

## **BAB 6**

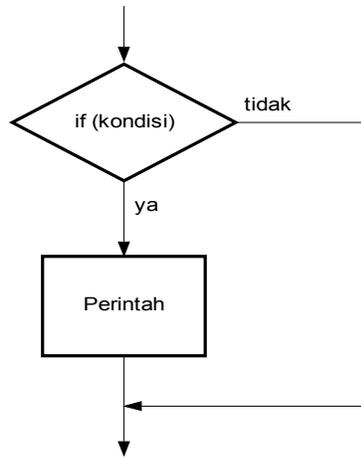
### **KONTROL ALIRAN**

Ada delapan pernyataan kontrol kendali yang disediakan di dalam Matlab. Kedelapan pernyataan tersebut antara lain

1. `if`, termasuk di dalamnya pernyataan `else` dan `elseif`. Pernyataan ini menjalankan kelompok pernyataan berdasarkan pada syarat logika.
2. `switch`, termasuk di dalamnya adalah `case` dan `otherwise`. Statement ini mengeksekusi kelompok pernyataan berbeda bergantung pada harga syarat kondisi.
3. `while`, menjalankan group pernyataan dengan jumlah iterasi tak terbatas berdasarkan pada syarat logika.
4. `for` menjalankan group pernyataan dengan jumlah iterasi telah ditentukan.
5. `continue` melewati kendali ke iterasi berikutnya untuk loop `for` atau `while`.
6. `break` berfungsi menghentikan eksekusi looping `for` atau `while`.
7. `try...catch` mengubah kendali aliran apabila ditemukan kesalahan selama proses eksekusi.
8. `return` menyebabkan eksekusi kembali ke fungsi invoking. Semua aliran membangun pemnggunaan `end` untuk menunjukkan akhir dari blok kontrol aliran.

#### **1. Pernyataan Bersyarat `if`, `else`, dan `elseif`**

Pernyataan `if` digunakan untuk menyeleksi suatu kondisi yang memungkinkan dua atau lebih pilihan. Bila proses yang diseleksi terpenuhi atau bernilai benar, maka pernyataan yang ada di dalam blok `if` akan diproses dan dikerjakan. Jika digambarkan dalam diagram alir, maka percabangan `if` dapat digambarkan sebagaimana terlihat pada gambar 4.4



Gambar 4.4 Diagram alir percabangan *if*

Bentuk umum struktur kondisi if adalah :  
 pernyataan;

```

if (kondisi)
    pernyataan/perintah
end
  
```

Apabila pernyataan logika berharga benar (true), maka seluruh pernyataan atau perintah yang berada diantara if dan end akan dijalankan. Sebaliknya, jika pernyataan logika berharga salah (false), maka seluruh perintah/pernyataan yang berada diantara if dan end tidak akan dijalankan karena Matlab akan langsung menuju end.

contoh:

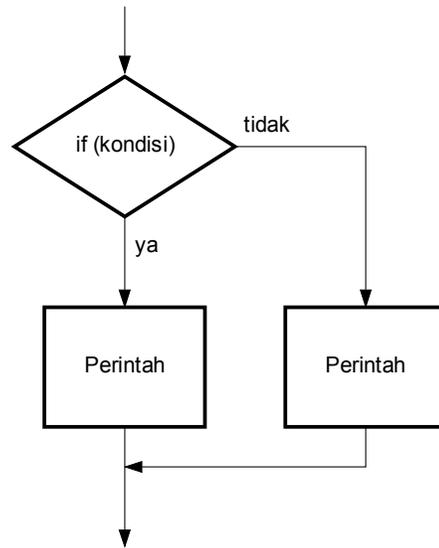
```

if rem(a,2) == 0
    disp('a adalah bilangan genap')
    b = a/2;
end
  
```

## 2. Pernyataan Bersyarat if-else

Dalam pernyataan bersyarat if-else paling tidak terdapat dua pernyataan. Jika kondisi yang diperiksa bernilai benar atau terpenuhi maka pernyataan pertama yang dilaksanakan

dan jika kondisi yang diperiksa bernilai salah maka pernyataan yang kedua yang dilaksanakan. Pernyataan ini dapat diilustrasikan seperti pada gambar 4.5.



