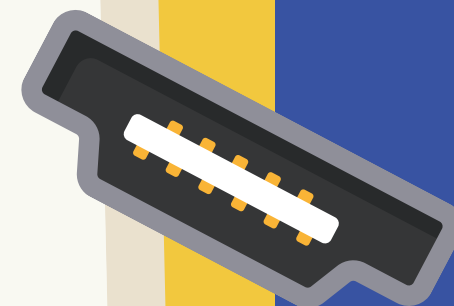
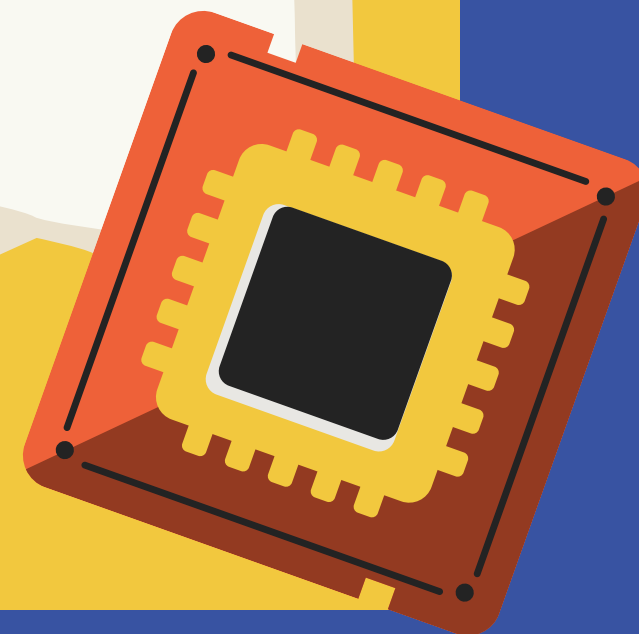
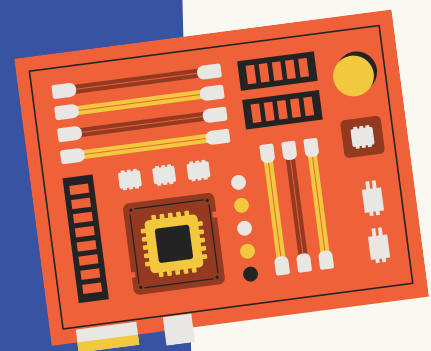
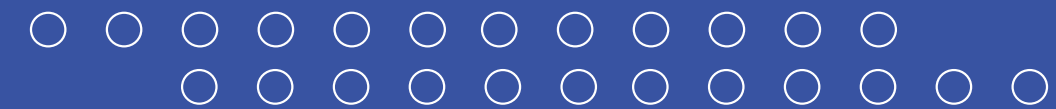


# WHITE BOX





# PENGERTIAN

Metode White box ini adalah suatu metode desain test case yang menggunakan struktur control desain procedural untuk memperoleh test case Dengan menggunakan metode pengujian white box

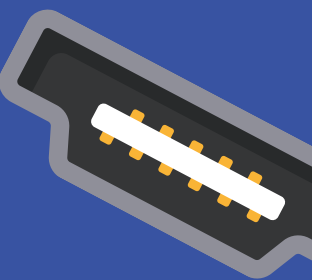
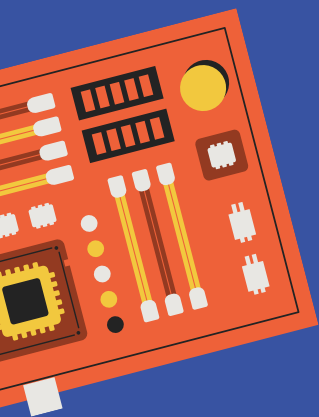


# PEREKAYASAAN SISTEM DAPAT DI LAKUKAN DENGAN TEST CASE

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 (perulangan) pada batasan mereka dan pada batas operasional mereka



# KOMPONEN DALAM PENGUJIAN WHITE BOX



Uji Coba Basis Path



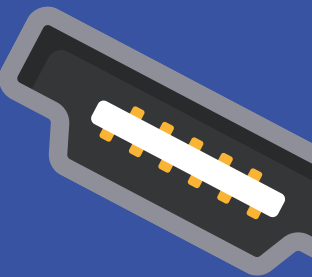
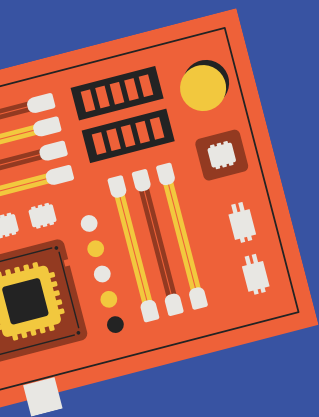
Cyclomatic Complexity



