , Data : [10, 1, 9, 3, 7, 8, 11]
is data[1] < data[0] ? True , Data : [1, 10, 9, 3, 7, 8, 11]
Outer Loop #2
is data[6] < data[5] ? False , Data : [1, 10, 9, 3, 7, 8, 11]
is data[5] < data[4] ? False , Data : [1, 10, 9, 3, 7, 8, 11]
is data[4] < data[3] ? False , Data : [1, 10, 9, 3, 7, 8, 11]
is data[3] < data[2] ? True , Data : [1, 10, 3, 9, 7, 8, 11]
is data[2] < data[1] ? True , Data : [1, 3, 10, 9, 7, 8, 11]
Outer Loop #3
is data[6] < data[5] ? False , Data : [1, 3, 10, 9, 7, 8, 11]
is data[5] < data[4] ? False , Data : [1, 3, 10, 9, 7, 8, 11]
is data[4] < data[3] ? True , Data : [1, 3, 10, 7, 9, 8, 11]
is data[3] < data[2] ? True , Data : [1, 3, 7, 10, 9, 8, 11]
Outer Loop #4
is data[6] < data[5] ? False , Data : [1, 3, 7, 10, 9, 8, 11]
is data[5] < data[4] ? True , Data : [1, 3, 7, 10, 8, 9, 11]
is data[4] < data[3] ? True , Data : [1, 3, 7, 8, 10, 9, 11]
Outer Loop #5
is data[6] < data[5] ? False , Data : [1, 3, 7, 8, 10, 9, 11]
is data[5] < data[4] ? True , Data : [1, 3, 7, 8, 9, 10, 11]
Outer Loop #6
is data[6] < data[5] ? False , Data : [1, 3, 7, 8, 9, 10, 11]
Sorted Data [1, 3, 7, 8, 9, 10, 11]

```

(b)

Gambar 1: Bubble Sort - Ascending

2.2 Kode Soal=1

Buat fungsi **Modifikasi Algoritma Sorting - Bubble Sort**, untuk pengurutan secara **descending**, dengan ketentuan :

- Proses pengurutan (perbandingan) dimulai dari indeks terakhir sampai dengan indeks awal
- Data yang memiliki nilai besar, diletakkan terlebih dahulu di indek-indeks awal, sehingga pada akhir iterasi indeks-indeks awal ditempati data yang memiliki nilai besar.
- Contoh Proses perbandingan yang dilakukan pada modifikasi algoritma bubble sort ini dan output yang harus dihasilkan pada fungsi yang akan dibuat, dapat dilihat pada Gambar 2

2.3 Kode Soal=2

Buat fungsi **Modifikasi Algoritma Sorting - Selection Sort**, untuk pengurutan secara **ascending**, dengan ketentuan :

- Seleksi/cari indeks data yang memiliki nilai maksimal dari suatu list (pencarian dilakukan mulai dari indeks-indeks akhir)
- Data yang memiliki nilai besar, diletakkan terlebih dahulu di indek-indeks akhir, sehingga pada akhir iterasi indeks-indeks akhir ditempati data yang memiliki nilai besar.
- Contoh Proses perbandingan yang dilakukan pada modifikasi algoritma selection sort ini dan output yang harus dihasilkan pada fungsi yang akan dibuat, dapat dilihat pada Gambar 3

2.4 Kode Soal=3

Buat fungsi **Modifikasi Algoritma Sorting - Selection Sort**, untuk pengurutan secara **descending**, dengan ketentuan :

```
a=[10,9,1,3,8,7,11]
bubbleSortDescending(a)

