

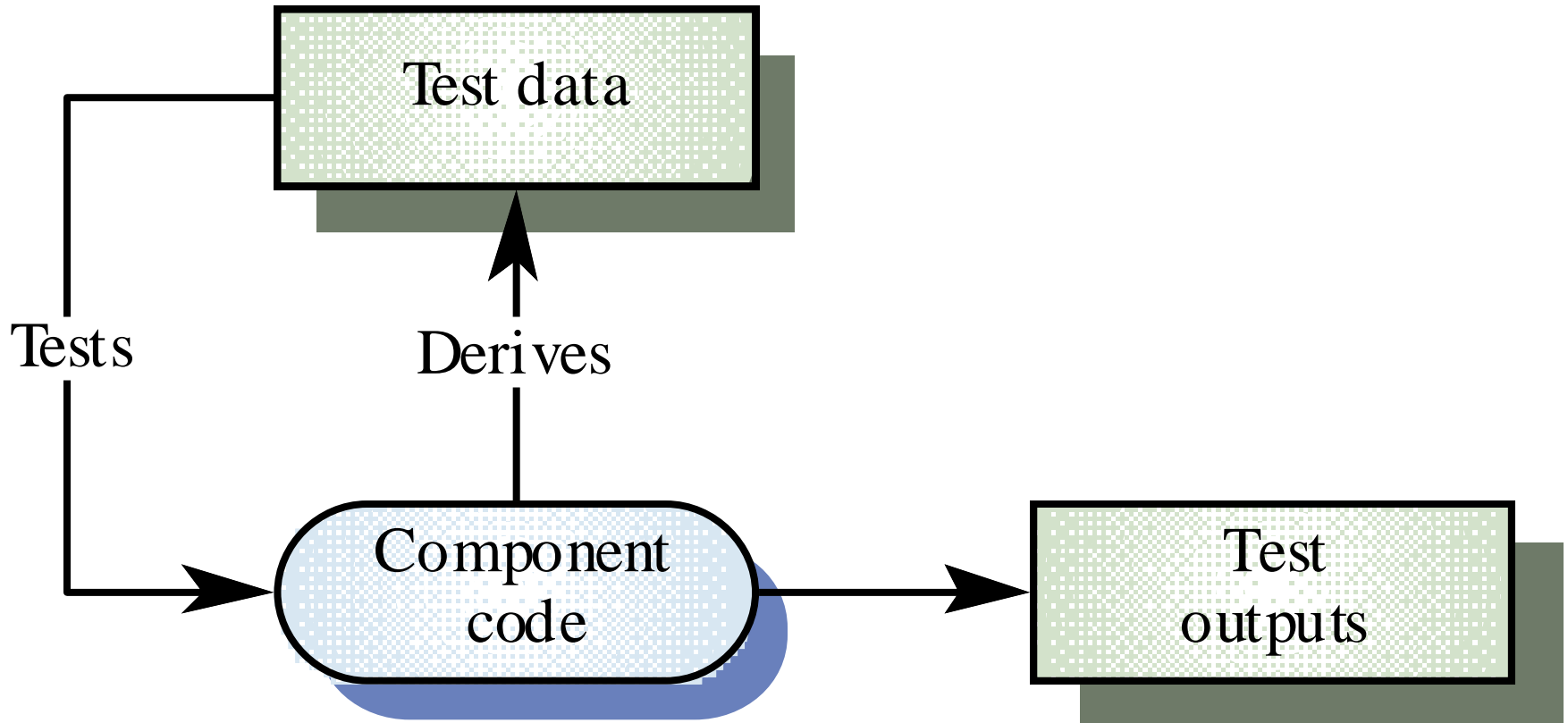
---

# **Metode Pengujian Perangkat Lunak (White Box)**

---

- Disebut juga dengan pengujian Glass Box.
- metode perancangan test case yang menggunakan struktur kontrol dari perancangan prosedural untuk mendapatkan test case.
- Digunakan untuk mengetahui cara kerja internal suatu perangkat lunak
- Pengujian dilakukan untuk menjamin operasi-operasi internal sesuai dengan spesifikasi yang telah ditetapkan dengan menggunakan struktur kendali dari prosedur yang dirancang.

