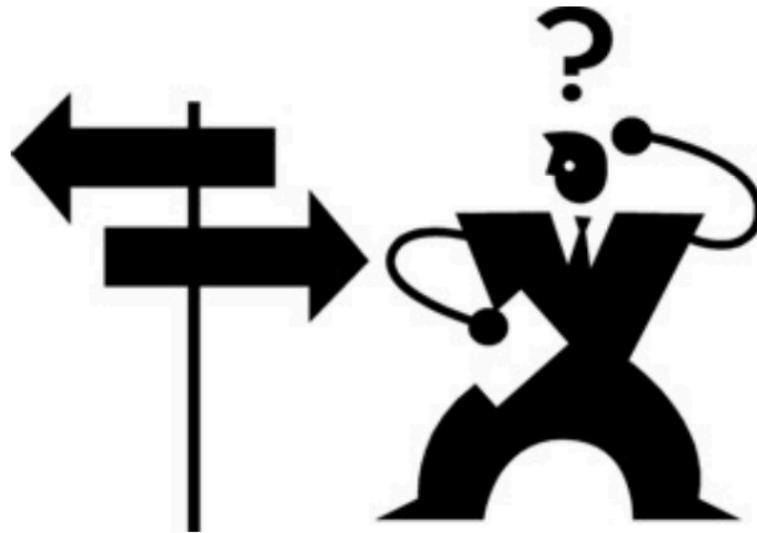


IF

Decision

Pilihan





Overview

Program dapat merepresentasikan situasi pemilihan yang sering dihadapi dalam dunia nyata. Berdasarkan satu atau beberapa kondisi, dapat ditentukan satu atau sejumlah aksi yang akan dilakukan. Dengan adanya struktur pemilihan, program dapat berjalan dengan jalur yang berbeda, berdasarkan hasil pengecekan kondisi yang dipenuhi.



Download

Tujuan

1. Memahami struktur pemilihan dalam program
2. Mengenal struktur IF dan CASE yang dapat digunakan dalam pemilihan
3. Memahami konsep kondisi dan aksi dalam struktur pemilihan
4. Menerapkan pemilihan dalam menyelesaikan berbagai kasus

Dalam kehidupan nyata, seringkali dihadapkan pada beberapa pilihan. Pada saat menghadapi pilihan, satu atau beberapa kondisi menjadi bahan pertimbangan dalam memutuskan untuk melakukan aksi tertentu. Contoh:

Jika cuaca mendung, maka saya membawa payung.

Pada contoh tersebut, 'cuaca mendung' merupakan kondisi yang menjadi bahan pertimbangan untuk melakukan aksi 'saya membawa payung'. Jika kondisi 'cuaca mendung' terpenuhi (bernilai benar), maka aksi 'saya membawa payung' dilakukan

Sebuah program komputer juga dapat mengenali situasi pemilihan. Pernyataan dalam contoh di atas dapat dituliskan dalam struktur pemilihan sebagai berikut:

```
IF cuaca mendung THEN  
    saya membawa payung  
END IF
```

Untuk selengkapnya penggunaan bentuk pemilihan akan dijelaskan berikut ini.

Bentuk Umum IF

Bentuk IF yang juga dikenal dengan istilah IF Statement, memiliki bentuk umum sebagai berikut :

```
If kondisi then  
    Aksi-1  
[else  
    Aksi-2]  
End if
```

Kondisi adalah ekspresi boolean yang bernilai benar atau salah, bisa berupa:

- Sebuah nilai boolean: true atau false

- Sebuah variabel boolean

- Sebuah perbandingan data

- Dua perbandingan data atau lebih yang digabung

Aksi berupa satu statement beberapa statement, dimana tiap statement dapat berupa:

- Statement pengisian nilai seperti `a ← 5`

- Statement input data

- Statement output data

- Statement pemilihan (If Statement atau Case Statement)

- Statement pengulangan (For Statement atau While Statement)

Download

[else Aksi-2], tanda [] menyatakan opsional (boleh ada/tidak ada),
dimana kalau tidak ada, berarti setelah Aksi-1 langsung selesai.

