

Mata Kuliah	:	Dasar Pemrograman
Bobot Sks	:	2
Dosen Pengembang	:	Riad Sahara, S.Si, M.T Syahid Abdullah, S.Si, M.Kom
Tutor	:	Syahid Abdullah, S.Si, M.Kom
Capaian Pembelajaran Mata Kuliah	:	1. Mahasiswa mampu memahami Struktur dan Strukturisasi Program 2. Mahasiswa mampu memahami Struktur Kendali Proses Percabangan 3. Mahasiswa mampu memahami Struktur Kendali Proses Perulangan 4. Mahasiswa mampu memahami Teknik Modularisasi 5. Mahasiswa mampu memahami Konsep dan Penggunaan Fungsi
Kompetensi Akhir di Setiap Tahap (Sub- Cpmk)		1. Review Materi 9-14
Minggu Perkuliahan Online Ke-		15

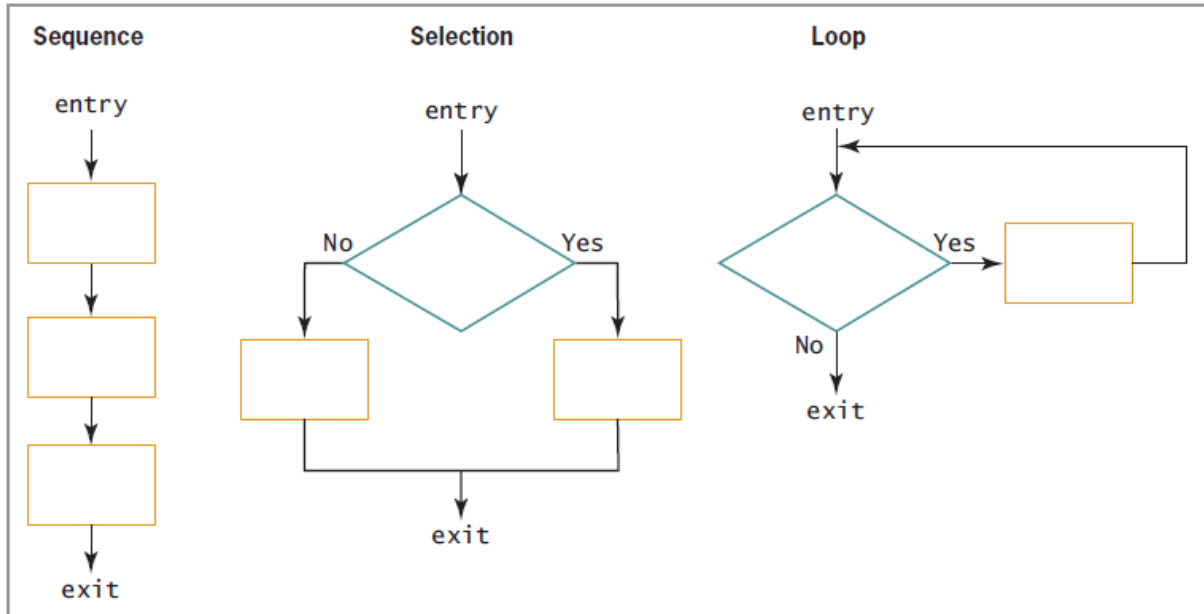
JUDUL TOPIK – Review

Review Struktur Kendali

Tiga Struktur Dasar

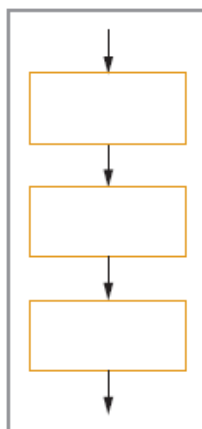
- Struktur merupakan unit dasar dalam pemrograman
- Jenis-jenis struktur dasar:
 - Sequence
 - Selection

– Loop



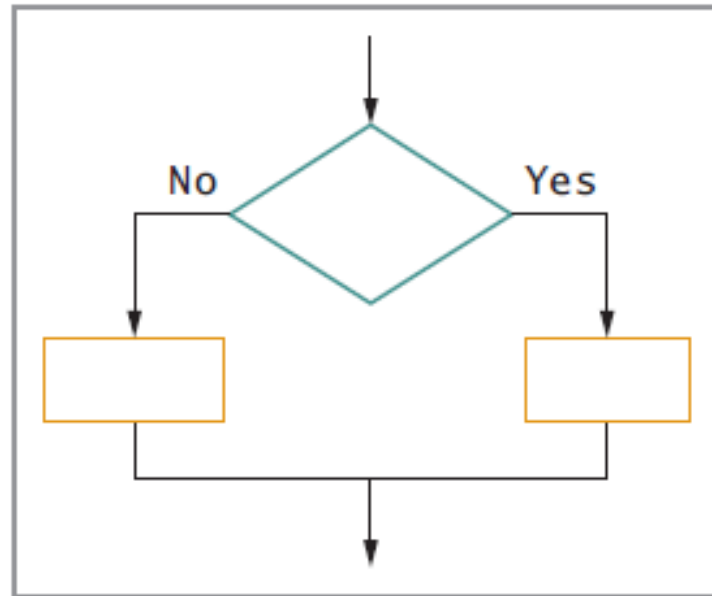
1. Sequence Structure

- Pada sequence structure, serangkaian tindakan dilakukan secara berurutan
- Misal: setelah tindakan A dilanjutkan dengan tindakan B, C, D, dst.



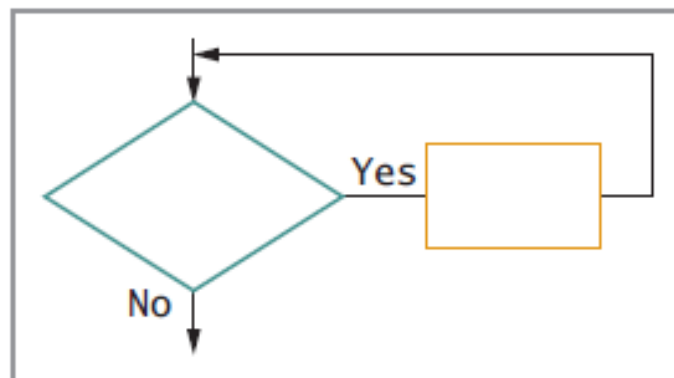
2. Selection Structure

- Pada selection structure, ada suatu decision yang harus dievaluasi.
- Berdasarkan hasil evaluasi, satu dari dua tindakan akan dieksekusi.

