

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 memahami tiga struktur dasar dalam pemrograman 2. Mahasiswa melakukan strukturisasi program dengan input dasar
Kompetensi Akhir di Setiap Tahap (Sub- Cpmk)		1. Mampu Memahami Struktur dan Strukturisasi Program
Minggu Perkuliahan Online Ke-		9

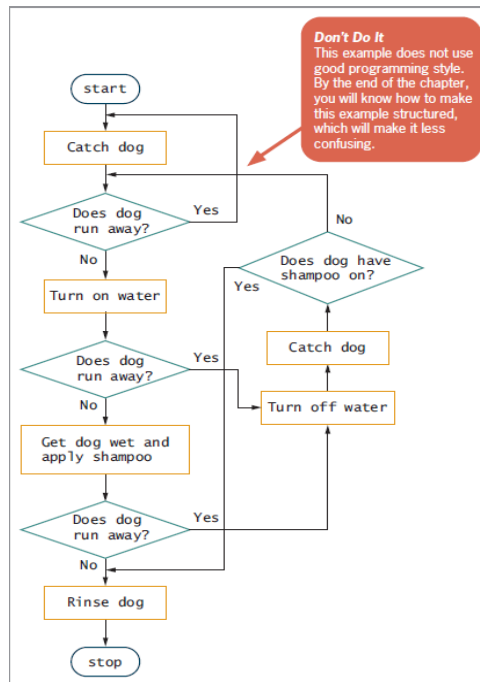
JUDUL TOPIK – Struktur dan Strukturisasi Program

Struktur

Spaghetti Code

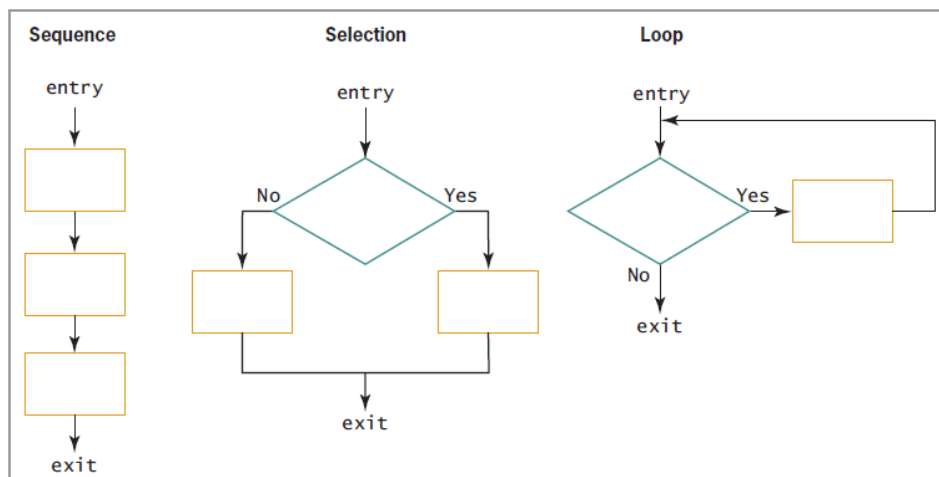
- “Spaghetti code” → istilah untuk pernyataan program yang rumit dan memiliki logika yang sulit untuk diikuti
- “Spaghetti code” → program tidak terstruktur
- Konsekuensi program tidak terstruktur:
 - Pernyataan sulit dibaca
 - Pernyataan sulit dipelihara
 - Logika pemrograman sulit diikuti