Dari bentuk umum yang telah dijelaskan, maka variasi bentuk IF ini banyak dan tidak berhingga. Di antaranya yang penting dapat disebutkan berikut:

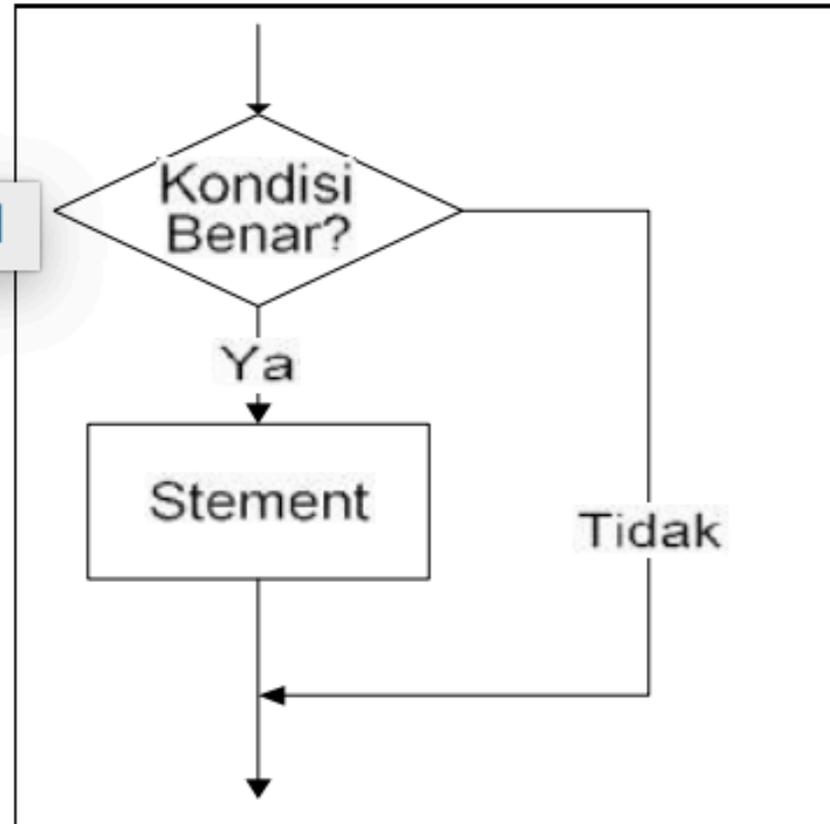
- if tanpa else (satu pilihan, mengerjakan atau tidak)
- if dengan else (dua pilihan)
- if bersarang dimana dalam if ada if lagi, karena Statement dapat berupa satu perintah pemilihan. Salah satu bentuk if bersarang adalah if untuk memilih salah satu dari banyak pilihan.

Contoh-contoh variasi:

