

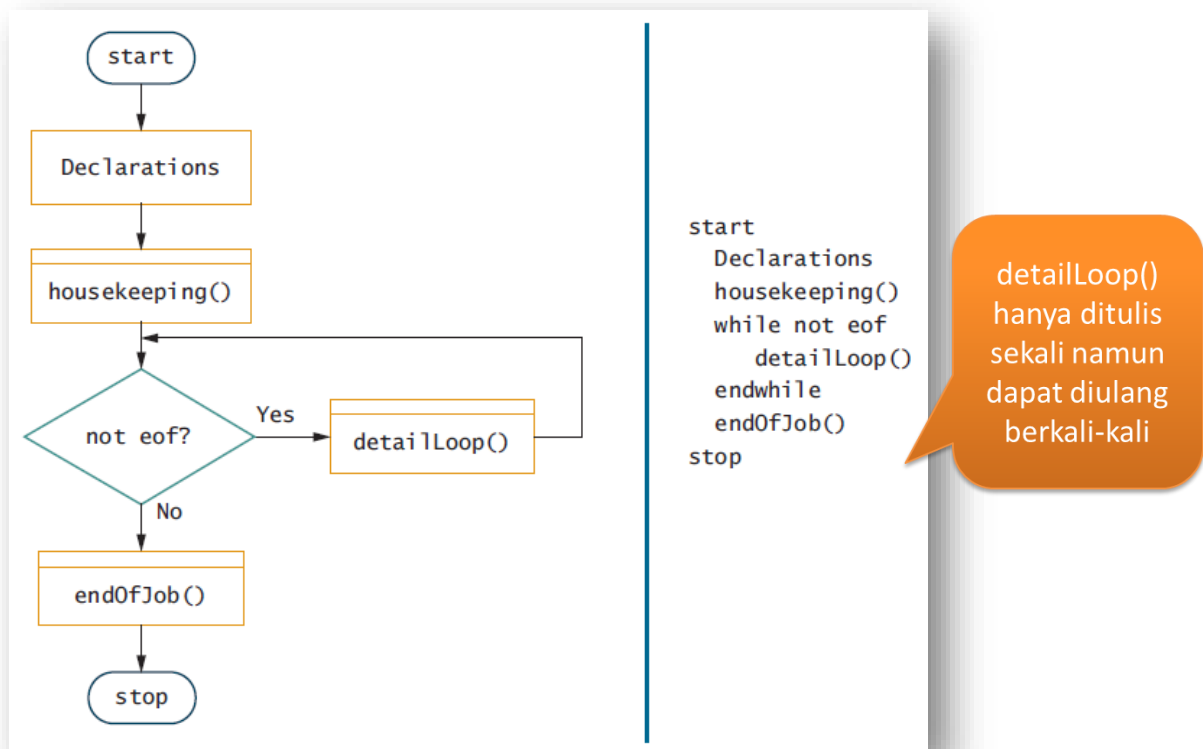
<b>Mata Kuliah</b>	<b>:</b>	<b>Dasar Pemrograman</b>
<b>Bobot Sks</b>	<b>:</b>	<b>2</b>
<b>Dosen Pengembang</b>	<b>:</b>	<b>Riad Sahara, S.SI, M.T Syahid Abdullah, S.Si, M.Kom</b>
<b>Tutor</b>	<b>:</b>	<b>Syahid Abdullah, S.Si, M.Kom</b>
<b>Capaian Pembelajaran Mata Kuliah</b>	<b>:</b>	<b>1. Mahasiswa memahami struktur kendali perulangan 2. Mahasiswa menggunakan perulangan FOR 3. Mahasiswa memahami Nested Loop (Perulangan Bersarang)</b>
<b>Kompetensi Akhir di Setiap Tahap (Sub- Cpmk)</b>	<b>:</b>	<b>1. Memahami Struktur Kendali Proses Percabangan</b>
<b>Minggu Perkuliahan Online Ke-</b>	<b>:</b>	<b>11</b>

## **JUDUL TOPIK – Struktur Kendali Proses Perulangan**

### **Perulangan**

#### **Manfaat Struktur Kendali Perulangan**

- Struktur kendali perulangan membuat program menjadi lebih efisien
- Serangkaian instruksi hanya ditulis sekali namun dapat diulang tanpa batas