Contoh Spaghetti Code Logic: *Washing a dog*



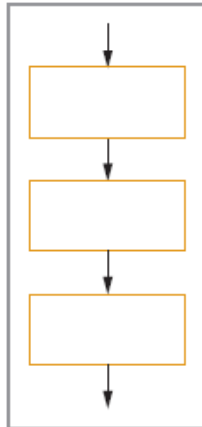
Tiga Struktur Dasar

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



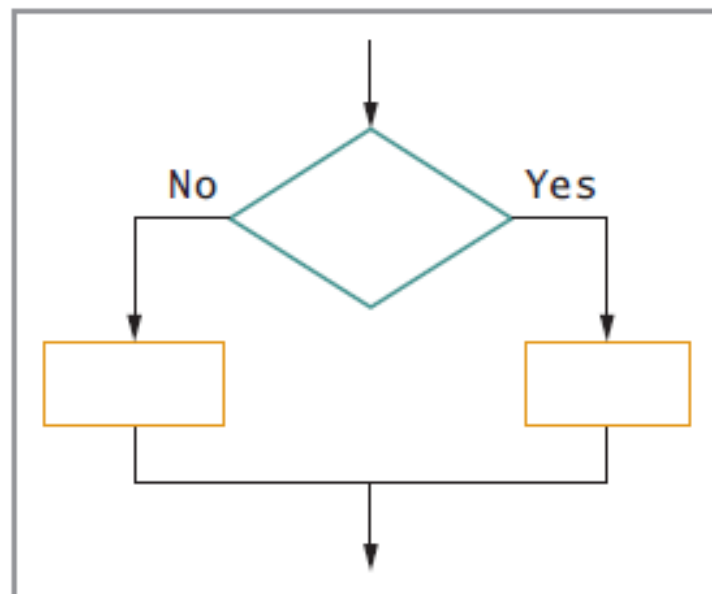
1. Sequence Structure

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



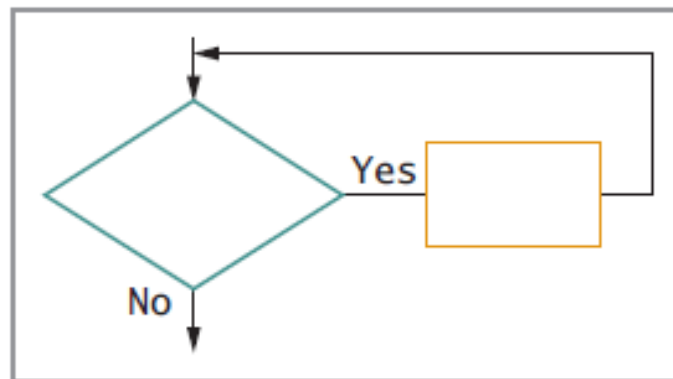
2. Selection Structure

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

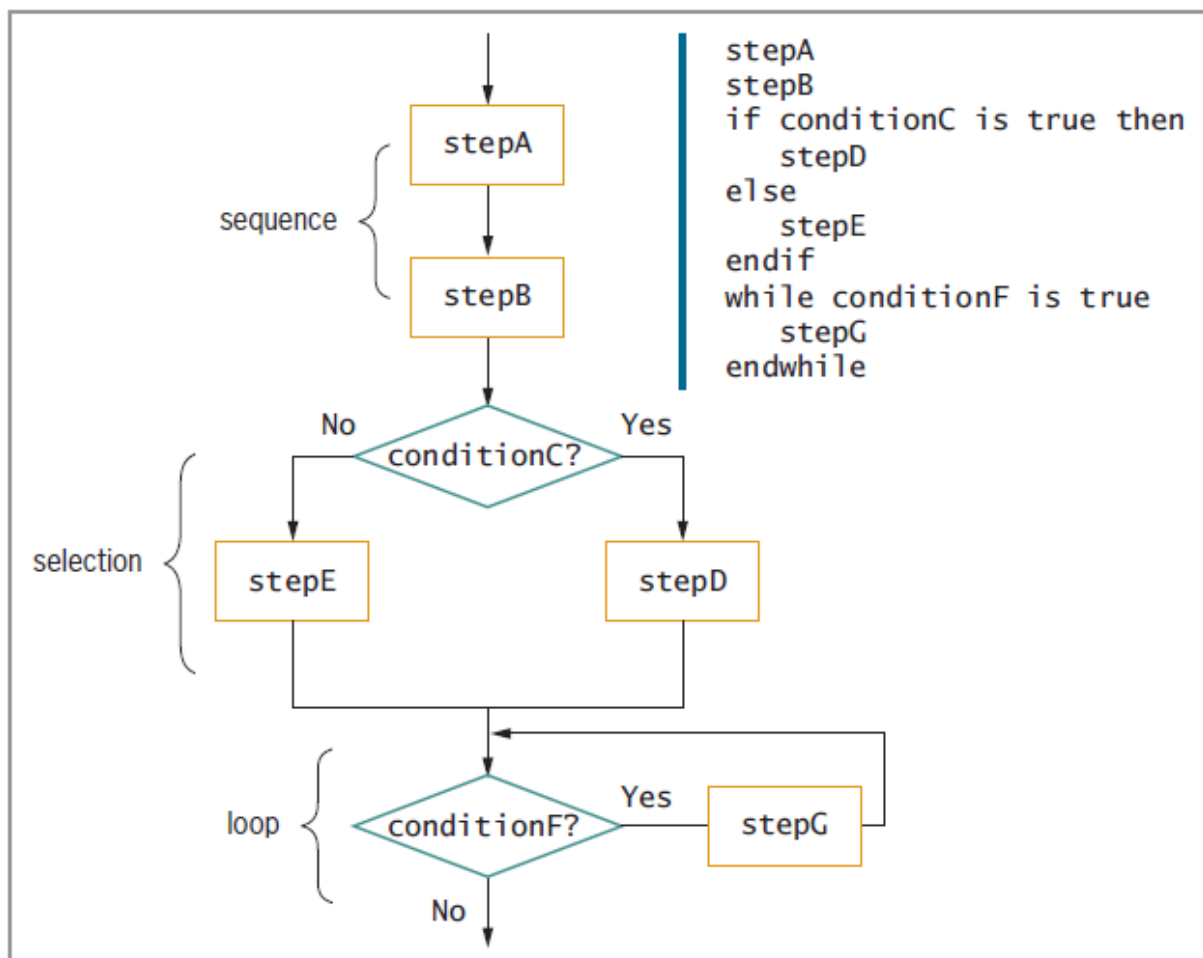


3. Loop Structure

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

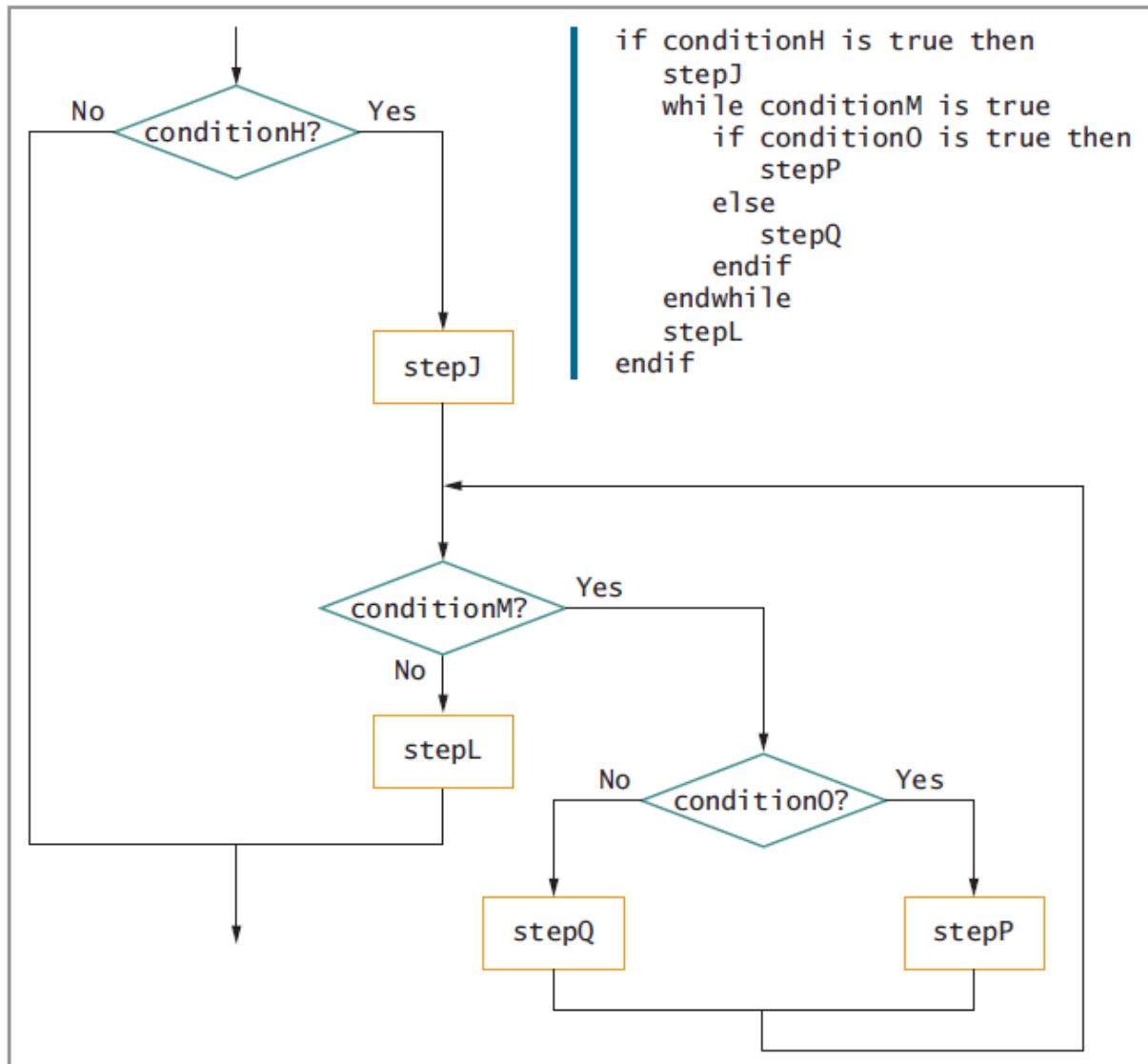


Flowchart dengan struktur kendali bertumpuk (stacked)



