

# Minggu 12

# Pengurutan 2

Algoritma Pemrograman – CII1F4  
Fakultas Informatika  
2021



# Question!

Bagaimana membuat suatu program sederhana bisa mengurutkan data mahasiswa berikut ini? Seperti apakah algoritmanya?

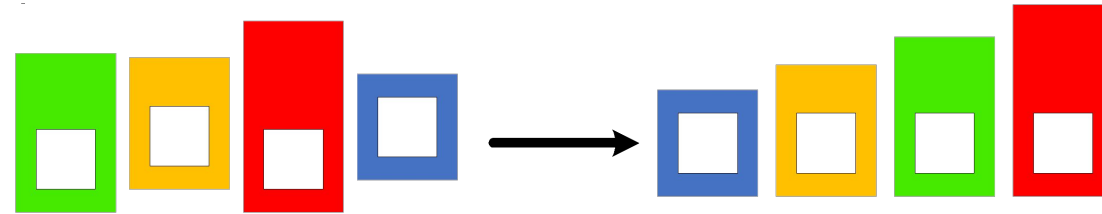
No	Student ID	Name	GPA
1	113210689	Harith	1.56
2	113212624	Johnson	3.19
3	113211834	Kimmy	1.32
4	113212925	Chou	3.68
5	113210520	Grock	1.45
6	113210223	Lunox	1.89
7	113212819	Karrie	1.05
8	113211273	Aldous	2.46
9	113211643	Franco	1.60
10	113211992	Selena	3.50



No	Student ID	Name	GPA
1	113212925	Chou	3.68
2	113211992	Selena	3.50
3	113212624	Johnson	3.19
4	113211273	Aldous	2.46
5	113210223	Lunox	1.89
6	113211643	Franco	1.60
7	113210689	Harith	1.56
8	113210520	Grock	1.45
9	113211834	Kimmy	1.32
10	113212819	Karrie	1.05



# Metode yang akan dipelajari



Secara umum ada 2 algoritma yang akan dipelajari:

1. Pengurutan dengan metode seleksi (Selection Sort)

## 2. Pengurutan dengan metode insersi

Mari kita mulai dengan contoh paling sederhana, yaitu dengan mengurutkan kumpulan bilangan.

(Insertion Sort)

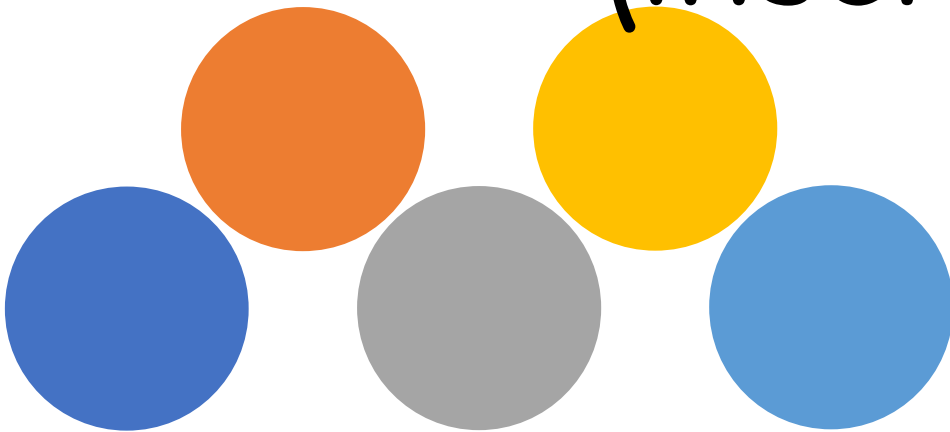
Misalnya didefinisikan tipe data **tabInt** dengan maksimum elemen **nMAX**

kamus

constant nMAX : integer = 1000

type tabInt : array [0..nMAX-1] of integer

# Metode Inseri (Insertion Sort)





# Metode Inserksi

Ide dari pengurutan secara inserksi:



Mencari tempat yang "tepat" untuk setiap nilai dengan cara sequential search



Kemudian menyisipkan nilai tersebut pada tempat yang seharusnya.