Data yang akan diurutkan : [10, 9, 1, 3, 8, 7, 11]
Outer Loop #1
is data[6] > data[5] ? True , Data : [10, 9, 1, 3, 8, 11, 7]
is data[5] > data[4] ? True , Data : [10, 9, 1, 3, 11, 8, 7]
is data[4] > data[3] ? True , Data : [10, 9, 1, 11, 3, 8, 7]
is data[3] > data[2] ? True , Data : [10, 9, 11, 1, 3, 8, 7]
is data[2] > data[1] ? True , Data : [10, 11, 9, 1, 3, 8, 7]
is data[1] > data[0] ? True , Data : [11, 10, 9, 1, 3, 8, 7]
Outer Loop #2
is data[6] > data[5] ? False , Data : [11, 10, 9, 1, 3, 8, 7]
is data[5] > data[4] ? True , Data : [11, 10, 9, 1, 8, 3, 7]
is data[4] > data[3] ? True , Data : [11, 10, 9, 8, 1, 3, 7]
is data[3] > data[2] ? False , Data : [11, 10, 9, 8, 1, 3, 7]
is data[2] > data[1] ? False , Data : [11, 10, 9, 8, 1, 3, 7]
Outer Loop #3
is data[6] > data[5] ? True , Data : [11, 10, 9, 8, 1, 7, 3]
is data[5] > data[4] ? True , Data : [11, 10, 9, 8, 7, 1, 3]
is data[4] > data[3] ? False , Data : [11, 10, 9, 8, 7, 1, 3]
is data[3] > data[2] ? False , Data : [11, 10, 9, 8, 7, 1, 3]
Outer Loop #4
is data[6] > data[5] ? True , Data : [11, 10, 9, 8, 7, 3, 1]
is data[5] > data[4] ? False , Data : [11, 10, 9, 8, 7, 3, 1]
is data[4] > data[3] ? False , Data : [11, 10, 9, 8, 7, 3, 1]
Outer Loop #5
is data[6] > data[5] ? False , Data : [11, 10, 9, 8, 7, 3, 1]
is data[5] > data[4] ? False , Data : [11, 10, 9, 8, 7, 3, 1]
Outer Loop #6
is data[6] > data[5] ? False , Data : [11, 10, 9, 8, 7, 3, 1]
Sorted Data [11, 10, 9, 8, 7, 3, 1]
```

(a)

```
a=[0,2,5,7,10,12]
bubbleSortDescending(a)

Data yang akan diurutkan : [0, 2, 5, 7, 10, 12]
Outer Loop #1
is data[5] > data[4] ? True , Data : [0, 2, 5, 7, 12, 10]
is data[4] > data[3] ? True , Data : [0, 2, 5, 12, 7, 10]
is data[3] > data[2] ? True , Data : [0, 2, 12, 5, 7, 10]
is data[2] > data[1] ? True , Data : [0, 12, 2, 5, 7, 10]
is data[1] > data[0] ? True , Data : [12, 0, 2, 5, 7, 10]
Outer Loop #2
is data[5] > data[4] ? True , Data : [12, 0, 2, 5, 10, 7]
is data[4] > data[3] ? True , Data : [12, 0, 2, 10, 5, 7]
is data[3] > data[2] ? True , Data : [12, 0, 10, 2, 5, 7]
is data[2] > data[1] ? True , Data : [12, 10, 0, 2, 5, 7]
Outer Loop #3
is data[5] > data[4] ? True , Data : [12, 10, 0, 2, 7, 5]
is data[4] > data[3] ? True , Data : [12, 10, 0, 7, 2, 5]
is data[3] > data[2] ? True , Data : [12, 10, 7, 0, 2, 5]
Outer Loop #4
is data[5] > data[4] ? True , Data : [12, 10, 7, 0, 5, 2]
is data[4] > data[3] ? True , Data : [12, 10, 7, 5, 0, 2]
Outer Loop #5
is data[5] > data[4] ? True , Data : [12, 10, 7, 5, 2, 0]
Sorted Data [12, 10, 7, 5, 2, 0]
```

(b)

Gambar 2: Bubble Sort - Descending

```
a=[10,9,1,3,8,7,11,0]
selectionSortAscending(a)

