

Struktur Data

Fitri Nuraeni, M.Kom

<http://latihan.nufi3.com>

2020

Pengurutan Data Array

Pertemuan ke-6

Apa itu “*Sorting*”?

- Pengurutan (sorting) mengacu pada **operasi atau teknik mengatur dan mengatur ulang set data dalam beberapa urutan tertentu.**
- Kumpulan data yang disebut *list* di mana setiap record data memiliki satu atau lebih *field*. *Field* yang berisi nilai unik untuk setiap record data disebut sebagai *key* atau *field* kunci.
- Kumpulan data tersebut diurutkan baik secara numerik atau alfanumerik. Catatan kemudian diatur dalam **urutan menaik** atau **menurun** tergantung pada nilai numerik kunci.
- Algoritma Sorting digunakan untuk mengatur ulang array atau elemen *list* sesuai dengan **operator perbandingan** pada elemen. **Operator perbandingan** digunakan untuk menentukan urutan elemen baru dalam struktur data masing-masing.

Kategori *Sorting*

- ***Internal Sorting*** : Jika semua data yang akan disortir dapat disesuaikan pada suatu waktu di **memori utama**, metode penyortiran internal sedang dilakukan.
- ***External Sorting*** : Ketika data yang akan disortir tidak dapat ditampung dalam memori pada saat yang sama dan beberapa harus disimpan dalam **memori tambahan** seperti hard disk, floppy disk, pita magnetik dll, maka metode penyortiran eksternal dilakukan.

Kompleksitas Algoritma Pengurutan

- Kompleksitas algoritma pengurutan menghitung **waktu berjalan** suatu fungsi dimana '**n**' **jumlah item** harus diurutkan. Pilihan metode penyortiran yang cocok untuk suatu masalah tergantung pada beberapa konfigurasi dependensi untuk masalah yang berbeda.
- Pertimbangan yang paling penting dari ini adalah:
 - **Lamanya waktu yang dihabiskan oleh programmer** dalam pemrograman suatu program penyortiran tertentu
 - **Jumlah waktu mesin** yang diperlukan untuk menjalankan program
 - **Jumlah memori** yang diperlukan untuk menjalankan program

Jenis-jenis Teknik Pengurutan

- *Bubble Sort*
- *Selection Sort*
- *Insertion Sort*
- *Merge Sort*
- *Quick Sort*
- *Heap Sort*

Bubble Sort

Pengertian

- **Bubble Sort** adalah algoritma pengurutan paling sederhana yang bekerja dengan berulang kali menukar elemen yang berdekatan jika urutannya tidak sesuai.

Contoh Teknik Buble Sort

Array A	Indeks	0	1	2	3	4
	elemen	54	16	42	4	21

Array A akan diurutkan secara ascending

Array A memiliki **5 elemen**, maka proses menemukan posisi yang tepat akan diulang sebanyak **(5-1) kali**

Pada setiap proses menemukan posisi yang tepat, akan dilakukan **pembandingan tiap elemen yang belum menemukan posisinya**

Proses 1 : Menemukan Elemen untuk Posisi 0

1.a. Bandingkan elemen ke-4 dengan elemen ke-3

Apakah elemen ke-4 **lebih kecil** dari elemen ke-3?

Tidak

Array A	Indeks	0	1	2	3	4
	elemen	54	16	42	4	21

1.b. Bandingkan elemen ke-3 dengan elemen ke-2

Apakah elemen ke-3 **lebih kecil** dari elemen ke-2?

YA

Array A	Indeks	0	1	2	3	4
	elemen	54	16	42	4	21

Tukar Posisi

Array A	Indeks	0	1	2	3	4
	elemen	54	16	4	42	21

Proses 1 : Menemukan Elemen untuk Posisi 0

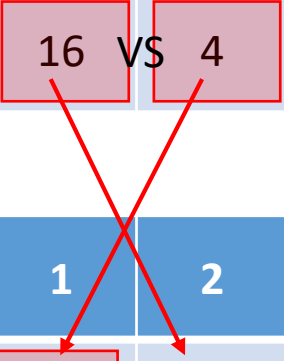
1.c. Bandingkan elemen ke-2 dengan elemen ke-1

Apakah elemen ke-2 **lebih kecil** dari elemen ke-1?

YA

Array A	Indeks	0	1	2	3	4
	elemen	54	16	4	42	21

Tukar Posisi



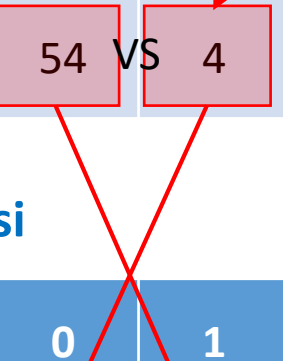
1.b. Bandingkan elemen ke-1 dengan elemen ke-0

Apakah elemen ke-1 **lebih kecil** dari elemen ke-0?

YA

Array A	Indeks	0	1	2	3	4
	elemen	54	4	16	42	21

Tukar Posisi



Array A	Indeks	0	1	2	3	4
	elemen	4	54	16	42	21


Proses 2 : Menemukan Elemen untuk Posisi 1

2.a. Bandingkan elemen ke-4 dengan elemen ke-3

Apakah elemen ke-4 **lebih kecil** dari elemen ke-3?

YA

Array A	Indeks	0	1	2	3	4
	elemen	4	54	16	42	21



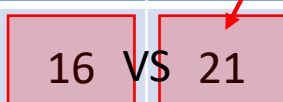
Tukar Posisi

2.b. Bandingkan elemen ke-3 dengan elemen ke-2

Apakah elemen ke-3 **lebih kecil** dari elemen ke-2?

Tidak

Array A	Indeks	0	1	2	3	4
	elemen	4	54	16	21	42



Array A	Indeks	0	1	2	3	4
	elemen	4	54	16	21	42

Proses 2 : Menemukan Elemen untuk Posisi 1

2.c. Bandingkan elemen ke-2 dengan elemen ke-1
Apakah elemen ke-2 **lebih kecil** dari elemen ke-1?

YA

Array A	Indeks	0	1	2	3	4
	elemen	4	54	16	21	42

Tukar Posisi

Array A	Indeks	0	1	2	3	4
	elemen	4	16	54	21	42

Proses 3 : Menemukan Elemen untuk Posisi 2

3.a. Bandingkan elemen ke-4 dengan elemen ke-3

Apakah elemen ke-4 **lebih kecil** dari elemen ke-3?

Tidak

Array A	Indeks	0	1	2	3	4
	elemen	4	16	54	21	42

1.b. Bandingkan elemen ke-3 dengan elemen ke-2

Apakah elemen ke-3 **lebih kecil** dari elemen ke-2?

YA

Array A	Indeks	0	1	2	3	4
	elemen	4	16	54	21	42

Tukar Posisi

Array A	Indeks	0	1	2	3	4
	elemen	4	16	21	54	42

Proses 4 : Menemukan Elemen untuk Posisi 3

4.a. Bandingkan elemen ke-4 dengan elemen ke-3
Apakah elemen ke-4 **lebih kecil** dari elemen ke-3?