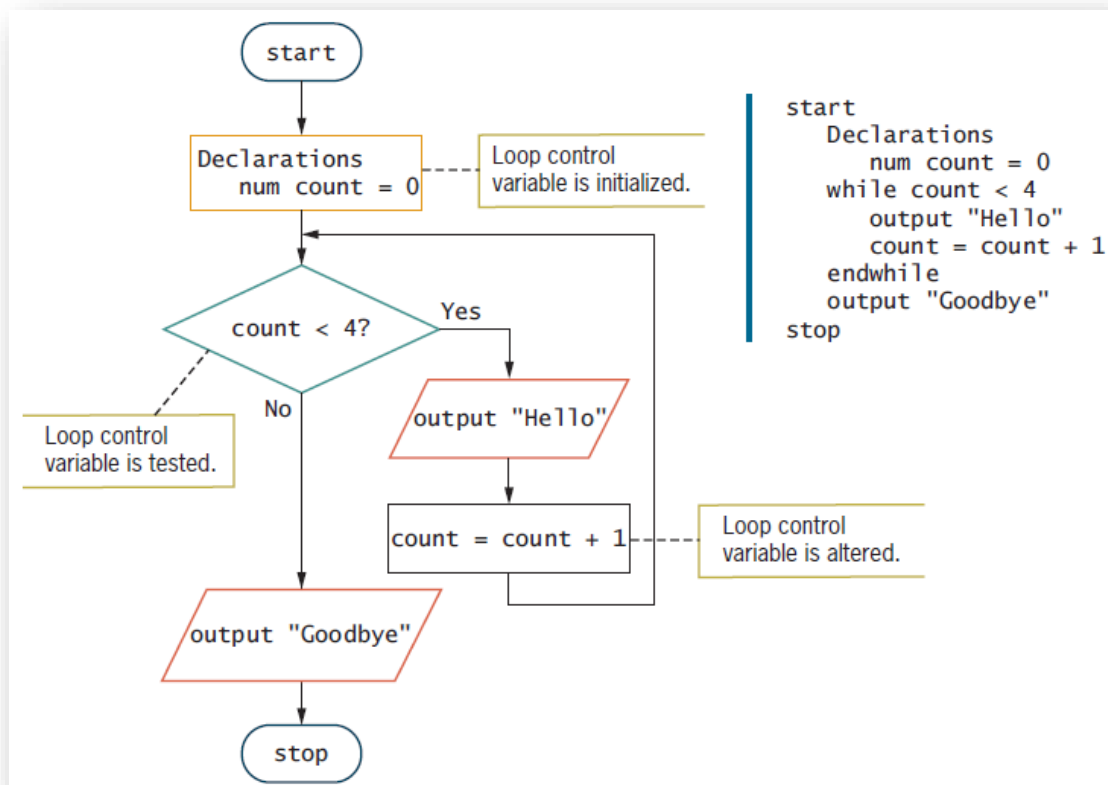
## Menggunakan Variabel Kendali Perulangan

- WHILE LOOP digunakan untuk mengeksekusi serangkaian instruksi yang sama selama kondisi tetap bernilai TRUE
- Agar WHILE LOOP berakhir dengan benar, hendaknya ada suatu variabel yang mengendalikan eksekusi loop
  - Variabel kendali perulangan diinisiasi sebelum masuk loop
  - Variabel kendali perulangan diuji, jika hasil TRUE maka masuk loop
  - Di dalam loop harus ada tindakan untuk mengubah nilai dari variabel kendali perulangan
- Jika menggunakan pengulangan, jumlah pengulangan harus dikendalikan
- Jika tidak dikendalikan → infinite loop
- Cara untuk mengendalikan pengulangan:
  - Menggunakan counter (counter-controlled loop) → definite loop
  - Menggunakan nilai penjaga (sentinel value) → indefinite loop

## Menggunakan Counter dalam Definite Loop

- Definite loop → jumlah pengulangan sudah ditentukan sebelumnya
- Dalam definite loop, ada variabel kendali pengulangan yang berfungsi untuk menghitung jumlah pengulangan
- Counter dapat dilakukan secara naik (incrementing) dan turun (decrementing)
  - Incrementing →  $\text{counter} = \text{counter} + 1$
  - Decrementing →  $\text{counter} = \text{counter} - 1$