Gambar 4.5 Diagram alir percabangan if...else

Bentuk umumnya adalah sebagai berikut :

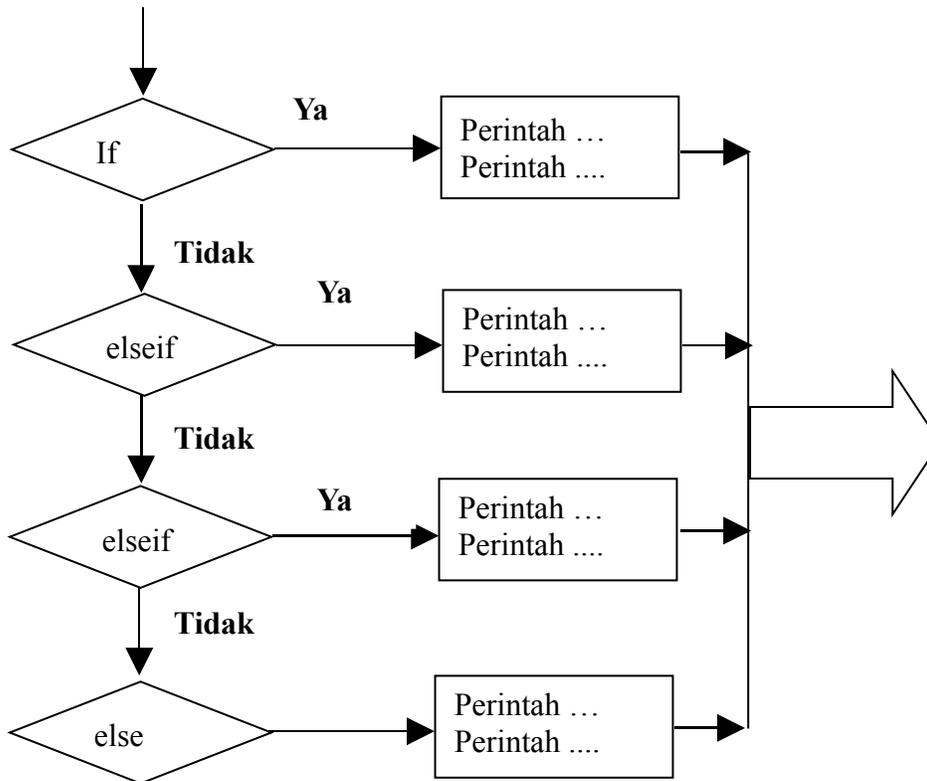
```
if(kondisi)
    perintah_1
else
    perintah_2
end
```

%Contoh program if-else

```
x = input('Masukkan harga x:');
if (x > 3)
    disp('Pernyataan benar. ');
    disp('Pernyataan benar sekali.. ');
else
    disp('Pernyataan salah. ');
    disp('Pernyataan salah sekali.. ');
end
```

### 3.Pernyataan Bersyarat elseif

Pernyaan elseif akan dieksekusi apabila syarat yang diberikan pada if sebelumnya (atau elseif sebelumnya) berharga salah (0). Pernyaan ini selanjutnya akan mengeksekusi perintah/pernyataan di dalamnya apabila syarat logikanya berharga benar (1). Pernyataan ini dapat digambarkan dengan diagram dibawah ini



Gambar 6.4 Diagram alir pernyataan elseif

%contoh penggunaan elseif

```
n = input('Masukkan harga n:')  
if n < 0 % jika n negatif, ada pesan salah.  
    disp('Masukan harus bilangan positif');  
elseif rem(n,2) == 0 % jika n positif dan bulat,  
    % kemudian bagi dengan 2  
    A = n/2;  
else  
    A = (n + 1)/2; % jika n positif dan ganjil
```