YA

Array A	Indeks	0	1	2	3	4
	elemen	4	16	21	54	42

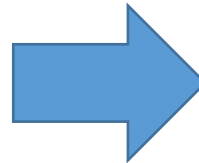
Tukar Posisi

Array A	Indeks	0	1	2	3	4
	elemen	4	16	21	42	54

Akhir Pengurutan

Array A	Indeks	0	1	2	3	4
	elemen	4	16	21	42	54

Array A	Indeks	0	1	2	3	4
	elemen	54	16	42	4	21



Array A	Indeks	0	1	2	3	4
	elemen	4	16	21	42	54

Selection Sort

Teknik Pengurutan 2

Pengertian

- Algoritma *Selection Sorting* mengurutkan array dengan berulang kali menemukan **elemen minimum** (mempertimbangkan **urutan naik**) dari bagian yang tidak disortir dan meletakkannya di awal. Algoritma mempertahankan dua sub-array dalam array yang diberikan.
 - Subarray yang sudah diurutkan.
 - Sisa subarray yang tidak disortir.
- Di setiap iterasi dari jenis seleksi, elemen minimum (mempertimbangkan urutan naik) dari subarray yang tidak disortir diambil dan dipindahkan ke subarray yang diurutkan.

Contoh Teknik Selection Sort

Array A	Indeks	0	1	2	3	4
	elemen	54	16	42	4	21

Array A akan diurutkan secara **ascending**

Array A memiliki **5 elemen**, maka proses menemukan **elemen terkecil** akan diulang sebanyak **(5-1) kali**

Proses 0 : Persiapan

Array A	Indeks	0	1	2	3	4
	elemen	54	16	42	4	21

Min	posisi	nilai
	0	9999

Proses 1 : Menemukan Elemen untuk Posisi 0

Min	posisi	nilai
	0	9999

VS

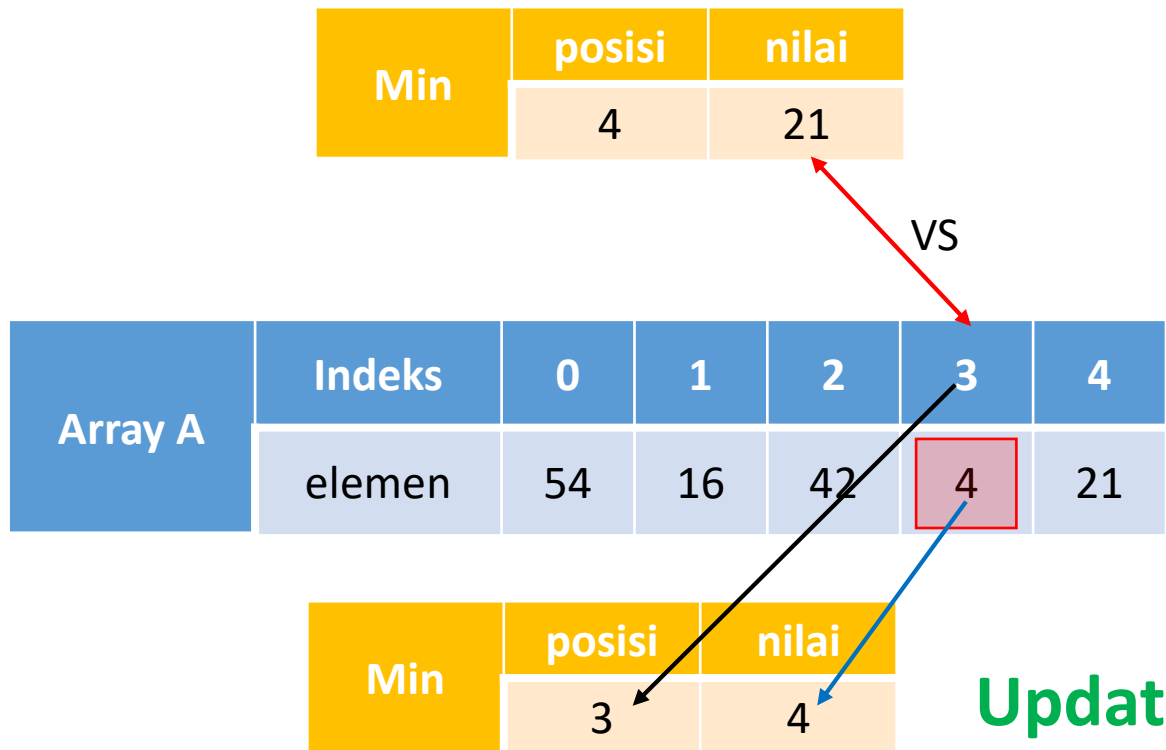
Array A	Indeks	0	1	2	3	4
	elemen	54	16	42	4	21