# White-box testing



# Pengujian dengan metode ini diharapkan memperoleh test case sbb:

memberikan jaminan bahwa semua jalur independen suatu modul digunakan minimal satu kali

menggunakan semua keputusan logis untuk semua kondisi *true* atau *false*

mengeksekusi semua perulangan pada batasan nilai dan operasional pada setiap kondisi.

menggunakan struktur data internal untuk menjamin validitas jalur keputusan

# Beberapa metode/teknik pengujian White Box

---

- Pengujian Basis Path
- Pengujian Kondisi
- Pengujian Loop

# 1. Pengujian Basis Path

## □ Basis Path Testing

- Buat *Flow Graph Notation*
- Hitung *Cyclomatic Complexity* = salah satu dari:
  - Jumlah *region*
  - $V(G) = E - N + 2$ 
    - $E$  = jumlah busur pada *flow graph*
    - $N$  = jumlah simpul pada *flow graph*
  - $V(G) = P + 1$ 
    - $P$  = simpul predikat (simpul yang memiliki kondisi = 2 atau lebih busur yang keluar)
- Tentukan jalur bebas (*independent path*) = jalur program yang merupakan satu kumpulan perintah pengolahan atau satu kondisi pengolahan
- Siapkan kasus uji untuk setiap jalur bebas
- *Graph Matrices* = *Connection Matrices* = representasi lain dari FGN

# Pengujian Basis Path...

---

- Merupakan teknik uji coba yang diusulkan oleh **Tom McCabe**.
- Digunakan untuk **mengukur kompleksitas logis** dari desain procedural dan menggunakannya sebagai pedoman untuk menetapkan himpunan basis dari semua jalur eksekusi.
- Test case yang didapat digunakan untuk **mengerjakan basis set** yang menjamin pengerjaan setiap perintah min 1x selama uji coba.

# Pengujian Basis Path...

---

Tujuannya meyakinkan bahwa **himpunan test case akan menguji setiap path** pada suatu program paling sedikit satu kali.

Titik awal untuk path testing adalah suatu program flow graph yang menunjukkan node-node yang menyatakan program decisions (mis.: if-then-else condition) dan busur menyatakan alur kontrol



## Program Flow Graphs

---

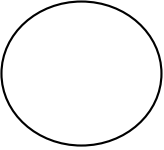

Menggambarkan alur kontrol. Setiap cabang ditunjukkan oleh path yg terpisah dan loop ditunjukkan oleh arrows looping kembali ke loop kondisi node.

Digunakan sebagai basis untuk menghitung cyclomatic complexity

Cyclomatic complexity = Jumlah edges - Jumlah Node + 2

Cyclomatic complexity menyatakan jumlah test untuk menguji control statements

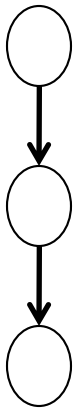
## 1. Flow Graph Notation/Grafik Alir

-  = Simpul/node, merepresentasikan satu atau lebih statement prosedural
-  = link/edge, merepresentasikan aliran kontrol

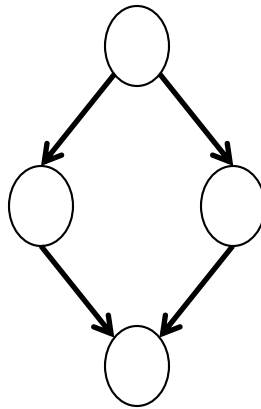
# Pengujian Basis Path...

