

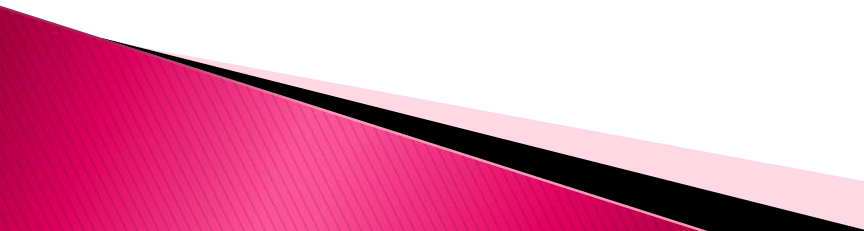
Pengujian Perangkat Lunak

Universitas Indraprasta PGRI

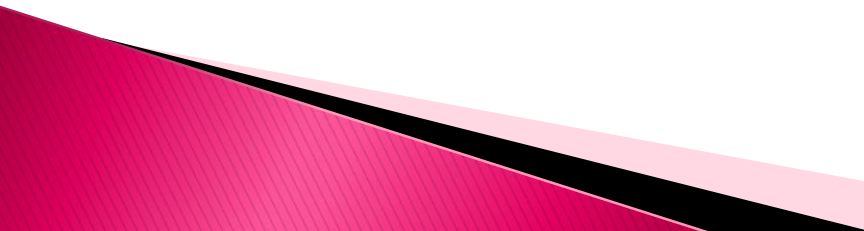
Tujuan

- ▶ membantu memastikan kelengkapan pengujian dan memberikan kemungkinan tertinggi untuk mengungkap kesalahan pada perangkat lunak.

White Box Testing

- ▶ Didasarkan pada pengamatan yang teliti terhadap detail prosedural
 - ▶ Jalur–jalur logika yang melewati perangkat lunak diuji dengan memberikan kasus uji yang menguji serangkaian kondisi dan atau loop tertentu
 - ▶ Status program tersebut dapat diuji pada berbagai titik untuk menentukan apakah status yang diharapkan sesuai dengan status yang sebenarnya
- 

Kasus Uji White Box Testing

- ▶ Memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali
 - ▶ Menggunakan semua keputusan logis pada sisi true dan false
 - ▶ Mengeksekusi semua loop pada batasan mereka dan pada batas operasional mereka
 - ▶ Menggunakan struktur data internal untuk menjamin validitasnya
- 

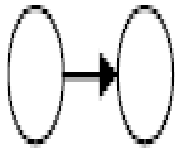
Jenis Pengujian White Box

- ▶ Basis path testing
- ▶ Control Structure Testing, yang terdiri dari:
 - Condition Testing
 - Data Flow Testing
 - Loop Testing