1.a. Apakah elemen ke-4 lebih kecil dibanding **nilai Min** ? **YA**

Min	posisi	nilai
	4	21

Update Min

Proses 1 : Menemukan Elemen untuk Posisi 0



1.b. Apakah elemen ke-**3** lebih kecil dibanding **nilai Min** ? **YA**

Update Min

Proses 1 : Menemukan Elemen untuk Posisi 0

Min	posisi	nilai
	3	4

VS

Array A	Indeks	0	1	2	3	4
	elemen	54	16	42	4	21

1.c. Apakah elemen ke-**2** lebih kecil dibanding **nilai Min** ? **Tidak**

Min	posisi	nilai
	3	4

Min Tetap

Proses 1 : Menemukan Elemen untuk Posisi 0

Min	posisi	nilai
	3	4

VS

Array A	Indeks	0	1	2	3	4
	elemen	54	16	42	4	21

1.d. Apakah elemen ke-**1** lebih kecil dibanding **nilai Min** ? **Tidak**

Min	posisi	nilai
	3	4

Min Tetap

Proses 1 : Menemukan Elemen untuk Posisi 0

Min	posisi	nilai
	3	4

VS

Array A	Indeks	0	1	2	3	4
	elemen	54	16	42	4	21

1.e. Apakah elemen ke-**0** lebih kecil dibanding **nilai Min** ? **Tidak**

Min	posisi	nilai
	3	4

Min Tetap

Proses 1 : Menemukan Elemen untuk Posisi 0

Min	posisi	nilai
	3	4

Tukar Posisi

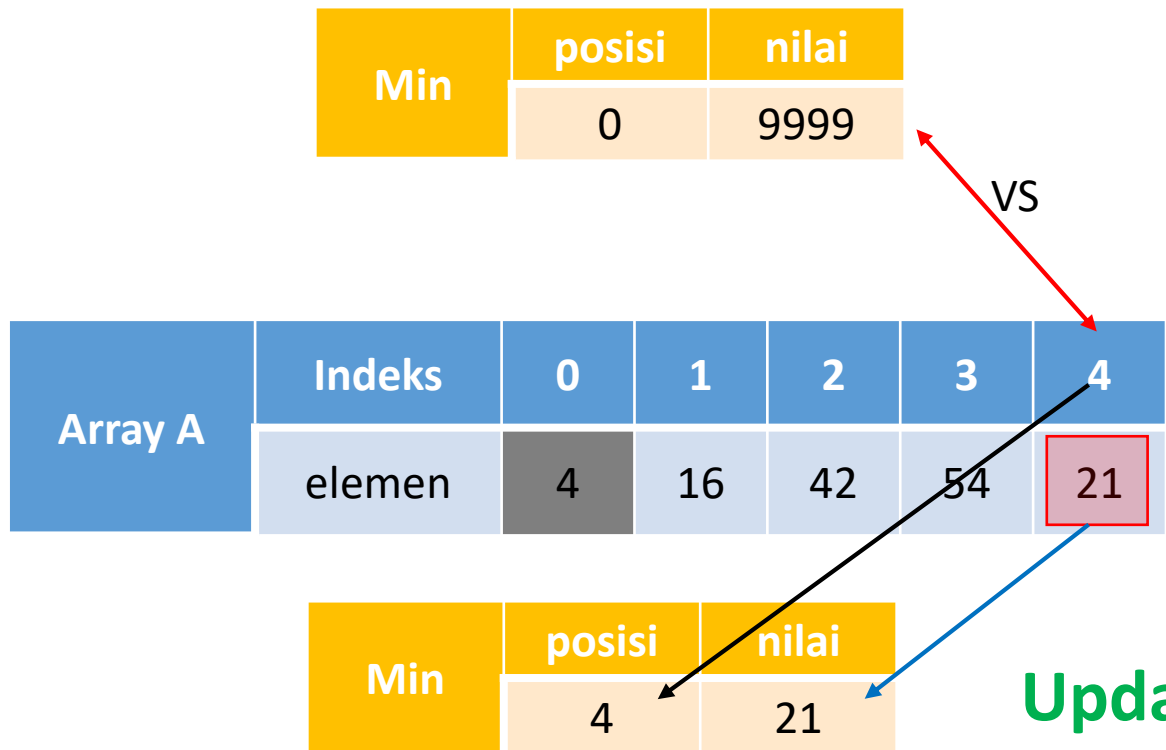
Array A	Indeks	0	1	2	3	4
	elemen	54	16	42	4	21

Array A	Indeks	0	1	2	3	4
	elemen	4	16	42	54	21

Tidak ada lagi elemen Array A yang dapat dibandingkan

1.f. Tukar **elemen posisi Min** dengan **elemen Posisi Ke-0**?

Proses 2 : Menemukan Elemen untuk Posisi 1



2.a. Apakah elemen ke-4 lebih kecil dibanding **nilai Min** ? **YA**

Update Min

Proses 2 : Menemukan Elemen untuk Posisi 2

Min	posisi	nilai
	4	21

VS

Array A	Indeks	0	1	2	3	4
	elemen	4	16	42	54	21

2.b. Apakah elemen ke-**3** lebih kecil dibanding **nilai Min** ? **Tidak**

Min	posisi	nilai
	4	21

Min Tetap

Proses 2 : Menemukan Elemen untuk Posisi 2

Min	posisi	nilai
	4	21

VS

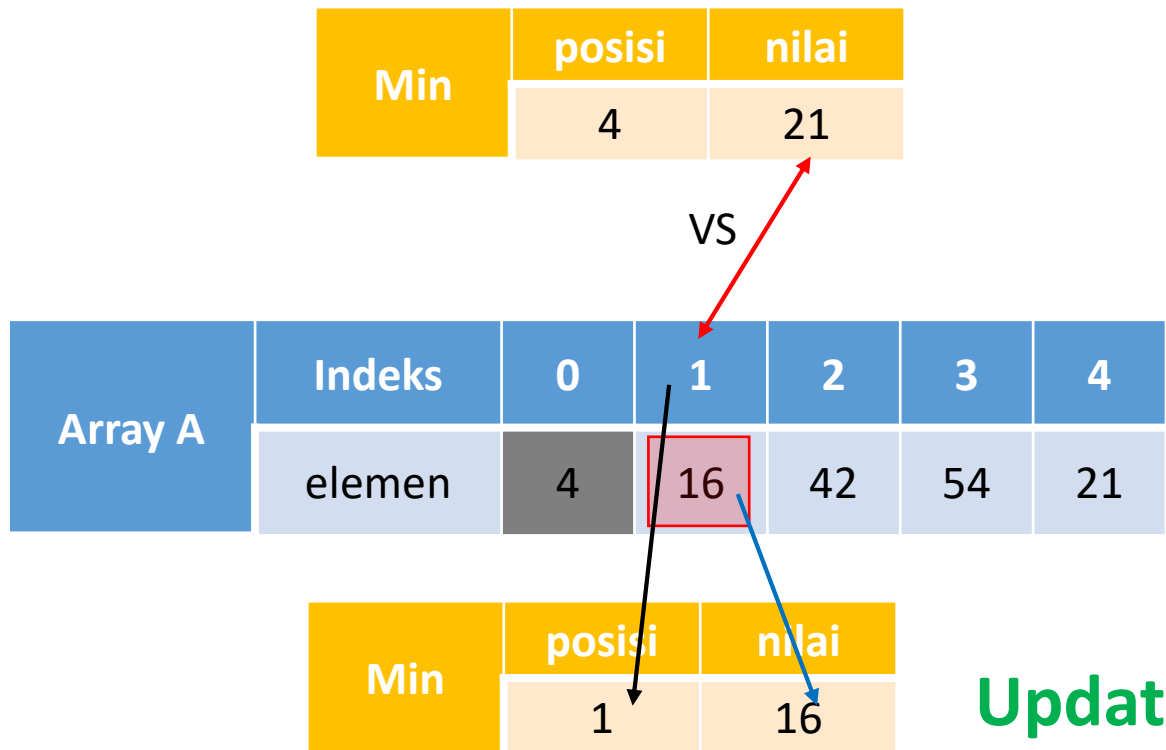
Array A	Indeks	0	1	2	3	4
	elemen	4	16	42	54	21

2.c. Apakah elemen ke-**2** lebih kecil dibanding **nilai Min** ? **Tidak**

Min	posisi	nilai
	4	21

Min Tetap

Proses 2 : Menemukan Elemen untuk Posisi 1



2.d. Apakah elemen ke-**1** lebih kecil dibanding **nilai Min** ? **YA**

Update Min

Proses 2 : Menemukan Elemen untuk Posisi 1

Min	posisi	nilai
	1	16

Tukar Posisi

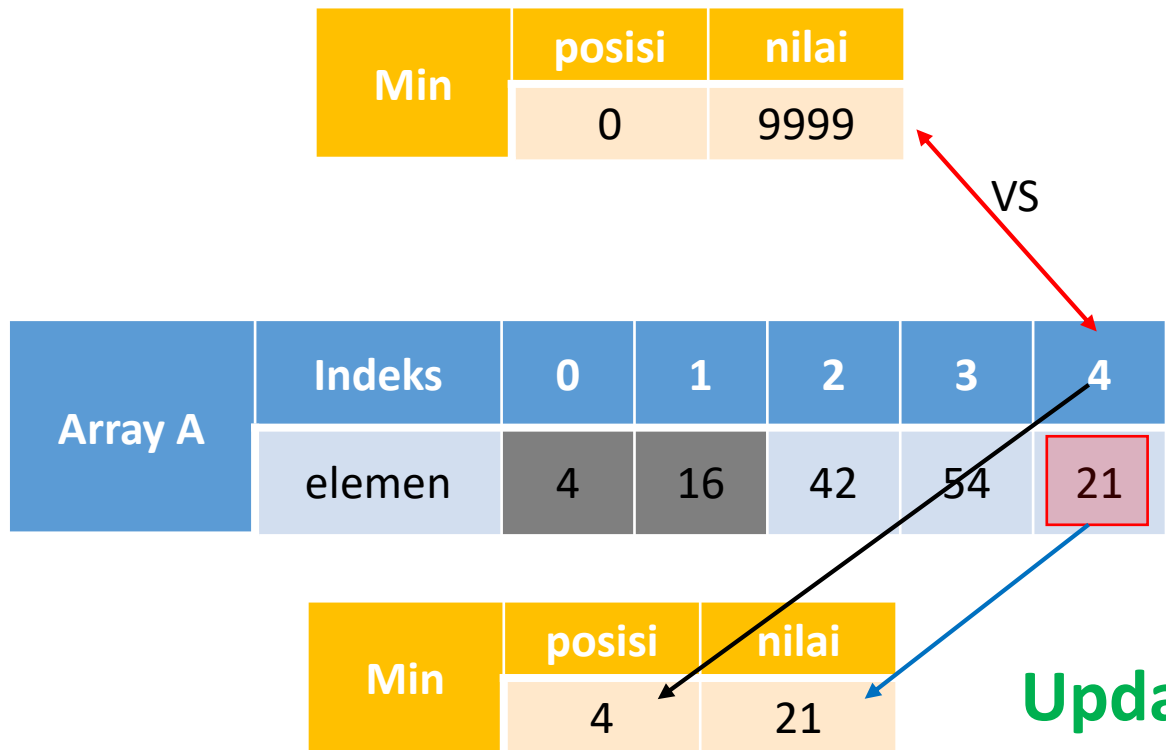
Array A	Indeks	0	1	2	3	4
	elemen	4	16	42	54	21

