

# STRUKTUR PEMILIHAN (SELECTION)

Temu 7

# Konsep Dasar Kondisi IF Bersarang (Nested IF)

- Secara sederhana, **IF bersarang** atau **nested IF** adalah penggunaan struktur **IF di dalam IF**. Kondisi seperti ini sering digunakan untuk kode program yang sudah cukup kompleks.
- Terdapat banyak variasi dari **nested IF**, tergantung kode program yang ingin kita rancang. Salah satunya adalah sebagai berikut:

# Contoh 1

```
IF (kondisi 1) THEN
  begin
    (kode program 1)
    IF (kondisi 1.1) THEN
      begin
        (kode program 1.1)
      end;
    end
  ELSE
  begin
    (kode program 2)
  end;
```

- Perhatikan bahwa (**kode program 1.1**) hanya akan dijalankan jika **kondisi 1** dan **kondisi 1.1** terpenuhi.
- Jika kondisi 1 **tidak terpenuhi**, program akan langsung lompat ke bagian **ELSE**.

## Contoh 2

- Pascal dan mayoritas bahasa pemrograman lain tidak membatasi jumlah IF di dalam IF. Sebagai contoh struktur yang lebih rumit seperti disamping ini:
- Kali ini struktur **IF di dalam IF di dalam IF** (*2 level nested IF*). Untuk kode program yang panjang seperti ini, harus hati-hati dalam menentukan kode program mana yang masuk ke dalam blok IF.
- Penulisan *indenting* (menjorokkan kode program beberapa spasi di awal) juga akan membantu untuk membedakan kode program ini masuk IF yang ini, dan kode program itu masuk ke IF yang itu.

```
IF (kondisi 1) THEN
  begin
    (kode program 1)
    IF (kondisi 1.1) THEN
      begin
        (kode program 1.1)
        IF (kondisi 1.1.1) THEN
          begin
            (kode program 1.1.1)
          end;
        end;
      end;
    end;
  end
ELSE
  begin
    (kode program 2)
    IF (kondisi 2.1) THEN
      begin
        (kode program 2.1)
      end;
    end;
  end;
```

# Struktur Case

- Struktur **CASE** dapat digunakan **untuk** menganalisis penyeleksian dua kasus atau lebih dan bentuknya adalah lebih sederhana dari pada struktur **IF-THEN-ELSE** yang memiliki bentuk bertingkat-tingkat
- **Kondisi<sub>1</sub>, kondisi<sub>2</sub>, ... kondisi<sub>N</sub>** dapat bernilai benar atau salah. Tiap kondisi diperiksa nilai pertama sampai ditemukan kondisi yang benar.
  - Jika **kondisi** ke-**k** benar, maka **aksi** ke-**k** dilaksanakan, selanjutnya keluar dari struktur **CASE**
  - **Aksi** yang dipasangkan dengan **kondisi** ke-**k** dapat lebih dari satu, karena itu ia berupa runtunan
  - Jika tidak ada satupun **kondisi** yang benar, maka **aksi** sesudah **otherwise** (**optional**) dikerjakan.

```
case (nama)
    kondisi1 : aksi1
    kondisi2 : aksi2
    kondisi3 : aksi3
    .
    .
    .
    kondisiN : aksiN
    [otherwise aksiX]
endcase
```

# Contoh 1

- Buatlah algoritma yang membaca sebuah angka bulat yang nilainya terletak antara 1 sampai 4, lalu menuliskan ke piranti keluaran angka tersebut dalam kata-kata. Misalkan bila dibaca angka 1, maka tercetak tulisan “satu”, bila dibaca 2, maka tercetak tulisan “dua”, demikian seterusnya. Jika angka yang dimasukkan selain 1 sampai 4 tuliskan pesan bahwa angka yang dimasukkan salah.