Bentuk penulisan **Grafik Alir** :

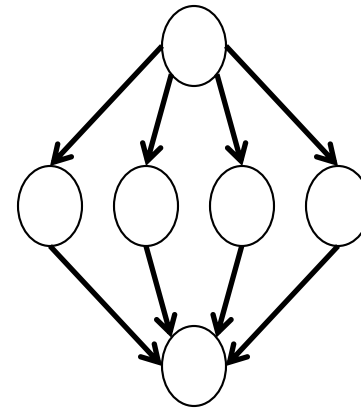
---



1.sekuensial



2.if

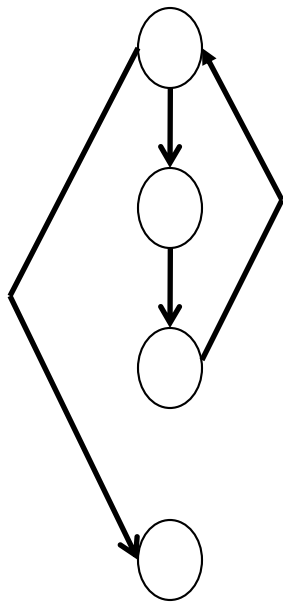


3.case

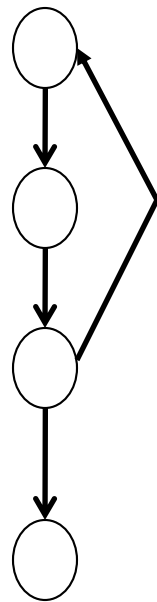
# Pengujian Basis Path...

Bentuk penulisan **Grafik Alir** :

---



4.while



5. until

# Pengujian Basis Path...

---

- ❑ Lingkaran (node/simpul), menggambarkan satu/lebih perintah prosedural dan yang berisi sebuah kondisi yang ditandai dengan dua/lebih link yang berasal darinya(Predikat). Urutan proses dan keputusan dapat dipetakan dalam satu node.
- ❑ Tanda panah (edge/link), menggambarkan aliran kontrol. Setiap node harus mempunyai tujuan node.
- ❑ Region adalah daerah yang dibatasi oleh edge dan node.  
Termasuk daerah diluar grafik alir.

## 2. Kompleksitas Siklomatis (V)

metrik perangkat lunak yang memberikan **pengukuran kuantitatif** terhadap kompleksitas logis suatu program, nilai yang didapat akan menentukan jumlah jalur independen dalam himpunan path, serta akan memberi nilai batas atas bagi jumlah pengujian yang harus dilakukan, untuk memastikan bahwa semua pernyataan telah dieksekusi sedikitnya satu kali.

### Jalur Independen

Jalur yang melalui program yang mengintroduksi sedikitnya satu rangkaian statement proses baru atau suatu kondisi baru

### Basis Set

Kumpulan dari jalur-jalur independen yang ada pada suatu PL

# Pengujian Basis Path...

---

Catatan :

Nilai yang dihitung dari Kompleksitas Siklomatis digunakan untuk menentukan jumlah jalur independen dalam basis set

Menentukan jalur independen menggunakan metrik Kompleksitas Siklomatis :