Array A	Indeks	0	1	2	3	4
	elemen	4	16	42	54	21

Tidak ada lagi elemen Array A yang dapat dibandingkan

2.e. Tukar **elemen posisi Min** dengan **elemen Posisi Ke-1**?

Proses 3 : Menemukan Elemen untuk Posisi 2



3.a. Apakah elemen ke-4 lebih kecil dibanding **nilai Min** ? **YA**

Update Min

Proses 3 : Menemukan Elemen untuk Posisi 2

Min	posisi	nilai
	4	21

VS

Array A	Indeks	0	1	2	3	4
	elemen	4	16	42	54	21

3.b. Apakah elemen ke-**3** lebih kecil dibanding **nilai Min** ? **Tidak**

Min	posisi	nilai
	4	21

Min Tetap

Proses 3 : Menemukan Elemen untuk Posisi 2

Min	posisi	nilai
	4	21

VS

Array A	Indeks	0	1	2	3	4
	elemen	4	16	42	54	21

3.c. Apakah elemen ke-**2** lebih kecil dibanding **nilai Min** ? **Tidak**

Min	posisi	nilai
	4	21

Min Tetap

Proses 3 : Menemukan Elemen untuk Posisi 2

Min	posisi	nilai
	4	21

Tukar Posisi

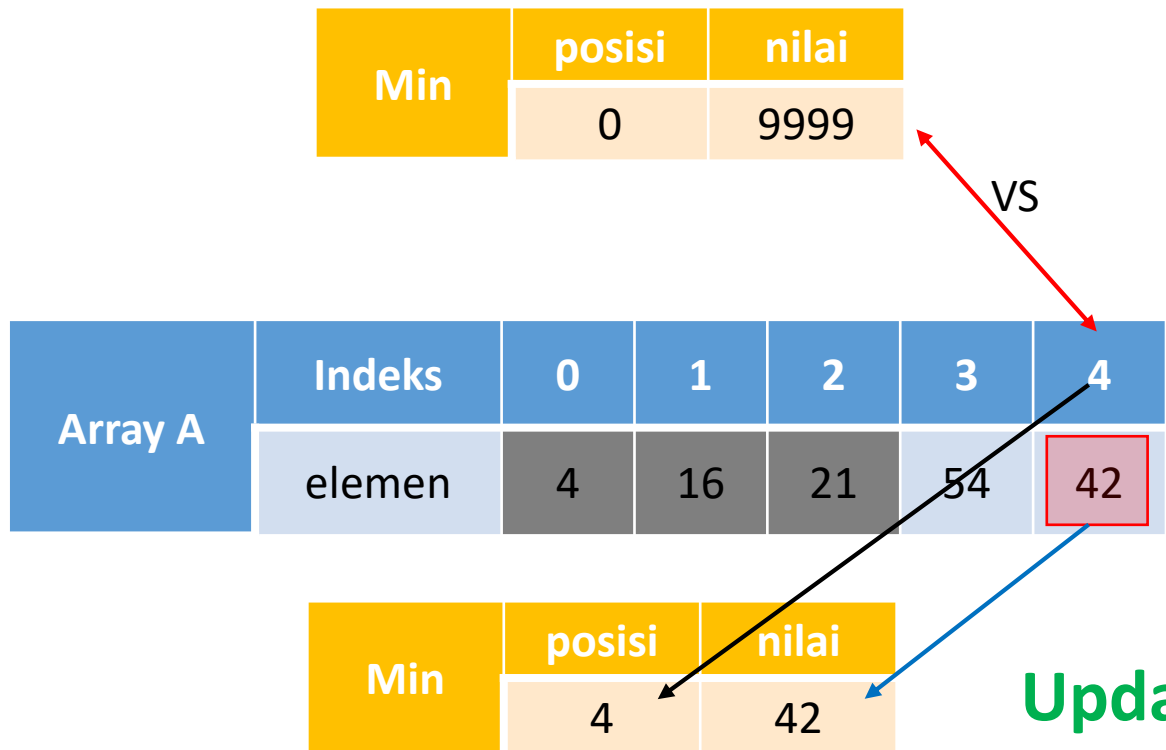
Array A	Indeks	0	1	2	3	4
	elemen	4	16	42	54	21

Array A	Indeks	0	1	2	3	4
	elemen	4	16	21	54	42

Tidak ada lagi elemen Array A yang dapat dibandingkan

3.d. Tukar **elemen posisi Min** dengan **elemen Posisi Ke-2**?

Proses 4 : Menemukan Elemen untuk Posisi 3



4.a. Apakah elemen ke-4 lebih kecil dibanding **nilai Min** ? **YA**

Update Min

Proses 4 : Menemukan Elemen untuk Posisi 3

Min	posisi	nilai
	4	42

VS

Array A	Indeks	0	1	2	3	4
	elemen	4	16	21	54	42

3.b. Apakah elemen ke-**3** lebih kecil dibanding **nilai Min** ? **Tidak**

Min	posisi	nilai
	4	42

Min Tetap

Proses 4 : Menemukan Elemen untuk Posisi 3

Min	posisi	nilai
	4	42

Tukar Posisi

Array A	Indeks	0	1	2	3	4
	elemen	4	16	21	54	42

Array A	Indeks	0	1	2	3	4
	elemen	4	16	21	42	54

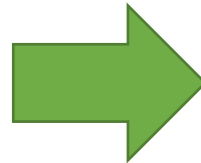
Tidak ada lagi elemen Array A yang dapat dibandingkan

3.d. Tukar **elemen posisi Min** dengan **elemen Posisi Ke-3**?

Akhir Pengurutan

Array A	Indeks	0	1	2	3	4
	elemen	4	16	21	42	54

Array A	Indeks	0	1	2	3	4
	elemen	54	16	42	4	21



Array A	Indeks	0	1	2	3	4
	elemen	4	16	21	42	54

Insertion Sort

Teknik Pengurutan 3

Pengertian

- Pengurutan dengan teknik penyisipan adalah algoritma penyortiran sederhana yang berfungsi seperti cara kita menyortir kartu remi di tangan kita.