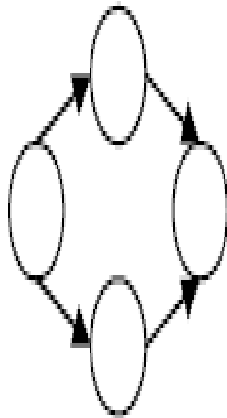
Uji Coba Basis Path

▶ Notasi Diagram

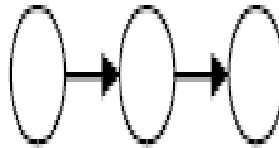
Sequence



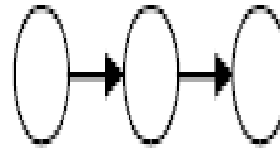
if



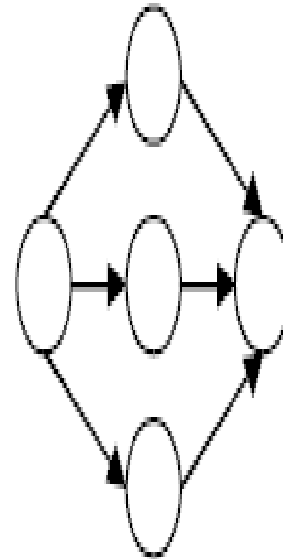
while



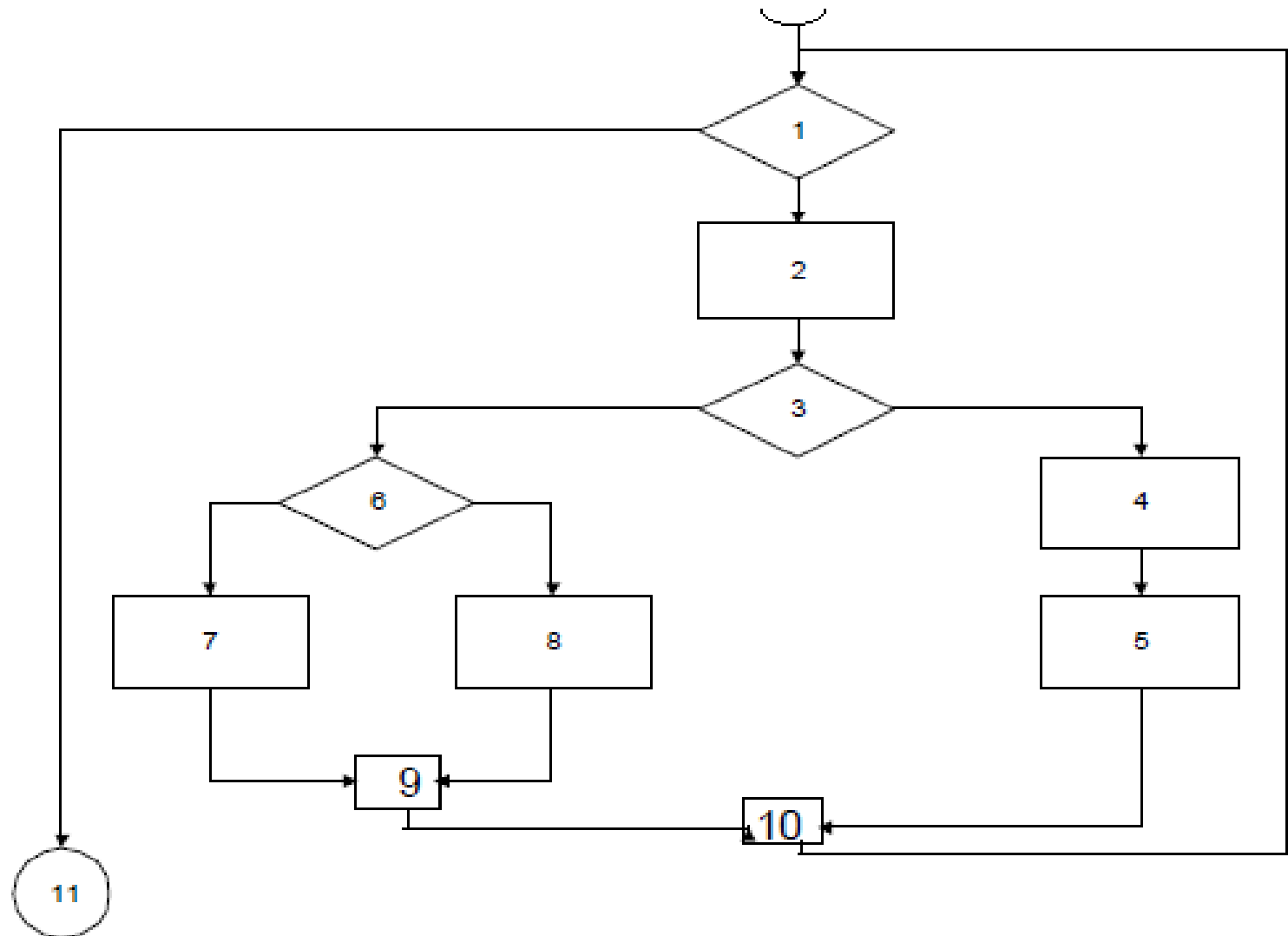
until



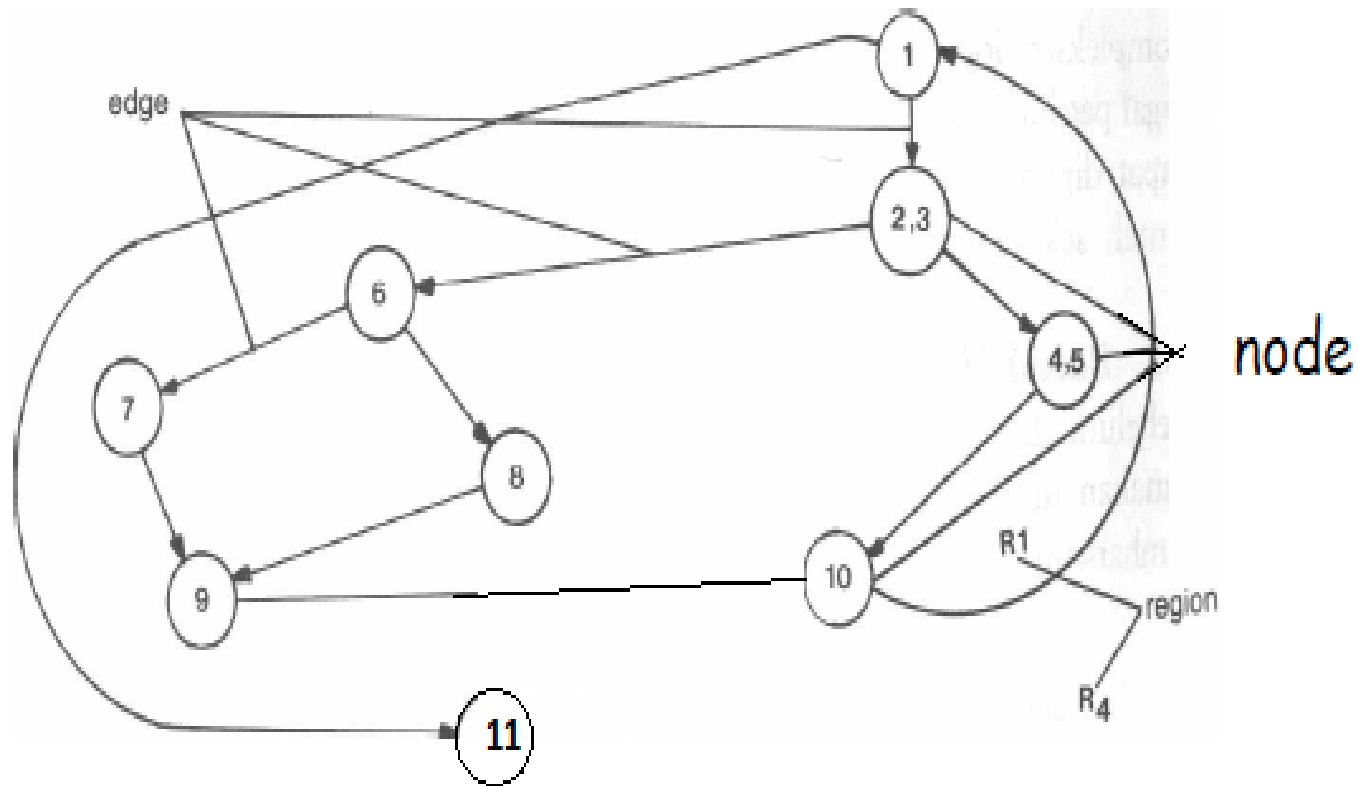
case



Digambarkan dengan Flowchart



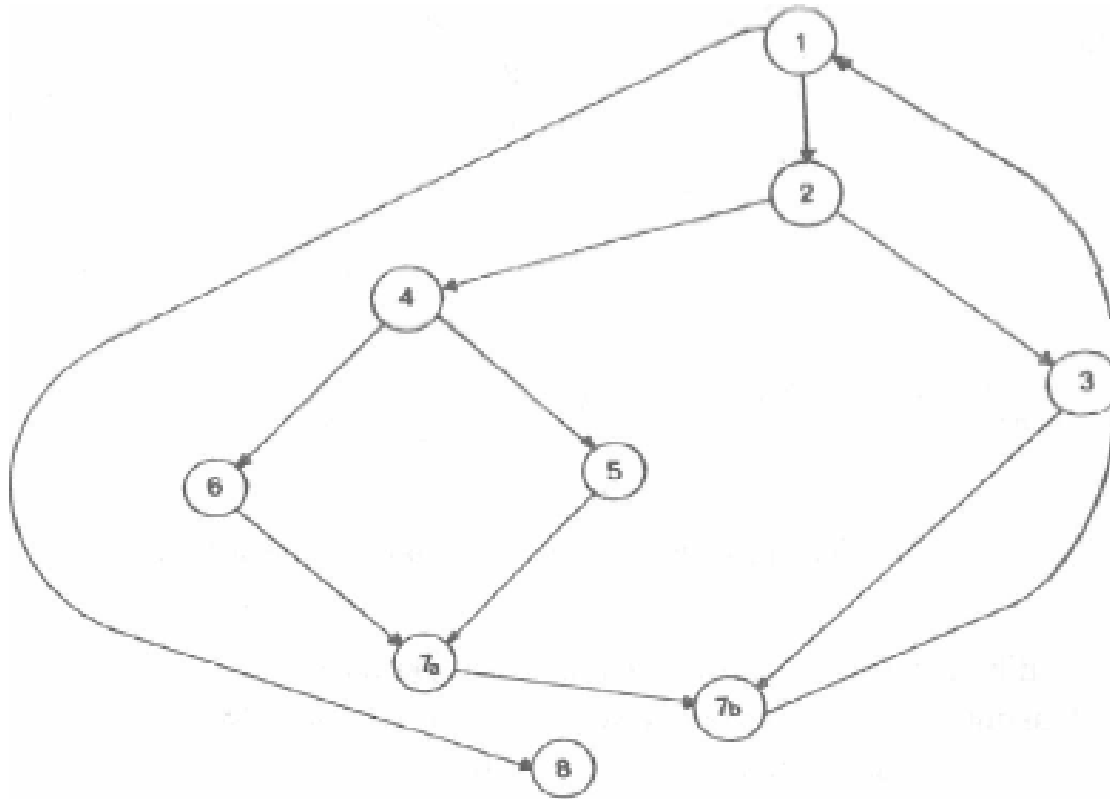
Digambarkan dengan Flowgraph



Penerjemahan Pseudocode ke FlowGraph