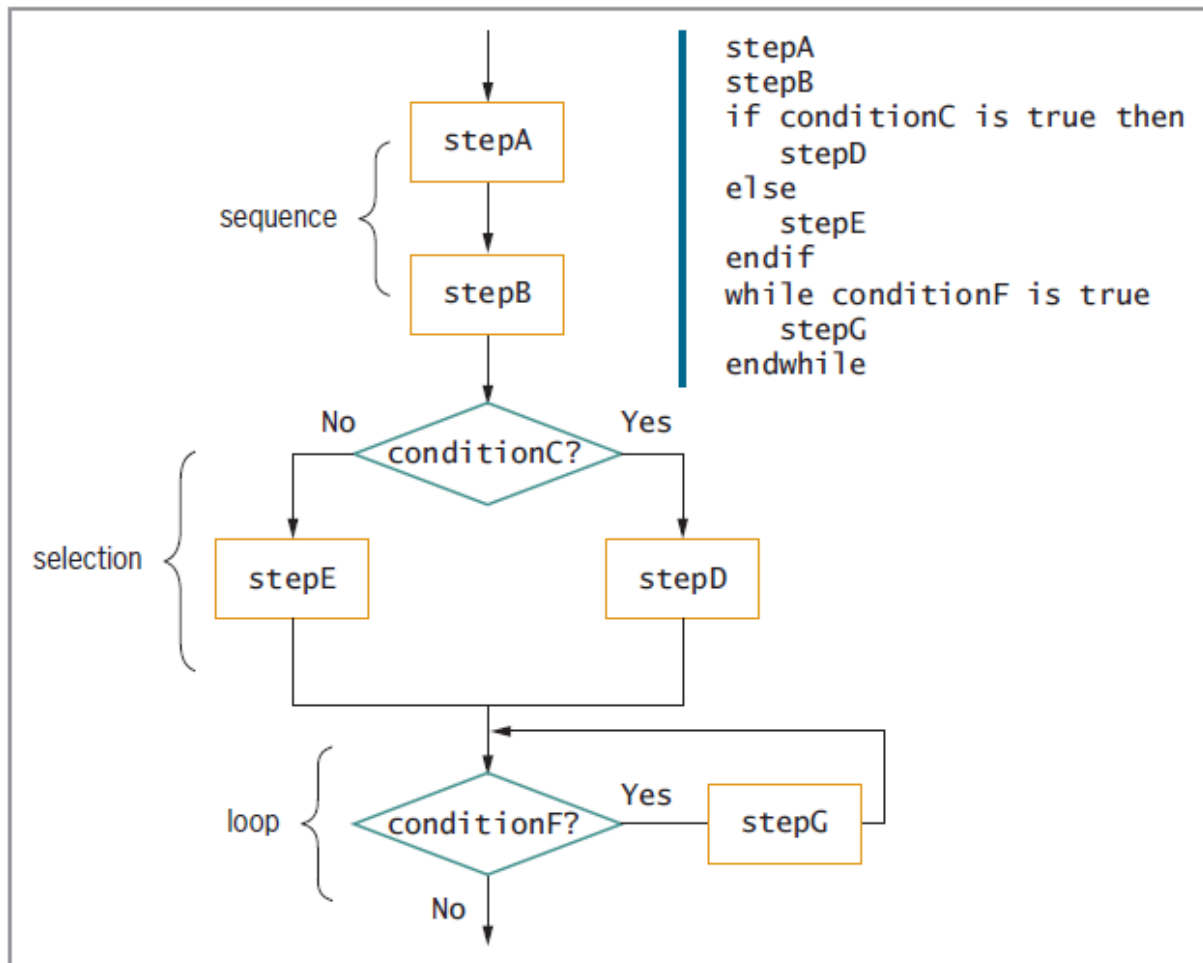


3. Loop Structure

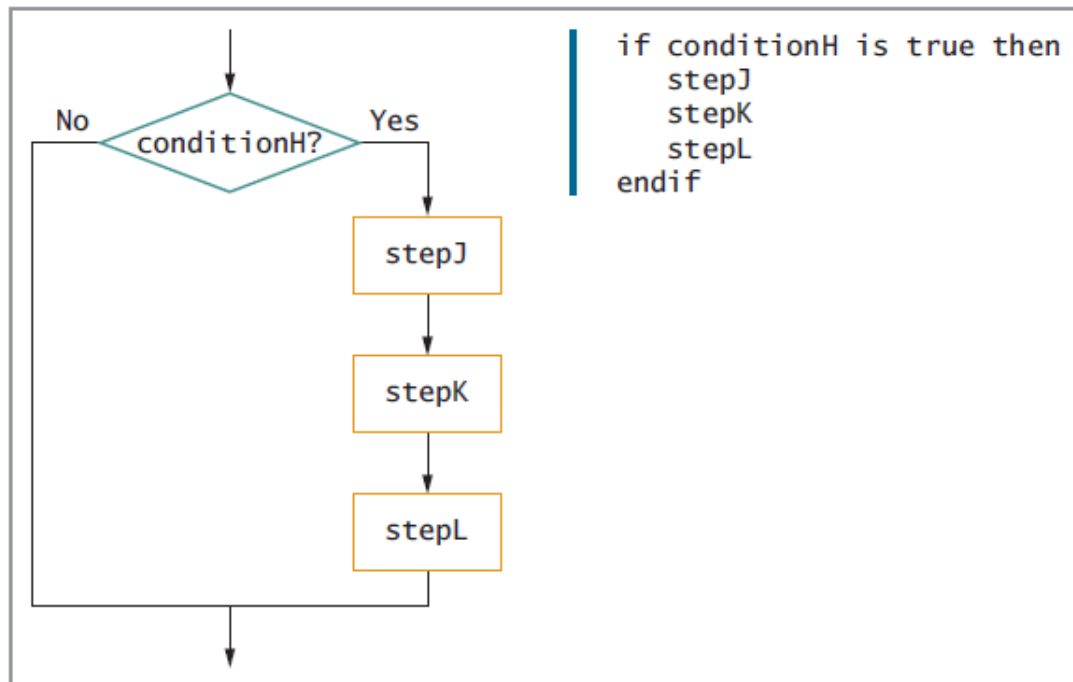
- Pada loop structure, suatu tindakan akan dilakukan secara terus-menerus selama kondisi masih valid



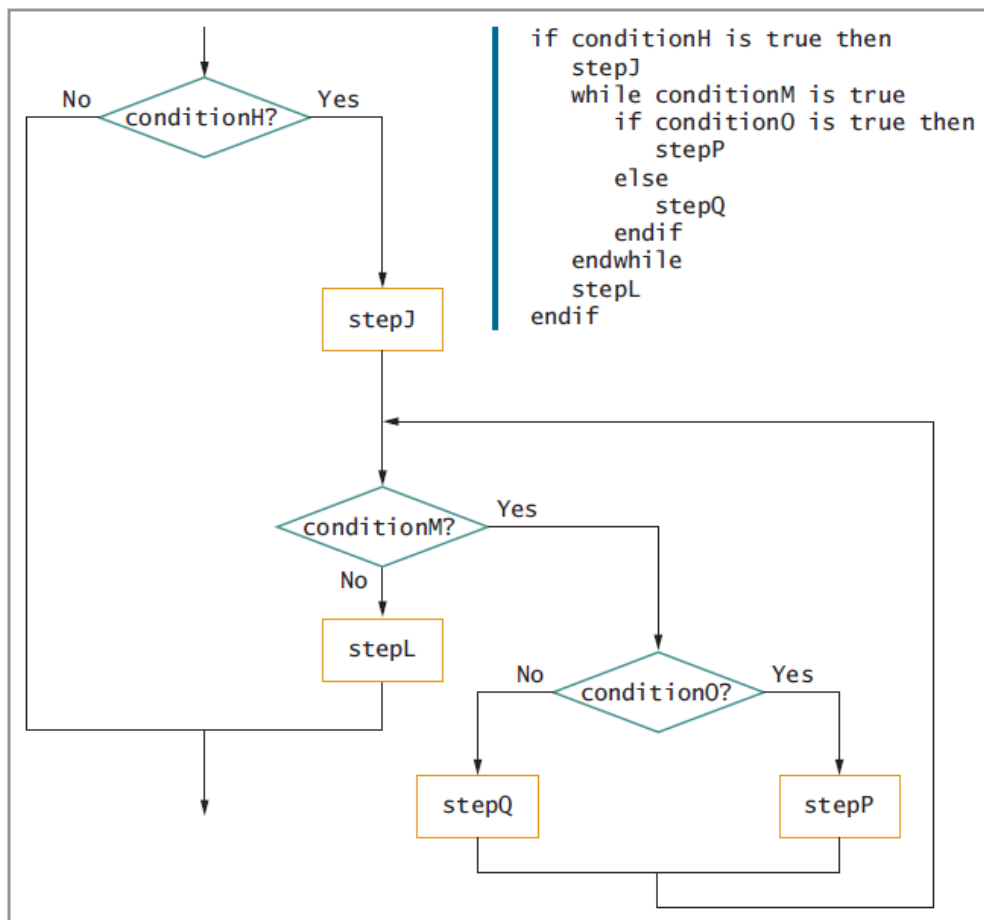
Flowchart dengan struktur kendali bertumpuk (stacked)



