
6 PEMILIHAN

Analisis kasus adalah salah satu elemen primitif pembangun algoritma. Analisis kasus diperlukan dalam sebuah program ketika terdapat suatu instruksi yang hanya dikerjakan jika memenuhi persyaratan atau kondisi tertentu. Contoh pada algoritma Ibu Tati mengupas kentang

Penulisan Algoritma Yang Mengandung Analisa Kasus/Pemilihan

1. Flowchart

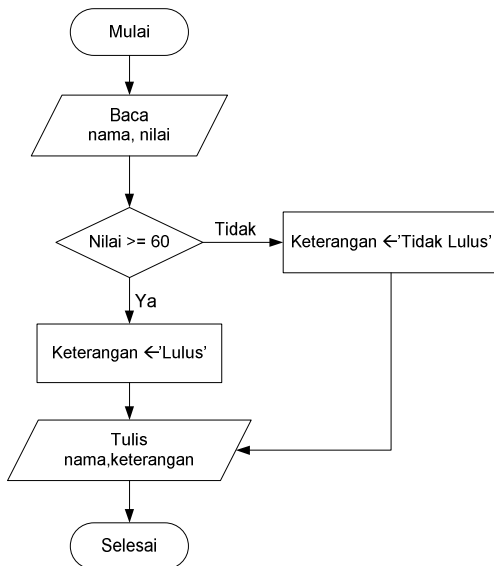
Permasalahan : Diberikan nama dan nilai mahasiswa, jika nilai tersebut lebih besar atau sama dengan 60 maka mahasiswa tersebut dinyatakan lulus jika nilai lebih kecil dari 60 maka dinyatakan tidak lulus.

– Uraian kalimat deskriptif (narasi)

DESKRIPSI :

- o baca nama dan nilai mahasiswa.
- o jika nilai ≥ 60 maka keterangan = lulus
- o tetapi jika nilai < 60 maka keterangan = tidak lulus.
- o tulis nama dan keterangan

– Flow chart



2. Pseudocode

Penulisan algoritma yang mengandung analisis kasus menggunakan pseudocode terdiri dari dua struktur umum : IF-THEN dan DEPEND-ON

Mendefinisikan analisis kasus adalah mendefinisikan :

- **kondisi** boolean, berupa suatu ekspresi yang menghasilkan nilai true atau false
- **aksi** yang akan dilaksanakan jika kondisi yang dipasangkan dengan aksi yang bersangkutan dipenuhi.

Ungkapan **Kondisi** dapat dihasilkan dengan operator perbandingan dan operator logika.

contoh kondisi : $x > 100$, $kar = '*'$, $(a \neq 0)$ or $(b = 0)$, $ketemu = true$, $not\ ketemu$

IF-THEN

a. Satu Kasus

```
if <kondisi> then  
    aksi  
endif
```

Contoh – contoh :

```
a. if x > 100 then  
    x ← x + 1  
endif  
b. if (a ≠ 0) or (b = 0) then  
    b ← a * b  
    write(b)  
endif  
c. if (ketemu) then  
    if a ≤ 10 then  
        read(b)  
    endif  
endif
```

Contoh Kasus 1 :

```
Algoritma HURUF_VOKAL  
{mencetak pesan "huruf vokal" bila sebuah karakter yang dibaca  
merupakan huruf vokal. Asumsi huruf kecil}  
Kamus  
huruf : char  
Deskripsi  
read(huruf)  
if (huruf='a') or (huruf='i') or (huruf='u') or (huruf='e') or  
    (huruf='o') then  
    write('Huruf Vokal')  
endif
```