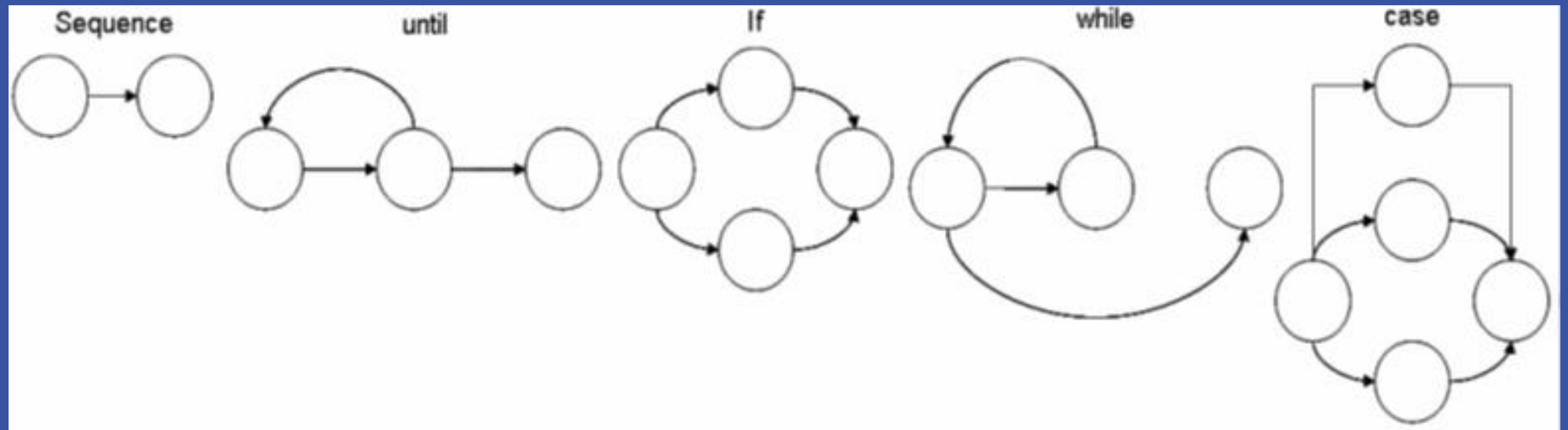
Melakukan Test Case

# UJI COBA BASIS PATH

Merupakan teknik uji coba white box yang diusulkan Tom McCabe. Metode ini memungkinkan perancang test case mendapatkan ukuran kompleksnya logical dari perancangan prosedural dan menggunakan ukuran ini sebagai petunjuk untuk mendefinisikan basis set dari jalur pengerjaan. Pengujian white box perangkat lunak didasarkan pada pengamatan yang teliti terhadap detail procedural. Test case yang didapat digunakan untuk mengerjakan basis set yang menjamin pengerjaan setiap perintah minimal satu kali selama uji coba



# ***DIAGRAM ALIR***



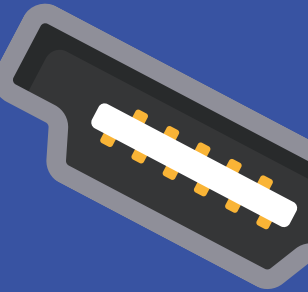
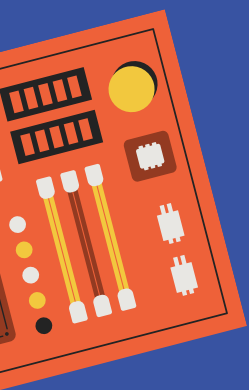
# CYCLOMATIC COMPLEXITY

Cyclomatic complexity adalah metrik software yang menyediakan ukuran kuantitatif dari kekompleksan logikal program. Apabila digunakan dalam konteks metode uji coba basis path, nilai yang dihitung untuk cyclomatic complexity 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