I. Satu pilihan (tanpa ELSE)

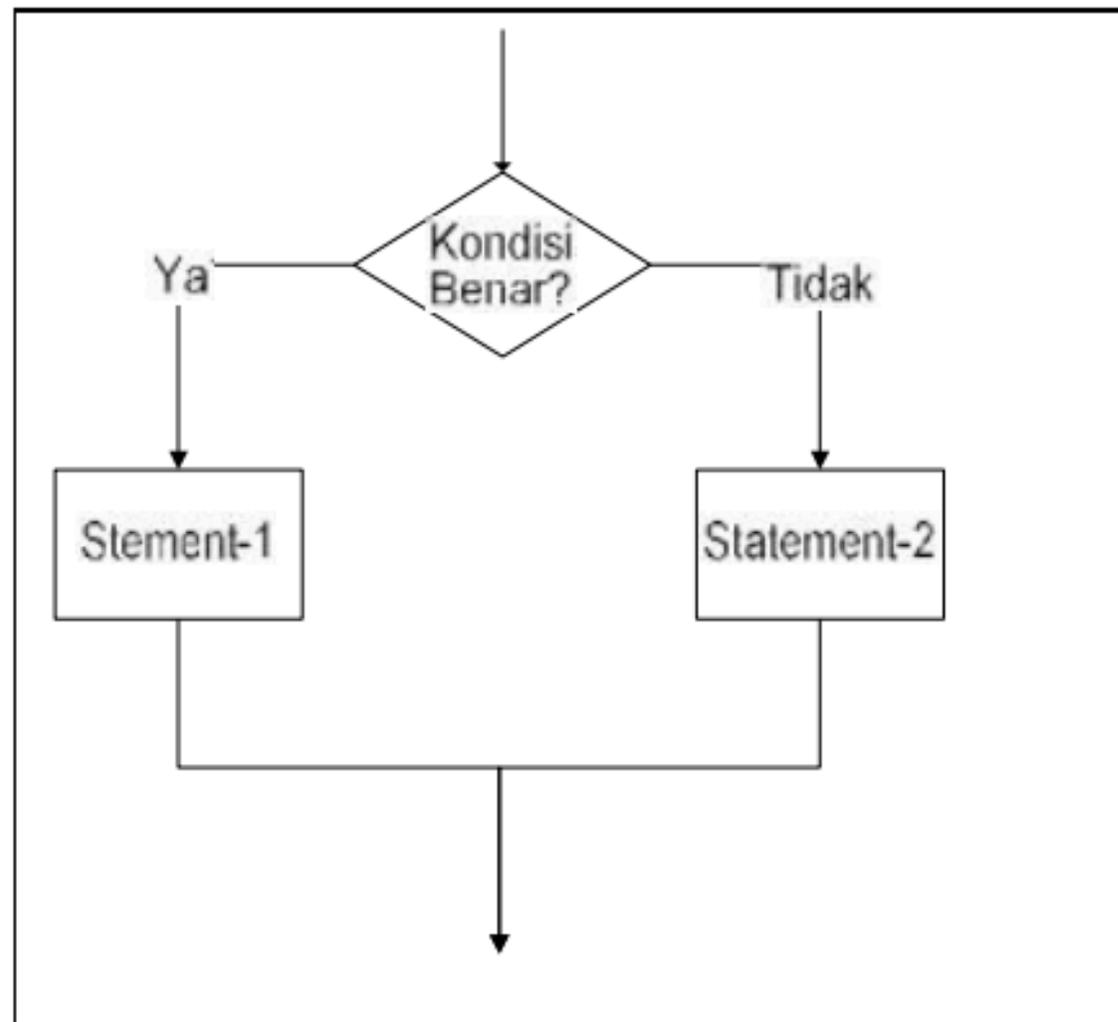
|If kondisi the
|Statement
|End if

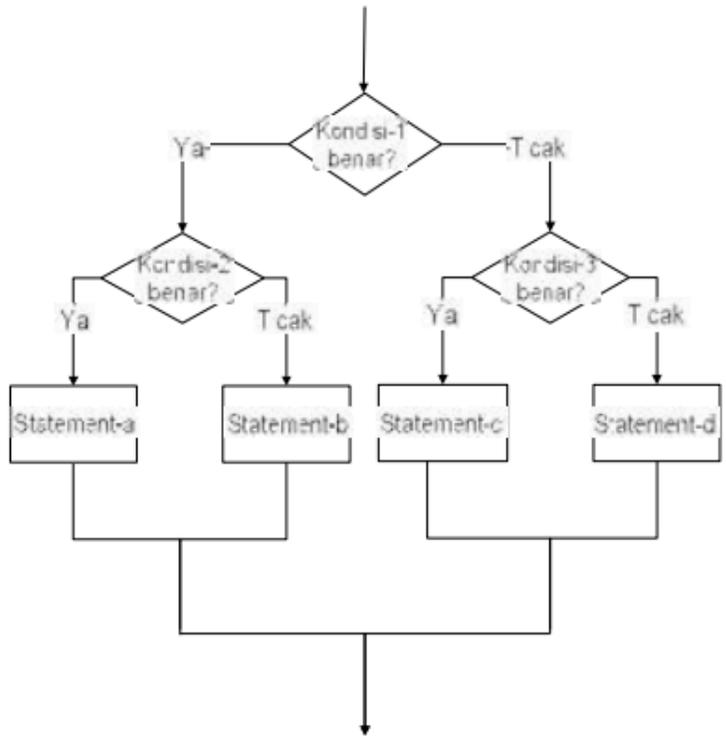
Download



2. Dua pilihan (dengan ELSE)

```
| if kondisi then  
|   Statement-1  
| else  
|   Statement-2  
| End if
```