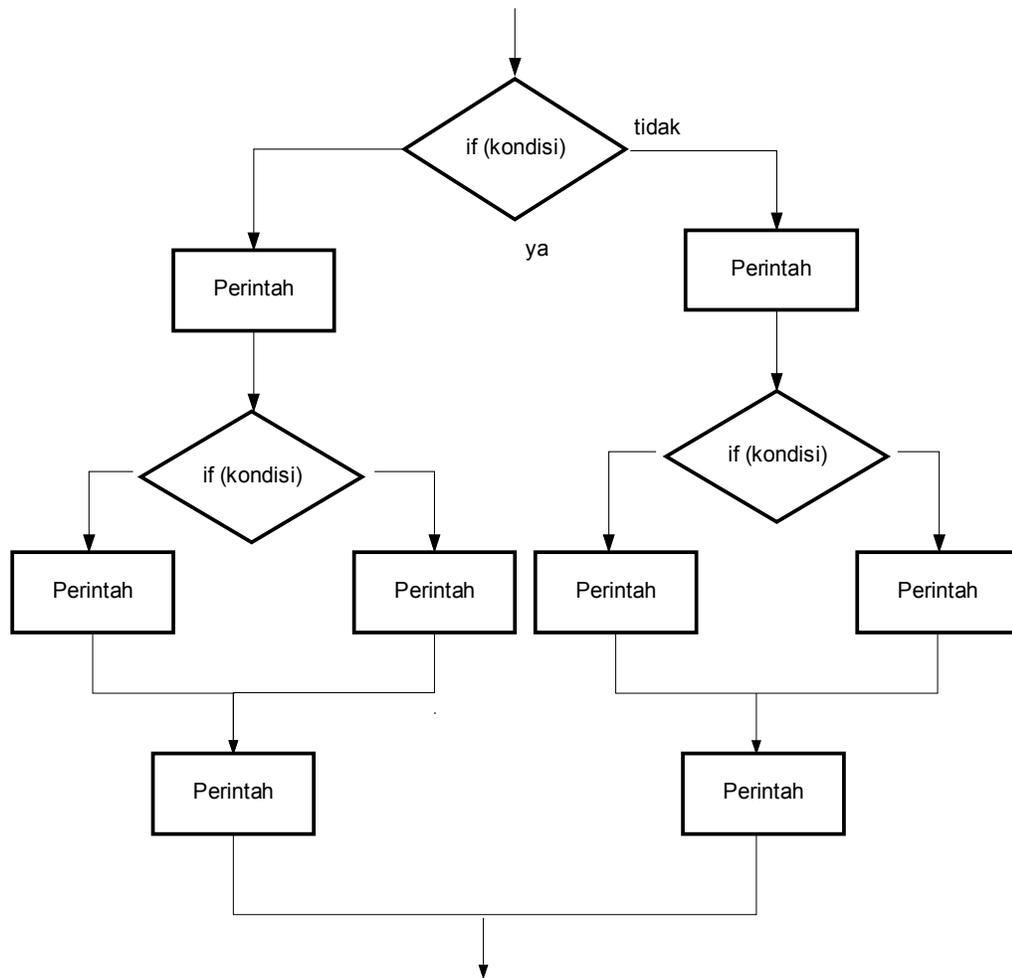
Gambar 6.6 Diagram alir pernyataan elseif

```
% tambahkan 1,kemudian bagi
% dg 2.
end
```

#### 4. Pernyataan bersyarat if bersarang

Pernyataan ini sangat penting untuk masalah-masalah yang memiliki lebih dari dua cabang. Mengapa pernyataan ini disebut pernyataan bersarang, karena di dalam pernyataan if ada pernyataan if lagi. If yang kedua ini dapat berada di dalam if sendiri atau berada di dalam else. Untuk lebih jelasnya lihatlah diagram alir 6.5. Bentuk umum dari pernyataan if bersarang adalah

```
if (syarat/kondisi)
    perintah/ Pernyataan
    if (kondisi)
        perintah/ Pernyataan
    else
        perintah/ Pernyataan
    end
else
    perintah/ Pernyataan
    if (kondisi)
        perintah/ Pernyataan
    else
        perintah/ Pernyataan
    end
end
```