Proses dilakukan untuk semua nilai yang ada.

<https://visualgo.net/en/sorting?slide=1>

## INSERTION SORT





# Ilustrasi Insertion Sort

- Misalnya terdapat array of integer A

	0	1	2	3	4
A =	5	1	4	2	8

- Pengurutan secara membesar/ascending, artinya bilangan paling kiri adalah terkecil dan paling kanan adalah terbesar
- Attribut
  - i : indeks elemen array A
  - N : jumlah data, pada contoh N = 5
  - idx : indeks nilai ekstrim berada
  - Pass : tahapan dalam satu siklus pencarian nilai ekstrim dan tukar





# Ilustrasi Insertion Sort

A = 


0	1	2	3	4
5	2	4	10	8

 array A secara Ascending

Pass = 1 

0	1	2	3	4
2	5	4	10	8

 temp = 2





A = 

0	1	2	3	4
2	5	4	10	8

Pass = 2 

0	1	2	3	4
2	4	5	10	8

 temp = 4




A = 

0	1	2	3	4
2	4	5	10	8

Pass = 3 

0	1	2	3	4
2	4	5	10	8

 temp = 10





A = 

0	1	2	3	4
2	4	5	10	8

Pass = 4 

0	1	2	3	4
2	4	5	8	10

 temp = 8



A = 

0	1	2	3	4
2	4	5	8	10



# Diskusi A

Berdasarkan ilustrasi sebelumnya dapat disimpulkan

1. Jumlah pass =  $N - 1$
2. Setiap pass akan membagi array menjadi 2 bagian: (a) bagian terurut; (b) bagian tidak terurut.
3. Pada pass pertama, data pertama dianggap telah terurut.
4. Proses sequential search dilakukan dari kanan ke kiri, di mana selama posisi bilangan belum ditemukan, dilakukan proses pergeseran (copy).
5. Pada pengurutan ascending, pergeseran dilakukan apabila  $temp < \text{nilai elemen}$
6. Apabila  $temp \geq \text{nilai elemen}$ , maka temp disisipkan setelah nilai elemen.
7. Temp selalu disisipkan ke indeks ke-0 (pertama) apabila telah dilakukan pergeseran pada elemen pertama tersebut.
8. Variasi Algoritma
  - Arah pengurutan bisa dimulai dari indeks paling kiri ataupun kanan





# Latihan Pemahaman 1

- Buatlah matrik berukuran 5x5 seperti gambar di bawah!
- Lakukan pengurutan array B dengan secara DESCENDING.
- Tuliskan isi array setelah melewati suatu Pass.

- - - - -


## SOLUSI

- - - - -


Buatlah matrik berukuran 10x10 untuk mengurutkan 10 digit NIM (9 pass) anda (ascending/descending),



# Algoritma Insertion Sort

Algoritma lengkap dalam bentuk subprogram adalah

Proses terbagi:

- Pergeseran nilai
- Penyisipan

```
procedure InsertionSort(in/out A : tabInt, in N : integer)  
  {I.S. terdefinisi array A yang berisi N bilangan bulat  
   F.S. array A terurut secara ASCENDING dengan menggunakan algoritma  
   Insertion Sort}  
  kamus  
    pass, i, temp : integer  
  algoritma  
    pass <- 1  
    while pass <= N-1 do  
      i <- pass  
      temp <- A[pass]  
      while i > 0 and temp < A[i-1] do  
        A[i] <- A[i-1]  
        i <- i - 1  
      endwhile  
      A[i] <- temp  
      pass <- pass + 1  
    endwhile  
  endprocedure
```

# Kembali ke pertanyaan ini

Bagaimana mengurutkan data berikut dengan menggunakan algoritma insertion sort?