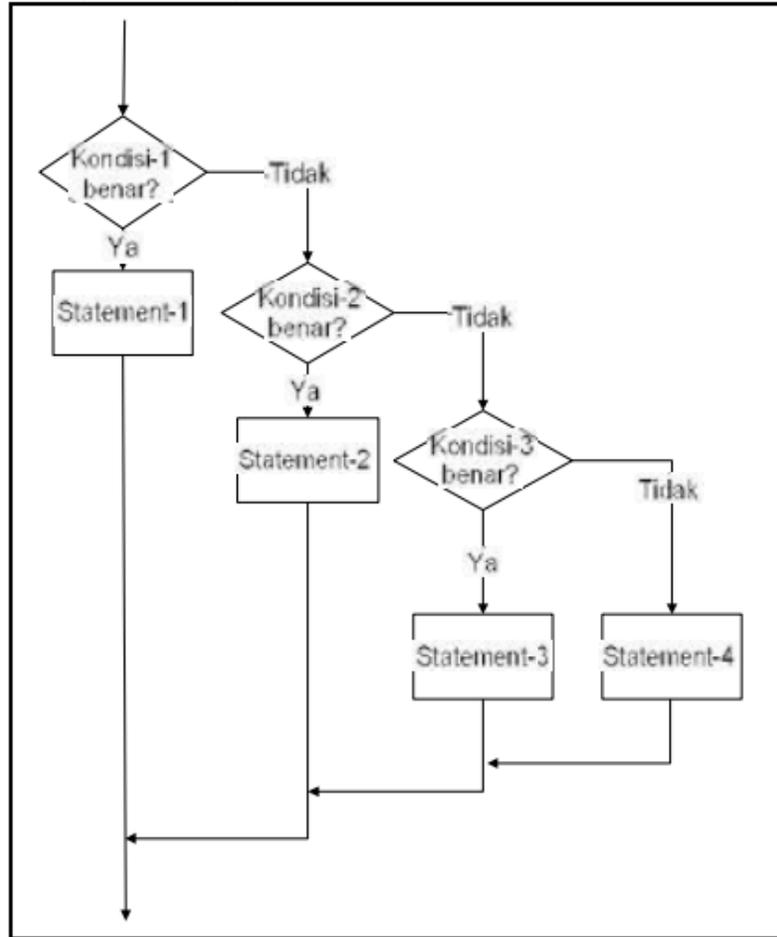




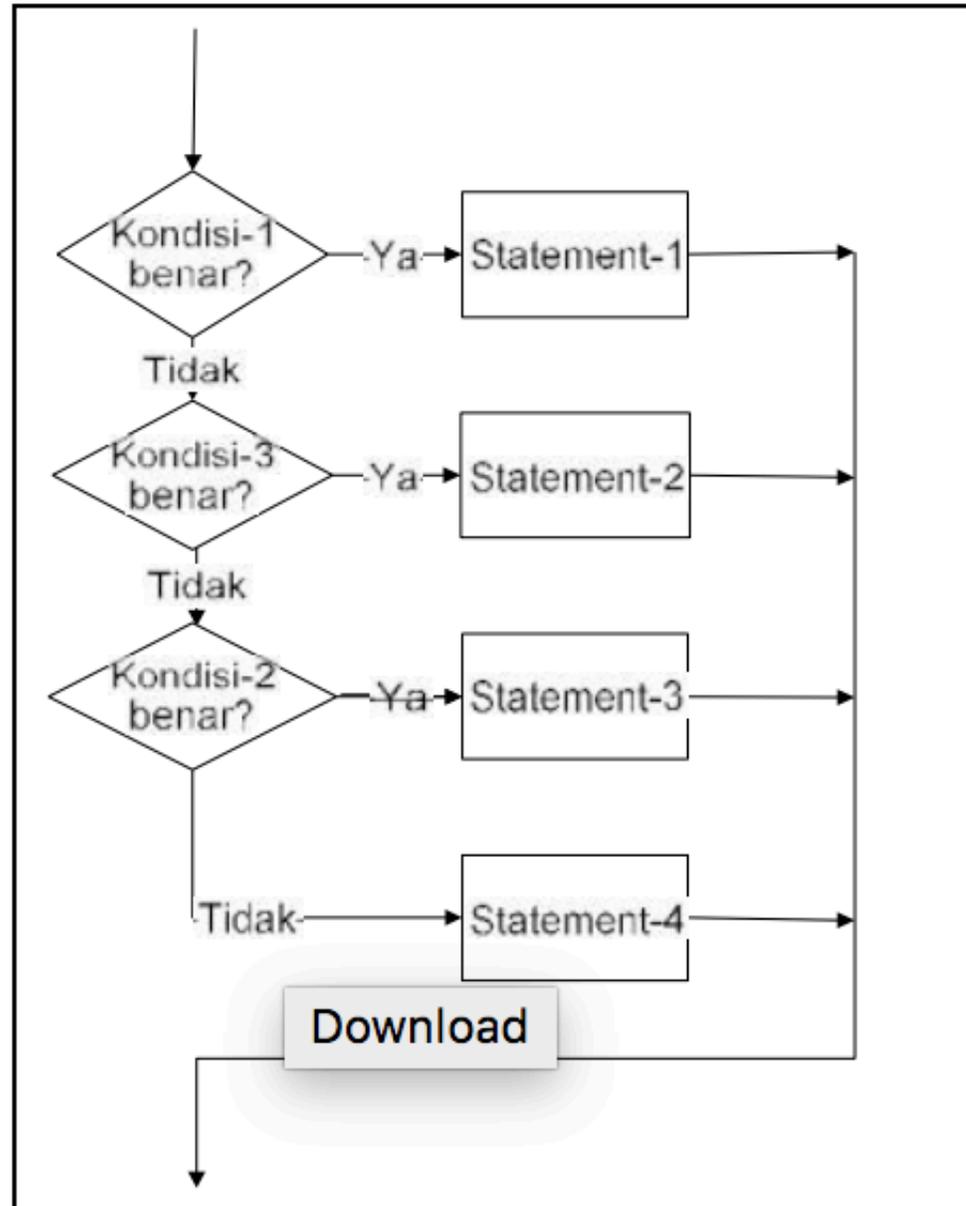
memeriksa kondisi-3. Apabila kondisi-3 bernilai benar maka Statement-c yang dikerjakan dan apabila kondisi-3 bernilai salah maka Statement-d yang dikerjakan. (dari 4 statement yang ada hanya salah satu yang dikerjakan).