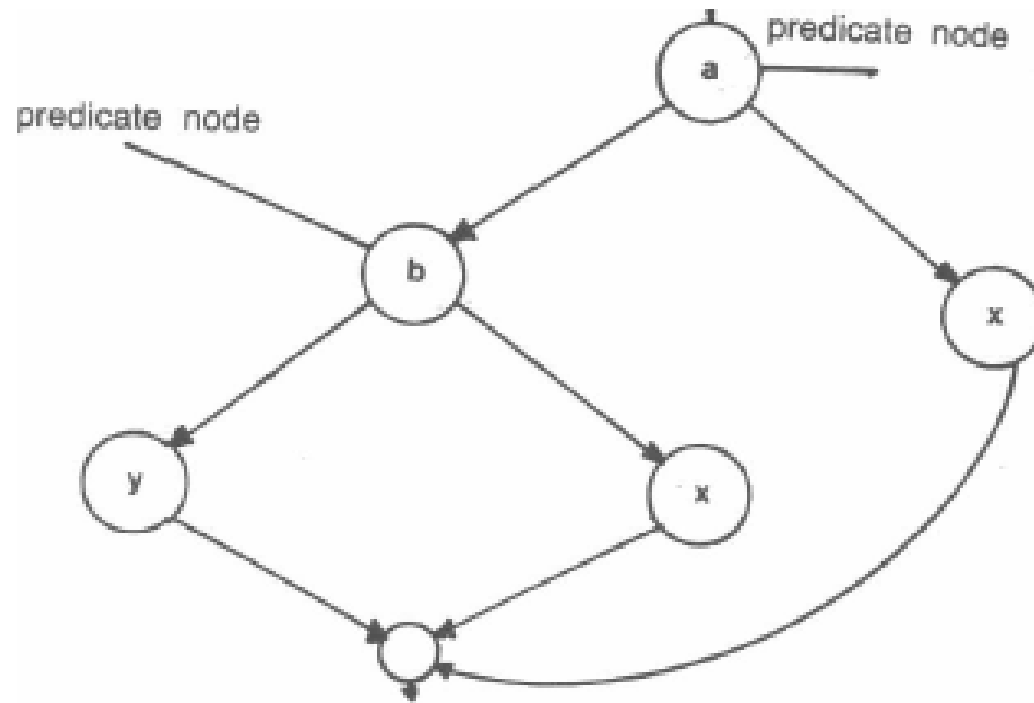
```
1:   do while record masih ada
      baca record
2:   if record ke 1 = 0
3:   then proses record
      simpan di buffer
      naikan kounter
4:   else if record ke 2 = 0
5:   then reser kounter
6:   proses record
      simpan pada file
7a:  endif
      endif
7b:  enddo
8:   end
```

Penerjemahan Pseudocode ke FlowGraph (Lanjutan)



Kondisi Majemuk

- ▶ Boolean (AND, OR, NAND, NOR) yg dipakai pada perintah if.



Cyclomatic Complexity

- ▶ metrik PL yang menyediakan ukuran kuantitatif dari kekompleksan logikal program.
- ▶ menentukan jumlah jalur independen dalam basis set suatu program dan memberi batas atas untuk jumlah uji coba yang harus dikerjakan untuk menjamin bahwa seluruh perintah sekurang-kurangnya telah dikerjakan sekali.
- ▶ Jalur independent adalah jalur yang melintasi atau melalui program dimana sekurang-kurangnya terdapat proses perintah yang baru atau kondisi yang baru.