Nested structure: sequence nested within a selection

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

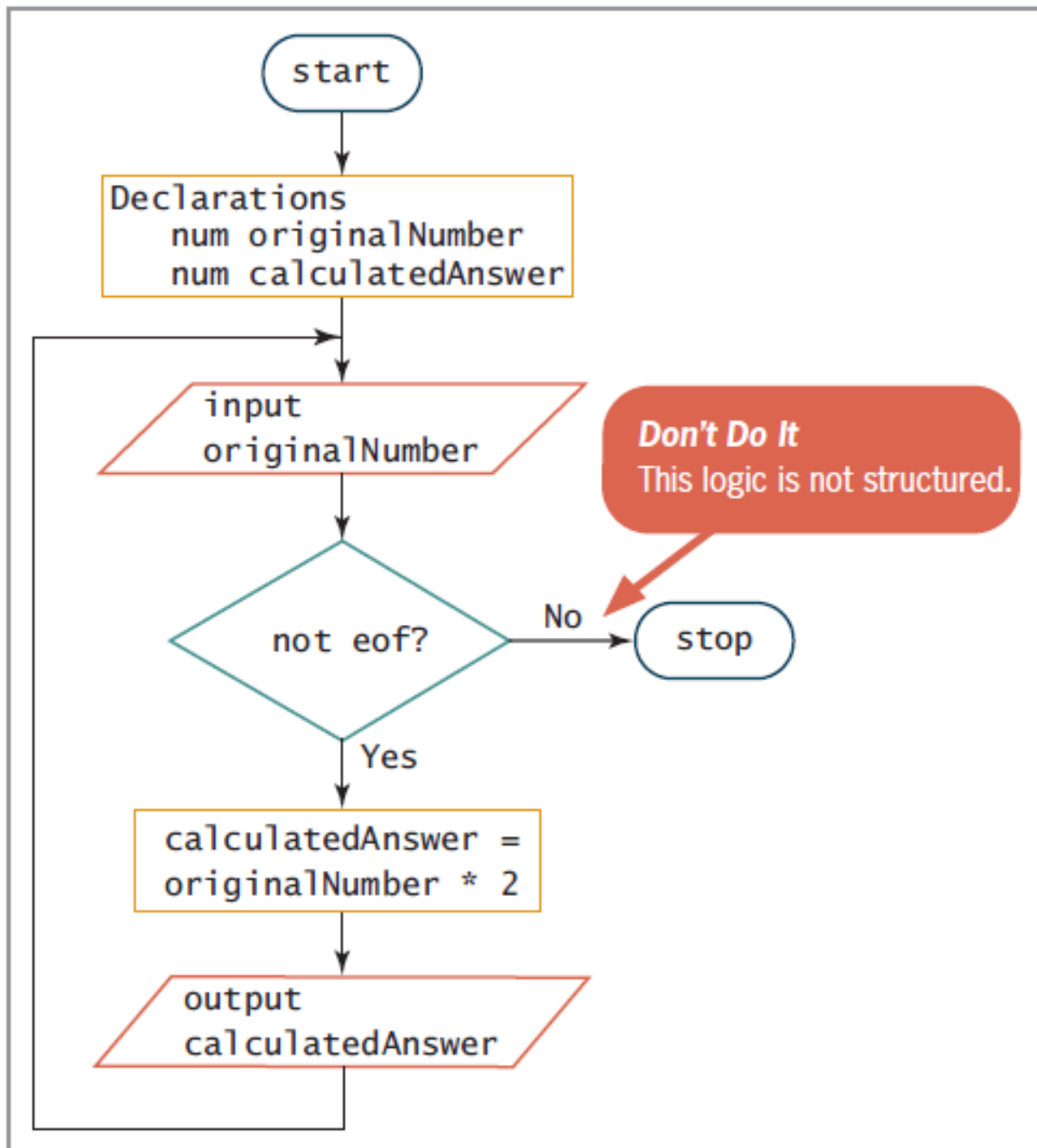


Strukturisasi Program dengan Input Dasar

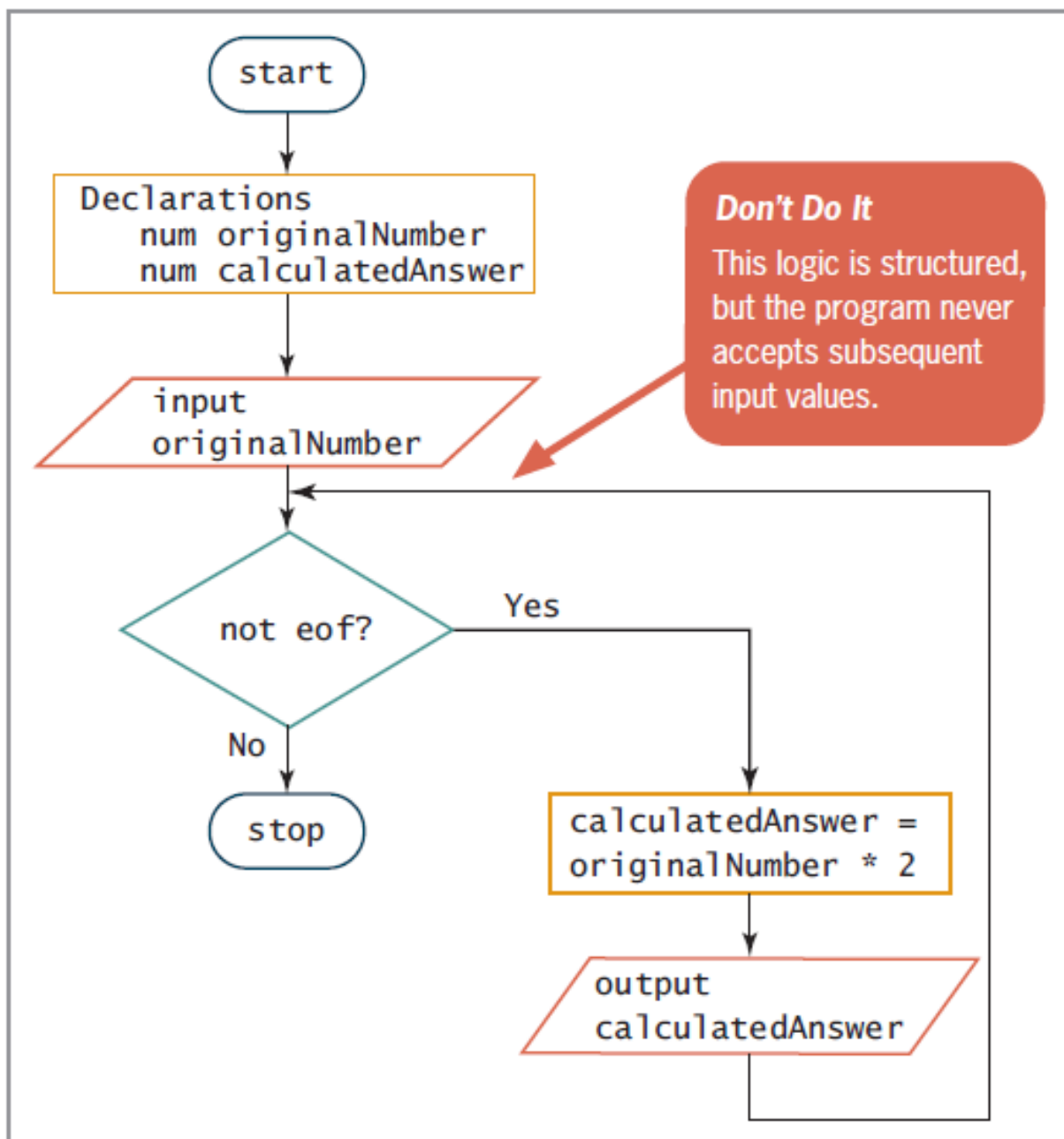
- Input dasar / "priming input" / "priming feed" → pernyataan yang ditambahkan untuk memperoleh nilai dari input pertama