4. Tiga pilihan atau lebih, dengan mengembangkan Statement setelah ELSE

```
| If kondisi-1 then  
|   Statement-1  
| else  
|   | if kondisi-2 then  
|   |   Statement-2  
|   | Else  
|   |   | If kondisi-3 then  
|   |   |   Statement-3  
|   |   |   Else  
|   |   |   | Statement-4  
|   |   |   End if  
|   |   End if  
|   End if
```



If kondisi-1 then
| Statement-1
Else if kondisi-2 then
| Statement-2
Else if kondisi-3 then
| Statement-3
Else
| Statement-4
End if



Sebuah masalah terkadang dapat diselesaikan dengan berbagai cara, seperti penggunaan “if tanpa else” dan “if dengan else”. Sebagai contoh dapat dilihat pada kasus berikut:

Kasus 4.1 : Menentukan apakah bilangan yang diinput positif atau negatif

Solusi : ada beberapa cara berikut

Solusi-1

```
|Input(bil)
|If (bil>=0) then
|  Output('positip')
|Else
|  Output('negatip')
|End if
```

Download

Solusi-2

```
|Input(bil)
|If (bil<0) then
|  Output('negatip')
|Else
|  Output('positip')
|End if
```

Solusi-3

```
|Input(bil)
|Ket ← 'positip'
|If (bil<0) then
|  Ket ← 'negatip'
|End if
|Output(Ket)
```

Solusi-4

```
|Input(bil)
|Ket ← 'negatip'
|If (bil>=0) then
|  Ket ← 'positip'
|End if
|Output(Ket)
```

Solusi-5

```
|Input(bil)
|If (bil>=0) then
|  Output('positip')
|End if
|If (bil<0) then
|  Output('negatip')
|End if
```

Solusi-6

```
|Input(bil)
|If (bil<0) then
|  Output('negatip')
|End if
|If (bil>=0) then
|  Output('positip')
|End if
```

Solusi-7

```
|Input(bil)
|positip ← bil>=0
|If (positip=true) then
|  Output('positip')
|else
|  Output('negatip')
|End if
```

Solusi-8

```
|Input(bil)
|positip ← bil>=0
|If (positip) then
|  Output('positip')
|else
|  Output('negatip')
|End if
```

Download

Ulasan dari beberapa solusi:

Solusi-1 dan Solusi-2 adalah solusi yang sama, digunakan kondisi berkebalikan sehingga posisi perintah tampilan ditukar.

Solusi-3 dan Solusi-4 juga sama, keduanya menggunakan “if tanpa else”, dengan cara variabel Ket diinisialisasi (diberi nilai awal) dengan salah satu kemungkinan hasilnya, kemudian diubah bila memenuhi kondisi.

Solusi-5 dan Solusi-6 juga sama, pada solusi ini dibuat 2 buah “if tanpa else” secara terpisah. Dengan cara ini, berarti akan dilakukan pemeriksaan kondisi 2 kali (padahal sebenarnya cukup satu kali).

Solusi-7 dan Solusi-8 keduanya menggunakan variabel bertipe boolean bernama positip untuk mencatat hasil perbandingan $bil \geq 0$. Penulisan “if (positip=true)” sama saja dengan menuliskan “if (positip)” cara yang terakhir lebih cepat waktu eksekusinya.

TUGAS

- Buatlah Solusi Untuk
 - Pilihlah bilangan tertinggi dari 4 buah data (A,B,C,D) yang di inputkan.