Rumus Cyclomatic Complexity

- ▶ Jumlah region grafik alir sesuai dengan cyclomatic complexity.
- ▶ Cyclomatic complexity $V(G)$ untuk grafik alir dihitung dengan rumus:

$$V(G) = E - N + 2$$

dimana:

E = jumlah edge pada grafik alir

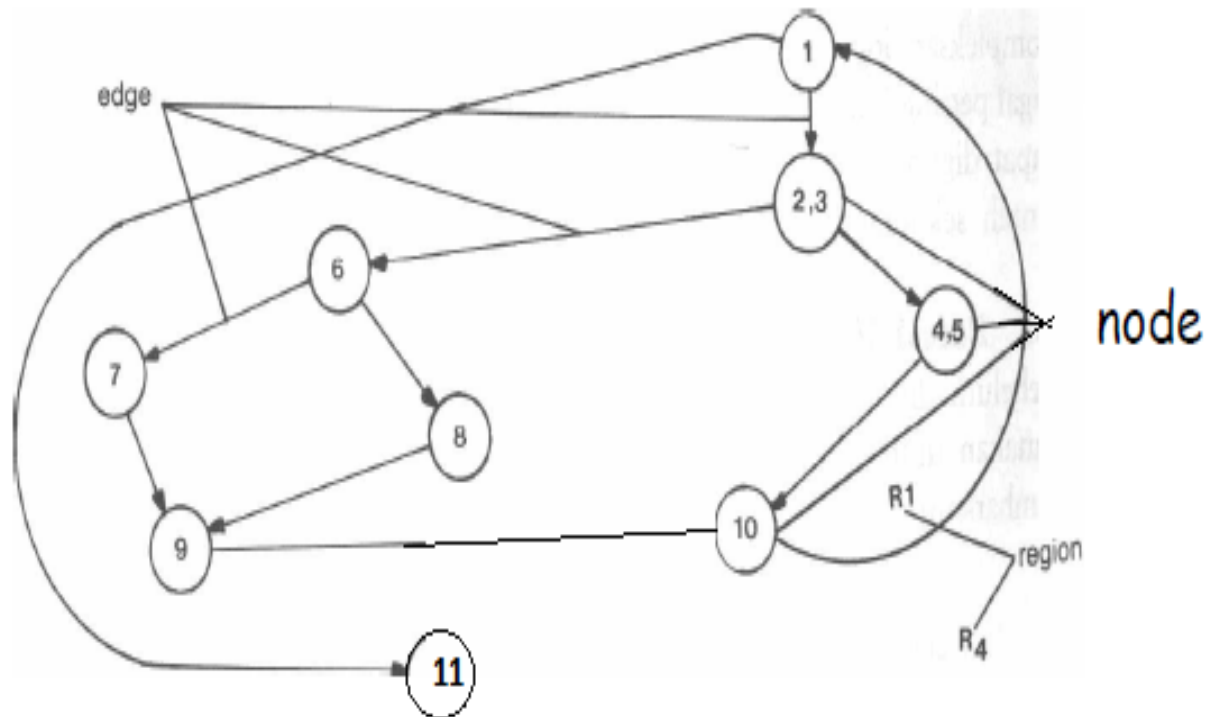
N = jumlah node pada grafik alir

- ▶ Cyclomatic complexity $V(G)$ juga dapat dihitung dengan rumus:

$$V(G) = P + 1$$

dimana P = jumlah predicate node pada grafik alir

Hitung Cyclomatic Complexity pada flowgraph di bawah ini!



Jawaban

- ▶ Flowgraph mempunyai 4 region

$$V(G) = 11 \text{ edge} - 9 \text{ node} + 2 = 4$$

$$V(G) = 3 \text{ predicate node} + 1 = 4$$

- ▶ Independent Path

Path 1 : 1 - 11

Path 2 : 1 - 2 - 3 - 4 - 5 - 10 - 1 - 11

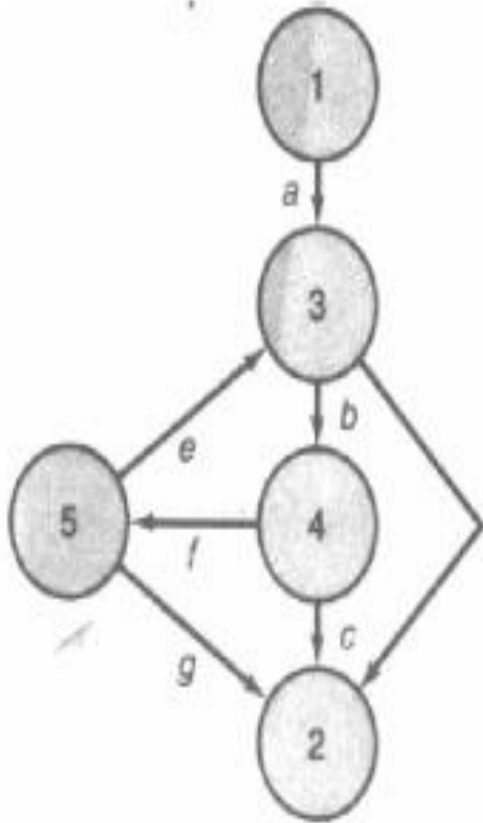
Path 3 : 1 - 2 - 3 - 6 - 8 - 9 - 10 - 1 - 11