- Input dasar penting untuk menghasilkan program yang terstruktur dan fungsional (sesuai dengan kebutuhan yang dimaksud)

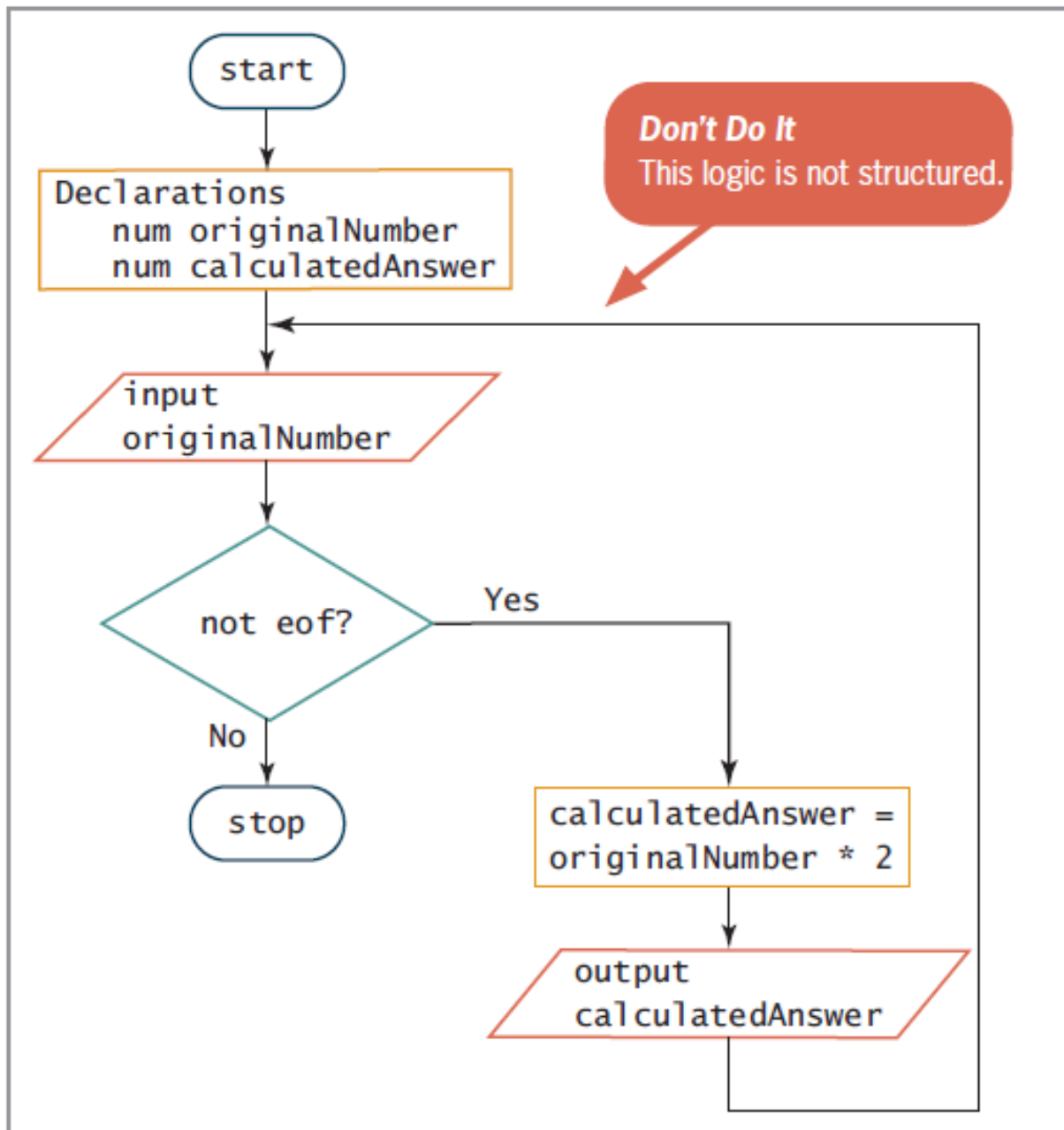
Program tidak terstruktur



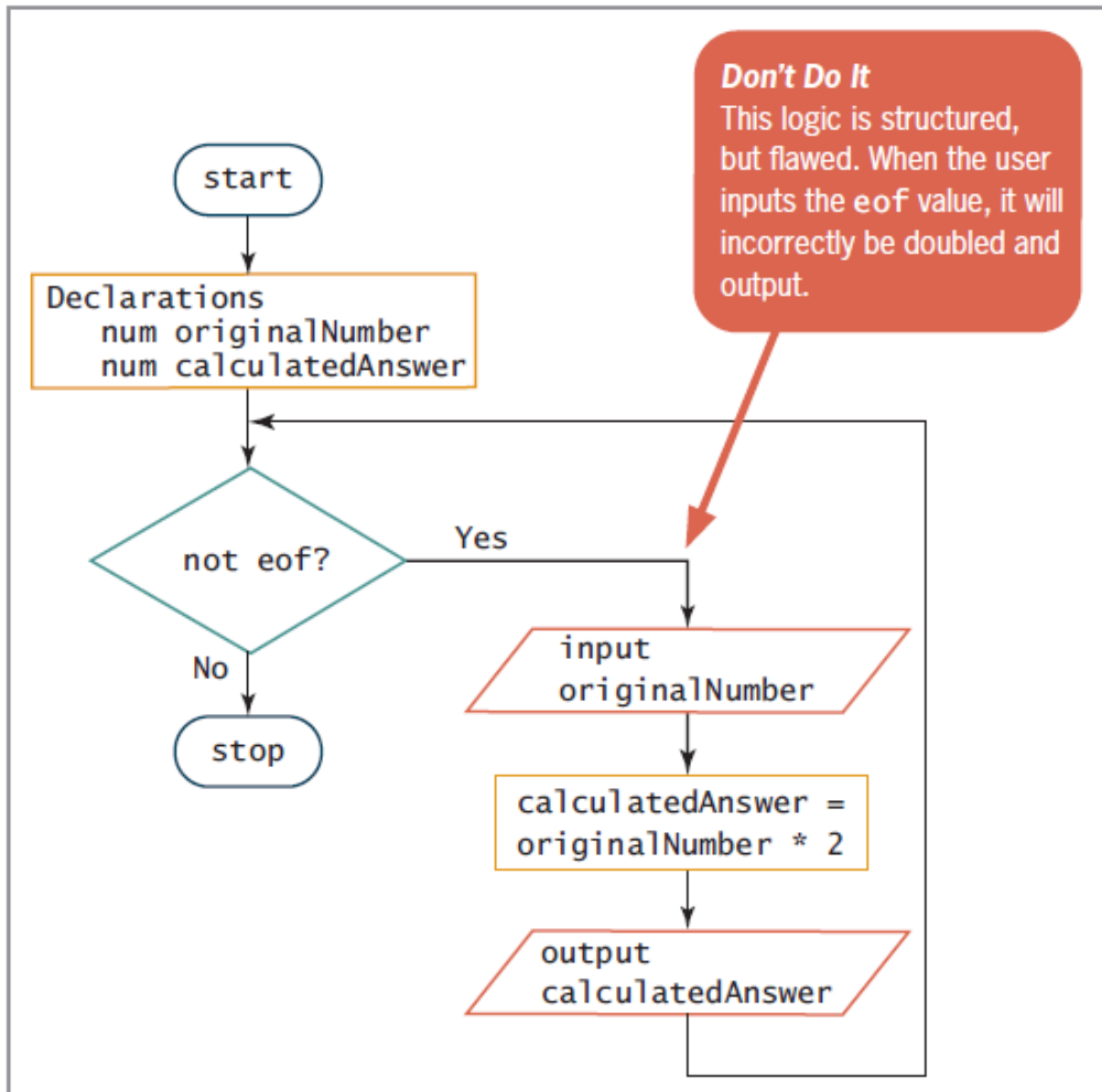
Terstruktur tapi tidak fungsional



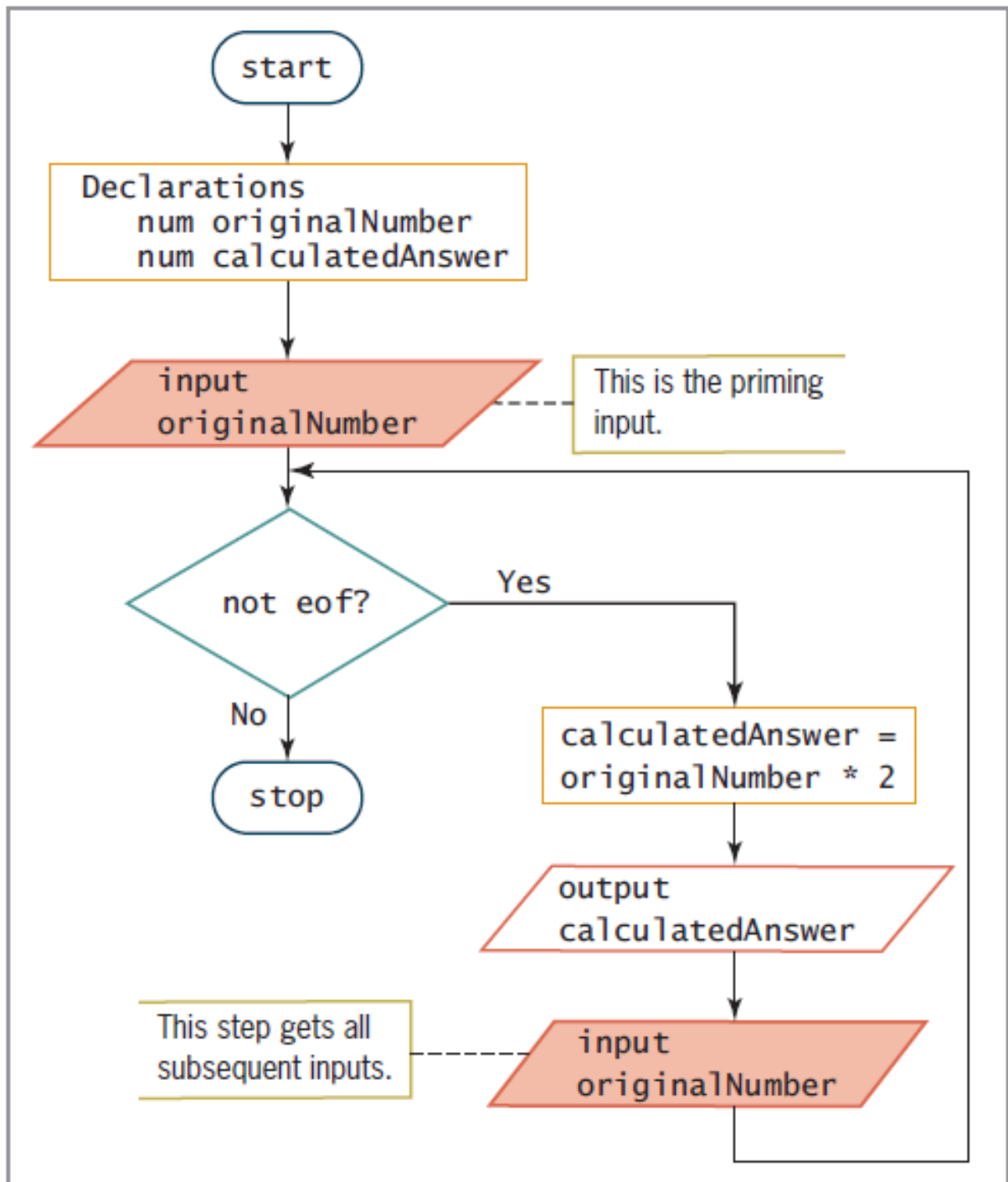