### Contoh Definite Loop



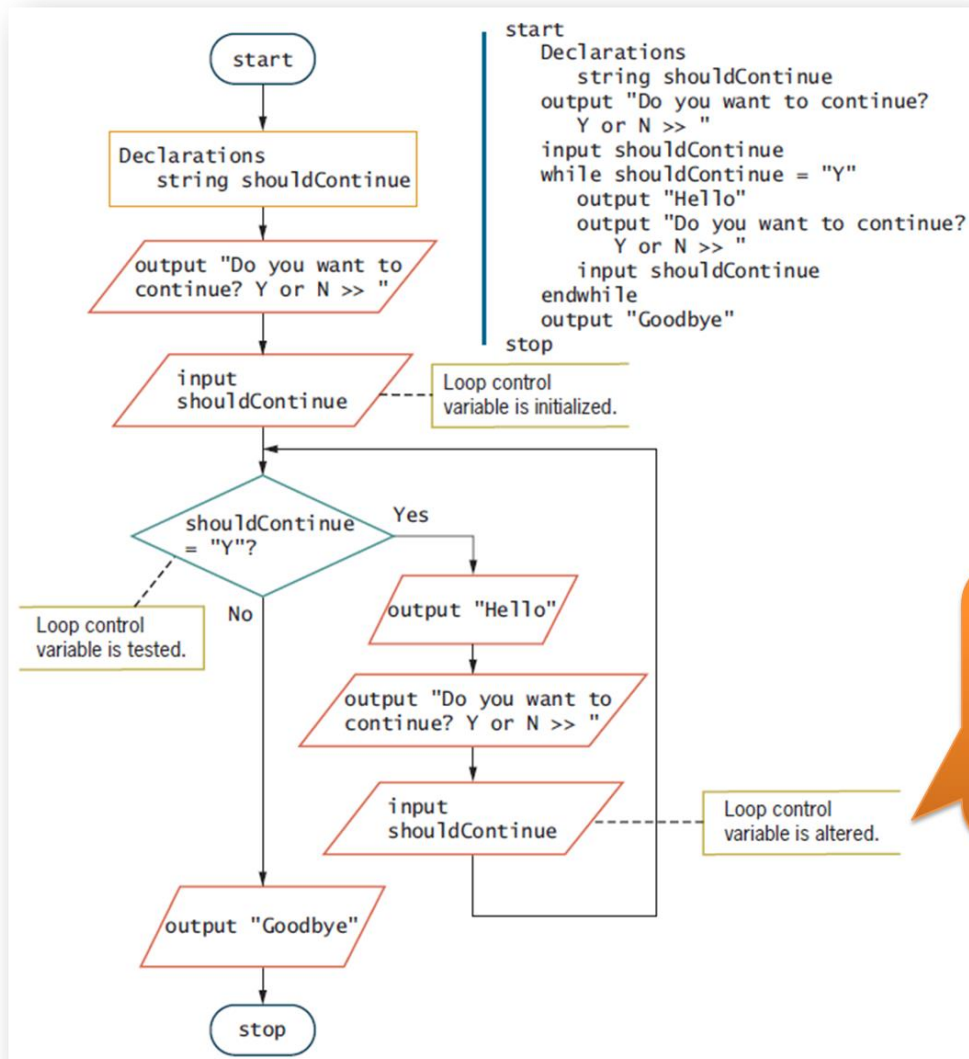
Tabel Penelusuran

METHOD	Line	count
num count = 0	2	0
while count < 4	3	0 (TRUE)
output "Hello"	4 output "Hello"	0

count = count + 1	5	1
while count < 4	3	1 (TRUE)
output "Hello"	4 output "Hello"	1
count = count + 1	5	2
while count < 4	3	2 (TRUE)
output "Hello"	4 output "Hello"	2
count = count + 1	5	3
while count < 4	3	3 (TRUE)
output "Hello"	4 output "Hello"	3
count = count + 1	5	4
while count < 4	3	4 (FALSE)
output "Goodbye"	7 output "Goodbye"	4

### Menggunakan Nilai Penjaga dalam Indefinite Loop

- Indefinite loop → jumlah pengulangan tidak dapat ditentukan / berbeda setiap kali eksekusi
- Biasanya banyaknya pengulangan tidak ditentukan oleh jumlah aritmatika, namun oleh input dari pengguna → jumlah pengulangan tidak dapat ditentukan secara pasti
- Dibutuhkan nilai penjaga untuk menghentikan pengulangan