Path 4 ' : 1 - 2 - 3 - 6 - 7 - 9 - 10 - 1 - 11

Graph Metrik

- ▶ PL yang dikembangkan untuk membantu uji coba basis path atau struktur data.
- ▶ matrik empat persegi yang mempunyai ukuran (sejumlah baris dan kolom) yang sama dengan jumlah node pada flowgraph

Contoh Graph Metrik



Terhubung ke simpul

Simpul	1	2	3	4	5
1			a		
2					
3		d		b	
4		c			f
5		g	e		

Hubungan Bobot

Simpul	Terhubung ke simpul				
	1	2	3	4	5
1			1		
2					
3		1		1	
4		1			1
5		1	1		

Koneksi :

$$1 - 1 = 0$$

$$2 - 1 = 1$$

$$2 - 1 = 1$$

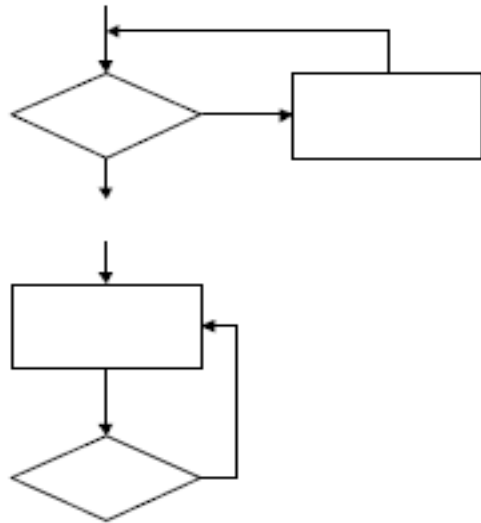
$$2 - 1 = 1$$

$$3 + 1 = 4 \text{ cyclomatic complexity}$$

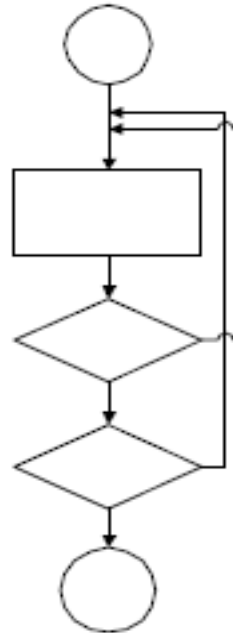
PENGUJIAN LOOP

- ▶ **Loop Sederhana**
 - ▶ **Loop Tersarang**
 - ▶ **Loop Terangkai**
 - ▶ **Loop Tidak Terstruktur**
- 