Fungsional, tapi tidak terstruktur



Terstruktur dan fungsional, tapi memiliki kelemahan dalam input



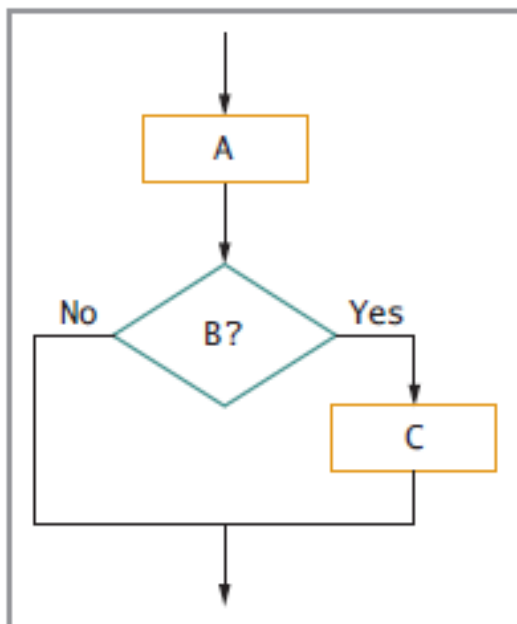
Terstruktur, fungsional sesuai dengan kebutuhan



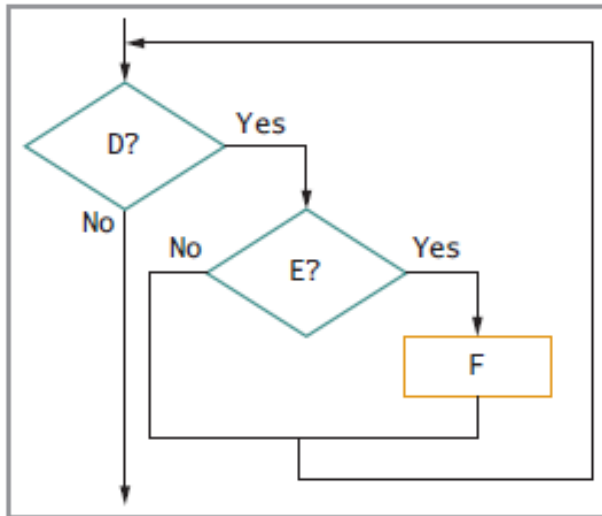
Mengenali Struktur

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

