

# 14. PENGUJIAN PERANGKAT LUNAK

**14.1 Dasar-dasar Pengujian**

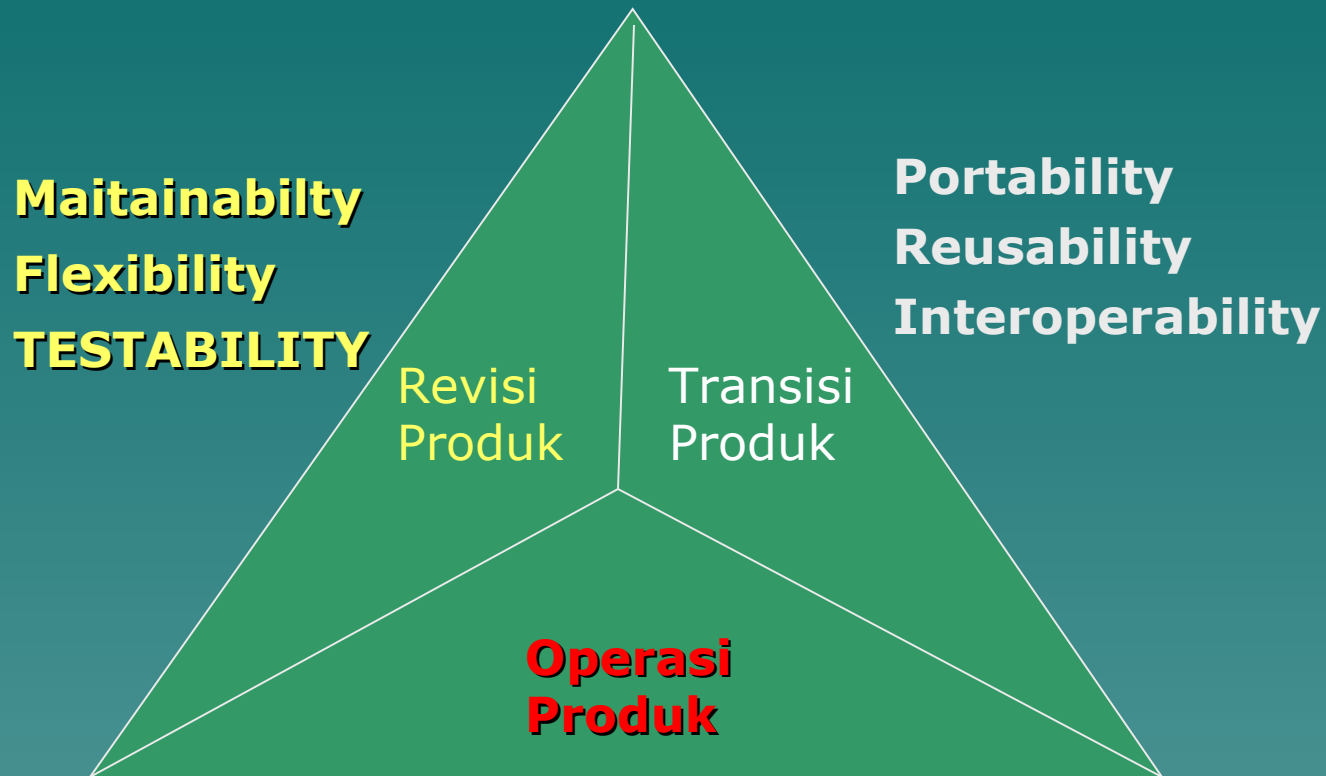
**14.2 Teknik Pengujian**

**14.3 Strategi Pengujian dan V&V**



# 14.1 Dasar-dasar Pengujian

Metrik Kualitas PL



**Correctness, Reliability, Usability, Integrity, Efisiensi**

**Faktor Kualitas PL McCall**

# Testability

Karakteristik PL yang dapat diuji :

- ◆ Operabilitas
- ◆ Observabilitas
- ◆ Kontrolabilitas
- ◆ Dekomposabilitas
- ◆ Kesederhanaan
- ◆ Stabilitas
- ◆ Kemampuan untuk dapat dipahami

Sebelum melakukan pengujian perlu dipersiapkan **Test Case** terlebih dahulu agar diperoleh kemungkinan tertinggi dalam menemukan kesalahan dengan waktu dan usaha yang minimum. Desain Test Case dapat dilakukan melalui berbagai teknik pengujian sbb.