No	Student ID	Name	GPA
1	113210689	Harith	1.56
2	113212624	Johnson	3.19
3	113211834	Kimmy	1.32
4	113212925	Chou	3.68
5	113210520	Grock	1.45
6	113210223	Lunox	1.89
7	113212819	Karrie	1.05
8	113211273	Aldous	2.46
9	113211643	Franco	1.60
10	113211992	Selena	3.50



No	Student ID	Name	GPA
1	113212925	Chou	3.68
2	113211992	Selena	3.50
3	113212624	Johnson	3.19
4	113211273	Aldous	2.46
5	113210223	Lunox	1.89
6	113211643	Franco	1.60
7	113210689	Harith	1.56
8	113210520	Grock	1.45
9	113211834	Kimmy	1.32
10	113212819	Karrie	1.05

## kamus

**constant** nMAX : **integer** = 2022

**type** student < name, sid : **string** ; gpa : **real** >

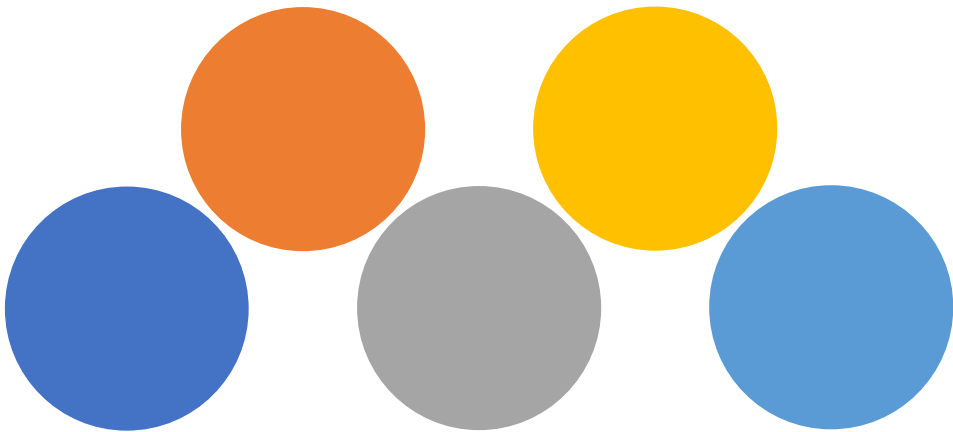
**type** tabMhs : **array** [0..nMAX-1] **of** student

# Jawaban

- Mengurutkan berdasarkan ipk secara ascending atau membesar.
- Hanya terdapat sedikit modifikasi.

```
procedure SortigGPA(in/out Mhs : tabMhs, in N : integer)  
{I.S. terdefinisi array Mhs yang berisi N data mahasiswa  
  F.S. array A terurut secara ASCENDING berdasarkan  
  IPK/GPA menggunakan Insertion Sort}  
kamus  
    pass, i: integer  
    temp : student  
algoritma  
    pass <- 1  
    while pass <= N-1 do  
        i <- pass  
        temp <- A[pass]  
        while i > 0 and temp.gpa < A[i-1].gpa do  
            A[i] <- A[i-1]  
            i <- i - 1  
        endwhile  
        A[i] <- temp  
        pass <- pass + 1  
    endwhile  
endprocedure
```

# Latihan Soal-Soal





# Soal 1 Median

Buatlah program yang menerima masukan berupa kumpulan angka, kemudian menampilkan median atau nilai tengahnya.

**Masukan** terdiri dari dua baris. Baris pertama adalah banyaknya angka, sedangkan baris kedua adalah angka-angka yang dipisahkan oleh spasi.