Data = [10, 9, 1, 3, 8, 7, 11, 0]
data[6] is the max value, swap data[6] and data[7], New Data : [10, 9, 1, 3, 8, 7, 0, 11]
data[0] is the max value, swap data[0] and data[6], New Data : [0, 9, 1, 3, 8, 7, 10, 11]
data[1] is the max value, swap data[1] and data[5], New Data : [0, 7, 1, 3, 8, 9, 10, 11]
data[4] is the max value, swap data[4] and data[4], New Data : [0, 7, 1, 3, 8, 9, 10, 11]
data[1] is the max value, swap data[1] and data[3], New Data : [0, 3, 1, 7, 8, 9, 10, 11]
data[1] is the max value, swap data[1] and data[2], New Data : [0, 1, 3, 7, 8, 9, 10, 11]
data[1] is the max value, swap data[1] and data[1], New Data : [0, 1, 3, 7, 8, 9, 10, 11]
data[0] is the max value, swap data[0] and data[0], New Data : [0, 1, 3, 7, 8, 9, 10, 11]
Sorted Data= [0, 1, 3, 7, 8, 9, 10, 11]
```

(a)

```
a=[10,7,6,5,3,1,0,-5]
selectionSortAscending(a)

Data = [10, 7, 6, 5, 3, 1, 0, -5]
data[0] is the max value, swap data[0] and data[7], New Data : [-5, 7, 6, 5, 3, 1, 0, 10]
data[1] is the max value, swap data[1] and data[6], New Data : [-5, 0, 6, 5, 3, 1, 7, 10]
data[2] is the max value, swap data[2] and data[5], New Data : [-5, 0, 1, 5, 3, 6, 7, 10]
data[3] is the max value, swap data[3] and data[4], New Data : [-5, 0, 1, 3, 5, 6, 7, 10]
data[3] is the max value, swap data[3] and data[3], New Data : [-5, 0, 1, 3, 5, 6, 7, 10]
data[2] is the max value, swap data[2] and data[2], New Data : [-5, 0, 1, 3, 5, 6, 7, 10]
data[1] is the max value, swap data[1] and data[1], New Data : [-5, 0, 1, 3, 5, 6, 7, 10]
data[0] is the max value, swap data[0] and data[0], New Data : [-5, 0, 1, 3, 5, 6, 7, 10]
Sorted Data= [-5, 0, 1, 3, 5, 6, 7, 10]
```

(b)

Gambar 3: Selection Sort - Ascending

- Seleksi/cari indeks data yang memiliki nilai minimal dari suatu list (pencarian dilakukan mulai dari indeks-indeks akhir)
- Data yang memiliki nilai kecil, diletakkan terlebih dahulu di indeks-indeks akhir, sehingga pada akhir iterasi indeks-indeks akhir ditempati data yang memiliki nilai kecil.
- Contoh Proses perbandingan yang dilakukan pada modifikasi algoritma selection sort ini dan output yang harus dihasilkan pada fungsi yang akan dibuat, dapat dilihat pada Gambar 4

```

a=[10,7,6,5,3,1,0,-5,5,8]
selectionSortDescending(a)

Data = [10, 7, 6, 5, 3, 1, 0, -5, 5, 8]
data[7] is the min value, swap data[7] and data[9], New Data : [10, 7, 6, 5, 3, 1, 0, 8, 5, -5]
data[6] is the min value, swap data[6] and data[8], New Data : [10, 7, 6, 5, 3, 1, 5, 8, 0, -5]
data[5] is the min value, swap data[5] and data[7], New Data : [10, 7, 6, 5, 3, 8, 5, 1, 0, -5]
data[4] is the min value, swap data[4] and data[6], New Data : [10, 7, 6, 5, 5, 8, 3, 1, 0, -5]
data[3] is the min value, swap data[3] and data[5], New Data : [10, 7, 6, 8, 5, 5, 3, 1, 0, -5]
data[4] is the min value, swap data[4] and data[4], New Data : [10, 7, 6, 8, 5, 5, 3, 1, 0, -5]
data[2] is the min value, swap data[2] and data[3], New Data : [10, 7, 8, 6, 5, 5, 3, 1, 0, -5]
data[1] is the min value, swap data[1] and data[2], New Data : [10, 8, 7, 6, 5, 5, 3, 1, 0, -5]
data[1] is the min value, swap data[1] and data[1], New Data : [10, 8, 7, 6, 5, 5, 3, 1, 0, -5]
data[0] is the min value, swap data[0] and data[0], New Data : [10, 8, 7, 6, 5, 5, 3, 1, 0, -5]
Sorted Data= [10, 8, 7, 6, 5, 5, 3, 1, 0, -5]

```

(a)