# 14.2 Teknik Pengujian

## 14.2.1 Pengujian White-Box (Glass Box)

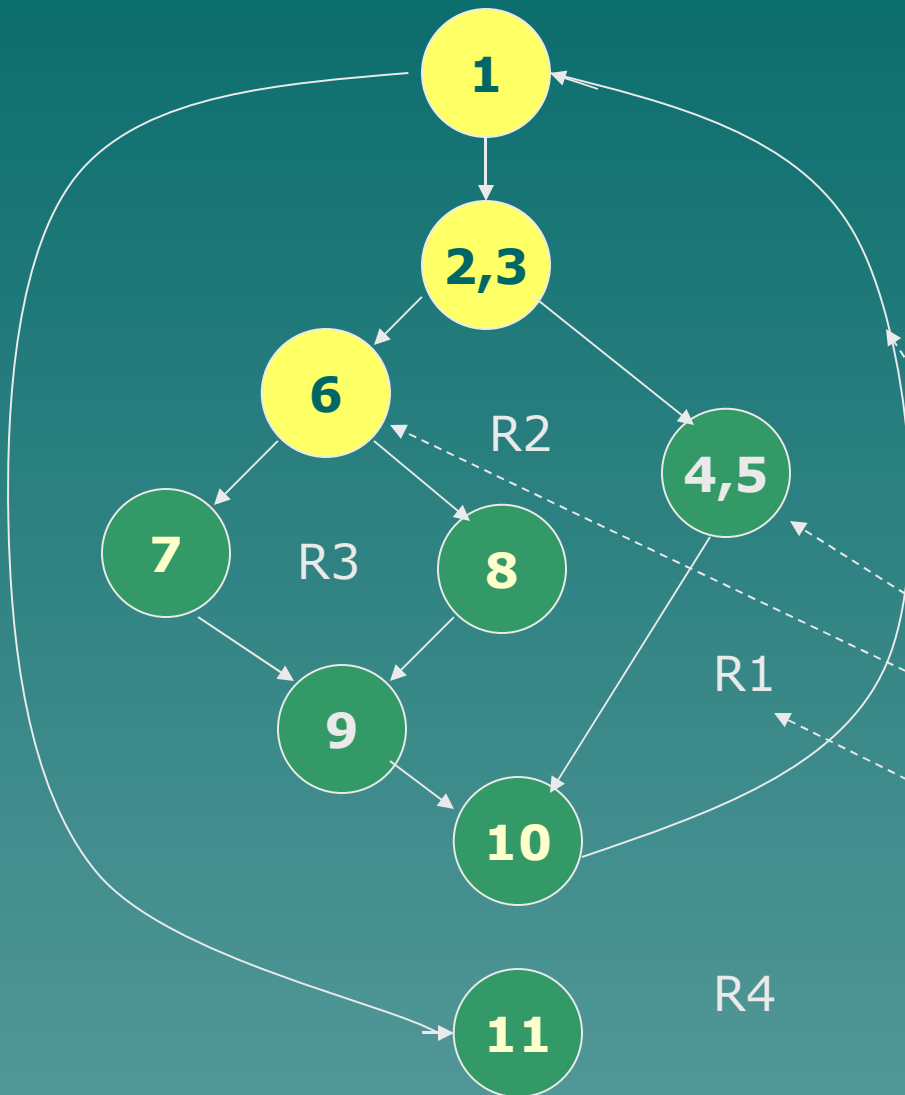
- Semua jalur independen pada suatu modul ditelusuri minimal 1 kali
- Semua jalur keputusan logis True / False dilalui
- Semua loop dieksekusi pada batas yang tercantum dan batas operasionalnya
- Struktur data internal digunakan agar validitas terjamin.

### 2. Pengujian Basis Path

- Merupakan salah satu teknik White Box
- Merupakan salah satu teknik pengujian struktur kontrol
- Menjamin semua statemen dalam setiap jalur independen program dieksekusi minimal 1 kali.

Perhitungan jalur independen dapat dilakukan melalui metrik Cyclomatic Complexity

# Cyclomatic Complexity



- ◆ Didasarkan pada teori graph.
  - ◆ Desain prosedural diterjemahkan dalam suatu grafik alir.
  - ◆ Dihitung melalui 4 cara
    - $V(G)$  = jumlah region grafik alir
    - $V(G)$  = jumlah path
    - $V(G) = E - N + 2$
    - $V(G) = P + 1$
- E : Edge  
N : Node  
P : simpul predikat (IF / percabangan)  
 $R_i$  : Region ke I [MCCL92]

Contoh  $V(G) = 4$   
 $V(G) > 10$  : more troublesome and less reliable[MART88].

## 1. **Pengujian Struktur Kontrol**

Teknik ini merupakan perbaikan dari basis path yang meliputi :

- Pengujian **Kondisi** : didasarkan pada struktur kontrol (=, <, >, not, and, dsb.)
- Pengujian **Aliran Data** : didasarkan pada adanya hubungan antar-statement pada program.
- Pengujian **Loop** : berfokus pada validitas konstruksi loop.