Proses 0 : Persiapan

Array A	Indeks	0	1	2	3	4
	elemen	54	16	42	4	21

Kunci	nilai
	0

Proses 1 : Menemukan Posisi untuk Kunci ke-1



1.a. Kunci = **nilai elemen ke-1**

Array A	Indeks	0	1	2	3	4
	elemen	54	16	42	4	21

1.b. Apakah **nilai elemen sebelah kiri kunci** lebih > kunci? **YA** **Pindahkan ke kanan**

Array A	Indeks	0	1	2	3	4
	elemen		54	42	4	21

1.c. Apakah masih ada **nilai elemen sebelah kiri kunci**? **Tidak**

Array A	Indeks	0	1	2	3	4
	elemen	16	54	42	4	21

Sisipkan kunci ke posisi kosong

Proses 2 : Menemukan Posisi untuk Kunci ke-2



2.a. Kunci = **nilai elemen ke-2**

Array A	Indeks	0	1	2	3	4
	elemen	16	54	42	4	21

2.b. Apakah **nilai elemen sebelah kiri kunci** lebih > kunci? **YA Pindahkan ke kanan**

Array A	Indeks	0	1	2	3	4
	elemen	16		54	4	21

2.c. Apakah masih ada **nilai elemen sebelah kiri kunci**? **Ya**

Proses 2 : Menemukan Posisi untuk Kunci ke-2

		Kunci		nilai		
				42		
Array A	Indeks	0	1	2	3	4
	elemen	16		54	4	21

VS

2.d. Apakah **nilai elemen sebelah kiri kunci** lebih > **kunci**? **Tidak**

Array A	Indeks	0	1	2	3	4
	elemen	16		54	4	21

2.e. Apakah masih ada **nilai elemen sebelah kiri kunci**? **Tidak**

Array A	Indeks	0	1	2	3	4
	elemen	16	42	54	4	21

Sisipkan kunci ke posisi kosong

Proses 3 : Menemukan Posisi untuk Kunci ke-3



3.a. Kunci = **nilai elemen ke-3**

vs

Array A	Indeks	0	1	2	3	4
	elemen	16	42	54	4	21

3.b. Apakah **nilai elemen sebelah kiri kunci** lebih > kunci? **YA Pindahkan ke kanan**

Array A	Indeks	0	1	2	3	4
	elemen	16	42		54	21

3.c. Apakah masih ada **nilai elemen sebelah kiri kunci**? **Ya**

Proses 3 : Menemukan Posisi untuk Kunci ke-2

						Kunci	nilai
							4
						VS	
Array A	Indeks	0	1	2	3	4	
	elemen	16	42		54	21	

3.d. Apakah **nilai elemen sebelah kiri kunci** lebih > kunci? **YA Pindahkan ke kanan**

Array A	Indeks	0	1	2	3	4
	elemen	16		42	54	21

3.e. Apakah masih ada **nilai elemen sebelah kiri kunci**? **Ya**

Proses 3 : Menemukan Posisi untuk Kunci ke-3

		Kunci		nilai		
				4		
Array A	Indeks	0	1	2	3	4
	elemen	16		42	54	21

VS

3.f. Apakah **nilai elemen sebelah kiri kunci** lebih > **kunci**? **YA** **Pindahkan ke kanan**

Array A	Indeks	0	1	2	3	4
	elemen		16	42	54	21

3.g. Apakah masih ada **nilai elemen sebelah kiri kunci**? **Tidak**

Array A	Indeks	0	1	2	3	4
	elemen	4	16	42	54	21

Sisipkan kunci ke posisi kosong

Proses 4 : Menemukan Posisi untuk Kunci ke-4



4.a. Kunci = **nilai elemen ke-4**

Array A	Indeks	0	1	2	3	4
	elemen	4	16	42	54	21

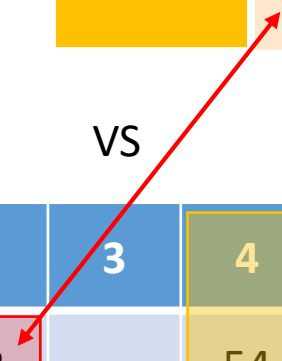
4.b. Apakah **nilai elemen sebelah kiri kunci** lebih > kunci? **YA Pindahkan ke kanan**

Array A	Indeks	0	1	2	3	4
	elemen	4	16	42		54

4.c. Apakah masih ada **nilai elemen sebelah kiri kunci**? **Ya**

Proses 4 : Menemukan Posisi untuk Kunci ke-4

		Kunci		nilai		
				21		
				VS		
Array A	Indeks	0	1	2	3	4
	elemen	4	16	42		54



4.d. Apakah **nilai elemen sebelah kiri kunci** lebih > kunci? **YA** **Pindahkan ke kanan**

Array A	Indeks	0	1	2	3	4
	elemen	4	16		42	54

4.e. Apakah masih ada **nilai elemen sebelah kiri kunci**? **Ya**

Proses 4 : Menemukan Posisi untuk Kunci ke-4

						Kunci	nilai
							21
						VS	
Array A	Indeks	0	1	2	3	4	
	elemen	4	16		42	54	

4.f. Apakah **nilai elemen sebelah kiri kunci** lebih > **kunci**? **Tidak**

Array A	Indeks	0	1	2	3	4
	elemen	4	16		42	54

4.g. Apakah masih ada **nilai elemen sebelah kiri kunci**? **Ya**

Proses 4 : Menemukan Posisi untuk Kunci ke-4

		Kunci		nilai		
				21		
Array A	Indeks	0	1	2	3	4
	elemen	4	16		42	54

VS

4.h. Apakah **nilai elemen sebelah kiri kunci** lebih > **kunci**? **Tidak**

Array A	Indeks	0	1	2	3	4
	elemen	4	16		42	54

4.i. Apakah masih ada **nilai elemen sebelah kiri kunci**? **Tidak**

Array A	Indeks	0	1	2	3	4
	elemen	4	16	21	42	54

Sisipkan kunci ke posisi kosong

Akhir Pengurutan

Array A	Indeks	0	1	2	3	4
	elemen	4	16	21	42	54

Array A	Indeks	0	1	2	3	4
	elemen	54	16	42	4	21



Array A	Indeks	0	1	2	3	4
	elemen	4	16	21	42	54

Untuk lebih memahami

- Saya sediakan website untuk simulasi pengurutan:

<http://uji-coba.nufi3.com/StrukturData>

Dashboard

Masukan Deret Angka *pisahkan dengan tanda koma (,)*

Pilih metode sorting

Buble Sorting ▼

Proses

Masukan Jumlah Deret Angka

Pilih metode sorting

Buble Sorting ▼

Buat Deret Random & Proses

Kembali

Proses Sorting

Mari Praktikum...