# MELAKUKAN TEST CASE

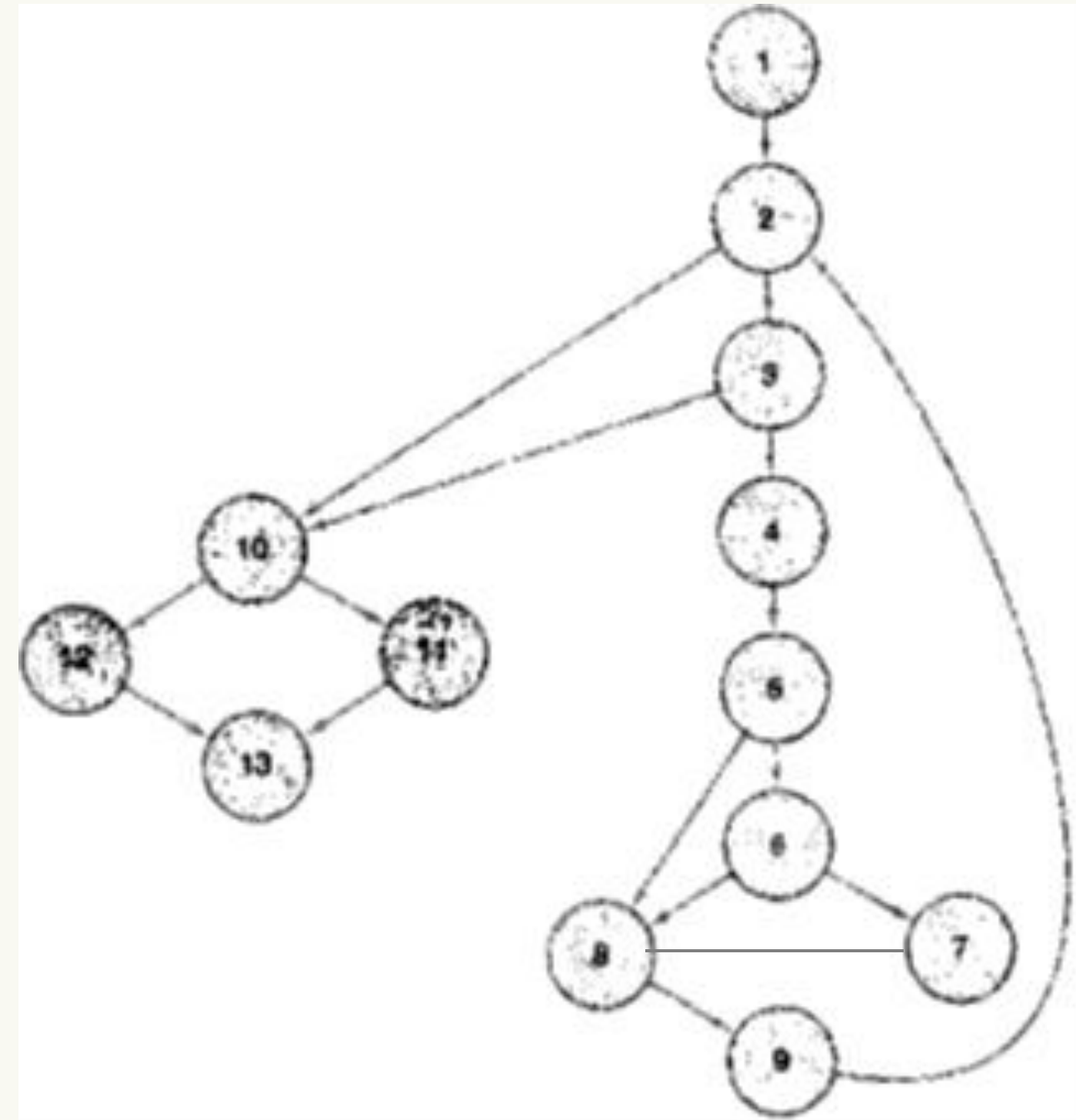
Metode uji coba basis path juga dapat diterapkan pada perancangan prosedural rinci atau program sumber. Prosedur rata-rata pada bagian berikut akan digunakan sebagai contoh dalam pembuatan test case