Nested structure: sequence nested within a selection



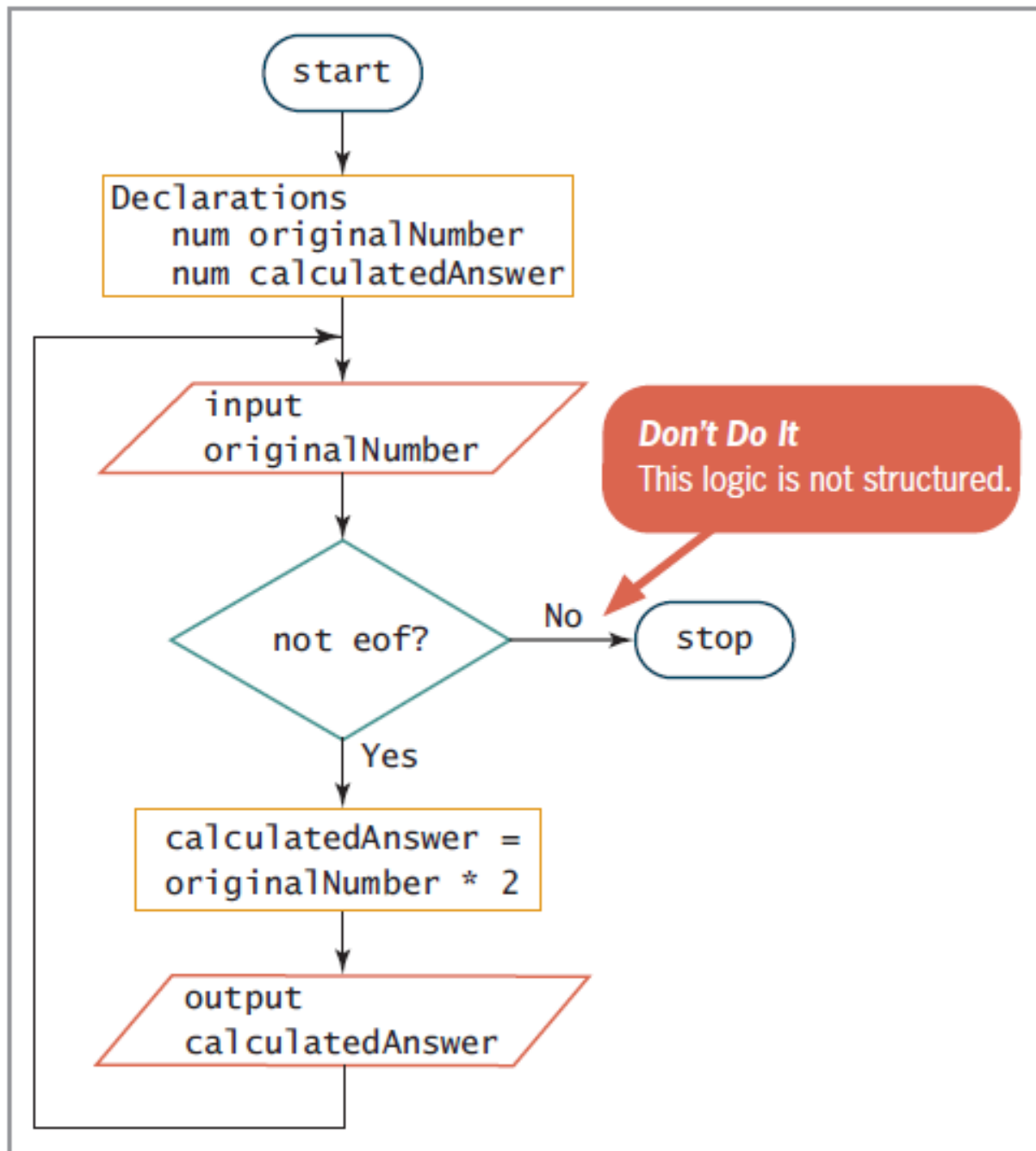
Nested structure: a loop nested within a selection, nested within a sequence, nested within a selection



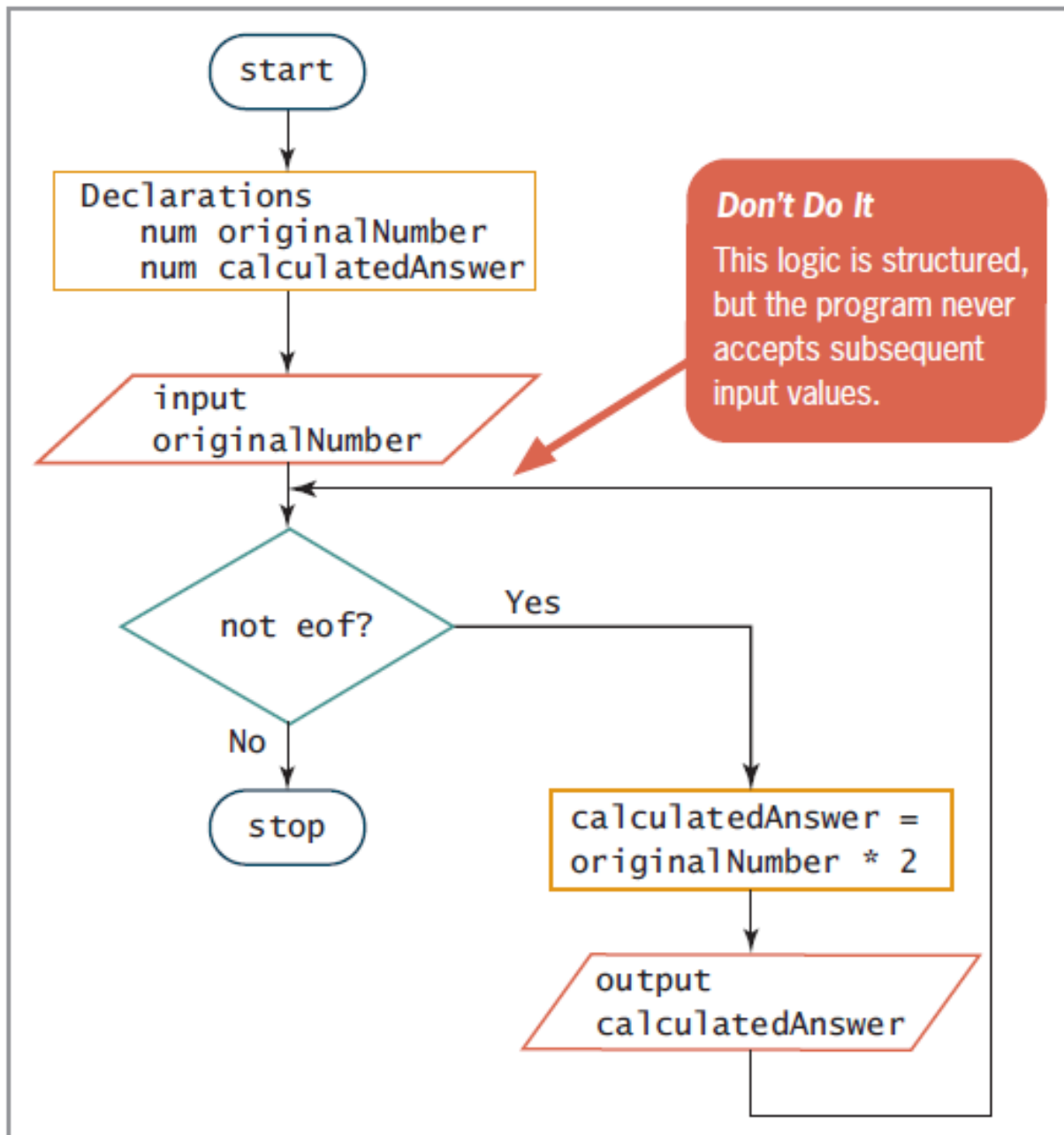
Strukturisasi Program dengan Input Dasar

- Input dasar / “priming input” / “priming feed” → pernyataan yang ditambahkan untuk memperoleh nilai dari input pertama
- Input dasar penting untuk menghasilkan program yang terstruktur dan fungsional (sesuai dengan kebutuhan yang dimaksud)