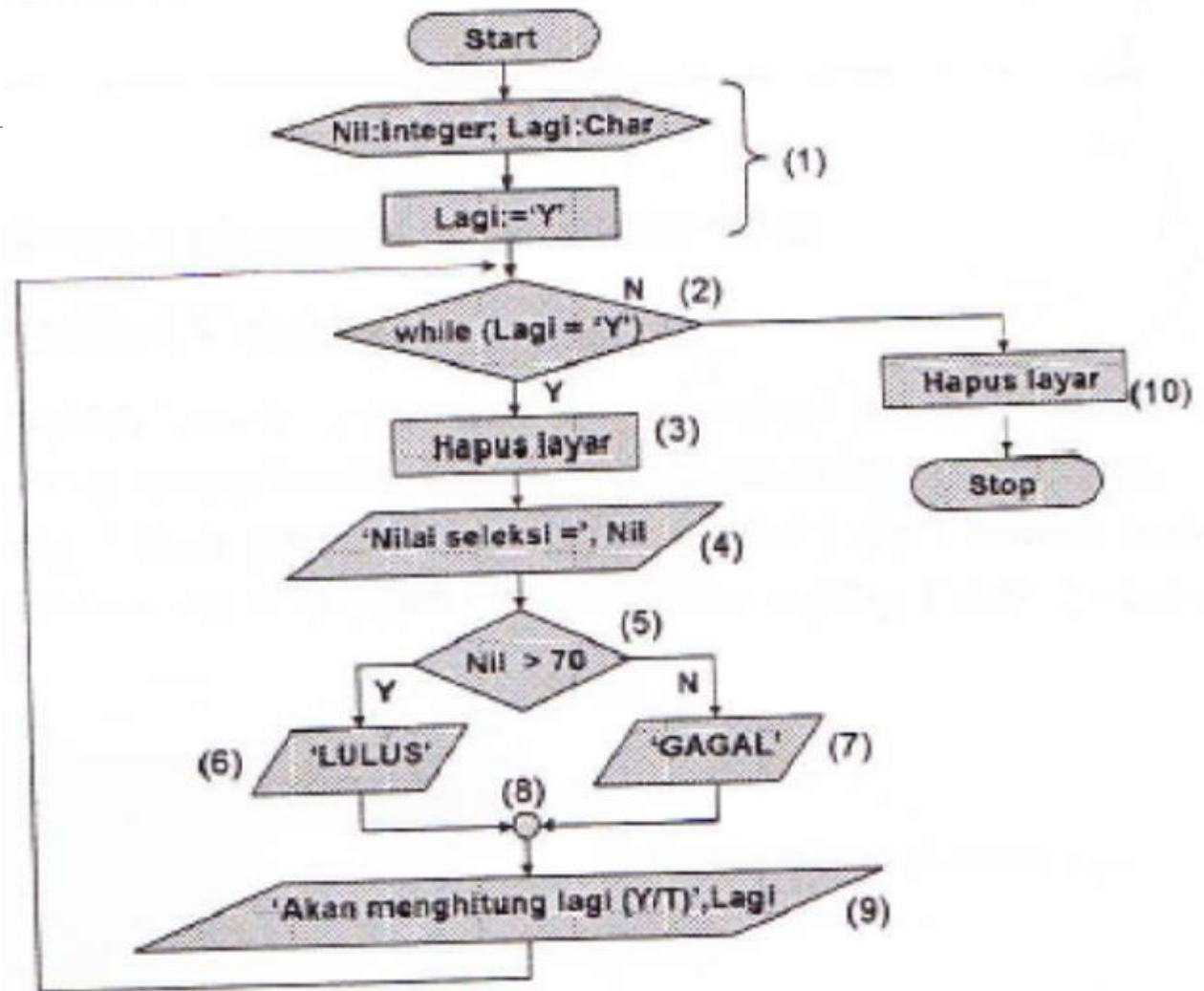
1.  $V(G)$  = jumlah region
2.  $V(G) = E - N + 2$
3.  $V(G) = P + 1$

dimana  $E$  = jml busur/link

$N$  = jml simpul

# Pengujian Basis Path...

## Contoh 1 :





## Pengujian Basis Path...

```
uses Crt;
var
  Nil:Integer; Lagi:Char;
begin
  Lagi:='Y'
  while(upcase(Lagi) = 'Y') do
  begin
    clrscr;
    write('Nilai seleksi = ');readln(Nil);
    if Nil > 70 then
    begin
      writeln('LULUS');
    end;
    else
    begin
      writeln('GAGAL');
    end;
    end;
    write('Akan menghitung lagi(Y/T) = ');readln(Lagi);
  end;
  clrscr;
end.
```

(1)

(2)

(3)

(4)

(5)

(6)

(7)