**Keluaran** berupa sebuah bilangan yang menyatakan median dari kumpulan angka tersebut.

**Petunjuk:** Gunakan subprogram untuk pengurutan dan pencarian posisi median

**Contoh masukan dan keluaran:**

No	Masukan	Keluaran
1	5 1 5 2 4 3	3
2	8 2 4 7 1 5 6 3 8	4.5



# Soal 2 Himpunan

Himpunan adalah kumpulan obyek yang berbeda, dimana tidak ada duplikasi obyek dan obyek tidak harus dalam urutan tertentu. Diberikan dua buah himpunan, program dibawah ini akan memeriksa apakah kedua himpunan tersebut sama atau berbeda. Lengkapilah agar program dapat bekerja dengan benar.

Contoh interaksi (underlined berarti input)

No.	Contoh	Penjelasan
1	Anggota himpunan 1: <u>8 2 4 1 5 4</u> Anggota himpunan 2: <u>8 2 4 1 5 1</u> Himpunan 1 = Himpunan 2? True	Input berakhir pada saat ada input yang duplikat, pada baris pertama angka 4 muncul dua kali, dan pada baris kedua angka 1 yang muncul dua kali. Keduanya mempunyai 5 obyek yang sama, yaitu {8, 2, 4, 1, 5}.
2	Anggota himpunan 1: <u>8 2 4 1 5 4</u> Anggota himpunan 2: <u>2 1 5 8 4 1</u> Himpunan 1 = Himpunan 2? True	Seperti contoh 1, tetapi urutan data antara kedua himpunan tidak sama, {8, 2, 4, 1, 5} sama saja dengan himpunan {2, 1, 5, 8, 4}. Karena itu output adalah kedua himpunan sama (true)
3	Anggota himpunan 1: <u>8 2 4 1 8</u> Anggota himpunan 2: <u>2 1 5 8 2</u> Himpunan 1 = Himpunan 2? False	Contoh ketiga ini masing-masing himpunan mempunyai 4 obyek saja, {8, 2, 4, 1} dan {2, 1, 5, 8}. Kedua himpunan berbeda, maka output adalah false.
4	Anggota himpunan 1: <u>8 2 4 1 8</u> Anggota himpunan 2: <u>2 1 5 2</u> Himpunan 1 = Himpunan 2? False	Contoh terakhir ini jumlah obyek dalam kedua himpunan berbeda, 4 obyek {8, 2, 4, 1} dan 3 obyek {2, 1, 5}. Karena itu tentunya output adalah false.

Buatlah program dengan melengkapi potongan algoritma berikut!

```
const nMax: integer=37
type tHimpunan: <
    anggota : array [1..nMax] of integer
    panjang : integer >

procedure bacaMasukan(in/out set: tHimpunan)
{IS. data himpunan telah siap pada piranti masukan
FS. array anggota dari set berisi sejumlah bilangan}

function ada(set: tHimpunan, x: integer) □ boolean
{mengembalikan true apabila x ada di dalam array anggota dari set, atau false apabila
sebaliknya}

procedure urut(in/out set: tHimpunan)
{IS. terdefinisi himpunan set
FS. array anggota dari set terurut menggunakan algoritma Insertion sort
(ascending/descending)}

function sama(set1, set2: tHimpunan) □ Boolean
{mengembalikan true apabila set1 dan set2 adalah sama, atau false apabila sebaliknya}

Program Himpunan
```





# Soal 3 Bunga

Buatlah sebuah tipe array of string dengan jumlah elemen maksimum sebesar 1000. Array ini akan digunakan untuk mengolah nama-nama bunga.

**Masukan** terdiri dari beberapa baris: Baris pertama adalah bilangan bulat  $N$  ( $1 \leq N \leq 1000$ ), yang menyatakan jumlah data yang akan diisi.  $N$  baris selanjutnya adalah nama bunga yang akan disimpan kedalam array. (gunakan underscore sebagai pengganti spasi, dan nama bunga selalu diakhiri oleh titik).

**Keluaran** terdiri dari  $N$  baris yang menampilkan nama bunga terurut secara ascending (membesar) berdasarkan panjang dari karakter dari nama bunga tersebut.