Tabel Penelusuran

METHOD	Line	shouldContinue
string shouldContinue	2	
output "Do you want to continue? Y or N >> "	3	
input shouldContinue	4	Y

while shouldContinue = "Y"	5	Y (TRUE)
output "Hello"	6 output "Hello"	Y
output "Do you want to continue? Y or N >> "	7	Y
input shouldContinue	8	Y
output "Hello"	6 output "Hello"	Y
output "Do you want to continue? Y or N >> "	7	Y
input shouldContinue	8	N
while shouldContinue = "Y"	5	N (FALSE)
output "Goodbye"	10 output "Goodbye"	N

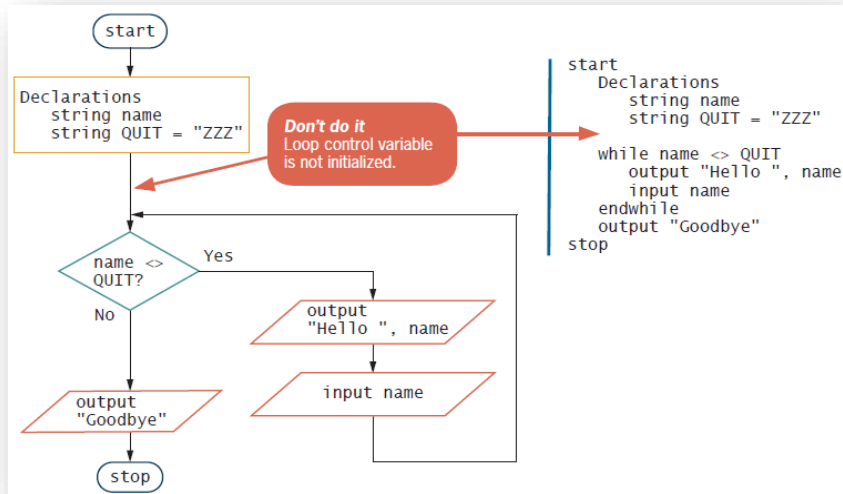
### Nested Loop (Perulangan Bersarang)

- Nested loop → di dalam struktur perulangan, ada perulangan
- Inner loop → struktur perulangan pada bagian dalam
- Outer loop → struktur perulangan pada luar
- Perhatikan variabel kendali perulangan pada setiap struktur loop untuk menghindari infinite loop

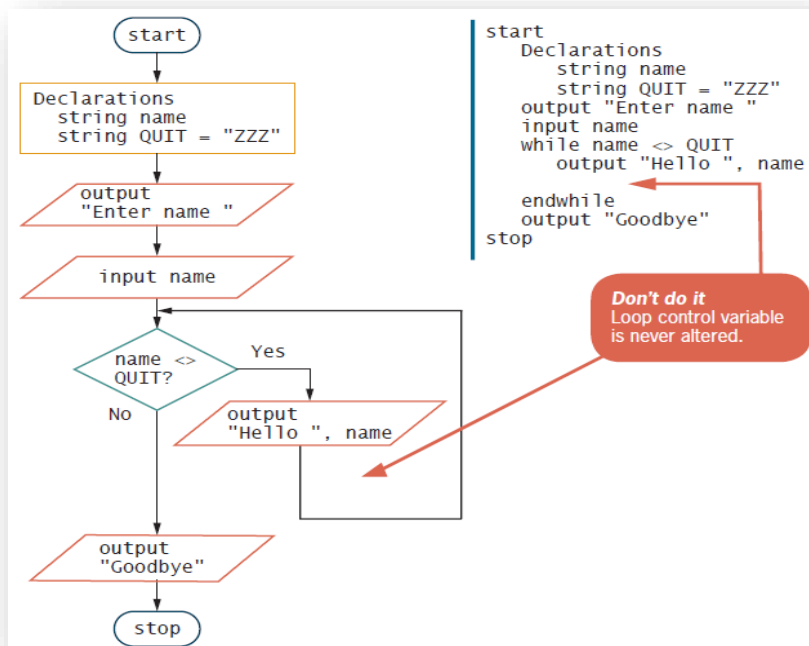
### Kesalahan dalam Perulangan

- Tidak menginisiasi variabel kendali perulangan
- Tidak ada perubahan nilai variabel kendali perulangan
- Menggunakan operator perbandingan yang salah pada variabel kendali perubahan