Kerjakan soal sesuai dengan angka terakhir NPM masing-masing

Soal 1 : 0, 1, 2

Soal 2 : 3, 4, 5

Soal 3 : 6, 7, 8, 9

Soal 1 : Lengkapi code bubble sort berikut:

```
#include <iostream>
#include <conio.h>

void main(){
    int deret[] = {15, 10, 13, 7, 22, 6, 17, 5, 7, 1};
    int n = 10;
    int i,j,temp;
    int tukar = 0;

    cout<<"Deret tak terurut = ";
    for(int id=0; id<n; id++){
        cout<< deret[id] <<" ";
    }
    cout<<endl<<endl;

    for(i=0; i < n; i++){
        cout<<" i = " << i <<endl;
        for(j=0; j < n-i-1; j++){
            cout<<"deret["<<(j)<<"] > deret["<<(j+1)<<"] ?";
            if(deret[j] > deret[j+1]){
                cout<<" Tukarkan ";
                temp = deret[j];
                deret[j] = deret[j+1];
                deret[j+1] = temp;
            }
        }
        cout<< endl;
    }
```

```
    }
}

cout<<endl<<"Deret terurut = ";
for(int id=0; id<n; id++){
    cout<< deret[id] <<" ";
}
cout<<endl;

getch();
}
```


Running Soal 1

```
i = 0
deret[0] > deret[1] ? Tukarkan
10 15 13 7 22 6 17 5 7 1
deret[1] > deret[2] ? Tukarkan
10 13 15 7 22 6 17 5 7 1
deret[2] > deret[3] ? Tukarkan
10 13 7 15 22 6 17 5 7 1
deret[3] > deret[4] ?
10 13 7 15 22 6 17 5 7 1
deret[4] > deret[5] ? Tukarkan
10 13 7 15 6 22 17 5 7 1
deret[5] > deret[6] ? Tukarkan
10 13 7 15 6 17 22 5 7 1
deret[6] > deret[7] ? Tukarkan
10 13 7 15 6 17 5 22 7 1
deret[7] > deret[8] ? Tukarkan
10 13 7 15 6 17 5 7 22 1
deret[8] > deret[9] ? Tukarkan
10 13 7 15 6 17 5 7 1 22
deret[9] > deret[10] ?
10 13 7 15 6 17 5 7 1 22
i = 1
deret[0] > deret[1] ?
10 13 7 15 6 17 5 7 1 22
deret[1] > deret[2] ? Tukarkan
10 7 13 15 6 17 5 7 1 22
deret[2] > deret[3] ?
10 7 13 15 6 17 5 7 1 22
deret[3] > deret[4] ? Tukarkan
10 7 13 6 15 17 5 7 1 22
deret[4] > deret[5] ?
10 7 13 6 15 17 5 7 1 22
deret[5] > deret[6] ? Tukarkan
10 7 13 6 15 5 17 7 1 22
deret[6] > deret[7] ? Tukarkan
10 7 13 6 15 5 7 17 1 22
deret[7] > deret[8] ? Tukarkan
10 7 13 6 15 5 7 1 17 22
deret[8] > deret[9] ?
10 7 13 6 15 5 7 1 17 22
```

```
i = 2
deret[0] > deret[1] ? Tukarkan
7 10 13 6 15 5 7 1 17 22
deret[1] > deret[2] ?
7 10 13 6 15 5 7 1 17 22
deret[2] > deret[3] ? Tukarkan
7 10 6 13 15 5 7 1 17 22
deret[3] > deret[4] ?
7 10 6 13 15 5 7 1 17 22
deret[4] > deret[5] ? Tukarkan
7 10 6 13 5 15 7 1 17 22
deret[5] > deret[6] ? Tukarkan
7 10 6 13 5 7 15 1 17 22
deret[6] > deret[7] ? Tukarkan
7 10 6 13 5 7 1 15 17 22
deret[7] > deret[8] ?
7 10 6 13 5 7 1 15 17 22
i = 3
deret[0] > deret[1] ?
7 10 6 13 5 7 1 15 17 22
deret[1] > deret[2] ? Tukarkan
7 6 10 13 5 7 1 15 17 22
deret[2] > deret[3] ?
7 6 10 13 5 7 1 15 17 22
deret[3] > deret[4] ? Tukarkan
7 6 10 5 13 7 1 15 17 22
deret[4] > deret[5] ? Tukarkan
7 6 10 5 7 13 1 15 17 22
deret[5] > deret[6] ? Tukarkan
7 6 10 5 7 1 13 15 17 22
deret[6] > deret[7] ?
7 6 10 5 7 1 13 15 17 22
```

```
i = 4
deret[0] > deret[1] ? Tukarkan
6 7 10 5 7 1 13 15 17 22
deret[1] > deret[2] ?
6 7 10 5 7 1 13 15 17 22
deret[2] > deret[3] ? Tukarkan
6 7 5 10 7 1 13 15 17 22
deret[3] > deret[4] ? Tukarkan
6 7 5 7 10 1 13 15 17 22
deret[4] > deret[5] ? Tukarkan
6 7 5 7 1 10 13 15 17 22
deret[5] > deret[6] ?
6 7 5 7 1 10 13 15 17 22
i = 5
deret[0] > deret[1] ?
6 7 5 7 1 10 13 15 17 22
deret[1] > deret[2] ? Tukarkan
6 5 7 7 1 10 13 15 17 22
deret[2] > deret[3] ?
6 5 7 7 1 10 13 15 17 22
deret[3] > deret[4] ? Tukarkan
6 5 7 1 7 10 13 15 17 22
deret[4] > deret[5] ?
6 5 7 1 7 10 13 15 17 22
i = 6
deret[0] > deret[1] ? Tukarkan
5 6 7 1 7 10 13 15 17 22
deret[1] > deret[2] ?
5 6 7 1 7 10 13 15 17 22
deret[2] > deret[3] ? Tukarkan
5 6 1 7 7 10 13 15 17 22
deret[3] > deret[4] ?
5 6 1 7 7 10 13 15 17 22
```

```
i = 7
deret[0] > deret[1] ?
5 6 1 7 7 10 13 15 17 22
deret[1] > deret[2] ? Tukarkan
5 1 6 7 7 10 13 15 17 22
deret[2] > deret[3] ?
5 1 6 7 7 10 13 15 17 22
i = 8
deret[0] > deret[1] ? Tukarkan
1 5 6 7 7 10 13 15 17 22
deret[1] > deret[2] ?
1 5 6 7 7 10 13 15 17 22
i = 9
deret[0] > deret[1] ?
1 5 6 7 7 10 13 15 17 22
Deret terurut = 1 5 6 7 7 10 13 15 17 22
```

Soal 2 : Lengkapi code selection sort berikut:

```
#include <iostream>
#include <conio.h>

void main() {
    int deret[] = {15, 10, 13, 7, 22, 6, 17, 5, 7, 1};
    int n = 10;
    int i, j, temp, idMin;
    int min = 9999;

    for(int id=0; id<n; id++){
        cout<< deret[id] <<" ";
    }
    cout<<endl;
    cout<<"Angka Terkecil = " <<(min) <<endl;
    for(i=0; i < n; i++){
        min = 9999;
        cout<<" i = " << i <<endl;
        for(j=i; j<n; j++){
            cout<<"Lebih kecil deret["<<j<<"] ("<<deret[j]<<") dibanding "<<min<<" ? ";
            if(deret[j] < min){
                cout<<" Ya ; Rubah ";
                min = deret[j];
                idMin = j;
                cout<<"Angka Terkecil = deret["<<idMin<<"] = "<<min<<endl;
            }else{
                cout<<" Tidak ; " <<endl;
            }
        }
    }
}
```