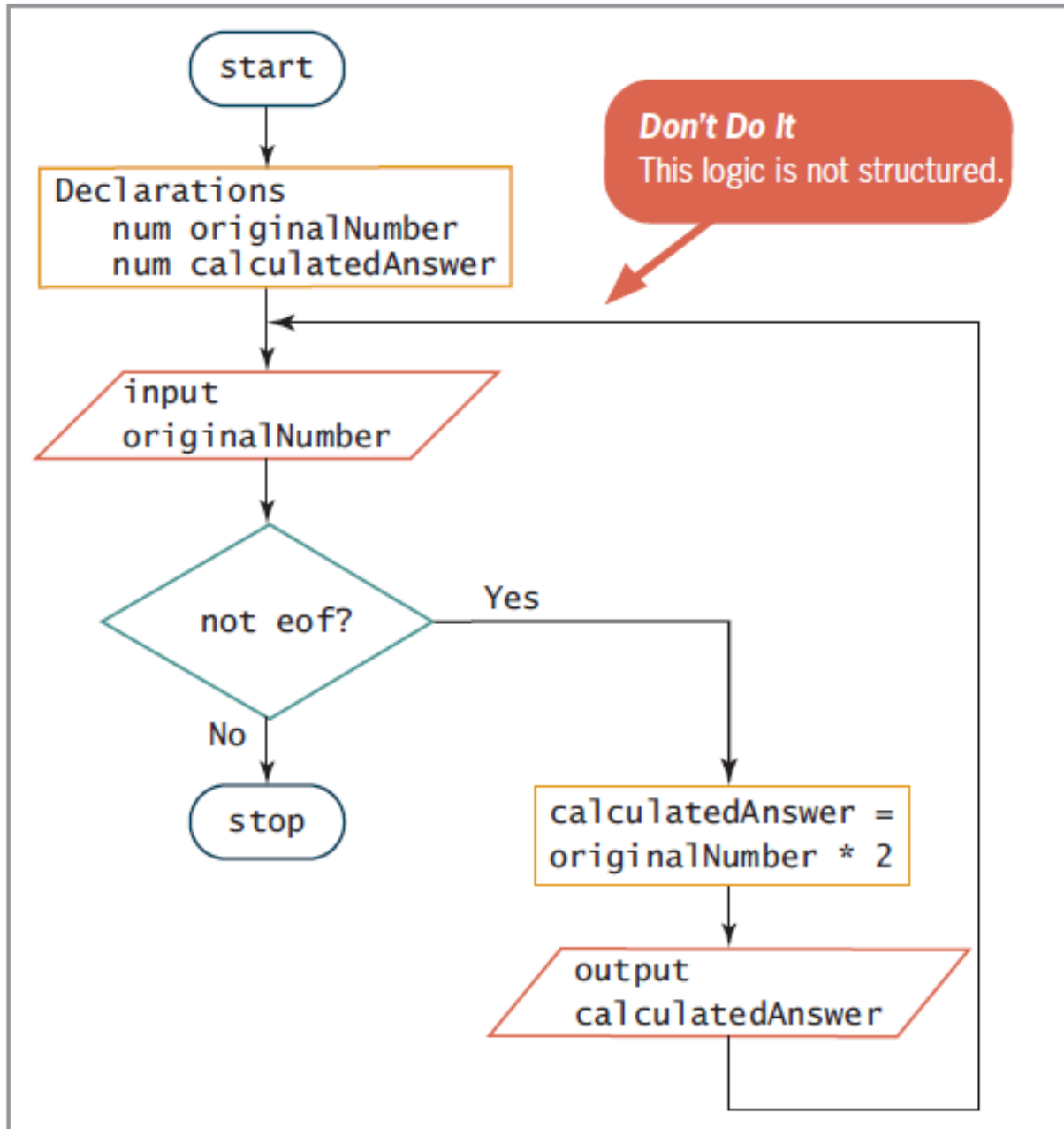
Program tidak terstruktur



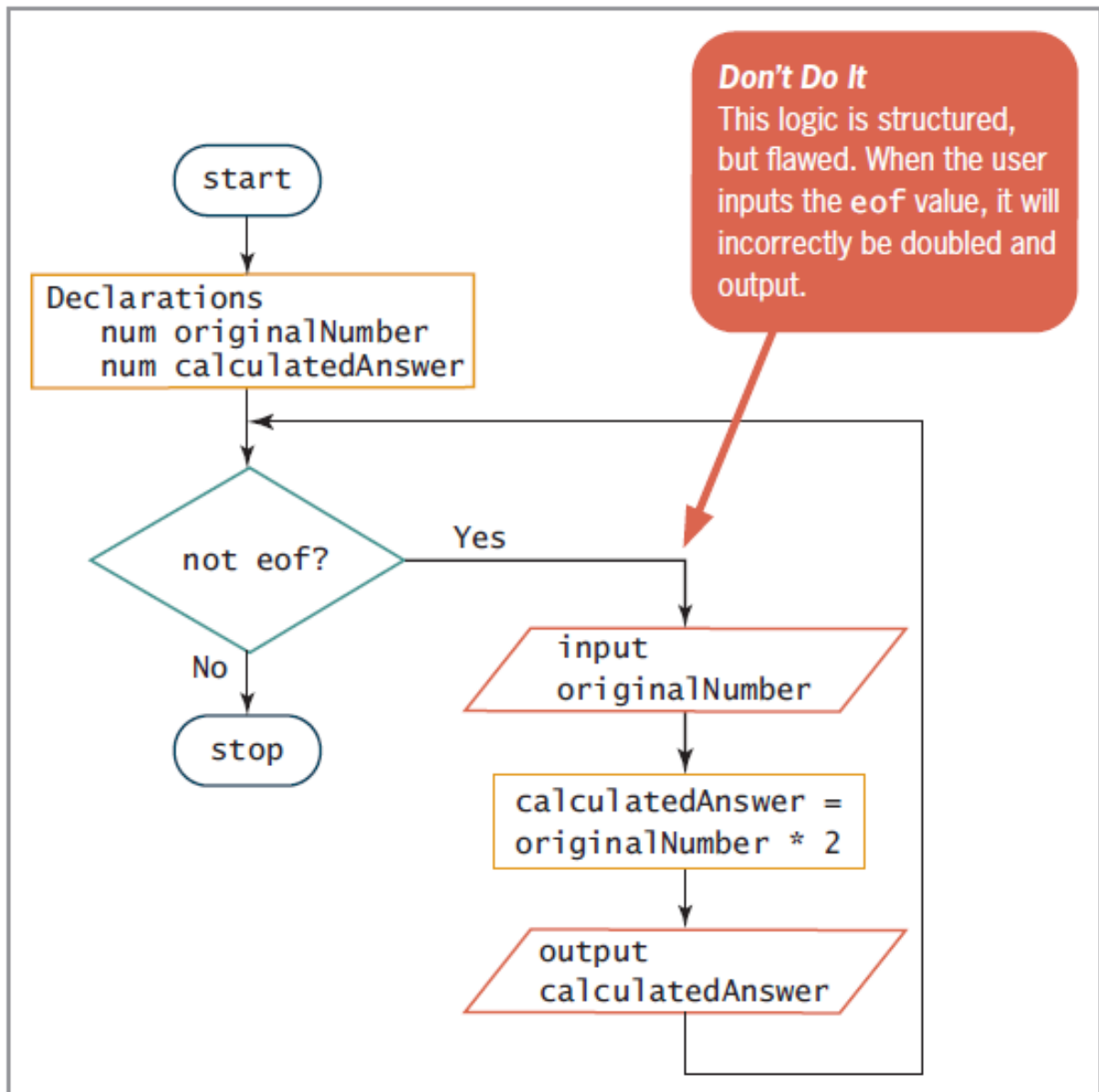
Terstruktur tapi tidak fungsional



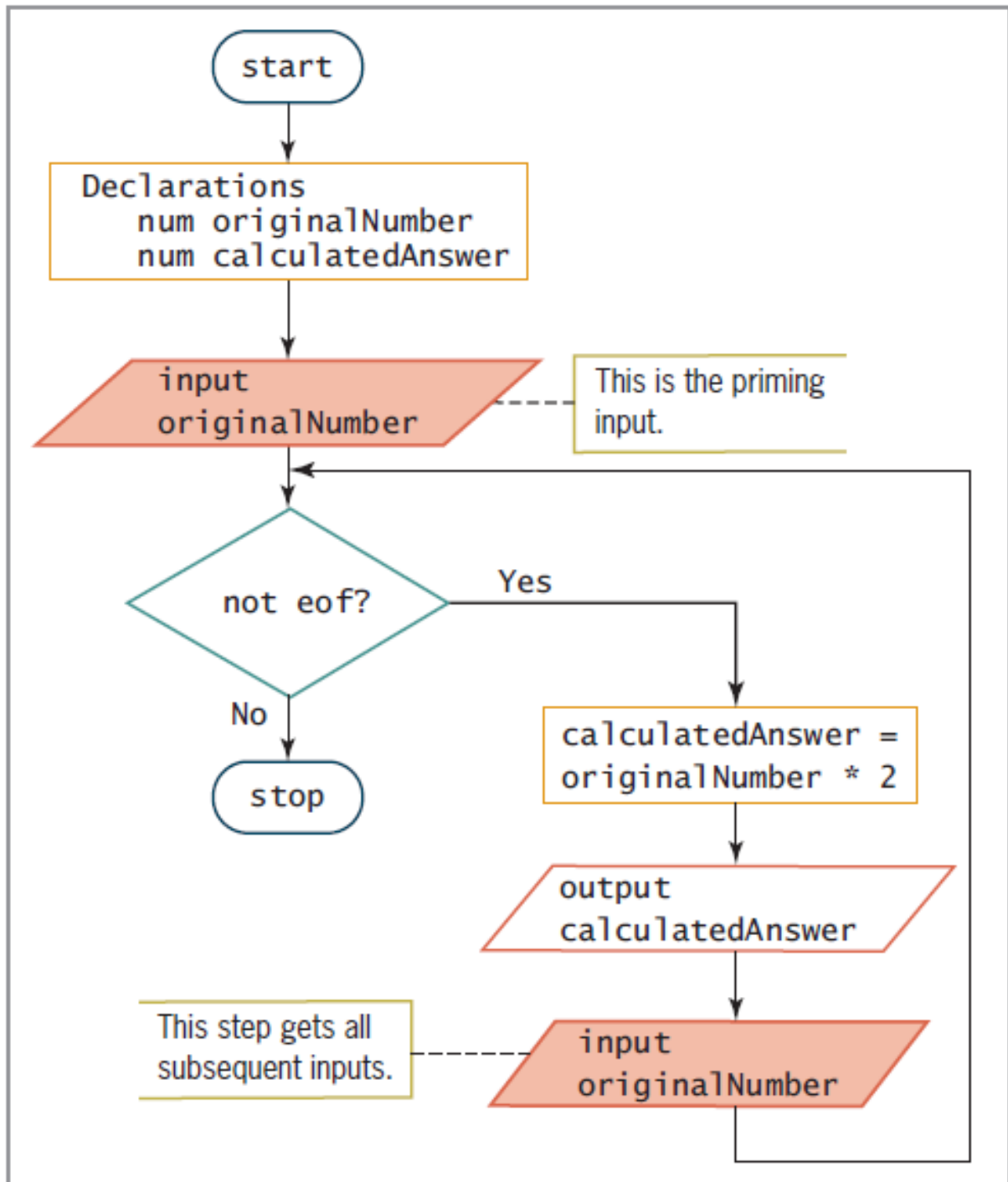
Fungsional, tapi tidak terstruktur



Terstruktur dan fungsional, tapi memiliki kelemahan dalam input