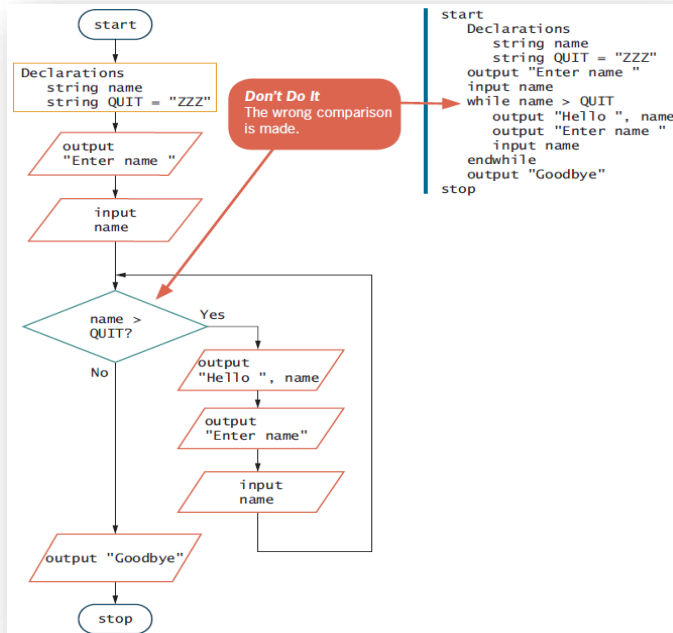
**Kesalahan 1: Tidak ada inisiasi variabel kendali perulangan**



**Kesalahan 2: Tidak ada perubahan nilai variabel kendali pengulangan**



### Kesalahan 3: Menggunakan operator perbandingan yang salah



### Menggunakan Perulangan FOR

- FOR digunakan pada definite loop jika diketahui secara pasti berapa kali loop yang dibutuhkan
- Variabel kendali pengulangan:
  - Inisiasi variabel
  - Evaluasi variabel
  - Perubahan nilai variabel (incrementing/decrementing)

### Perbandingan WHILE dan FOR

```
count = 0
while count <= 3
  output "Hello"
  count = count + 1
endwhile
```

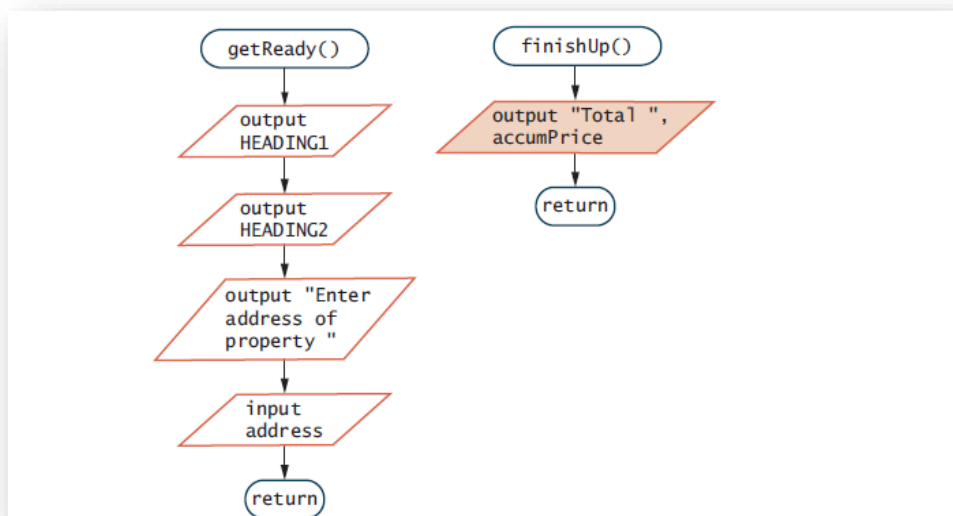
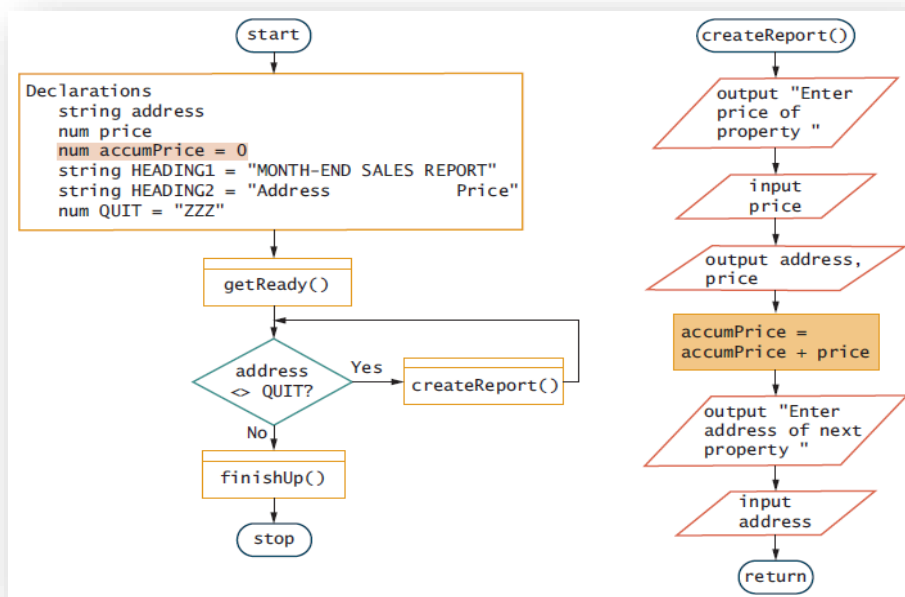
```
for count = 0 to 3 step 1
  output "Hello"
endfor
```