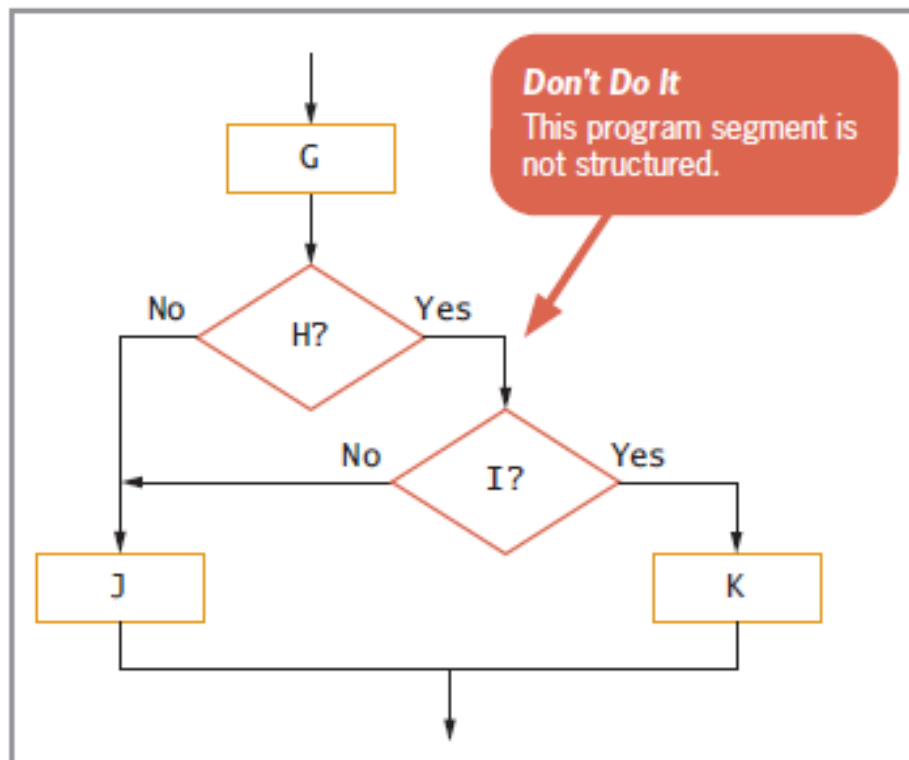
Struktur dengan sequence dan selection



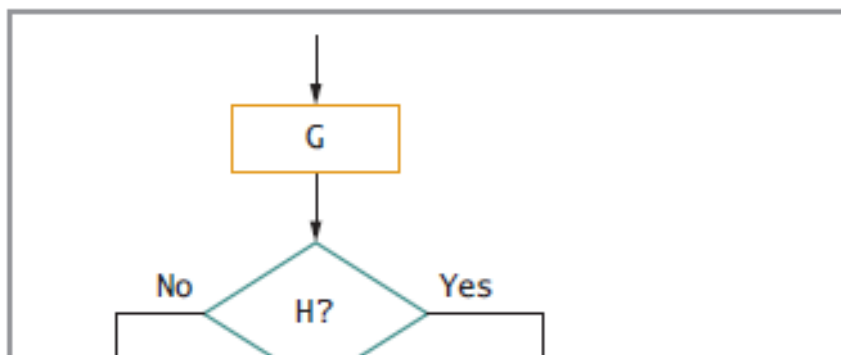
Struktur dengan loop dan di dalam loop ada selection



Program tidak terstruktur



Program terstruktur dengan memisahkan statement J



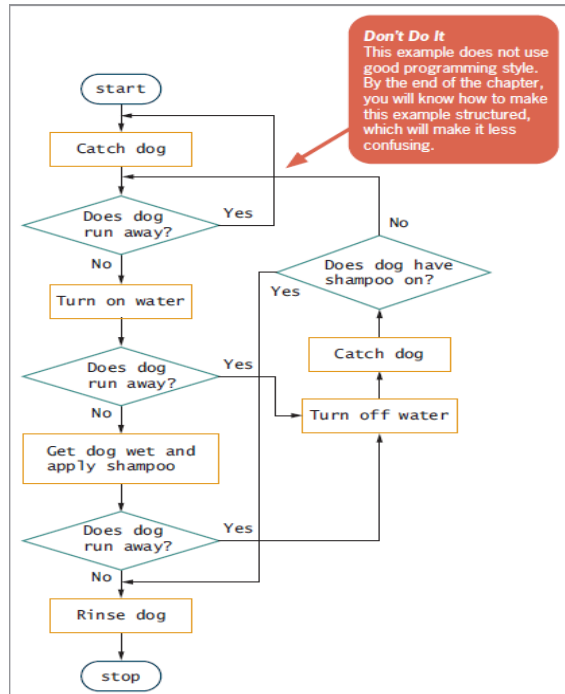
Mengapa Harus Terstruktur?