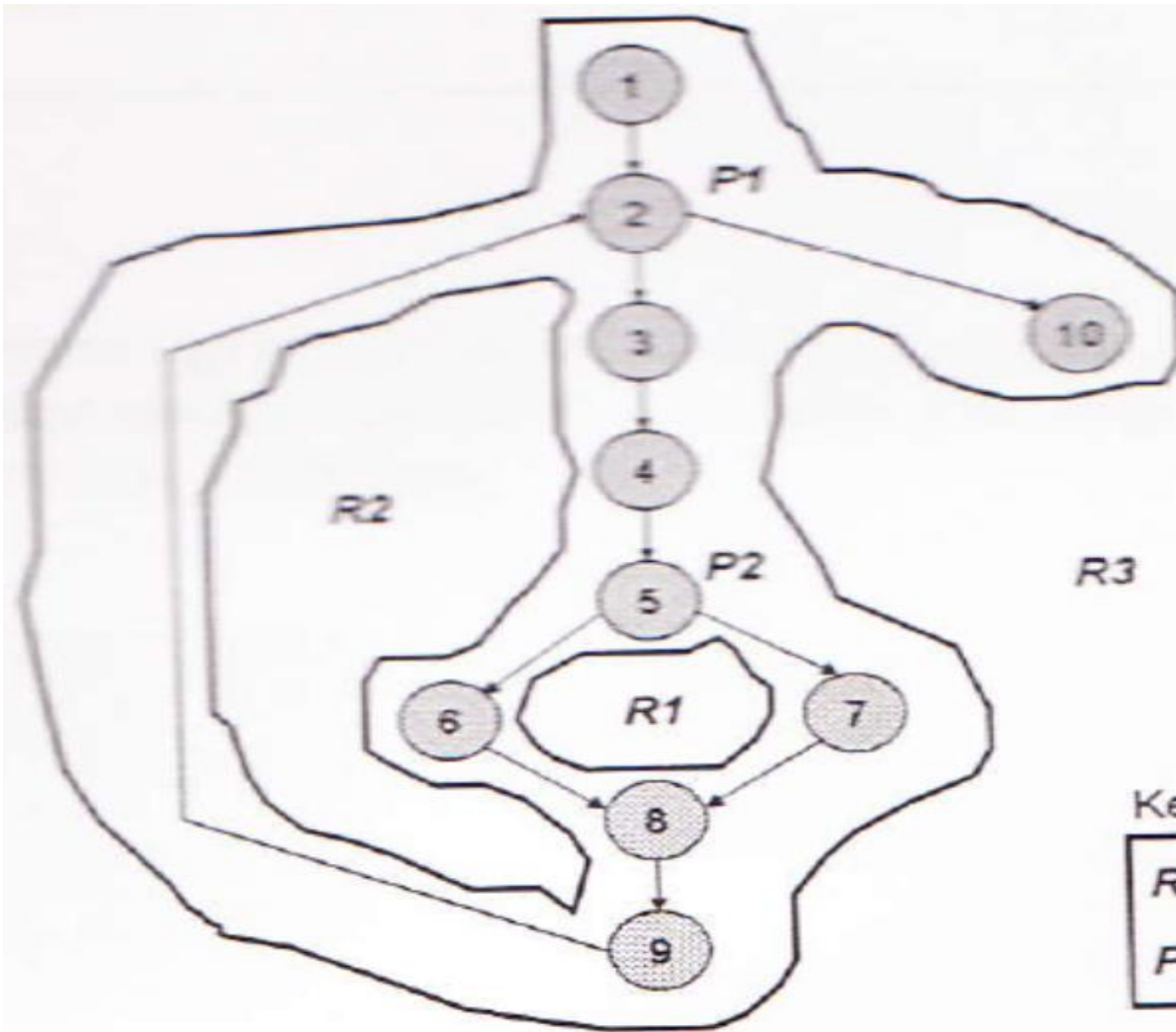
(8)

(9)

(10)

10-Jan-11

## Pengujian Basis Path...



Ket:

*R1, R2, R3 = Region*

*P1, P2 = Predikat*

# Pengujian Basis Path...

---

Menentukan jalur independen menggunakan metrik Kompleksitas Siklomatis :