```

a=[-3,0,1,4,6,8,10,12]
selectionSortDescending(a)

Data = [-3, 0, 1, 4, 6, 8, 10, 12]
data[0] is the min value, swap data[0] and data[7], New Data : [12, 0, 1, 4, 6, 8, 10, -3]
data[1] is the min value, swap data[1] and data[6], New Data : [12, 10, 1, 4, 6, 8, 0, -3]
data[2] is the min value, swap data[2] and data[5], New Data : [12, 10, 8, 4, 6, 1, 0, -3]
data[3] is the min value, swap data[3] and data[4], New Data : [12, 10, 8, 6, 4, 1, 0, -3]
data[3] is the min value, swap data[3] and data[3], New Data : [12, 10, 8, 6, 4, 1, 0, -3]
data[2] is the min value, swap data[2] and data[2], New Data : [12, 10, 8, 6, 4, 1, 0, -3]
data[1] is the min value, swap data[1] and data[1], New Data : [12, 10, 8, 6, 4, 1, 0, -3]
data[0] is the min value, swap data[0] and data[0], New Data : [12, 10, 8, 6, 4, 1, 0, -3]
Sorted Data= [12, 10, 8, 6, 4, 1, 0, -3]

```

(b)

Gambar 4: Selection Sort - Descending

3 Implementasi-2

Kerjakan untuk semua mahasiswa.

Modifikasi algoritma Bubble Sort agar iterasi berhenti ketika tidak ada lagi data yang akan ditukar posisinya (semua data sudah berada di tempat yang tepat). Contoh Bubble Sort tanpa modifikasi dapat dilihat pada 5a, sedangkan modifikasi bubble sort dapat dilihat pada 5b.

Selamat Mengerjakan, Selalu Latihan, Jujur
 harus dimulai kapanpun, Bertanya jika kurang
 mengerti, #StayAtHome,
 #LearningFromHome

Struktur Data - 2022
Indah Agustien Siradjuddin

```

b=[10,2,5,8,1,20,2,2,4]
bubbleSort(b)

Data yang akan diurutkan : [10, 2, 5, 8, 1, 20, 2, 2, 4]
Iterasi ke- 1 :
1 = [2, 10, 5, 8, 1, 20, 2, 2, 4]
2 = [2, 5, 10, 8, 1, 20, 2, 2, 4]
3 = [2, 5, 8, 10, 1, 20, 2, 2, 4]
4 = [2, 5, 8, 1, 10, 20, 2, 2, 4]
5 = [2, 5, 8, 1, 10, 20, 2, 2, 4]
6 = [2, 5, 8, 1, 10, 2, 20, 2, 4]
7 = [2, 5, 8, 1, 10, 2, 2, 20, 4]
8 = [2, 5, 8, 1, 10, 2, 2, 4, 20]
Iterasi ke- 2 :
1 = [2, 5, 8, 1, 10, 2, 2, 4, 20]
2 = [2, 5, 8, 1, 10, 2, 2, 4, 20]
3 = [2, 5, 1, 8, 10, 2, 2, 4, 20]
4 = [2, 5, 1, 8, 10, 2, 2, 4, 20]
5 = [2, 5, 1, 8, 2, 10, 2, 4, 20]
6 = [2, 5, 1, 8, 2, 2, 10, 4, 20]
7 = [2, 5, 1, 8, 2, 2, 4, 10, 20]
Iterasi ke- 3 :
1 = [2, 5, 1, 8, 2, 2, 4, 10, 20]
2 = [2, 1, 5, 8, 2, 2, 4, 10, 20]
3 = [2, 1, 5, 8, 2, 2, 4, 10, 20]
4 = [2, 1, 5, 2, 8, 2, 4, 10, 20]
5 = [2, 1, 5, 2, 2, 8, 4, 10, 20]
6 = [2, 1, 5, 2, 2, 4, 8, 10, 20]
Iterasi ke- 4 :
1 = [1, 2, 5, 2, 2, 4, 8, 10, 20]
2 = [1, 2, 5, 2, 2, 4, 8, 10, 20]
3 = [1, 2, 2, 5, 2, 4, 8, 10, 20]
4 = [1, 2, 2, 2, 5, 4, 8, 10, 20]
5 = [1, 2, 2, 2, 4, 5, 8, 10, 20]
Iterasi ke- 5 :
1 = [1, 2, 2, 2, 4, 5, 8, 10, 20]
2 = [1, 2, 2, 2, 4, 5, 8, 10, 20]
3 = [1, 2, 2, 2, 4, 5, 8, 10, 20]
4 = [1, 2, 2, 2, 4, 5, 8, 10, 20]
Iterasi ke- 6 :
1 = [1, 2, 2, 2, 4, 5, 8, 10, 20]
2 = [1, 2, 2, 2, 4, 5, 8, 10, 20]
3 = [1, 2, 2, 2, 4, 5, 8, 10, 20]
Iterasi ke- 7 :
1 = [1, 2, 2, 2, 4, 5, 8, 10, 20]
2 = [1, 2, 2, 2, 4, 5, 8, 10, 20]
Iterasi ke- 8 :
1 = [1, 2, 2, 2, 4, 5, 8, 10, 20]
Iterasi ke- 9 :
Data urut- [1, 2, 2, 2, 4, 5, 8, 10, 20]

```

(a) Bubble Sort Tanpa Modifikasi