# Dengan struktur IF-THEN-ELSE

## Algoritma KATA\_UNTUK\_ANGKA

*{ Mencetak kata untuk angka 1 sampai 4, angka dibaca dari piranti masukan }*

### DEKLARASI:

Angka : integer { angka yang dibaca }

### DESKRIPSI:

```
read(angka)
if Angka=1 then
  write('satu')
else
  if Angka=2 then
    write('dua')
  else
    if Angka=3 then
      write('tiga')
    else
      if Angka=4 then
        write('empat')
      else
        write('angka yang dimasukkan salah')
      endif
    endif
  endif
endif
```

# Dengan STRUKTUR CASE

---

Algoritma KATA\_UNTUK\_ANGKA

{ Mencetak kata untuk angka 1 sampai 4, angka dibaca dari piranti masukan }

DEKLARASI:

A : integer { angka yang dibaca }

DESKRIPSI:

read(A)

case(A)

A=1 : write('satu')

A=2 : write('dua')

A=3 : write('tiga')

A=4 : write('empat')

otherwise write('angka yang dimasukkan salah')

endcase

## Contoh 2:

Buatlah algoritma untuk menentukan nama bulan berdasarkan pada nomor bulannya.

**Algoritma NAMA\_BULAN**

{ Mencetak nama bulan berdasarkan nomor bulannya }

DEKLARASI

AngkaBulan : integer { 1..12 }

DESKRIPSI :

read(AngkaBulan)

case(AngkaBulan)

    AngkaBulan=1 : write('Januari')

    AngkaBulan=2 : write('Februari')

    AngkaBulan=3 : write('Maret')

    AngkaBulan=4 : write('April')

    AngkaBulan=5 : write('Mei')

    AngkaBulan=6 : write('Juni')

    AngkaBulan=7 : write('Juli')

    AngkaBulan=8 : write('Agustus')

    AngkaBulan=9 : write('September')

    AngkaBulan=10 : write('Oktober')

    AngkaBulan=11 : write('Nopember')

    AngkaBulan=12 : write('Desember')

otherwise write('Bukan nomor bulan yang benar')

endcase

```
program NAMA_BULAN;  
{ Mencetak nama bulan berdasarkan nomor bulannya }  
(* DEKLARASI *)  
var  
    AngkaBulan : integer;  
(* DESKRIPSI *)  
begin  
    write('Masukkan angka bulan (1 – 12) : ');  
    readln(AngkaBulan);  
    case AngkaBulan of  
        1 : writeln('Januari');  
        2 : writeln('Februari');  
        3 : writeln('Maret');  
        4 : writeln('April');  
        5 : writeln('Mei');  
        6 : writeln('Juni');  
        7 : writeln('Juli');  
        8 : writeln('Agustus');  
        9 : writeln('September');  
    10 : writeln('Oktober');  
    11 : writeln('Nopember');  
    12 : writeln('Desember');  
    else writeln('Bukan nomor bulan yang benar');  
    end;  
end.
```

## Keberadaan CASE

- Tidak semua bahasa pemrograman menyediakan struktur CASE (misalnya Bahasa Fortran).
- Bahasa Pascal dan C menyediakan struktur Case.
- Bahasa pemrograman tidak menyediakan struktur CASE, maka struktur CASE dapat diganti dengan struktur IF-THEN-ELSE yang ekuivalen

# Latihan

1. Buatlah algoritma yang membaca angka bulan (integer), lalu menuliskan nama bulan sesuai angka bulannya. Misalnya jika dibaca bulan 8, maka tercetak 'Agustus'.

2. Sebuah program mempunyai menu sebagai berikut:

```
===== MENU =====
```

1. Baca data
2. Cetak data
3. Ubah data
4. Hapus data
5. Keluar program

```
=====
```

Pilihan Anda =

```
=====
```

Buatlah algoritma yang mencetak menu tersebut dan membaca nomor pilihan menu. Untuk nomor menu yang dipilih, tuliskan pesan seperti contoh berikut

Anda memilih menu nomor

Yang dalam hal ini adalah nomor menu yang dipilih oleh pengguna

# Latihan

3. Buatlah algoritma untuk menentukan: luas empat persegi panjang, keliling empat persegi panjang, dan panjang diagonal, lengkapi program dengan pilihan menu: 1) Luas, 2) Keliling, 3) Panjang diagonal, 4) Keluar program. Pada setiap pilihan menu, dibaca panjang dan lebar empat persegi panjang.

Luas persegi panjang =  $P * L$

Keliling persegi panjang =  $2 * \text{panjang} + 2 * \text{lebar}$

Diagonal =  $\text{sqrt}(P * P + \text{lebar} * \text{lebar})$