1.  $V(G) = \text{jumlah region} \rightarrow 3$

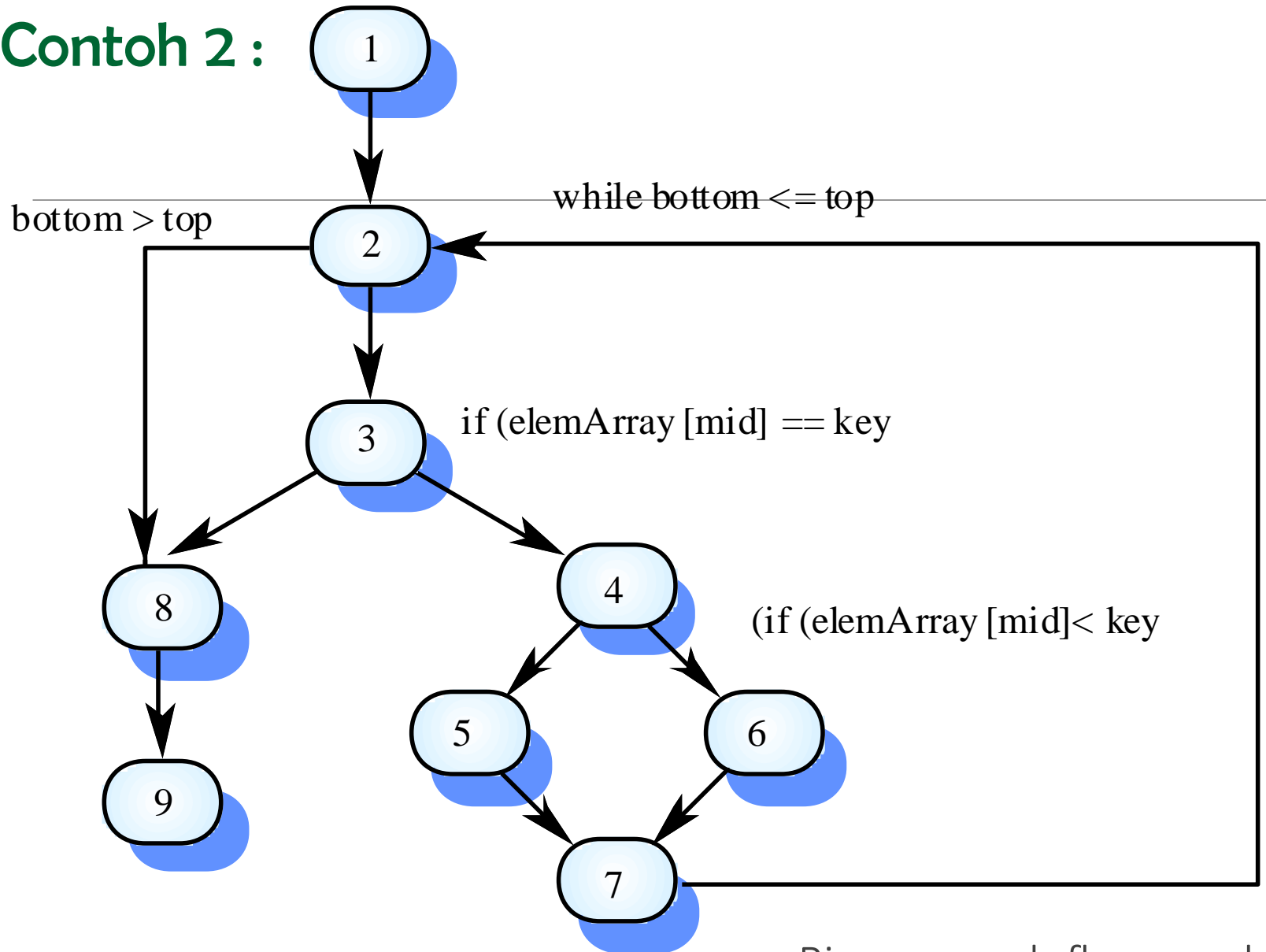
2.  $V(G) = E - N + 2 \rightarrow (11-10) + 2 = 3$

3.  $V(G) = P + 1 \rightarrow 2 + 1 = 3$

dimana  $E$  = jml busur/link

$N$  = jml simpul

## Contoh 2 :



Binary search flow graph

## Independent paths

---

1, 2, 3, 8, 9

1, 2, 3, 4, 6, 7, 2, 8, 9

1, 2, 3, 4, 5, 7, 2, 8, 9

1, 2, 8, 9

Test cases harus ditentukan sehingga semua path tsb tereksekusi.