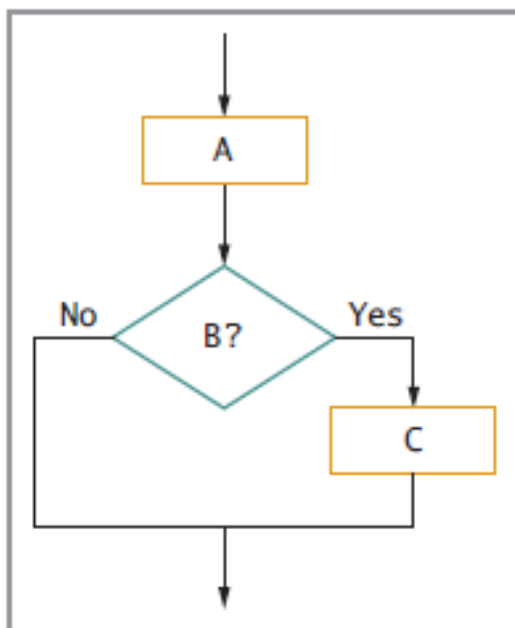
Terstruktur, fungsional sesuai dengan kebutuhan



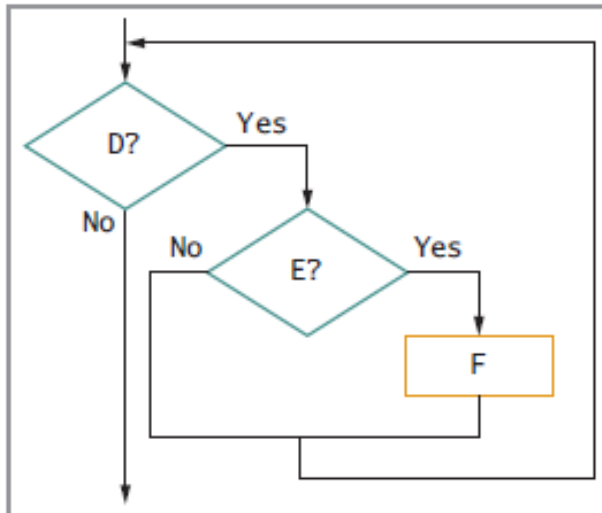
Mengenali Struktur

- Cara untuk membuat program terstruktur adalah dengan mengenali struktur
- Cek apakah program terdiri dari ketiga jenis struktur:
 - Sequence
 - Selection
 - Loop