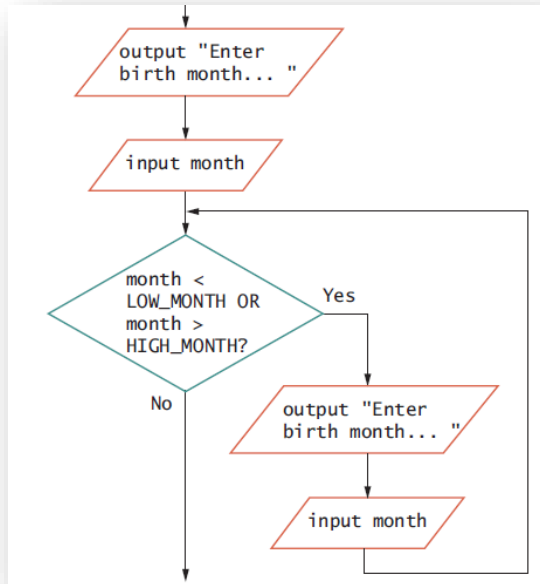
- FOR memiliki fungsi yang sama seperti WHILE:
  - Ada inisiasi variabel kondisi pengulangan
  - Ada evaluasi variabel kondisi pengulangan
  - Ada perubahan nilai variabel kondisi pengulangan

