



- Dasar Pemrograman – Pertemuan 5

Tim Bahan Ajar Dasar Pemrograman
Teknik Informatika - S1
Fakultas Ilmu Komputer



Review Materi





Capaian Kuliah Pertemuan 5



Capaian Pembelajaran

- Setelah mengikuti matakuliah ini mahasiswa dapat menjelaskan, membuat, dan mempraktikkan analisis dua kasus komplemen diterapkan pada aksi sekuensial permasalahan komputasional sederhana.



ANALISIS DUA KASUS KOMPLEMENTER



Analisis Dua Kasus Komplementer

- Pengecekan terhadap satu kondisi yang memenuhi syarat.
 - Jika benar lakukan aksi, jika salah lakukan aksi lainnya
- Menggunakan keyword **if ... then else ...**

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



Perhatikan
indentasi !

Contoh Analisa Dua Kondisi

- Algoritma untuk mengecek bilangan genap / ganjil dari masukan **x**

Program GenapGanjil

{Program untuk mengecek apakah a bilangan genap atau ganjil}

KAMUS

x : integer

ALGORITMA

input(x)

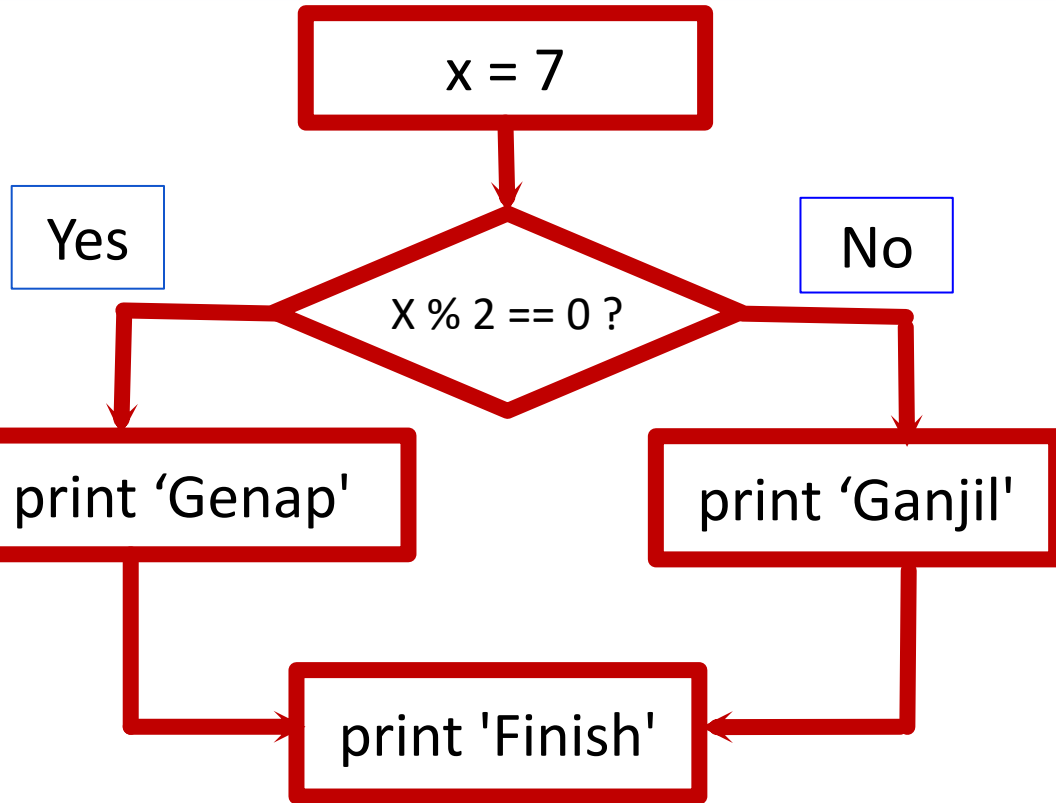
if x mod 2 eq 0 then

output("genap")

else

output("ganjil")