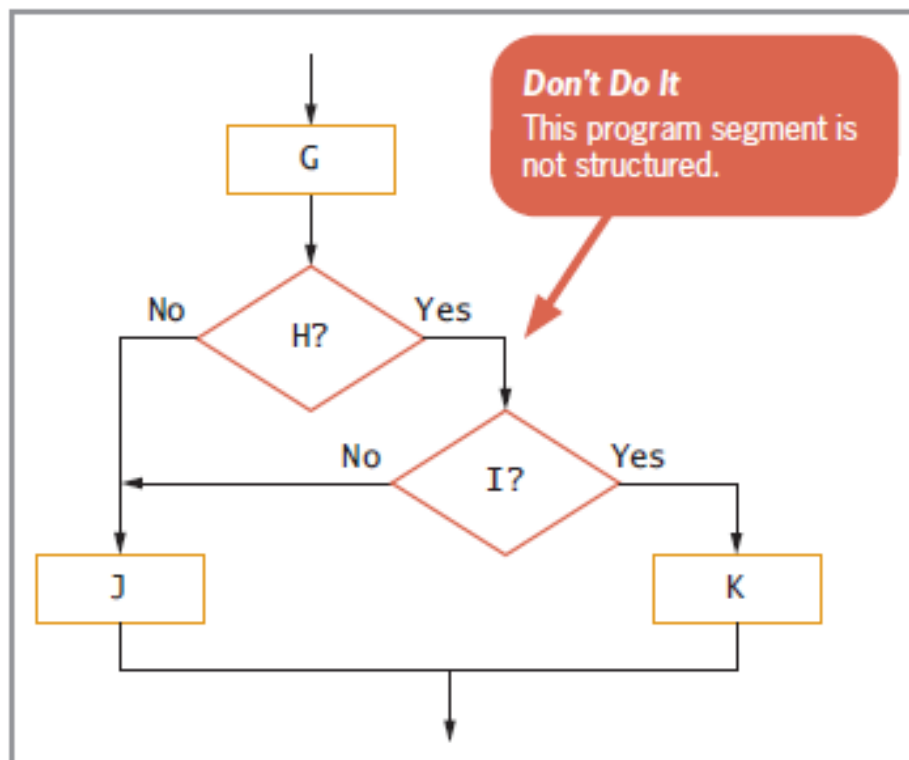
Struktur dengan sequence dan selection



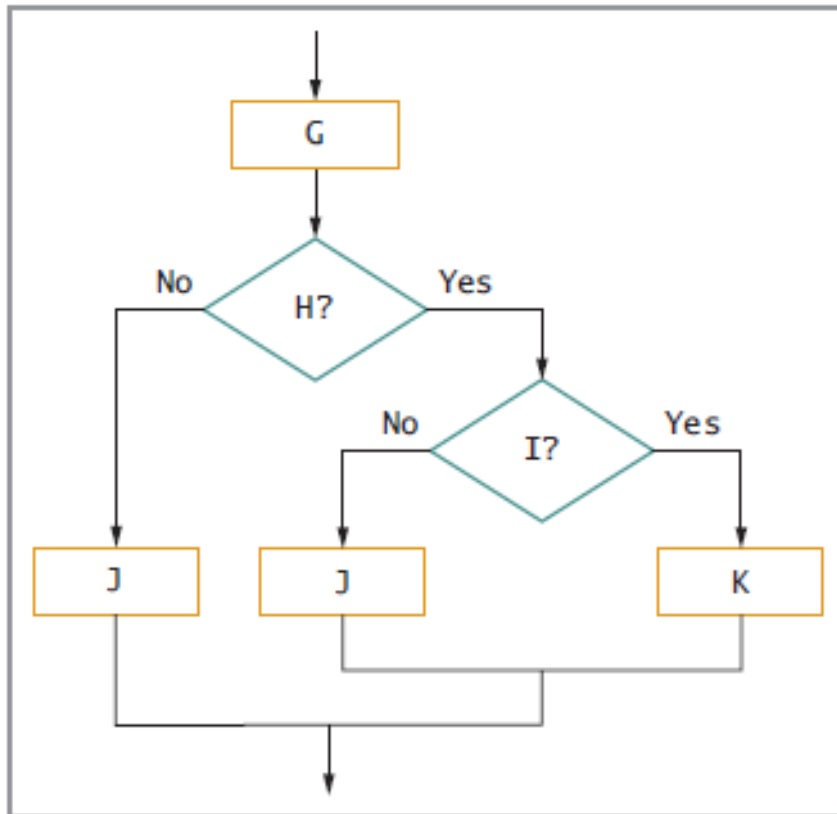
Struktur dengan loop dan di dalam loop ada selection



Program tidak terstruktur



Program terstruktur dengan memisahkan statement J



Memahami Sub Program (Method)

- Method merupakan modul program yang terdiri dari serangkaian pernyataan yang menjalankan suatu tugas
- Method dapat dipanggil dari program atau method lain
- Suatu program dapat memiliki banyak method, dan setiap method dapat dipanggil berkali-kali
- Penamaan method berbeda pada setiap bahasa pemrograman, tapi biasanya mirip dengan penamaan variabel
- Method terdiri dari method header (bagian deklaratif) dan method body. Baris terakhir pada method body merupakan **return statement**
- Variabel dan konstanta yang dideklarasikan dalam method header berlaku secara lokal di dalam method tersebut

Hal Penting Mengenai Method

Terkadang beberapa bagian dari program mengakses data yang sama. Ketika method harus berbagi data, data dapat masuk ke dalam method dan dikembalikan (return) ke luar method. Beberapa hal yang harus diperhatikan:

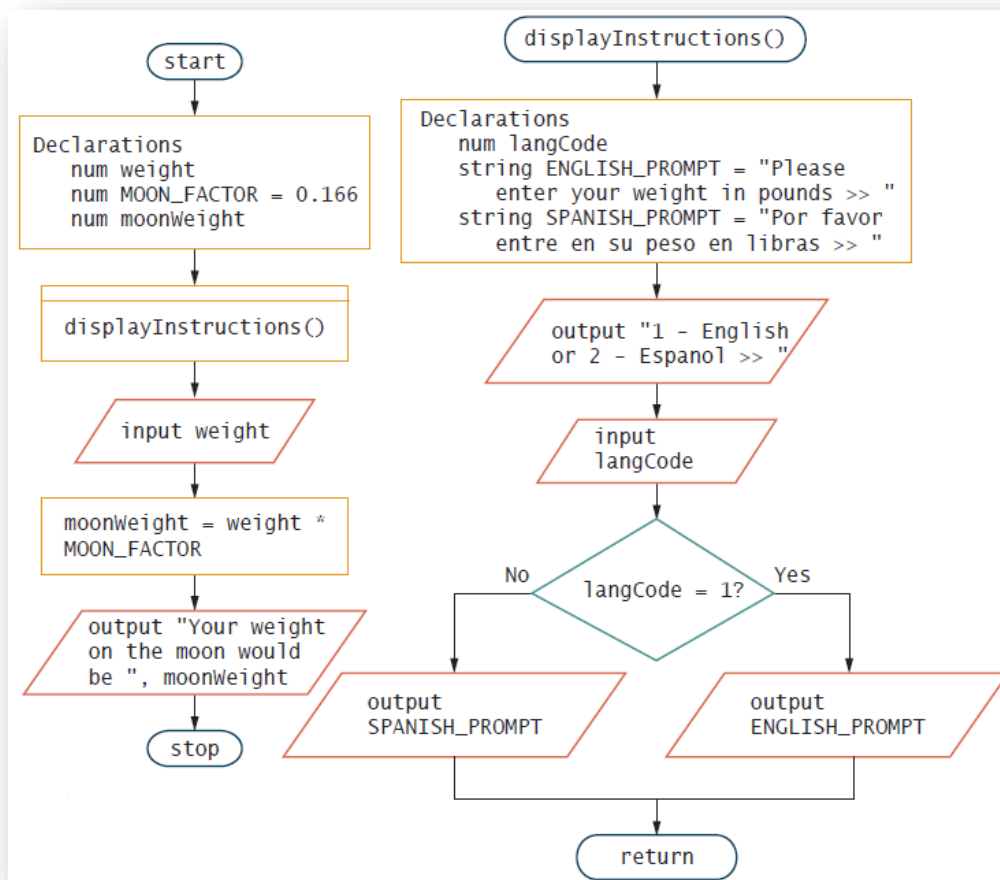
- Nama method
- Data & tipe data yang dikirim ke method
- Data & tipe data yang keluar dari method

Method Tanpa Parameter

- Method tanpa parameter → tidak ada data yang masuk/keluar method
- Cara memanggil method → panggil menggunakan nama method dengan data item kosong, contoh: `count()`

Contoh Method Tanpa Parameter

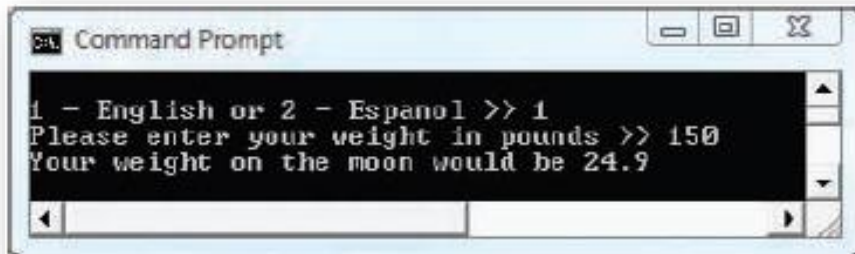
- Berikut adalah contoh penggunaan method tanpa parameter untuk menampilkan berat badan pengguna di bulan.
- Program memberikan pilihan dalam bahasa Inggris dan Spanyol.
- Program utama memiliki 2 variabel dan 1 konstanta.
- Method memiliki 1 variabel lokal.



```

start
  Declarations
    num weight
    num MOON_FACTOR = 0.166
    num moonWeight
  displayInstructions()
  input weight
  moonWeight = weight * MOON_FACTOR
  output "Your weight on the moon would be ", moonWeight
stop

displayInstructions()
  Declarations
    num langCode
    string ENGLISH_PROMPT = "Please enter your weight in pounds >> "
    string SPANISH_PROMPT = "Por favor entre en su peso en libras >> "
  output "1 - English or 2 - Espanol >> "
  input langCode
  if langCode = 1 then
    output ENGLISH_PROMPT
  else
    output SPANISH_PROMPT
  endif
  return
  