Soal 2 : Lengkapi code selection sort berikut:

```
        cout<<endl;
        cout<<"Tukar deret["<<i<<"] dg Angka Terkecil pada deret["<<idMin<<"]"<<endl;
            =
            = deret[idMin];
            = temp;

        for(int id=0; id<n; id++){
            cout<< deret[id] <<" ";
        }
        cout<<endl<<endl;
    }

    for(int id=0; id<n; id++){
        cout<< deret[id] <<" ";
    }
    cout<<endl;

    getch();
}
```

Running Soal 2

```
E:\TEACHING\Struktur Data\mingguK5\selection_sort.exe
15 10 13 7 22 6 17 5 7 1
Angka Terkecil = 9999
i = 0
Lebih kecil deret[0] (15) dibanding 9999 ? Ya ; Rubah Angka Terkecil = deret[0] = 15
Lebih kecil deret[1] (10) dibanding 15 ? Ya ; Rubah Angka Terkecil = deret[1] = 10
Lebih kecil deret[2] (13) dibanding 10 ? Tidak ;
Lebih kecil deret[3] (7) dibanding 10 ? Ya ; Rubah Angka Terkecil = deret[3] = 7
Lebih kecil deret[4] (22) dibanding 7 ? Tidak ;
Lebih kecil deret[5] (6) dibanding 7 ? Ya ; Rubah Angka Terkecil = deret[5] = 6
Lebih kecil deret[6] (17) dibanding 6 ? Tidak ;
Lebih kecil deret[7] (5) dibanding 6 ? Ya ; Rubah Angka Terkecil = deret[7] = 5
Lebih kecil deret[8] (7) dibanding 5 ? Tidak ;
Lebih kecil deret[9] (1) dibanding 5 ? Ya ; Rubah Angka Terkecil = deret[9] = 1

Tukar deret[0] dg Angka Terkecil pada deret[9]
1 10 13 7 22 6 17 5 7 15

i = 1
Lebih kecil deret[1] (10) dibanding 9999 ? Ya ; Rubah Angka Terkecil = deret[1] = 10
Lebih kecil deret[2] (13) dibanding 10 ? Tidak ;
Lebih kecil deret[3] (7) dibanding 10 ? Ya ; Rubah Angka Terkecil = deret[3] = 7
Lebih kecil deret[4] (22) dibanding 7 ? Tidak ;
Lebih kecil deret[5] (6) dibanding 7 ? Ya ; Rubah Angka Terkecil = deret[5] = 6
Lebih kecil deret[6] (17) dibanding 6 ? Tidak ;
Lebih kecil deret[7] (5) dibanding 6 ? Ya ; Rubah Angka Terkecil = deret[7] = 5
Lebih kecil deret[8] (7) dibanding 5 ? Tidak ;
Lebih kecil deret[9] (15) dibanding 5 ? Tidak ;

Tukar deret[1] dg Angka Terkecil pada deret[7]
1 5 13 7 22 6 17 10 7 15
```

Running Soal 2

```
i = 2
Lebih kecil deret[2] (13) dibanding 9999 ? Ya ; Rubah Angka Terkecil = deret[2] = 13
Lebih kecil deret[3] (7) dibanding 13 ? Ya ; Rubah Angka Terkecil = deret[3] = 7
Lebih kecil deret[4] (22) dibanding 7 ? Tidak ;
Lebih kecil deret[5] (6) dibanding 7 ? Ya ; Rubah Angka Terkecil = deret[5] = 6
Lebih kecil deret[6] (17) dibanding 6 ? Tidak ;
Lebih kecil deret[7] (10) dibanding 6 ? Tidak ;
Lebih kecil deret[8] (7) dibanding 6 ? Tidak ;
Lebih kecil deret[9] (15) dibanding 6 ? Tidak ;

Tukar deret[2] dg Angka Terkecil pada deret[5]
1 5 6 7 22 13 17 10 7 15

i = 3
Lebih kecil deret[3] (7) dibanding 9999 ? Ya ; Rubah Angka Terkecil = deret[3] = 7
Lebih kecil deret[4] (22) dibanding 7 ? Tidak ;
Lebih kecil deret[5] (13) dibanding 7 ? Tidak ;
Lebih kecil deret[6] (17) dibanding 7 ? Tidak ;
Lebih kecil deret[7] (10) dibanding 7 ? Tidak ;
Lebih kecil deret[8] (7) dibanding 7 ? Tidak ;
Lebih kecil deret[9] (15) dibanding 7 ? Tidak ;

Tukar deret[3] dg Angka Terkecil pada deret[3]
1 5 6 7 22 13 17 10 7 15

i = 4
Lebih kecil deret[4] (22) dibanding 9999 ? Ya ; Rubah Angka Terkecil = deret[4] = 22
Lebih kecil deret[5] (13) dibanding 22 ? Ya ; Rubah Angka Terkecil = deret[5] = 13
Lebih kecil deret[6] (17) dibanding 13 ? Tidak ;
Lebih kecil deret[7] (10) dibanding 13 ? Ya ; Rubah Angka Terkecil = deret[7] = 10
Lebih kecil deret[8] (7) dibanding 10 ? Ya ; Rubah Angka Terkecil = deret[8] = 7
Lebih kecil deret[9] (15) dibanding 7 ? Tidak ;

Tukar deret[4] dg Angka Terkecil pada deret[8]
1 5 6 7 7 13 17 10 22 15
```

Running Soal 2

```
i = 5
Lebih kecil deret[5] (13) dibanding 9999 ? Ya ; Rubah Angka Terkecil = deret[5] = 13
Lebih kecil deret[6] (17) dibanding 13 ? Tidak ;
Lebih kecil deret[7] (10) dibanding 13 ? Ya ; Rubah Angka Terkecil = deret[7] = 10
Lebih kecil deret[8] (22) dibanding 10 ? Tidak ;
Lebih kecil deret[9] (15) dibanding 10 ? Tidak ;

Tukar deret[5] dg Angka Terkecil pada deret[7]
1 5 6 7 7 10 17 13 22 15

i = 6
Lebih kecil deret[6] (17) dibanding 9999 ? Ya ; Rubah Angka Terkecil = deret[6] = 17
Lebih kecil deret[7] (13) dibanding 17 ? Ya ; Rubah Angka Terkecil = deret[7] = 13
Lebih kecil deret[8] (22) dibanding 13 ? Tidak ;
Lebih kecil deret[9] (15) dibanding 13 ? Tidak ;

Tukar deret[6] dg Angka Terkecil pada deret[7]
1 5 6 7 7 10 13 17 22 15

i = 7
Lebih kecil deret[7] (17) dibanding 9999 ? Ya ; Rubah Angka Terkecil = deret[7] = 17
Lebih kecil deret[8] (22) dibanding 17 ? Tidak ;
Lebih kecil deret[9] (15) dibanding 17 ? Ya ; Rubah Angka Terkecil = deret[9] = 15

Tukar deret[7] dg Angka Terkecil pada deret[9]
1 5 6 7 7 10 13 15 22 17

i = 8
Lebih kecil deret[8] (22) dibanding 9999 ? Ya ; Rubah Angka Terkecil = deret[8] = 22
Lebih kecil deret[9] (17) dibanding 22 ? Ya ; Rubah Angka Terkecil = deret[9] = 17

Tukar deret[8] dg Angka Terkecil pada deret[9]
1 5 6 7 7 10 13 15 17 22

i = 9
Lebih kecil deret[9] (22) dibanding 9999 ? Ya ; Rubah Angka Terkecil = deret[9] = 22

Tukar deret[9] dg Angka Terkecil pada deret[9]
1 5 6 7 7 10 13 15 17 22

1 5 6 7 7 10 13 15 17 22
```