Contoh Kasus 2 :

```
Algoritma Bilangan Genap  
{mencetak pesan "bilangan genap" kemudian kalikan bilangan tersebut  
dengan angka 2 jika bilangan bulat yang dimasukkan dari piranti  
masukan merupakan bilangan genap}  
Kamus  
bil : integer  
Deskripsi  
read(bil)  
if bil mod 2 = 0 then  
    write('bilangan genap')  
    write(bil*2)  
endif
```

b. Dua Kasus

```
if <kondisi> then  
    aksi1  
else  
    aksi2  
endif
```

Contoh – contoh :

```
a. if a > 0 then  
    write('bilangan positif')  
    else  
        write('bilangan negatif')  
    endif  
b. if (k > 4) and (k div 2 = 4) then  
    read(n)  
    z ← n * k  
    else  
        read(m)  
        z ← n / k  
    endif
```

Contoh Kasus 1 :

Algoritma Kelulusan Mhs {Menerima masukan nama dan nilai mahasiswa, jika nilai tersebut lebih besar atau sama dengan 60 maka mahasiswa tersebut dinyatakan lulus jika tidak maka dinyatakan tidak lulus}
Kamus Nama : string Nilai : integer Keterangan : string
Deskripsi <u>read</u> (nama, nilai) <u>if</u> nilai >= 60 <u>then</u> keterangan ← 'lulus' <u>else</u> keterangan ← 'tidak lulus' <u>endif</u> <u>write</u> (nama, keterangan)

Contoh Kasus 2 :

Algoritma Bilangan Genap dan Ganjil {Mencetak "bilangan genap" jika bilangan bulat yang dibaca merupakan bilangan genap dan "bilangan ganjil" jika bukan bilangan genap}
Kamus Bil : integer
Deskripsi <u>read</u> (bil) <u>if</u> (bil mod 2 = 0) <u>then</u> <u>write</u> ('Bilangan Genap') <u>else</u> <u>write</u> ('Bilangan Ganjil') <u>endif</u>

c. Banyak Kasus

```
if <kondisi1> then  
aksi1  
else  
    if <kondisi2> then  
aksi2  
else  
    if <kondisi3> then  
  
        aksi3  
    endif  
endif  
endif
```

Contoh Kasus 1 :

Membaca dua buah nilai integer, jika nilai pertama lebih besar atau sama dengan nilai kedua kerjakan nilai pertama/nilai kedua, jika nilai kedua lebih besar dari nilai pertama kerjakan nilai kedua/nilai pertama, tetapi jika nilai kedua = 0 maka tampilkan pesan error

```
Algoritma WUJUD_AIR  
{Menentukan hasil pembagian}  
Kamus  
    n1, n2 : integer  
    hasil : real  
Deskripsi  
read(n1,n2)  
if n2=0 then  
    write('Error')  
else  
    if n1>n2 then  
        hasil ← n1/n2  
    else  
        hasil ← n2/n1  
    endif  
    write(hasil)  
endif
```

DEPEND-ON

Untuk menyederhanakan pola IF-THEN-ELSE jika terdapat banyak kasus. Strukturnya :

depend on (nama)

```
<kondisi1> : aksi1  
<kondisi2> : aksi2  
<kondisi3> : aksi3  
.....  
<kondisiN> : aksiN  
[otherwise aksiX]
```

Tiap langkah diperiksa kebenarannya. Jika kondisi ke-k benar maka aksi ke-k dilaksanakan. Kondisi berikutnya tidak dipertimbangkan lagi. Jika tidak ada satupun aksi yang benar maka aksi sesudah *otherwise* yang dikerjakan.

Contoh Kasus 1:

Dibaca nomor dari 1-7 untuk menunjukkan hari. Tuliskan nama hari sesuai nomor harinya

```
Algoritma NAMA_HARI
{Mencetak nama bulan }
Kamus
    Nomor_hari : integer
Deskripsi
read(nomor_hari)
depend on (nomor_ hari)
    nomor_hari=1 : write('Januari')
    nomor_hari =2 : write('Februari')
    nomor_hari =3 : write('Maret')
    nomor_hari =4 : write('April')
    nomor_hari =5 : write('Mei')
    nomor_hari =6 : write('Juni')
    nomor_hari =7 : write('Juli')
    otherwise write('Bukan nomor hari yang benar')
```