```

b=[10,2,5,8,1,20,2,2,4]
sorting.modifiedBubbleSort(b)

```

```

iterasi ke- 1 jumlah iterasi- 8
1 = [2, 10, 5, 8, 1, 20, 2, 2, 4]
2 = [2, 5, 10, 8, 1, 20, 2, 2, 4]
3 = [2, 5, 8, 10, 1, 20, 2, 2, 4]
4 = [2, 5, 8, 1, 10, 20, 2, 2, 4]
5 = [2, 5, 8, 1, 10, 20, 2, 2, 4]
6 = [2, 5, 8, 1, 10, 2, 20, 2, 4]
7 = [2, 5, 8, 1, 10, 2, 2, 20, 4]
8 = [2, 5, 8, 1, 10, 2, 2, 4, 20]
iterasi ke- 2 jumlah iterasi- 7
1 = [2, 5, 8, 1, 10, 2, 2, 4, 20]
2 = [2, 5, 8, 1, 10, 2, 2, 4, 20]
3 = [2, 5, 1, 8, 10, 2, 2, 4, 20]
4 = [2, 5, 1, 8, 10, 2, 2, 4, 20]
5 = [2, 5, 1, 8, 2, 10, 2, 4, 20]
6 = [2, 5, 1, 8, 2, 2, 10, 4, 20]
7 = [2, 5, 1, 8, 2, 2, 4, 10, 20]
iterasi ke- 3 jumlah iterasi- 6
1 = [2, 5, 1, 8, 2, 2, 4, 10, 20]
2 = [2, 1, 5, 8, 2, 2, 4, 10, 20]
3 = [2, 1, 5, 8, 2, 2, 4, 10, 20]
4 = [2, 1, 5, 2, 8, 2, 4, 10, 20]
5 = [2, 1, 5, 2, 2, 8, 4, 10, 20]
6 = [2, 1, 5, 2, 2, 4, 8, 10, 20]
iterasi ke- 4 jumlah iterasi- 5
1 = [1, 2, 5, 2, 2, 4, 8, 10, 20]
2 = [1, 2, 5, 2, 2, 4, 8, 10, 20]
3 = [1, 2, 2, 5, 2, 4, 8, 10, 20]
4 = [1, 2, 2, 2, 5, 4, 8, 10, 20]
5 = [1, 2, 2, 2, 4, 5, 8, 10, 20]
iterasi ke- 5 jumlah iterasi- 4
1 = [1, 2, 2, 2, 4, 5, 8, 10, 20]
2 = [1, 2, 2, 2, 4, 5, 8, 10, 20]
3 = [1, 2, 2, 2, 4, 5, 8, 10, 20]
4 = [1, 2, 2, 2, 4, 5, 8, 10, 20]

```

(b) Bubble Sort Modifikasi

Gambar 5: Bubble Sort Ascending