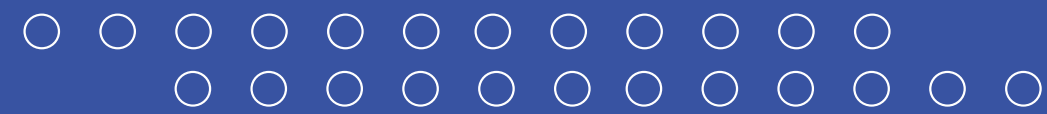


Langkah-langkah  
pembuatan test  
case

*Bagian 1*



Dengan mempergunakan perancangan prosedural atau program sumber sebagai dasar, digambarkan diagram alirnya



## Langkah-langkah pembuatan test case

### *Bagian 2*

Tentukan cyclomatic complexity untuk  
diagram alir yang telah dibuat:

$$V(G) = 6 \text{ Region}$$

$$V(G) = 17 \text{ Edge} - 13 \text{ Node} + 2 = 6$$

$$V(G) = 5 \text{ Predicate Node} + 1 = 6$$



## Langkah-langkah pembuatan test case

### *Bagian 3*

Tentukan independent path pada  
flowgraph

Dari hasil perhitungan cyclomatic  
complexity terdapat 6 independent path  
Yaitu :

Path 1 : 1-2-10-11-13

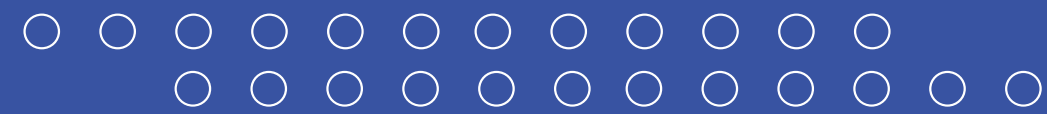
Path 2 : 1-2-10-12-13

Path 3 : 1-2-3-10-11-13

Path 4 : 1-2-3-4-5-8-9-2-..

Path 5 : 1-2-3-4-5-6-8-9-2-..

Path 6 : 1-2-3-4-5-6-7-8-9-2-...

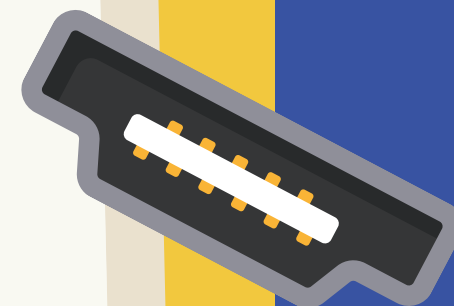
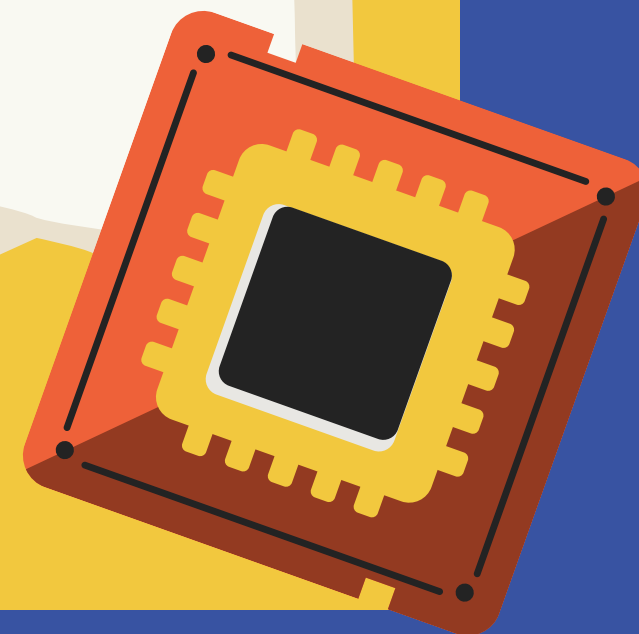
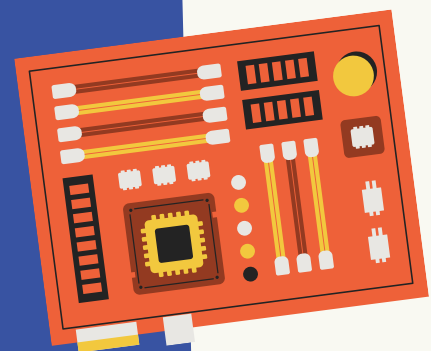


