

IF1210 Algoritma dan  
Pemrograman 1

# Skema Standar (Bag. 1): Skema Validasi

Tim Pengajar IF1210

Sekolah Teknik Elektro dan Informatika



# PENDAHULUAN

# Apa itu skema standar?

- Skema standar adalah skema umum yang sering digunakan untuk memecahkan kasus-kasus trivial (contoh: pemasukan nilai, penampilan nilai, pencarian nilai, pengurutan, dll)
- Skema standar yang dipelajari di IF1210:
  - Skema proses validasi (I dan II)
  - Skema pengulangan
  - Skema pemrosesan sekuensial: skema tanpa mark, skema dengan mark
  - Skema pemrosesan array: skema traversal, skema pencarian nilai ekstrim, skema searching, skema sorting
  - Skema pemrosesan file I/O

# Apa pentingnya skema standar?

- Penggunaan pola yang seragam antar programmer
- Mempermudah pembacaan dan pemeriksaan kode sumber oleh orang lain
- Mempercepat penulisan program untuk kasus-kasus yang umum (pengisian, penulisan, validasi, pencarian nilai ekstrim, dll)

# SKEMA VALIDASI I

# Skema Proses Validasi

## **SKEMA PROSES\_VALIDASI**

```
{ Skema program yang menerima input data,  
hanya melakukan proses jika data valid }
```

## **KAMUS**

```
{ deklarasi data input }
```

## **ALGORITMA**

```
input (<data>)  
if (<data_valid>) then  
    { <data_valid> adalah predikat, suatu ekspresi  
        boolean yang menyatakan validitas dari data }  
    { Lakukan_proses }  
else { data tidak valid }  
output (<pesan_kesalahan>)
```

# Contoh Persoalan: Rangking

- Buatlah program yang membaca 3 buah integer a, b, dan c dan menuliskan secara terurut mulai dari terkecil s.d. yang terbesar.
- Ketiga bilangan yang dibaca harus berlainan harganya. Jika tidak, maka tuliskan pesan kesalahan “Data salah, tidak sesuai spesifikasi”.

# Contoh

**Program RANGKING1**

```
{ Dibaca 3 integer a, b, c }  
{ Dituliskan dari terkecil s.d. terbesar }
```

**KAMUS**

a, b, c : integer

**ALGORITMA**

```
input(a,b,c)  
if (a ≠ b and a ≠ c and b ≠ c) then  
    depend on (a,b,c)  
        a < b < c : output(a,b,c)  
        a < c < b : output(a,c,b)  
        b < a < c : output(b,a,c)  
        b < c < a : output(b,c,a)  
        c < a < b : output(c,a,b)  
        c < b < a : output(c,b,a)  
else { ada yang sama: a=b or a=c or b=c }  
    output("Data salah, tidak sesuai spesifikasi")
```

# Latihan Soal

Semua soal dikerjakan dalam notasi algoritmik

# Latihan 1

- Buatlah program dalam notasi algoritmik yang menerima 3 buah bilangan integer yaitu h, m, dan s yang akan digunakan untuk membentuk data bertipe jam. Definisi type jam adalah sbb.

```
type jam : < HH : integer[0..23]; { bagian jam }
           MM : integer[0..59]; { bagian menit }
           SS : integer[0..59] > { bagian detik }
```

- Jika ketiga input valid, maka sebuah variabel J bertipe jam akan terbentuk (didefinisikan nilainya) dengan J.HH bernilai h, J.MM bernilai m, J.SS bernilai s.
- Nilai valid didefinisikan sebagai:  $0 \leq h \leq 23$ ;  $0 \leq m \leq 59$ ;  $0 \leq s \leq 59$
- Jika tidak valid, dituliskan pesan kesalahan ke layar “Tidak dapat membentuk jam”

# Latihan 2

- Buatlah program dalam notasi algoritmik yang menerima tiga buah nilai resistor  $R_1$ ,  $R_2$ , dan  $R_3$ , berupa bilangan integer  $\geq 0$ , dan menghitung nilai resistansi total. Ketiga resistor tersebut dapat dihubungkan secara serial maupun paralel.
- Pengguna bisa memilih untuk menghitung resistansi total (RT):
  - Jika dihubungkan serial (pilihan 1), yaitu  $RT = R_1 + R_2 + R_3$ ; dan
  - Jika dihubungkan paralel (pilihan 2), yaitu  $1/RT = 1/R_1 + 1/R_2 + 1/R_3$ .
- Buatlah program yang memvalidasi semua masukan dan selanjutnya menghitung dan menampilkan resistansi total.
- Validasi input pilihan pengguna, dan validasi input nilai resistor dari pengguna jika nilai resistor dihubungkan secara paralel (yaitu nilai resistor tidak boleh  $\leq 0$ ).
- Berikan pesan kesalahan jika ada masukan yang tidak valid.

# Latihan 3

- Buatlah program dalam notasi algoritmik yang membaca P1 dan P2 bertipe pecahan. Berikut definisi type pecahan di notasi algoritmik:

```
type pecahan : < pembilang : integer;  
                  penyebut : integer > 0 >
```

- Program akan membandingkan nilai pecahan P1 dan P2 dan menghitung selisihnya (dalam tipe **pecahan**), kemudian menuliskan hasilnya di layar.
- Sebelum melakukan proses perbandingan, program harus memeriksa bahwa pengguna memasukkan nilai pecahan yang valid, yaitu penyebut bernilai lebih besar daripada nol.
- Jika masukan tidak valid, program menampilkan pesan kesalahan di layar “Masukan tidak valid”. Masukan tidak perlu diulangi.