

PERCABANGAN

A. Konsep dan Struktur Dasar Algoritma Percabangan

Sebuah program tidak hanya berjalan dalam garis lurus atau hanya ada satu jalan dalam menyelesaikan suatu persoalan, adakalanya program itu harus memiliki sebuah alternatif jalan lain agar bisa berjalan. Hal ini bisa di ibaratkan seperti sebuah persimpangan di jalan raya, dimana seorang pengemudi bisa memilih jalur mana yang harus ia pilih untuk mencapai tujuannya, dan juga misalkan jalan yang biasanya ia lewati itu sedang ada perbaikan maka pengemudi itu dapat memilih alternatif jalan lain yang bisa ia gunakan.

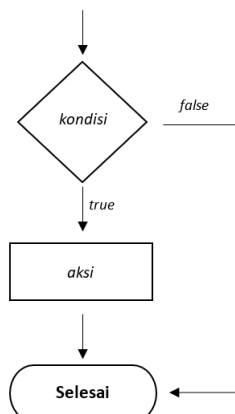
Pada struktur percabangan, program akan berpindah urutan pelaksanaan jika suatu kondisi yang disyaratkan dipenuhi. Maka dari itu dapat disimpulkan bahwa algoritma percabangan merupakan langkah-langkah logis yang disusun secara sistematis untuk menyelesaikan permasalahan dengan menggunakan alternatif dari persyaratan tertentu. Konsep dasar percabangan terletak pada persyaratan atau kondisi permasalahan serta tindakan atau aksi bila kondisi tertentu terpenuhi. Kondisi adalah suatu ekspresi *boolean* yang bernilai *true* atau *false* dan menentukan aksi yang dilakukan jika kondisi tersebut terpenuhi.

B. Algoritma Percabangan dengan Satu Kasus

Notasi algoritma untuk analisis dengan satu kasus adalah dengan menggunakan konstruksi

If *kondisi* **then**
 aksi
End if

Intruksi di atas berarti bahwa *aksi* hanya akan dilakukan jika *kondisi* benar(true). Bila kondisinya salah (false) maka tidak ada *aksi* apapun yang dieksekusi. **end if** sengaja ditambahkan sebagai penanda bahwa struktur dari **if-then** berakhir. Aksi yang dikerjakan bisa lebih dari satu.



Flow chart diatas dapat membantu dalam memvisualisasikan percabangan dengan satu kasus. Jika kondisi benar maka aliran intruksi akan bercabang dan mengeksekusi *aksi* yang ada, sedangkan jika kondisinya salah maka langsung selesai atau tidak ada aksi yang dieksekusi.

Contoh :

Buatlah algoritma yang menerima inputan bilangan bulat, jika bilangan yang di inputkan adalah genap maka akan muncul tulisan “Bilangan Genap”

Penyelesaian :

Bilangan genap adalah bilangan yang habis dibagi 2, maka dari itu kita bisa menggunakan operator modulus (sisanya bagi) dalam menyelesaikan persoalan diatas, dengan demikian jika bilangan yang di-inputkan bila dimodulus dengan 2 sama dengan 0 maka cetak tulisan “Bilangan Genap”.

PROGRAM BilanganGenap

{Mencetak tulisan “Bilangan Genap” jika bilangan yang di-inputkan adalah bilangan genap}

DEKLARASI :

bil : **integer**

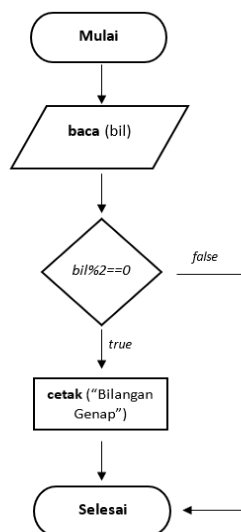
DESKRIPSI :

read(bil)

if bil **mod** 2 == 0 **then**

write(“Bilangan Genap”)

end if



Berikut adalah contoh penyelesaian dari bahasa python

```
1 # Program Bilangan Genap
2 bil = int(input("Masukan bilangan : "))
3
4 if bil %2 == 0 :
5     print("Bilangan Genap")
```

```
PS C:\Users\ACER\OneDrive\Desktop> py.exe .\main.py
Masukan bilangan : 2
Bilangan Genap
PS C:\Users\ACER\OneDrive\Desktop>
```

```
PS C:\Users\ACER\OneDrive\Desktop> py.exe .\main.py
Masukan bilangan : 1
PS C:\Users\ACER\OneDrive\Desktop>
```