## Langkah-langkah pembuatan test case

### *Bagian 4*

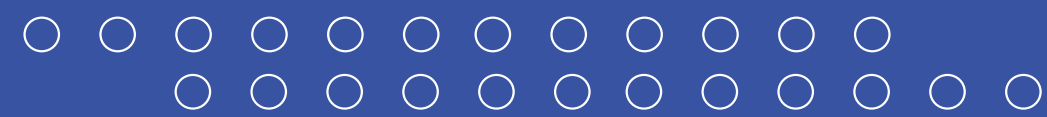
Buat test case yang akan mengerjakan masing-masing path pada basis set. Data yang dipilih harus tepat sehingga setiap kondisi dari predicate node dikerjakan semua

# BLACK BOX





Metode ujicoba black box memfokuskan pada keperluan fungsional dari software. Karna itu ujicoba black box memungkinkan pengembang software untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. pengujian black box berkaitan dengan pengujian yang dilakukan pada interface perangkat lunak. Ujicoba black box bukan merupakan alternatif dari ujicoba white box, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode white box

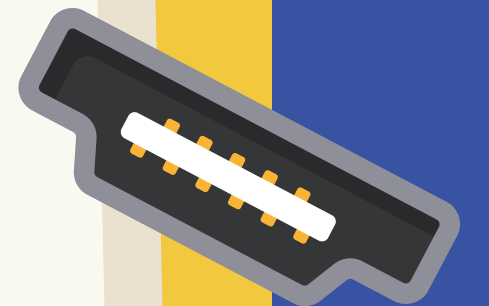
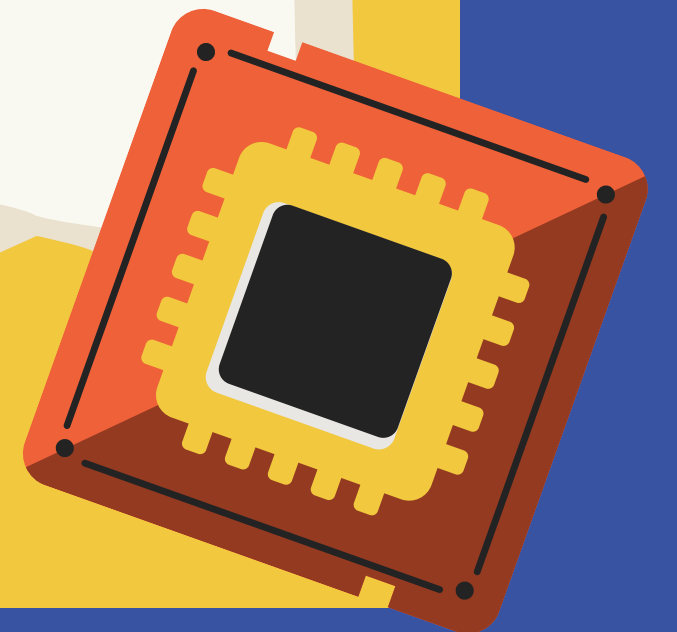
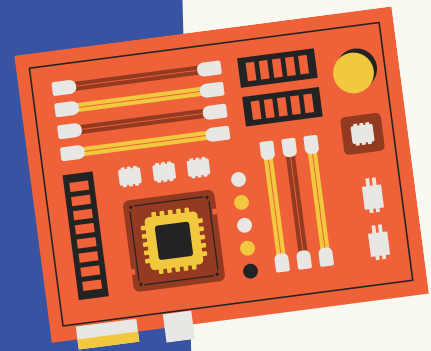


Ujicoba black box berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan performa
5. kesalahan inisialisasi dan terminasi