- **Clarity** → memudahkan dalam membaca program, semakin besar programnya maka akan sulit dibaca jika tidak terstruktur
- **Professionalism** → programmer dituntut untuk menghasilkan program yang terstruktur
- **Efficiency** → menggunakan struktur dapat meningkatkan efisiensi penulisan program
- **Maintenance** → mudah untuk dimodifikasi dan dipelihara
- **Modularity** → mudah untuk dipecah-pecah menjadi subprogram

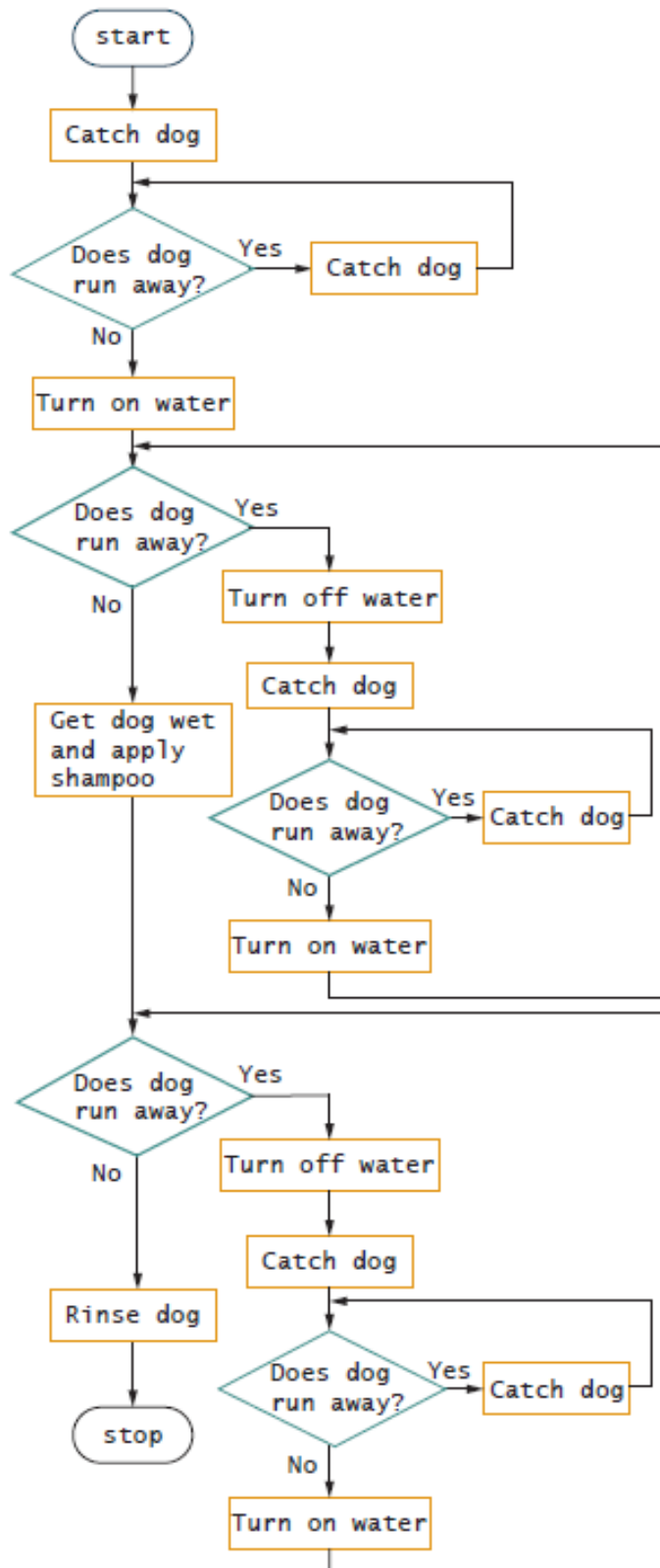
Strukturisasi dan Modularisasi

- Modularisasi merupakan cara untuk mensimplifikasi program, sehingga mudah dibaca
- Modularisasi hanya dapat dilakukan pada program terstruktur