Lengkapi program utama dan subprogram berikut ini sesuai dengan contoh masukan dan keluaran yang diberikan!

No	Masukan	Keluaran
1	7 Mawar. Lili. Kertas. Forget_me_not. Kamboja. Anggrek. Rafflesia.	Lili. Mawar. Kertas. Kamboja. Anggrek. Rafflesia. Forget_me_not.



Buatlah program dengan melengkapi potongan algoritma berikut!

```
function panjang(bunga : string) → integer
{mengembalikan jumlah karakter pada string bunga, titik dan underscore tidak
dihitung}

procedure mengurutkan(in/out tabBunga : arrayBunga, in N : integer)
{IS. tabBunga berisi N nama bunga, asumsi belum terurut
  FS. tabBunga terurut berdasarkan jumlah karakter pada masing-masing nama bunga
  secara ascending (membesar) menggunakan INSERTION SORT}

procedure isiArray(out tabBunga : arrayBunga, out N : integer)
{IS. -
  FS. array tabBunga berisi sejumlah N nama bunga yang berasal dari input user}

procedure tampilArray(in tabBunga : arrayBunga, in N: integer)
{IS. tabBunga berisi sejumlah N nama bunga
  FS. menampilkan nama bunga yang terdapat pada array tabBunga}
```



# Soal 4 Olimpiade

Sebuah program digunakan untuk menampilkan pemringkatan tim-tim yang ikut serta dalam suatu Olimpiade.

**Masukan** terdiri dari beberapa baris. Baris pertama adalah bilangan bulat  $n$  ( $1 \leq n \leq 100$ ) yang menyatakan jumlah tim yang berpartisipasi, sedangkan setiap baris pada  $n$  baris berikutnya adalah string  $t$ , dan bilangan bulat  $g$ ,  $s$ , dan  $b$ . Di mana  $t$  adalah nama tim, dan  $g$ ,  $s$  dan  $b$  adalah perolehan medali gold, silver, and bronze (gunakan tipe bentukan).

**Keluaran** adalah terdiri dari  $n$  baris yang masing-masing barisnya adalah nama tim dan perolehan medalnya (gold, silver, dan bronze) yang terurut berdasarkan perolehan poin para tim tersebut.

Rumus perhitungan poin adalah  $poin = (4 \times g) + (3 \times s) + b$

No	Masukan	Keluaran
1	5 mpvh 8 4 8 ptpc 2 7 10 omen 8 9 5 rmpw 2 8 7 dnba 7 8 1	omen 8 9 5 dnba 7 8 1 mpvh 8 4 8 rmpw 2 8 7 ptpc 2 7 10



- Buatlah program dengan melengkapi potongan algoritma berikut!

```
program Perlombaan
kamus
    type peserta ...
    type olimpiade ...
    tab : olimpiade
    n : integer
algoritma
    isiArray(tab,n)
    sorting(tab,n)
    tampilArray(tab,n)
endprogram

procedure isiArray(in/out t :olimpiade, in/out  :integer)
{ IS. n data tim dan perolehan medali telah siap pada piranti masukan
  FS. menerima input n dan array t berisi n data tim yang mengikuti olimpiade}

procedure tampilArray(in t :olimpiade, in n :integer)
{ IS. terdefinisi sebuah array t yang berisi n data tim dan perolehan medalnya
  FS. menampilkan array t ke layar monitor}

function poin(g,s,b : integer) -> integer
{mengembalikan nilai poin berdasarkan nilai g, s dan b yang diberikan}

procedure sorting(in/out t :olimpiade, in n :integer)
{ IS. terdefinisi sebuah array t yang berisi n data tim dan perolehan medalnya
  FS. array t terurut descending berdasarkan hasil perhitungan perolehan poinnya, gunakan insertion sort}
```



TERIMA KASIH