```

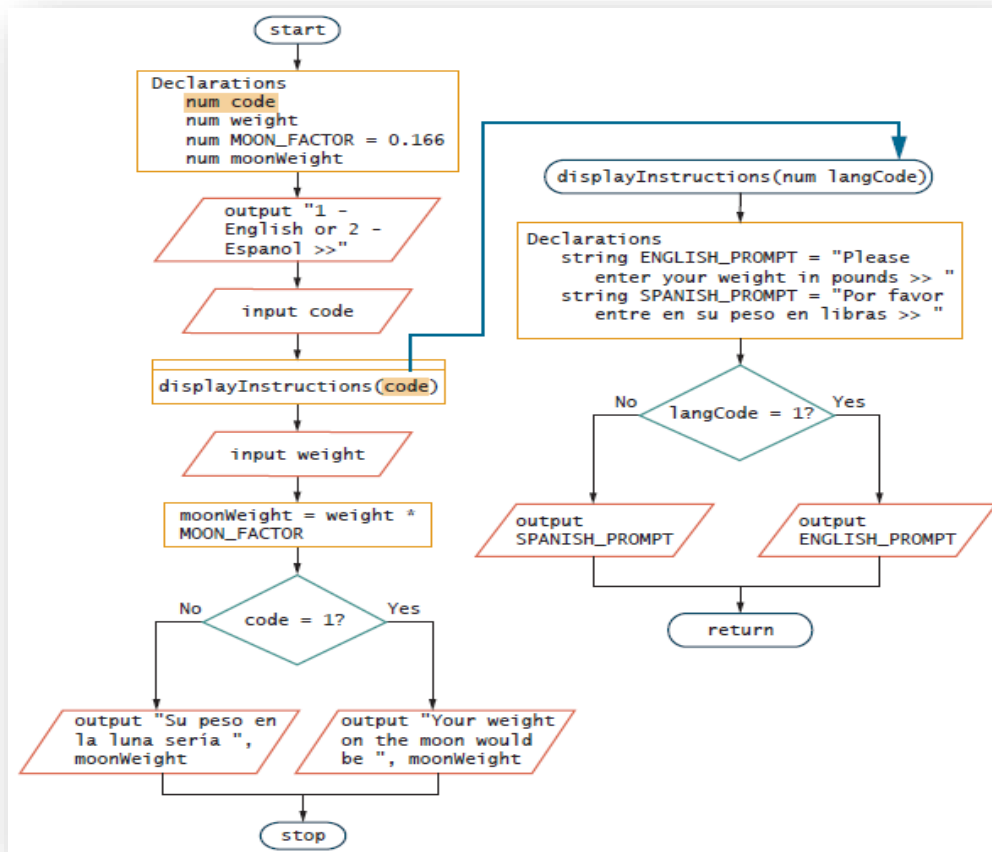



Method dengan Parameter Tunggal

- Method tanpa parameter tunggal → ada satu data yang masuk/keluar method
- Cara memanggil method → panggil menggunakan nama method beserta data item-nya, contoh: `count(c)`
- Method akan menerima data item sebagai sebuah parameter
- Method header → Definisikan tipe data dan nama lokal dari parameter, contoh: `count(num c1)`
- Pentingnya parameter bagi method:
 - Jika method tidak dapat menerima parameter, maka kita harus membuat seluruh kemungkinan method
- Jika method menerima parameter, berikut yang harus ditulis dalam bagian deklarasi method:
 - Tipe parameter (numerik atau string?)
 - Nama lokal parameter

Contoh Method dengan Parameter Tunggal

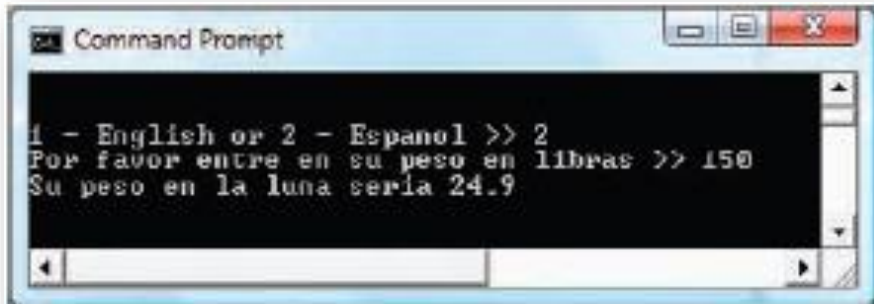
- Berikut adalah contoh program untuk menampilkan berat badan pengguna di bulan.
- Program memberikan pilihan dalam bahasa Inggris dan Spanyol.
- Program menggunakan method dengan parameter tunggal, yaitu pilihan bahasa.



```

start
  Declarations
    num code
    num weight
    num MOON_FACTOR = 0.166
    num moonWeight
  output "1 - English or 2 - Espanol >>"
  input code
  displayInstructions(code)
  input weight
  moonWeight = weight * MOON_FACTOR
  if code = 1 then
    output "Your weight on the moon would be ", moonWeight
  else
    output "Su peso en la luna sería ", moonWeight
  endif
stop

displayInstructions(num langCode)
  Declarations
    string ENGLISH_PROMPT = "Please enter your weight in pounds >> "
    string SPANISH_PROMPT = "Por favor entre en su peso en libras >> "
  if langCode = 1 then
    output ENGLISH_PROMPT
  else
    output SPANISH_PROMPT
  endif
return
  
```

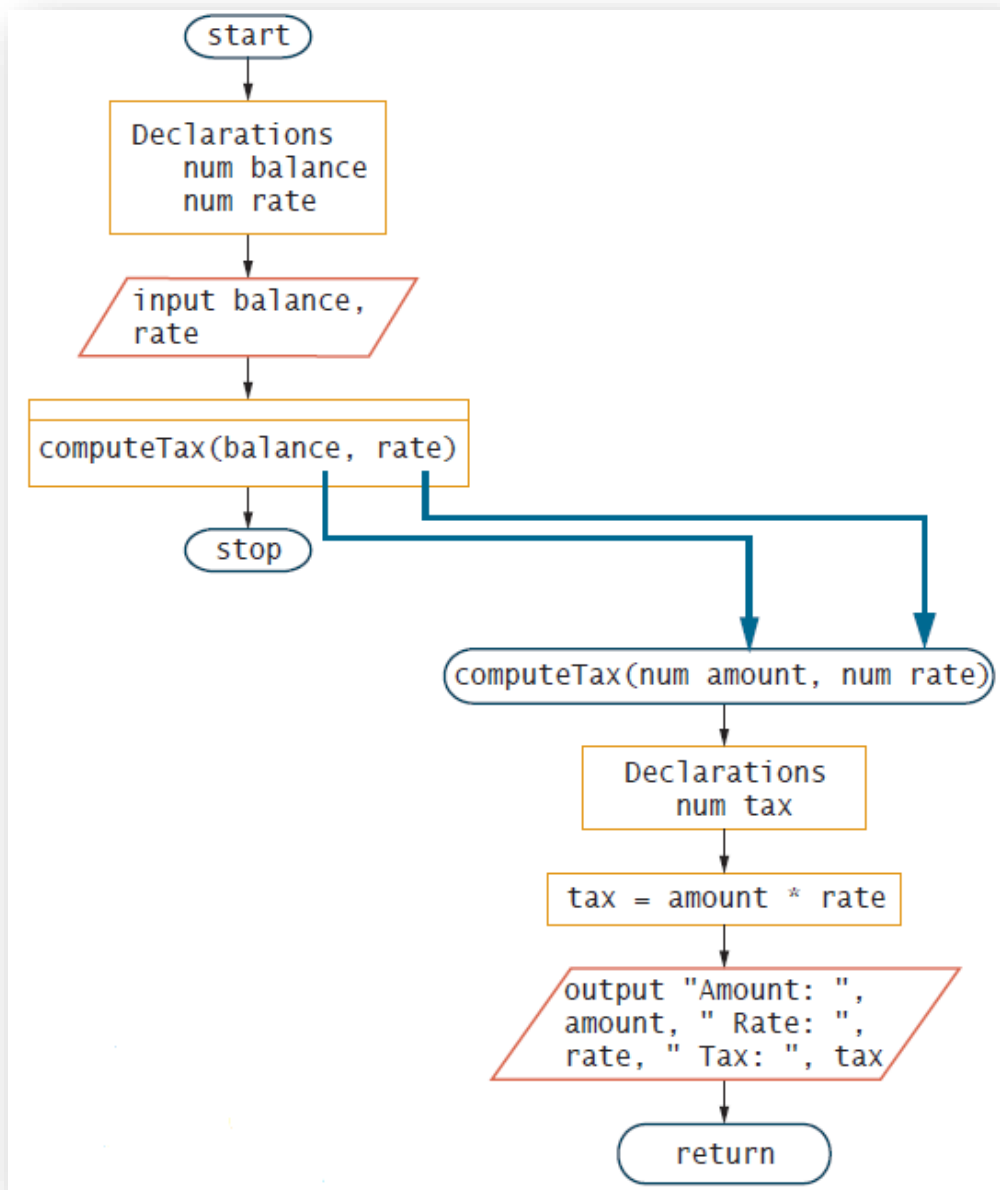


Method dengan Banyak Parameter

- Method dengan banyak parameter tunggal → ada lebih dari satu data yang masuk/keluar method
- Cara memanggil method → panggil menggunakan nama method beserta data item-nya dipisahkan dengan koma, contoh: count(x, y)
- Method akan menerima data item sebagai sebuah parameter
- Method header → Definisikan tipe data dan nama lokal dari parameter dipisahkan dengan koma, contoh: count(num x1, num y1)