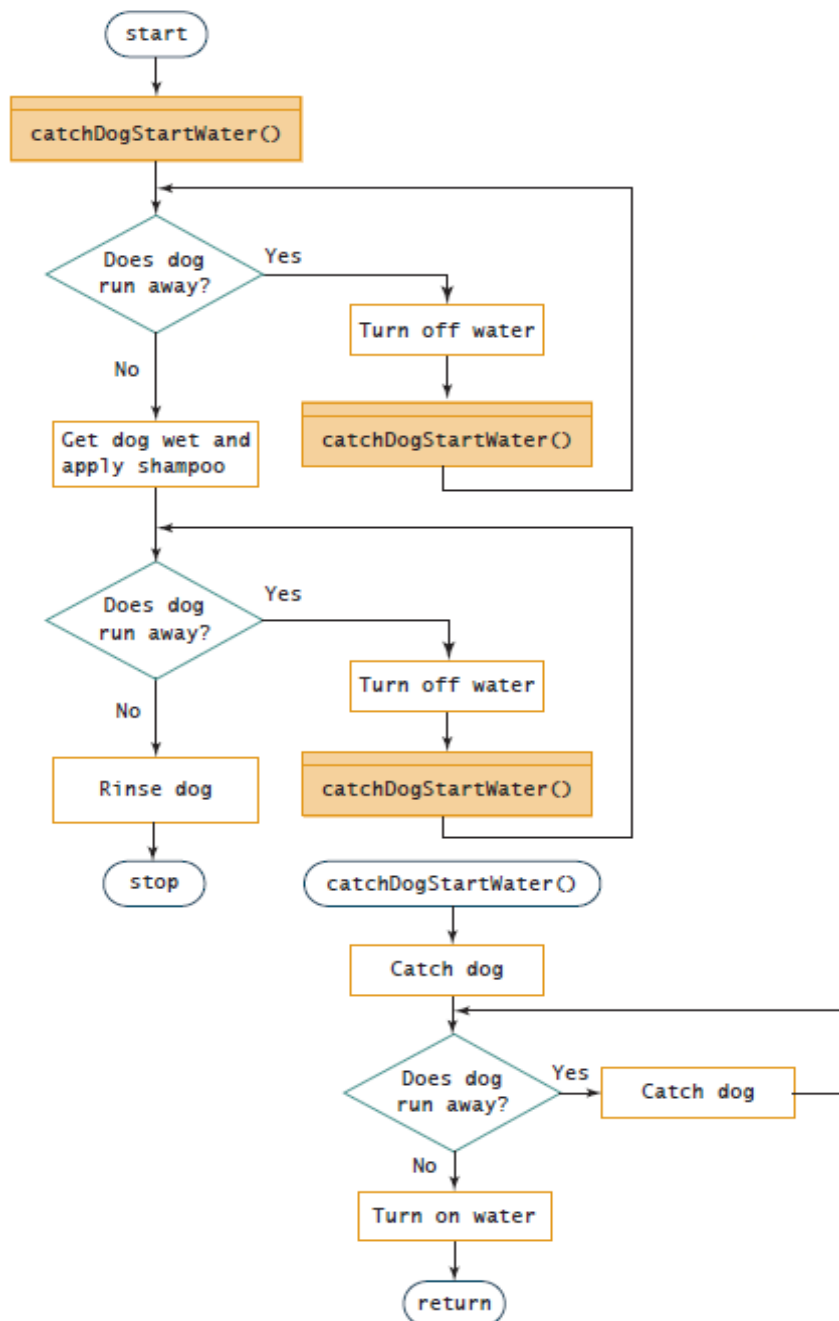
Contoh Spaghetti Code Logic: *Washing a dog*



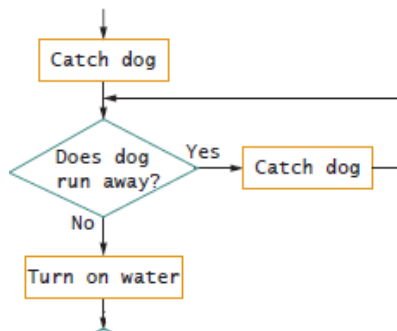
Program yang terstruktur dan fungsional



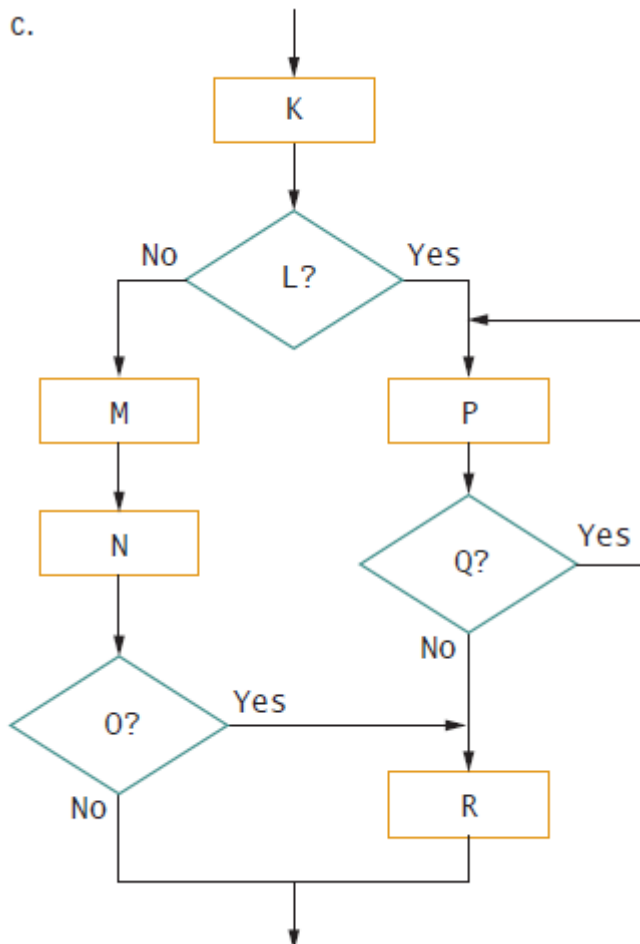
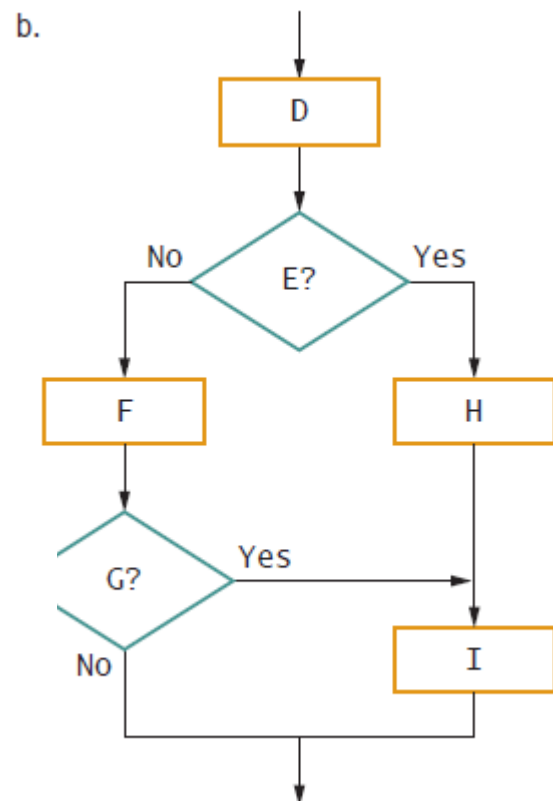
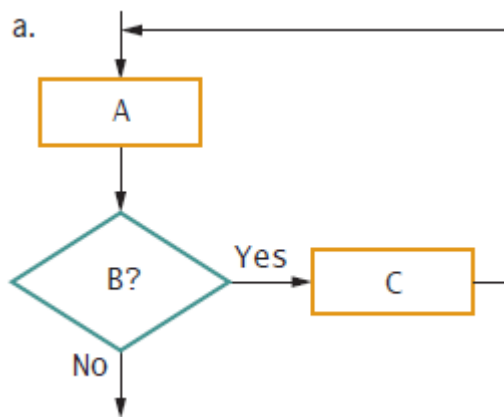
Program yang terstruktur dan fungsional dengan versi modular



catchDogStartWater()



Latihan



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.