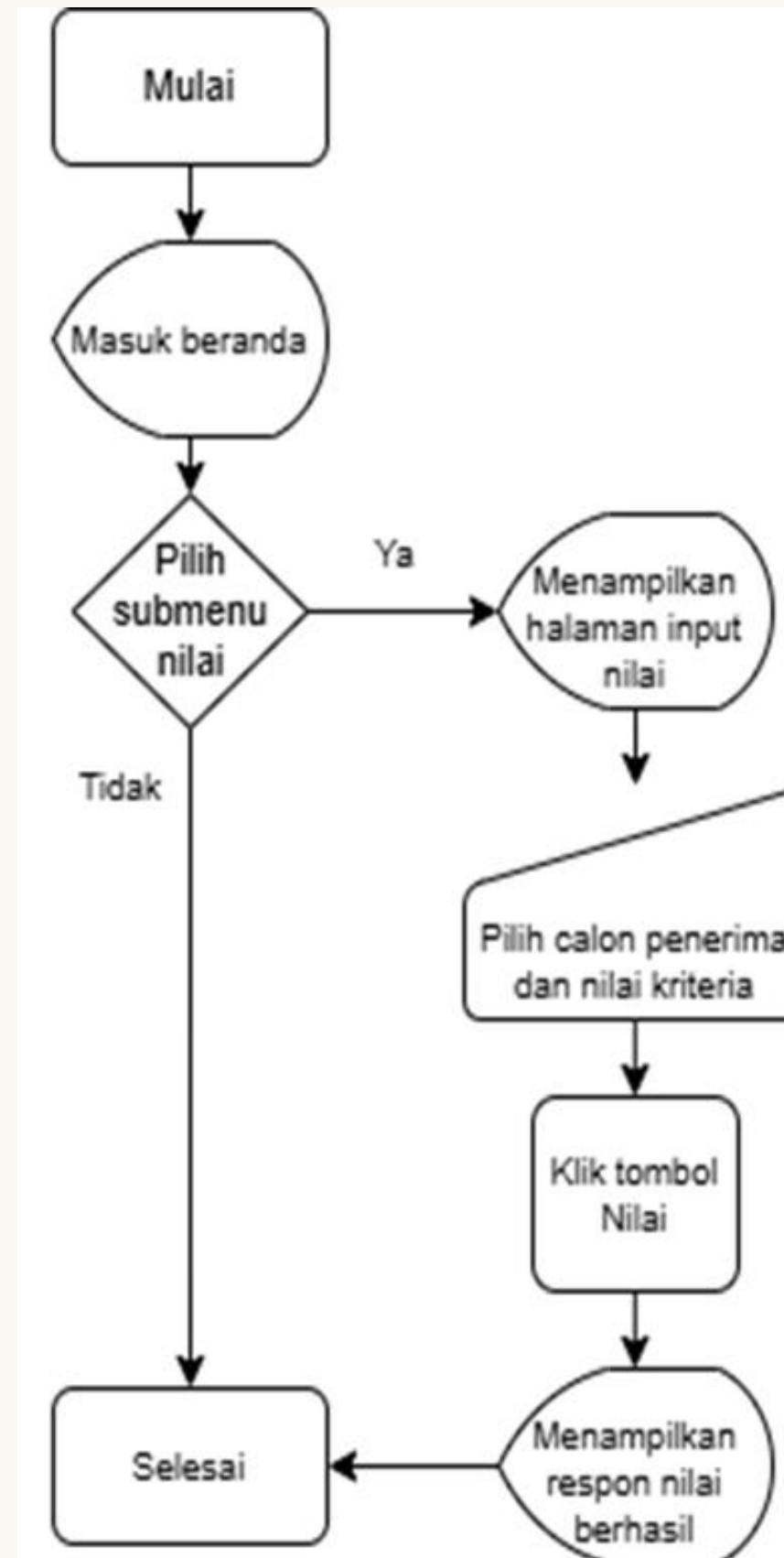
Tidak seperti metode white box yang dilaksanakan diawal proses, ujicoba black box diaplikasikan di beberapa tahapan berikutnya. Karena ujicoba black box dengan sengaja mengabaikan struktur kontrol, sehingga perhatiannya difokuskan pada informasi domain