Gambar 6.5 Diagram Alir Pernyataan If Bersarang

### switch

Pernyataan switch akan mengeksekusi sekelompok perintah/ Pernyataan berdasarkan pada harga variabel atau ekspresi. Bentuk dasar dari switch adalah

```

switch ekspresi (skalar atau string)
  case nilai1
    pernyataan % dieksekusi jika ekspresi adalah
    nilai1
  case nilai2
    pernyataan % dieksekusi jika ekspresi adalah
    nilai2
  .
  .
  otherwise
    pernyataan % dieksekusi jika ekspresi tidak cocok
    % dengan case manapun
  
```

Dari bentuk umum penggunaan switch di atas, maka pernyataan switch terdiri atas beberapa blok antara lain:

- switch diikuti oleh ekspresi yang mana dapat berupa skalar (bilangan) maupun string.
- Sejumlah grup case yang masing-masing diikuti oleh nilai ekspresi. Nilai ekspresi ini berada dalam satu baris dengan case. Baris dibawahnya berisi pernyataan-pernyataan yang akan dijalankan apabila ekspresi nilai sesuai dengan ekspresi pada switch. Blok case akan berakhir saat ditemukan blok otherwise.
- Blok otherwise. Blok ini ada untuk mengantisipasi apabila nilai ekspresi pada switch tidak sama dengan nilai ekspresi pada case manapun. Pernyataan-pernyataan yang berada dalam blok ini akan berakhir saat bertemu dengan pernyataan end.
- Pernyataan end.

Pernyataan switch bekerja dengan cara membandingkan masukan (input) ekspresi dengan harga pada tiap-tiap pernyataan case. Untuk ekspresi numerik, pernyataan case bernilai benar apabila (nilai=ekspresi). Untuk ekspresi string, pernyataan case bernilai benar apabila strcmp(nilai,ekspresi).

Dibawah ini diberikan sebuah contoh penggunaan pernyataan switch-case. Pernyataan switch akan mengecek variabel masukan untuk nilai tertentu. Apabila variabel masukan adalah 1, 2 atau 3 , maka pernyataan case akan menampilkan luas masing-masing untuk kubus, persegi panjang dan lingkaran. Sedangkan, apabila variabel masukan bukan 1,2 atau 3, maka switch akan langsung menuju ke otherwise.

% contoh menghitung luas bangun

```
fprintf('Pilih salah satu program..\n');  
fprintf('1: menghitung luas bujur sangkar: ');  
fprintf('2: menghitung luas persegi panjang: ');  
fprintf('3: menghitung luas lingkaran: ');  
  
no = input('Anda ingin menghitung luas 1,2 atau 3 \n');
```

```

switch no

    case 1
        a = input('masukkan panjang sisi bujur sangkar:');
        luas = a * a;
        fprintf('Luas bujur sangkar adalah %f \n',luas);

    case 2
        a = input('masukkan panjang persegi panjang: ');
        b = input('masukkan lebar persegi panjang: ');
        luas = a * b;
        fprintf('Luas persegi panjang adalah %f',luas);

    case 3
        a = input('masukkan jari-jari lingkaran :');
        luas = 3.14 * pow(r,2);
        fprintf('Luas lingkaran adalah %f ',luas);

    otherwise
        fprintf('Ulangi pilihan yang ada...!!')

end

```

Pernyataan switch juga dapat digunakan untuk menghandle multi syarat dalam satu case.

```

% contoh menghitung luas bangun
fprintf('Pilih salah satu program..\n');
fprintf('1: menghitung luas bujur sangkar: ');
fprintf('2: menghitung luas persegi panjang: ');
fprintf('3: menghitung luas persegi panjang: ');
fprintf('4: menghitung luas persegi panjang: ');
no = input('Anda ingin menghitung luas 1,2 atau 3 \n');
switch no