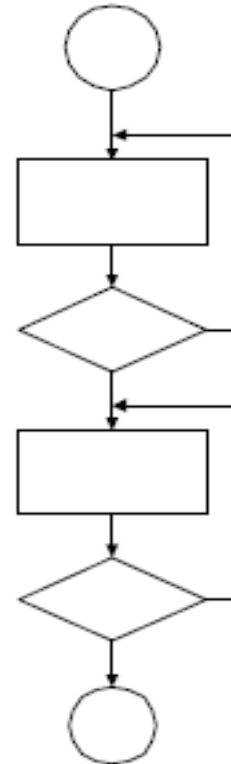
Gambar Loop



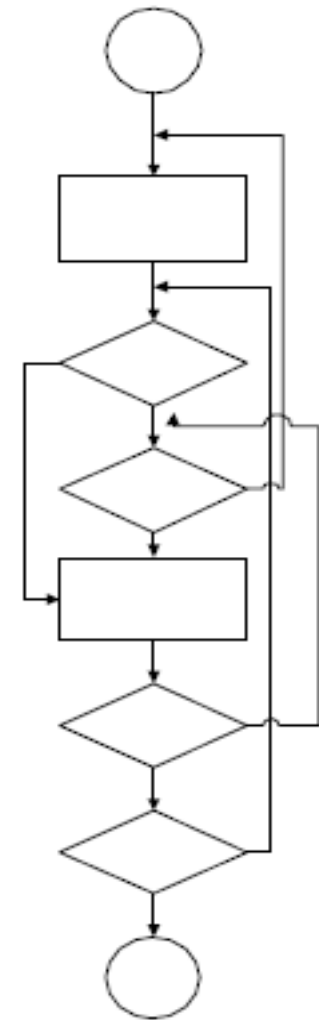
Loop sederhana



Loop tersarang



Loop terangkai



Loop tidak terstruktur

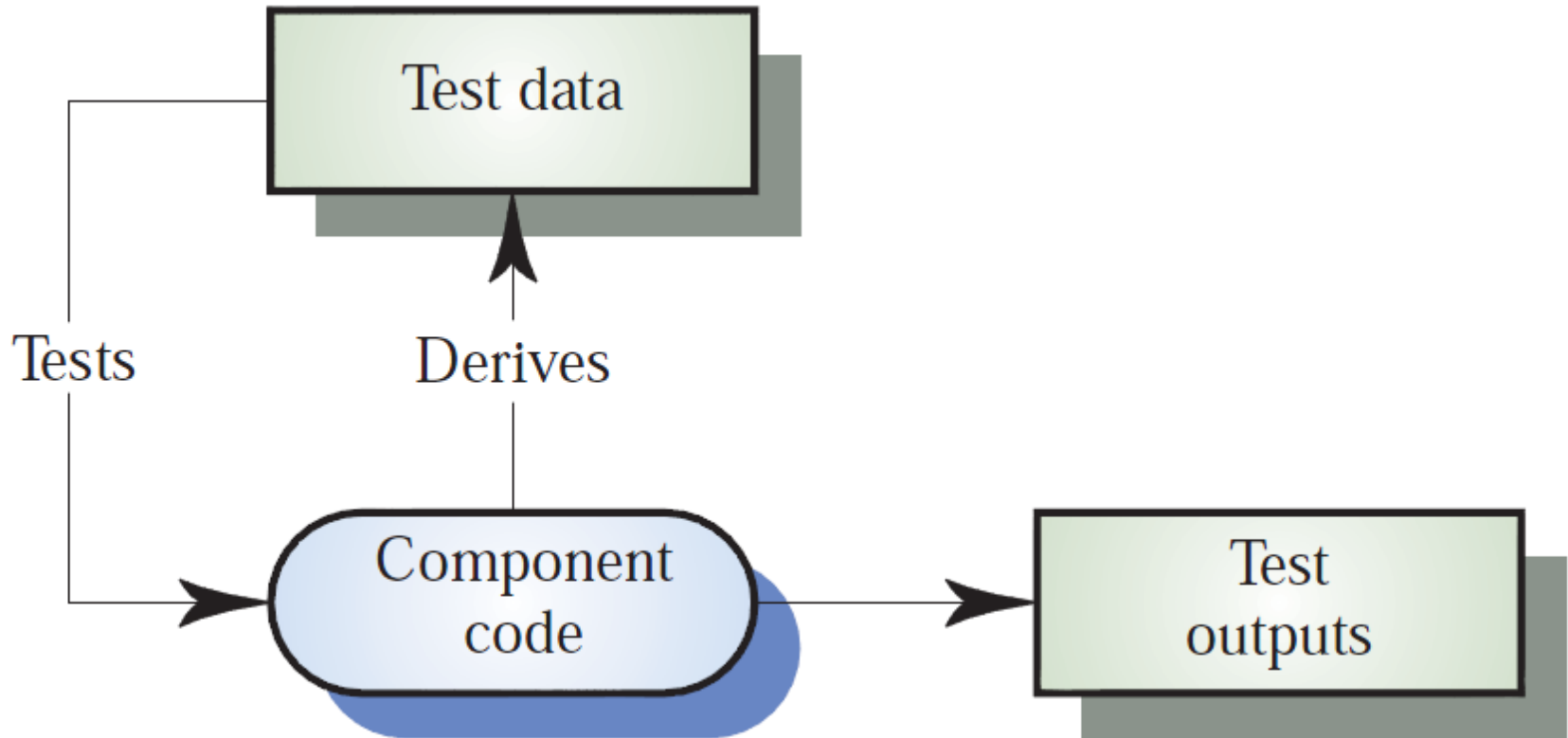
Black-Box Testing

- ▶ Pengujian ini fokus kepada persyaratan fungsional perangkat lunak. Pengujian ini memungkinkan pelaku RPL mendapatkan serangkaian kondisi input yang memenuhi persyaratan fungsional suatu program.
- ▶ Pengujian ini berusaha menemukan kesalahan dengan kategori sebagai berikut:
 - Fungsi-fungsi yang salah atau hilang
 - Kesalahan antarmuka
 - Kesalahan struktur data atau akses basisdata eksternal
 - Kesalahan kinerja
 - Kesalahan inisialisasi atau terminasi