Contoh Method dengan Banyak Parameter

- Berikut adalah contoh program untuk menghitung pajak.
- Program menghitung pajak berdasarkan jumlah saldo dan tarif pajak.
- Program menggunakan method dengan banyak parameter, yaitu balance dan rate.



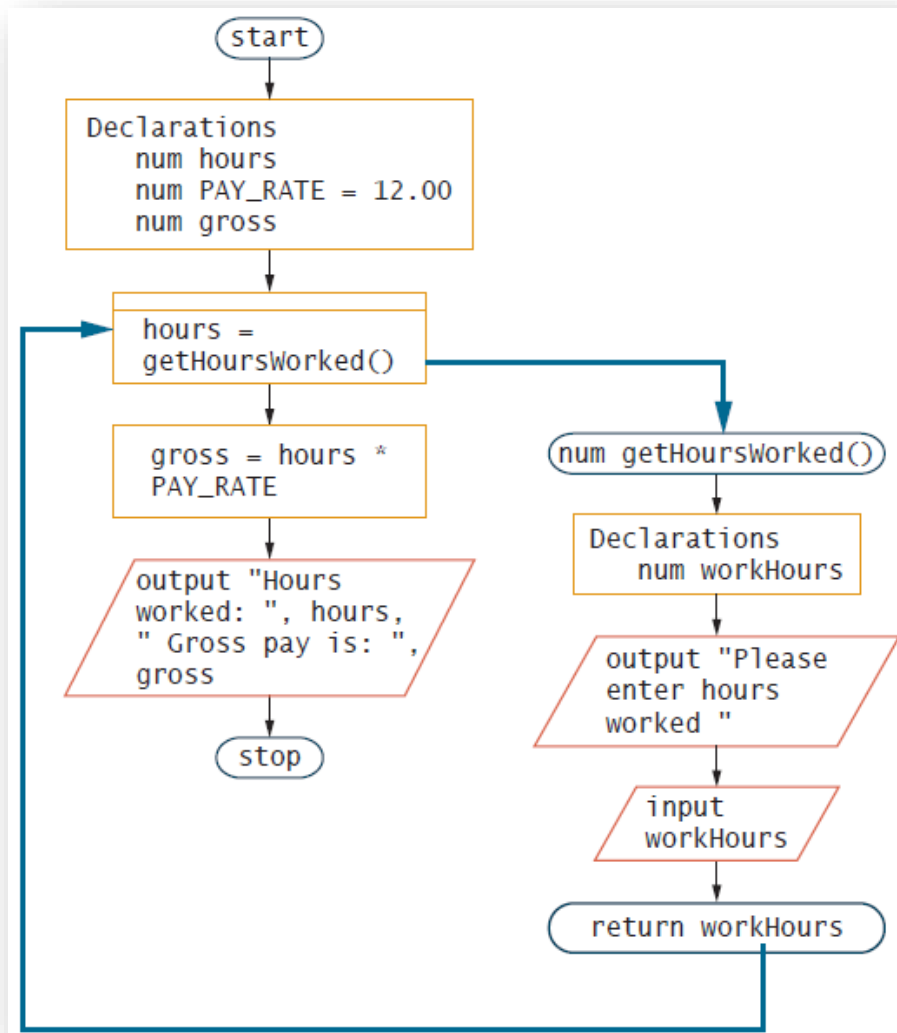
Method yang Mengembalikan Nilai

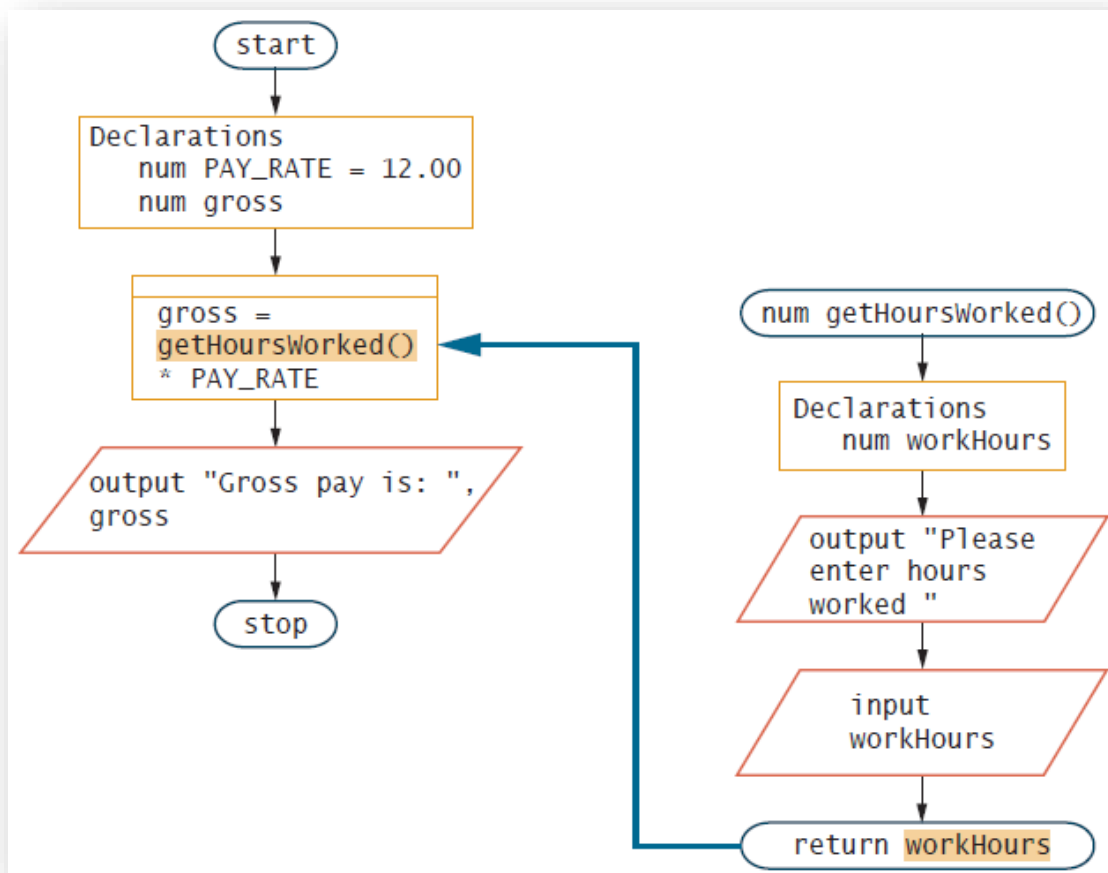
- Method tidak harus mengembalikan nilai (lihat tiga contoh sebelumnya)
- Apabila kita menghadapi ada suatu nilai yang dikembalikan oleh method, maka kita harus mendefinisikan tipe data dari nilai yang dikembalikan (return type)

- Method header → Definisikan tipe data dan nama lokal dari parameter dipisahkan dengan koma, contoh: num count()

Contoh Method Mengembalikan Nilai

- Berikut adalah contoh program untuk menghitung penghasilan kotor pegawai.
- Program menghitung penghasilan kotor berdasarkan jam kerja dan rate gaji.
- Program menggunakan method tanpa parameter dan mengembalikan nilai jam kerja.





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.