## Contoh Penggunaan Struktur Pengulangan

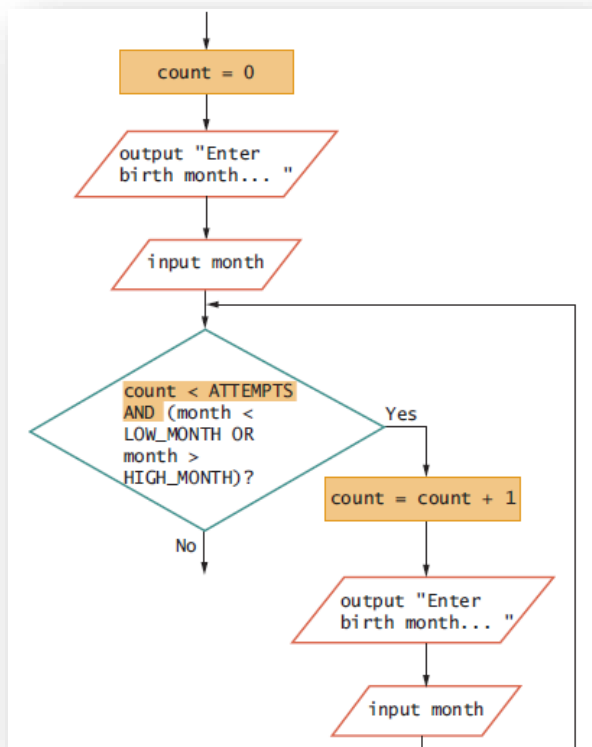
Menghitung total



## Validasi Data



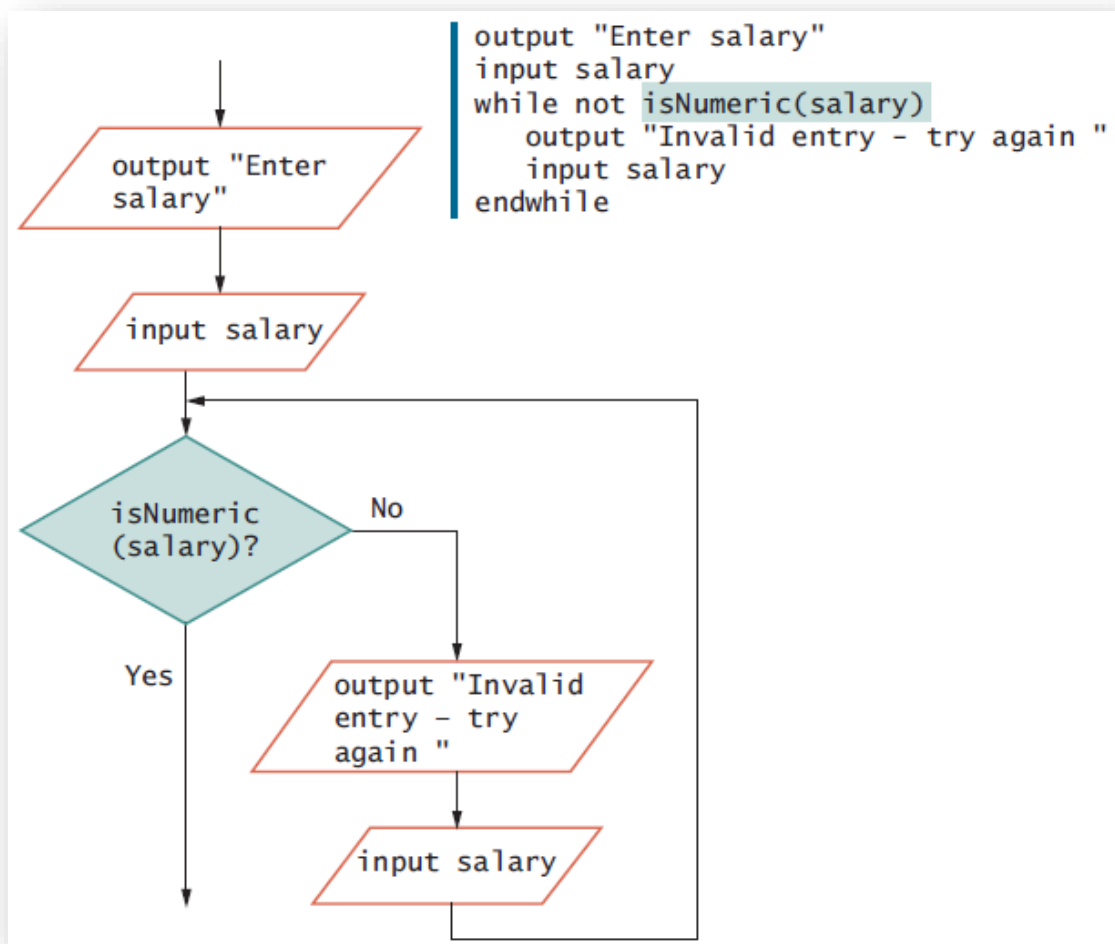
## Mendorong pembatasan pengulangan



**Significant declarations:**

```
num month  
num HIGH_MONTH = 12  
num LOW_MONTH = 1  
num count  
num ATTEMPTS = 3
```

Validasi tipe data



---

## Daftar Pustaka

Goodrich, Michael, Roberto Tamassia, and David Mount. *Data structures and algorithms in C++*. John Wiley & Sons, 2011.

Mehlhorn, Kurt, and Peter Sanders. *Algorithms and data structures: The basic toolbox*. Springer, 2010.