C. Algoritma Percabangan dengan Dua Kasus

Pada percabangan ini sebuah program memiliki dua buah alternatif kondisi yang memiliki aksi di masing – masing kondisi, berbeda dengan algoritma percabangan dengan satu kondisi yang jika suatu kondisi itu bernilai *false* maka tidak ada aksi yang dijalankan. Notasi algoritma dengan dua buah kasus menggunakan kontruksi *IF-THEN-ELSE* (jika-maka-kalau tidak) :

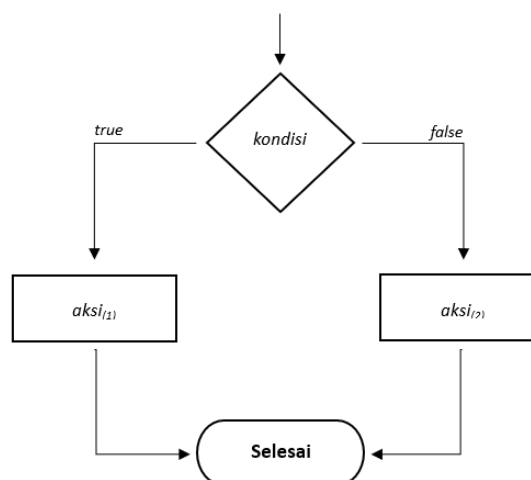
If kondisi then

$aksi_{(1)}$

else

$aksi_{(2)}$

End if



Flowchart diatas dapat memvisualisasikan percabangan dengan dua kasus. Jika kondisi benar, maka aliran intruksi bercabang pada $aksi_{(1)}$, tetapi jika kondisi salah maka aliran intruksi akan bercabang pada $aksi_{(2)}$

Contoh

Buatlah algoritma yang menerima inputan bilangan bulat, jika bilangan yang di inputkan adalah genap maka akan muncul tulisan “Bilangan Genap”, jika bilangan yang di inputkan ganjil maka akan muncul tulisan “Bilangan Ganjil”.

Penyelesaian :

Bilangan genap adalah bilangan yang habis dibagi 2, maka dari itu kita bisa menggunakan operator modulus (sisanya bagi) dalam menyelesaikan persoalan diatas, dengan demikian jika bilangan yang di-inputkan bila dimodulus dengan 2 sama dengan 0 maka cetak tulisan “Bilangan Genap” dan jika tidak maka cetak tulisan “Bilangan Ganjil”.

PROGRAM BilanganGenapGanjil

{Mencetak tulisan “Bilangan Genap” jika bilangan yang di-inputkan adalah bilangan genap dan mencetak tulisan “Bilangan Ganjil” jika bilangan yang di-inputkan adalah bilangan ganjil}

DEKLARASI :

 bil : **integer**

DESKRIPSI :

read(bil)

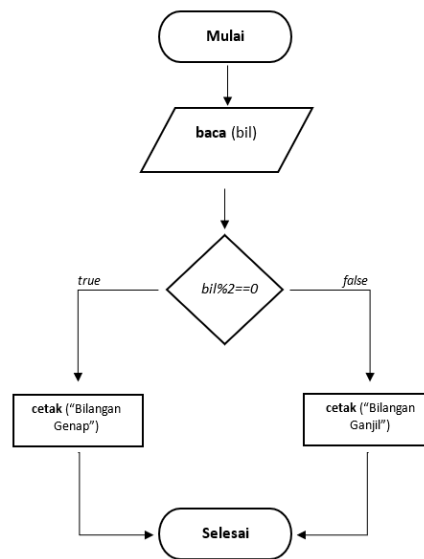
if bil **mod** 2 == 0 **then**

write(“Bilangan Genap”)

else

write(“Bilangan Ganjil”)

end if



Berikut adalah contoh penyelesaian dari bahasa python

```

1  bil = int(input("Masukan Bilangan : "))
2
3  if bil %2 == 0 :
4      print("Bilangan Genap")
5  else :
6      print("Bilangan Ganjil")
  
```

```

PS C:\Users\ACER\OneDrive\Desktop> py.exe .\main.py
Masukan Bilangan : 2
Bilangan Genap
PS C:\Users\ACER\OneDrive\Desktop>
  
```

```

PS C:\Users\ACER\OneDrive\Desktop> py.exe .\main.py
Masukan Bilangan : 5
Bilangan Ganjil
  
```

D. Algoritma Percabangan dengan Tiga Kasus atau Lebih