

Materi 4

LOGIKA & ALGORITMA

Operator Relasional

- Operator relasional, digunakan untuk membandingkan dua buah data.

- Simbol :

< lebih kecil

> lebih besar

>= lebih besar atau sama dengan

<= lebih kecil atau sama dengan

<> tidak sama

= sama (bedakan dengan tanda :=)

Pemilihan

- Dalam membuat program kadang-kadang kita perlu menganalisis kasus dan melakukan aksi berdasarkan status kasus tersebut.

Contoh:

- Tampilkan 'Anda Lulus' jika nilai ujian > 60
- Tampilkan 'Bilangan genap' jika anda memasukkan bilangan genap melalui keyboard.
- Tampilkan 'Salah Input' jika anda mencoba membagi sesuatu bilangan dengan 0
dsb

Pemilihan 1 kasus

- Bentuk dasar pemilihan 1 kasus:

```
if (kondisi) then
```

```
    (aksi)
```

```
endif
```

Aksi hanya dilakukan jika kondisi terpenuhi

Pseudocode vs Pascal

Pemilihan 1 kasus

Pseudocode :

- ⦿ if nilai>60 then
 write('Lulus')

endif

Pascal :

- ⦿ If nilai>60 then
 writeln('Lulus');

Latihan

- Buatlah sebuah program yang membaca nilai ujian dari keyboard. Jika nilai ujian ≥ 60 , tampilkan teks “Anda Lulus”. Jika nilai < 60 , program tidak menampilkan apa-apa.

pseudocode

- ⦿ deklarasi:

 - nilai : integer

- ⦿ deskripsi:

 - read(nilai)

 - if nilai \geq 60 then

 - write ('Anda Lulus')

 - endif

Notasi algoritma

- Notasi algoritma dapat berupa :
 - Uraian kalimat deskriptif (narasi)

Contoh

Algoritma Kelulusan_mhs

Diberikan nama dan nilai mahasiswa, jika nilai tersebut lebih besar atau sama dengan 60 maka mahasiswa tersebut dinyatakan lulus jika nilai lebih kecil dari 60 maka dinyatakan tidak lulus.

DESKRIPSI :

baca nama dan nilai mahasiswa.