# CONTOH WHITE BOX



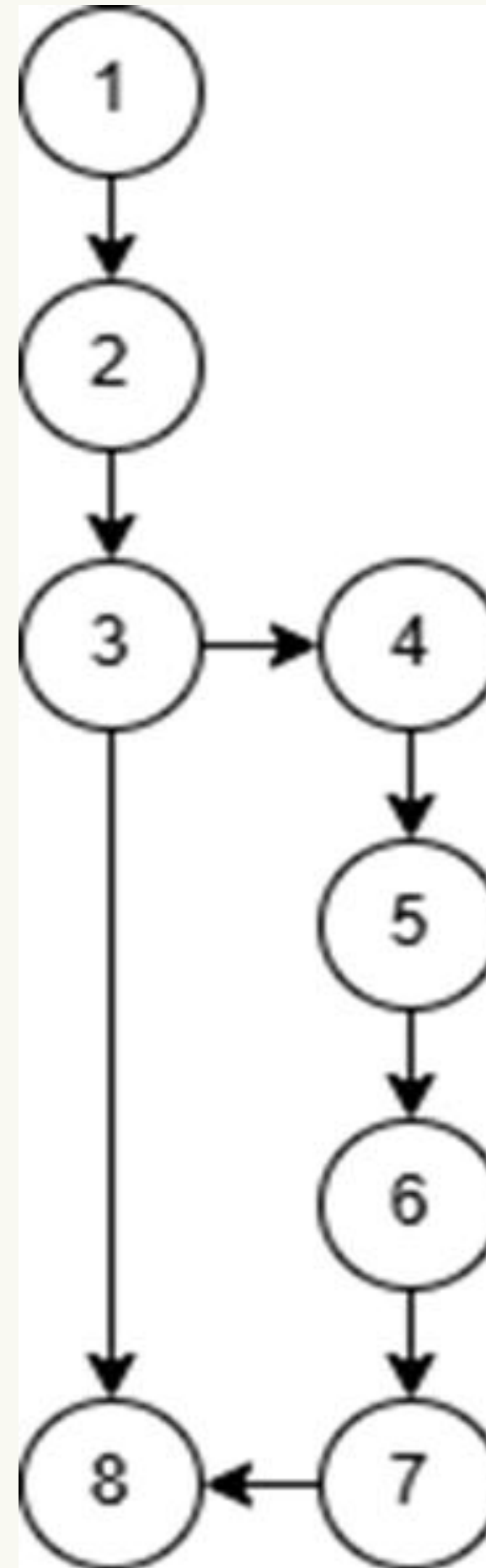


## CONTOH FLOWCHART PADA PENGUJIAN WHITE BOX

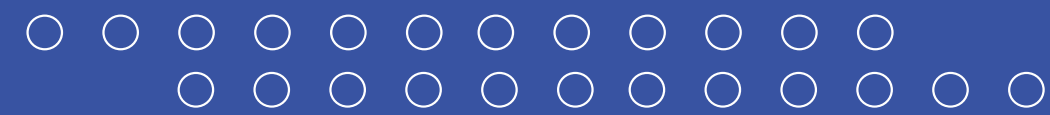


Flowchat penilaian  
calon penerima  
bantuan

CONTOH FLOWGRAPH  
PADA PENGUJIAN  
WHITE BOX

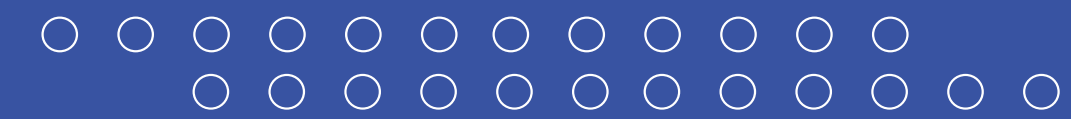


Flowgraph penilaian  
calon penerima  
bantuan



## Pseudocode

- Node 1 : Start / mulai
- Node 2 : Masuk Beranda
- Node 3 : Pilih submenu input nilai?  
Jika ya ke node 4, jika tidak ke node 8
- Node 4 : Menampilkan halaman input nilai
- Node 5 : Pilih calon penerima dan nilai kriteria
- Node 6 : Klik tombol nilai
- Node 7 : Menampilkan respon nilai berhasil
- Node 8 : Finish/ selesai



## Perhitungan Cyclomatic Complexity (CC)

Dari flowgraph untuk modul Input nilai di atas di ketahui bahwa nilai :

-Region (R) = 3 -> R1 dan R2

-Predicate Node (P) = 1

-Node = 8

-Edge = 8

$$\begin{aligned} 1.V(G) &= E - N + 2 \\ &= (8 - 8) + 2 = 2 \end{aligned}$$

$$\begin{aligned} 2.V(G) &= (\text{Predicate Node (P)} + 1) \\ &= 1 + 1 = 2 \end{aligned}$$

3. Cyclomatic Complexity (CC) = R1 dan R2 = 2

Independent Path untuk modul Input  
nilai yaitu:

R1 = 1-2-3-8

R2 = 1-2-3-4-5-6-7-8

Berdasarkan hasil pengujian di atas  
diperoleh :

- $V(G) = 2$

-Cyclometric Complexity (CC) = 2

Maka dapat disimpulkan bahwa alur  
logika untuk modul Input nilai adalah  
efektif dan efisien

Independent Path

**TERIMA  
KASIH..**