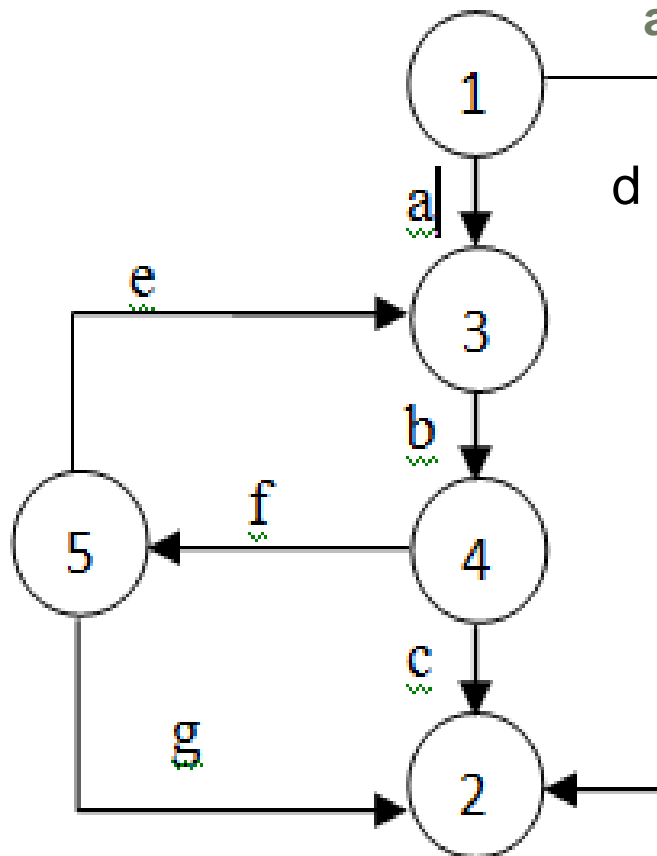
## 3. Graph Metrik

adalah matrik empat persegi yang mempunyai ukuran yang sama dengan jumlah node pada flowgraph

metrik merupakan software yang dikembangkan untuk membantu uji coba basis path atau struktur data

Masing-masing baris dan kolom mempunyai hubungan dengan node yang telah ditentukan dan pemasukan data matrik berhubungan dengan hubungan (edge) antar node.

# Pengujian Basis Path...



awal

Akhir(tujuan)

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

gambar Graph Matrik

$$\begin{aligned}
 2-1 &= 1 \\
 1-1 &= 0 \\
 2-1 &= 1 \\
 2-1 &= \frac{1}{3} + 1 \\
 &= 4
 \end{aligned}$$

# Pengujian Basis Path...

---

Hubungan bobot menyediakan tambahan informasi tentang aliran kontrol. Secara simpel hubungan bobot dapat diberi nilai 1 jika ada hubungan antara node atau nilai 0 jika tidak ada.