Contoh Kasus Analisa Dua Kondisi



Skenario
input:
 $x = 7$

Output:
Ganjil
Finish



Notasi Algoritmik untuk Aksi Sekuensial yang memanfaatkan Analisis dua kasus komplemen



Program MaxAB

- Spesifikasi:

Input: a dan b bilangan bulat

Proses: menuliskan nilai yang lebih besar dari kedua bilangan tersebut
cetak a jika $a \geq b$, selain itu cetak b (jika $a < b$)

Output: a atau b

Program MaxAB

Program MaxAB

{Program untuk mencetak bilangan dengan nilai maksimal dua bilangan bulat a dan b}

KAMUS

a , b : integer

ALGORITMA

input(a)

input(b)

if a >= b then

output(a)

else

output(b)

endif

Pertanyaan dan Jawaban

- Apakah bisa ekspresi boolean yang ada di kondisi analisis dua kasus komplement itu dengan operator and or atau ekspresi yang panjang?
 - Bisa
- Kapan analisis dua kasus komplemen ini sering digunakan?
 - Sering sekali, biasanya untuk kasus pengecekan dengan pilihan nilai jika kondisi terpenuhi dan yang lain tidak terpenuhi
- Apakah boleh menggunakan analisis satu kasus dan analisis dua kasus komplemen dalam suatu rangkaian program?
 - Tentu boleh

Kasus Komputasional

- Deskripsi: Mawar suka sekali dengan permainan bilangan. Dia ingin memastikan bilangan bulat yang dia dapat adalah bilangan yang absolut (tidak boleh bernilai negatif) kemudian dia juga ingin memastikan bahwa bilangannya itu merupakan bilangan kelipatan 5. Jika bilangan tadi kelipatan 5 maka dia ingin mencetak “hore ketemu bilangan kelipatan 5”, jika tidak dia ingin mencetak “yah.. bukan bilangan kelipatan 5”. Buatlah program untuk mawar.
- Input: a merupakan bilangan bulat boleh positif atau negatif
- Output: tulisan “hore ketemu bilangan kelipatan 5” atau “yah.. bukan bilangan kelipatan 5”
- Contoh Input-Output:
 - Input: -10
 - Output: hore ketemu bilangan kelipatan 5

Penyelesaian

1. Siapkan variabel a bertipe integer untuk menampung input user
2. Lakukan pengecekan absolut, jika ternyata yang di input user adalah bilangan negatif lakukan:
 2. 1 Rubah nilai a menjadi bernilai positif dengan cara $a \leftarrow a * -1$
3. Jika ternyata yang diinputkan oleh user sudah bilangan positif maka langkah 2 tidak perlu dilakukan
4. Lakukan pengecekan komplementer jika a habis dibagi 5 maka outputkan “hore ketemu bilangan kelipatan 5”
5. Jika tidak, outputkan “yah.. bukan bilangan kelipatan 5”

Program Mawar5

Program Mawar5

{Program untuk penyelesaian kasus kelipatan 5 si Mawar}

KAMUS

a : integer

ALGORITMA

input(a)

if a < 0 then

 a \leftarrow a * -1

endif

if a mod 5 eq 0 then

output("hore ketemu bilangan kelipatan 5")

else

output("yah.. bukan bilangan kelipatan 5")

endif

Referensi



Utama :

1. Liem, Inggriani. Diktat Pemrograman Prosedural Informatika ITB. IF-ITB. 2007
2. Bjarne Stroustrup, 2014, Programming: Principles and Practice Using C++ (Second Edition), Addison-Wesley Professional

Pendukung :

1. Introduction to Computer Science and Programming in Python, MIT
<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016>
2. Introduction to Computer Science and Programming, MIT <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00sc-introduction-to-computer-science-and-programming-spring-2011/index.htm>