Soal 3 : Lengkapi code Insertion sort berikut:

```
#include <iostream>
#include <conio.h>

void main(){
    int deret[] = {15, 10, 13, 7, 22, 6, 17, 5, 7, 1};
    int n = 10;
    int i,j;
    int kunci;

    cout<<"Deret tak terurut = ";
    for(int id=0; id<n; id++){
        cout<< deret[id] <<" ";
    }
    cout<<endl<<endl;

    for(i=1; i < n; i++){
        cout<<"i = "<< i <<endl;
        j = i;
        kunci = deret[j];
        cout<<"kunci = deret["<<j<<"] = "<<kunci<<endl;

        while(deret[j-1] > kunci){
            cout<<"deret["<<(j-1)<<"] ("<<deret[j-1]<<") > "<<kunci;
            cout<<" insert "<<deret[j-1]<<" pada posisi "<<j<<endl;
            deret[j] = deret[j-1];
            j--;

            for(int id=0; id<n; id++){
                cout<< deret[id] <<" ";
            }
        }
    }
}
```

Soal 3 : Lengkapi code Insertion sort berikut:

```
        j--;
    }

    =    ;
    cout<<"insert X ("<<kunci<<") pada posisi "<<j<<endl;

    for(int id=0; id<n; id++){
        cout<< deret[id] <<" ";
    }
    cout<<endl<<endl;
}

cout<<"Deret terurut = ";
for(int id=0; id<n; id++){
    cout<< deret[id] <<" ";
}
cout<<endl;

getch();
}
```


Running Soal 3

E:\TEACHING\Struktur Data\mingguK5\insert_sort.exe

Deret tak terurut = 15 10 13 7 22 6 17 5 7 1

```
i = 1
kunci = deret[1] = 10
deret[0] (15) > 10; insert 15 pada posisi 1
15 15 13 7 22 6 17 5 7 1
insert X (10) pada posisi 0
10 15 13 7 22 6 17 5 7 1
```

```
i = 2
kunci = deret[2] = 13
deret[1] (15) > 13; insert 15 pada posisi 2
10 15 15 7 22 6 17 5 7 1
insert X (13) pada posisi 1
10 13 15 7 22 6 17 5 7 1
```

```
i = 3
kunci = deret[3] = 7
deret[2] (15) > 7; insert 15 pada posisi 3
10 13 15 15 22 6 17 5 7 1
deret[1] (13) > 7; insert 13 pada posisi 2
10 13 13 15 22 6 17 5 7 1
deret[0] (10) > 7; insert 10 pada posisi 1
10 10 13 15 22 6 17 5 7 1
insert X (7) pada posisi 0
7 10 13 15 22 6 17 5 7 1
```

```
i = 4
kunci = deret[4] = 22
insert X (22) pada posisi 4
7 10 13 15 22 6 17 5 7 1
```

```
i = 5
kunci = deret[5] = 6
deret[4] (22) > 6; insert 22 pada posisi 5
7 10 13 15 22 22 17 5 7 1
deret[3] (15) > 6; insert 15 pada posisi 4
7 10 13 15 15 22 17 5 7 1
deret[2] (13) > 6; insert 13 pada posisi 3
7 10 13 13 15 22 17 5 7 1
deret[1] (10) > 6; insert 10 pada posisi 2
7 10 10 13 15 22 17 5 7 1
deret[0] (7) > 6; insert 7 pada posisi 1
7 7 10 13 15 22 17 5 7 1
insert X (6) pada posisi 0
6 7 10 13 15 22 17 5 7 1
```

```
i = 6
kunci = deret[6] = 17
deret[5] (22) > 17; insert 22 pada posisi 6
6 7 10 13 15 22 22 5 7 1
insert X (17) pada posisi 5
6 7 10 13 15 17 22 5 7 1
```

```
i = 7
kunci = deret[7] = 5
deret[6] (22) > 5; insert 22 pada posisi 7
6 7 10 13 15 17 22 22 7 1
deret[5] (17) > 5; insert 17 pada posisi 6
6 7 10 13 15 17 17 22 7 1
deret[4] (15) > 5; insert 15 pada posisi 5
6 7 10 13 15 15 17 22 7 1
deret[3] (13) > 5; insert 13 pada posisi 4
6 7 10 13 13 15 17 22 7 1
deret[2] (10) > 5; insert 10 pada posisi 3
6 7 10 10 13 15 17 22 7 1
deret[1] (7) > 5; insert 7 pada posisi 2
6 7 7 10 13 15 17 22 7 1
deret[0] (6) > 5; insert 6 pada posisi 1
6 6 7 10 13 15 17 22 7 1
insert X (5) pada posisi 0
5 6 7 10 13 15 17 22 7 1
```

```
i = 8
kunci = deret[8] = 7
deret[7] (22) > 7; insert 22 pada posisi 8
5 6 7 10 13 15 17 22 22 1
deret[6] (17) > 7; insert 17 pada posisi 7
5 6 7 10 13 15 17 17 22 1
deret[5] (15) > 7; insert 15 pada posisi 6
5 6 7 10 13 15 15 17 22 1
deret[4] (13) > 7; insert 13 pada posisi 5
5 6 7 10 13 13 15 17 22 1
deret[3] (10) > 7; insert 10 pada posisi 4
5 6 7 10 10 13 15 17 22 1
insert X (7) pada posisi 3
5 6 7 7 10 13 15 17 22 1
```

```
i = 9
kunci = deret[9] = 1
deret[8] (22) > 1; insert 22 pada posisi 9
5 6 7 7 10 13 15 17 22 22
deret[7] (17) > 1; insert 17 pada posisi 8
5 6 7 7 10 13 15 17 17 22
deret[6] (15) > 1; insert 15 pada posisi 7
5 6 7 7 10 13 15 15 17 22
deret[5] (13) > 1; insert 13 pada posisi 6
5 6 7 7 10 13 13 15 17 22
deret[4] (10) > 1; insert 10 pada posisi 5
5 6 7 7 10 10 13 15 17 22
deret[3] (7) > 1; insert 7 pada posisi 4
5 6 7 7 7 10 13 15 17 22
deret[2] (7) > 1; insert 7 pada posisi 3
5 6 7 7 7 10 13 15 17 22
deret[1] (6) > 1; insert 6 pada posisi 2
5 6 6 7 7 10 13 15 17 22
deret[0] (5) > 1; insert 5 pada posisi 1
5 5 6 7 7 10 13 15 17 22
insert X (1) pada posisi 0
1 5 6 7 7 10 13 15 17 22
```

Deret terurut = 1 5 6 7 7 10 13 15 17 22

Pengumpulan

- Screenshoot/ photo source code & bukti running aplikasi, simpan pada file word, cantumkan nama, npm & kelas.
- Simpan file word anda dengan format **Praktikum06_Kelas_Nama.doc/ .docx**
- **Submit** ke slot yang disediakan pada website <http://latihan.nufi3.com>
- **Tidak ada pengurangan nilai** sampai pengumpulan 8 April 2020 pukul 23.00 wib.

Sorting in Array (2)

Proyek UTS

Next