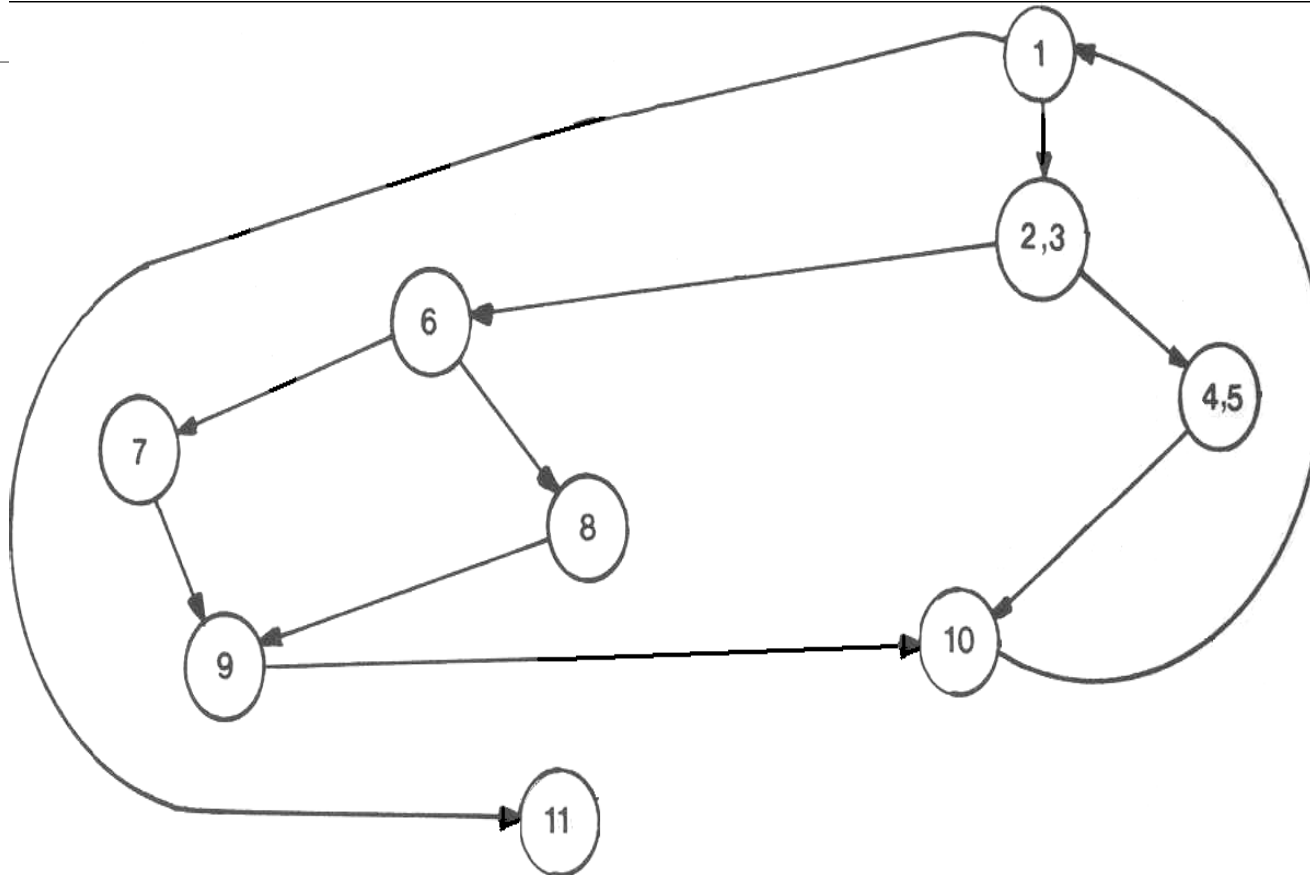
Dapat juga hubungan bobot diberi tanda dengan :

- Kemungkinan link (edge) dikerjakan
- Waktu yang digunakan untuk proses selama traversal dari link
- Memori yang diperlukan selama traversal link
- Sumber daya yang diperlukan selama traversal link



# Latihan Pengujian Basis Path...

Soal :



Tentukan independen path..?

## Latihan Pengujian Basis Path...

**Jawab :**

---

Dari gambar:

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

Path 1,2,3,4 yang telah didefinisikan diatas merupakan basis set untuk diagram alir.

# Latihan Pengujian Basis Path...

---

Flowgraph mempunyai 4 region

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

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

Jadi cyclomatic complexity untuk flowgraph adalah 4