Black – Box Testing

- Pengujian black box berusaha menemukan kesalahan dalam kategori :
 - Fungsi–fungsi yang tidak benar atau hilang
 - Kesalahan interface
 - Kesalahan dalam struktur data atau akses database eksternal
 - Kesalahan kinerja
 - Inisialisasi dan kesalahan terminasi

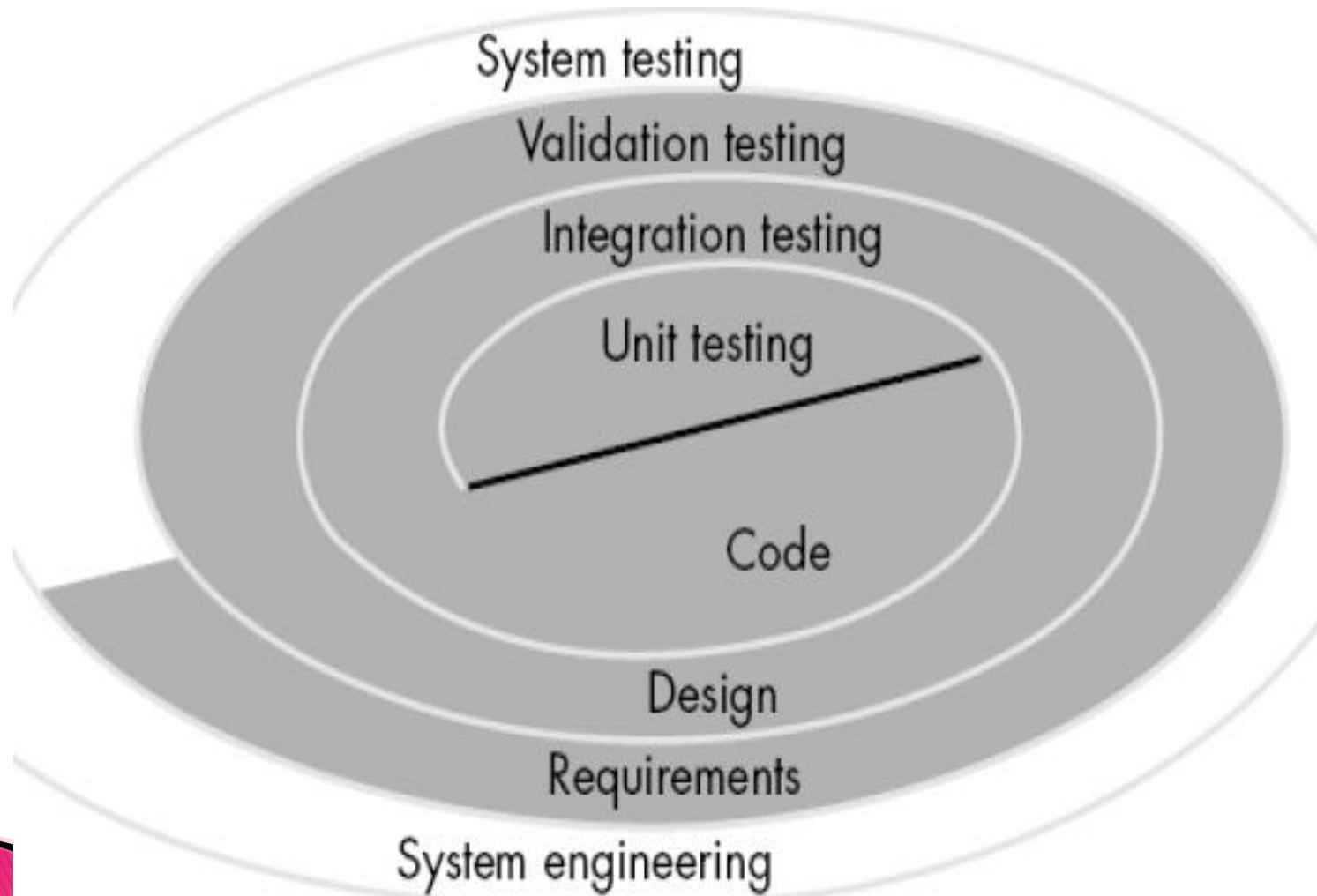
White - Box Testing



White – Box Testing (cont.)

- ▶ White – Box Testing adalah cara pengujian dengan melihat ke dalam modul untuk meneliti kode – kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak.
- ▶ Jika ada modul yang menghasilkan output yang tidak sesuai dengan proses bisnis yang dilakukan maka baris – baris program, variabel dan parameter yang terlibat pada unit tersebut akan dicek satu persatu dan diperbaiki kemudian di-compile ulang.

Strategi Pengujian Perangkat Lunak



Tugas 14

Tentukan Independent Path dari :