jika nilai ≥ 60 maka

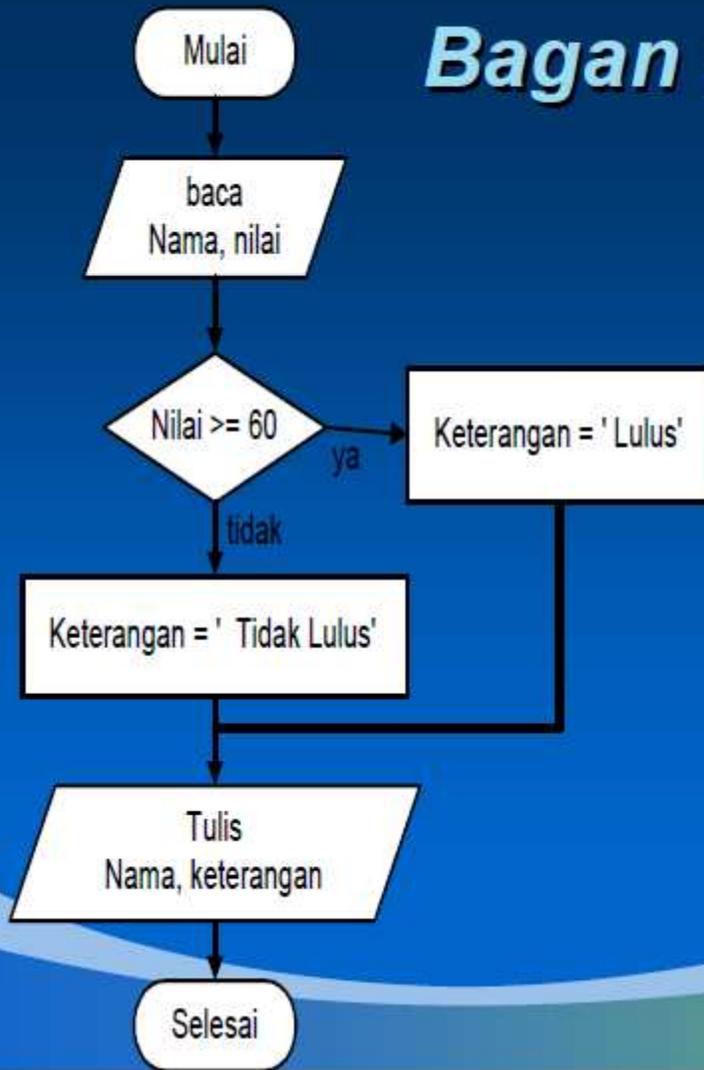
keterangan = lulus

tetapi jika

keterangan = tidak lulus.

tulis nama dan keterangan

Bagan alir (flow chart).



Pseudo-code.

Algoritma Kelulusan_mhs

{diberikan nama dan nilai mahasiswa, jika nilai tersebut lebih besar atau sama dengan 60 maka mahasiswa tersebut dinyatakan lulus jika tidak maka dinyatakan tidak lulus}

DEKLARASI :

Nama : string

Nilai : integer

Keterangan : string

DESKRIPSI :

read (nama, nilai)

if nilai \geq 60 then

keterangan = 'lulus'

else

keterangan = 'tidak lulus'

write(nama, keterangan)

Algoritma Luas_Kell_Lingkaran {<- ini judul algoritma}
{menghitung luas dan keliling lingkaran untuk ukuran jari-jari tertentu. Algoritma menerima masukan jari-jari lingkaran, menghitung luas dan kelilingnya, dan mencetak luas lingkaran ke piranti keluaran <- ini spesifikasi algoritma}

DEKLARASI :

const phi = 3.14 {nilai ? }
R : real {jari-jari lingkaran}
Luas : real {luas lingkaran}
Keliling : real {keliling lingkaran}

DESKRIPSI :

read (R)
Luas <- phi * R * R
Keliling <- 2 * phi * R
write(luas, keliling)

Latihan

- Buatlah sebuah program yang membaca sebuah bilangan bulat x dari keyboard. Jika angka yang dimasukkan adalah bilangan genap maka tampilkan teks “ x adalah bilangan genap”. Jika bukan bilangan genap maka program tidak menampilkan apa-apa.

pseudocode

- deklarasi:

A : integer

- deskripsi:

read(A)

if $A \bmod 2 = 0$ then

 write ('Bilangan genap')

endif

Latihan

- Tarif pemakaian internet di warnet 'Keong' adalah 3500/jam. Jika pemakaian lebih dari 5 jam akan mendapat diskon 10% dari total pemakaian.

Buatlah program console untuk menghitung biaya pemakaian internet. Lama pemakaian dimasukkan melalui keyboard.

Pseudocode (alternatif 1)

- ⦿ deklarasi:

lama, diskon, tarif : real

- ⦿ deskripsi:

read(lama)

diskon ← 0

if lama > 5 then

 diskon ← 0.1

endif

tarif ← 3500 * lama * (1-diskon)

write(tarif)

Pseudocode (alternatif 2)

- ⦿ deklarasi:

lama, biaya, tarif : real

- ⦿ deskripsi:

read(lama)

tarif ← 3500

if lama > 5 then

 tarif ← 0.9*3500

endif

biaya ← tarif * lama

write(biaya)

Pseudocode (alternatif 3)

- ⦿ deklarasi:

 - lama, biaya : real

- ⦿ deskripsi:

 - read(lama)

 - biaya \leftarrow 3500 * lama

 - if lama > 5 then

 - biaya \leftarrow 0.9*biaya

 - endif

 - write(biaya)