```

```

case 1
a=input('masukkan panjang sisi bujur sangkar:');
luas = a*a;
fprintf('Luas bujur sangkar adalah %f \n',luas);
case {2,3,4}
a=input('masukkan panjang persegi panjang: ');
b=input('masukkan lebar persegi panjang: ');
luas = a*b;
fprintf('Luas persegi panjang adalah %f',luas);
otherwise
fprintf('Ulangi pilihan yang ada...!!')
end

```

## 5. while

Pernyataan while akan menjalankan pernyataan atau sekelompok pernyataan secara berulang-ulang selama kontrol perulangan berharga benar (1). Secara umum, bentuk dari switch adalah

```

while expression
statements
end

```

Program di bawah ini akan mencari n bilangan pertama perjumlahan bilangan interger yang berharga < 100.

```

% contoh mencari n bilangan pertama perjumlahan
jum = 100;
n = 1;
while sum(1:n) < jum
disp(n)
n = n + 1;
end

```

```

%Program mencetak deret bilangan
i = 1;
while(i <= 6)
    x = 1;
    while(x <= i)
        fprintf('%i',x);
        x = x + 1;
    end
    fprintf('\n');
    i = i + 1;
end

```

## 5. Perulangan for

Struktur perulangan for biasa digunakan untuk mengulang suatu proses yang telah diketahui jumlah perulangannya. Dari segi penulisannya, struktur perulangan for tampaknya lebih efisien karena susunannya lebih simpel dan sederhana. Secara default, penambahan indeks adalah 1, tetapi kita dapat mengeset sendiri misalnya penambahannya 2, 3 atau berapapun bahkan dapat pula negatif. Untuk penambahan positif, perulangan akan berakhir manakala indeks sudah lebih dari yang tentukan. Sedangkan untuk penambahan negatif, perulangan berakhir saat indeks lebih kecil dari yang ditentukan. Secara umum, sintaks for adalah sebagai berikut

```

for indeks = mulai:penambahan:akhir
    pernyataan/perintah
end

```

Contoh program di bawah ini akan mencetak sebuah kalimat berbunyi "Aku mahasiswa semester 5" sebanyak lima kali dengan menggunakan pernyataan perulangan for.

```
%contoh perulangan for
N = 5;
for i = 1:N
    fprintf('%i. Aku mahasiswa semester 5 \n',i);
end
```

## 6. Pernyataan continue

Pernyataan continue berfungsi melewati kendali ke iterasi berikutnya di dalam perulangan for maupun while. Apabila pernyataan continue berada di dalam loop bersarang, maka pernyataan continue akan melewati kendali ke iterasi berikutnya. Sebagai contoh, perhatikan di bawah ini

```
N = 20;
for i = 1:N
    if (i < 5)
        continue;
    end
    fprintf('%i. Aku mahasiswa semester 5 \n',i);
end
```

Apabila skrip program di atas dieksekusi, maka yang keluar adalah tampilan tulisan “Aku mahasiswa semester 5” sebanyak 16 buah mulai dari nomor 5 hingga nomor 20. Dengan kata lain, kalimat dari nomor 1 hingga 4 tidak ditampilkan. Hal ini disebabkan perintah continue yang berada di dalam loop for.

## 8. Pernyataan break

Pernyataan break berfungsi untuk menghentikan eksekusi loop for maupun loop while. Apabila break ditemukan di dalam kalang for maupun while, maka eksekusi akan dihentikan apabila syarat yang diberikan terpenuhi. Untuk lebih jelasnya perhatikan contoh berikut ini

```
N = 20;
for i = 1:N
    if (i > 5)
        break;
    end
    fprintf('%i. Aku mahasiswa semester 5 \n', i);
end
```

Program di atas akan ditampilkan lima buah kalimat “Aku mahasiswa semester 5” mulai dari nomor 1 hingga 5. Setelah iterasi ke 5, program akan dihentikan karena pernyataan break di dalam pernyataan bersyarat dimana syaratnya telah terpenuhi. Dengan kata lain, kalimat pada nomor 6 hingga nomor 20 tidak akan dicetak

9.