## 14.2.2 Pengujian Black-Box

Pengujian ini berfokus pada persyaratan fungsional PL dan merupakan komplemen dari pengujian White-Box. Hal tsb dapat dicapai melalui :

- 3. Pengujian Graph-based** : dimulai dengan membuat grafik sekumpulan node yang mempresentasikan objek(misal New File, Layar baru dg atributnya), link (hubungan antarobjek), node-weight (misal nilai data tertentu seperti atribut layar, perilaku), dan link-weight (karakteristik suatu link, misal menu select)
- 4. Equivalence Partitioning** : membagi domain input untuk pengujian agar diperoleh kelas-kelas kesalahan (misal kelompok data karakter, atau atribut yang lain)
- 5. Analisis Nilai Batas** : pengujian berdasarkan nilai batas domain input.
- 6. Pengujian Perbandingan** : disebut juga pengujian back-to-back yang diterapkan pada pada suatu versi PL atau PL redundan untuk memastikan konsistensinya.

## 14.2.3 Pengujian untuk Aplikasi dan Lingkungan Khusus

2. **Pengujian GUI** : dilakukan melalui sederetan check list untuk menguji tampilan.
3. **Pengujian Arsitektur Client – Server** : berkaitan dengan sifat pemrosesan terdistribusi, kinerja, kehadiran sejumlah platform HW yang berbeda, kompleksitas komunikasi data dan jaringan, kebutuhan akan layanan client multiple, dan persyaratan koordinasi yang dibebankan pada server.
4. **Pengujian Dokumentasi dan Help** : didekati dalam 2 fase, yakni **FTR** dan **live test** yang dikaitkan langsung pada penggunaan nyata.
5. **Pengujian Sistem Real Time** : berkaitan dengan penanganan **interupsi, timing data, proses paralel**. Langkah<sup>2</sup> yang dilakukan meliputi Pengujian Task, Pengujian Perilaku, Pengujian antar-task, dan Pengujian Sistem.





# 14.3 Strategi Pengujian dan V&V

## 1. Strategi Pengujian

mengintegrasikan metode desain test case PL ke dalam sederetan langkah yang bertujuan untuk menghasilkan perangkat lunak yang berhasil.

## 3. Verifikasi & Validasi

- harus dilakukan pada tiap tahap pengembangan sistem,
- dengan dokumentasi dari tahap sebelumnya.

**Verifikasi** → *are we building the product right* : untuk memastikan bhw PL scr **tepat mengimplementasikan** suatu fungsi tertentu.

**Validasi** → *are we building the right product* : untuk memastikan bhw PL dpt ditelusuri hingga ke **persyaratan pelanggan**. Validasi mrpk seri akhir pengujian yang melibatkan user (**User Testing**) baik yang berlangsung dlm lingkungan pengembang (**Alpha Test**) maupun di luar lingkungan pengembang (**Beta Test**). Teknik pengujian Black-Box digunakan scr eksklusif dalam validasi.



## 1. Component Testing

Pengujian terhadap komponen sistem, seringkali menggunakan teknik pengujian **White-Box**.

### b. Unit Testing

- pengujian tahap awal
- pengujian komponen secara terpisah
- unit-unit terkecil diuji : function, procedure, subprogram, dll

### c. Module Testing (modul memadukan beberapa komponen)

- menguji interaksi antarunit
- menguji perilaku modul

## 1. Integration Testing

Pengujian terhadap integrasi antarmodul scr **Top-Down, Bottom-Up, dan Pengujian Regresi** (test case ulang pada suatu subset untuk memastikan tidak adanya perubahan akibat pengujian). Integration Test dilakukan baik dengan teknik pengujian **Black-Box** maupun sebagian **White-Box**.

### b. Sub-system Testing

Pengujian terhadap **antarmuka** pada modul-modul yang sudah diintegrasikan

### d. System Testing

- Pengujian terhadap perilaku sistem
- Apakah sistem sesuai dengan spesifikasi

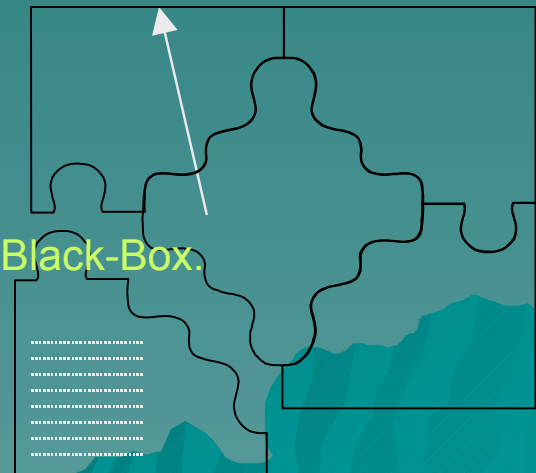
## 1. User Testing

Pengujian oleh user, seringkali menggunakan teknik **Black-Box**.

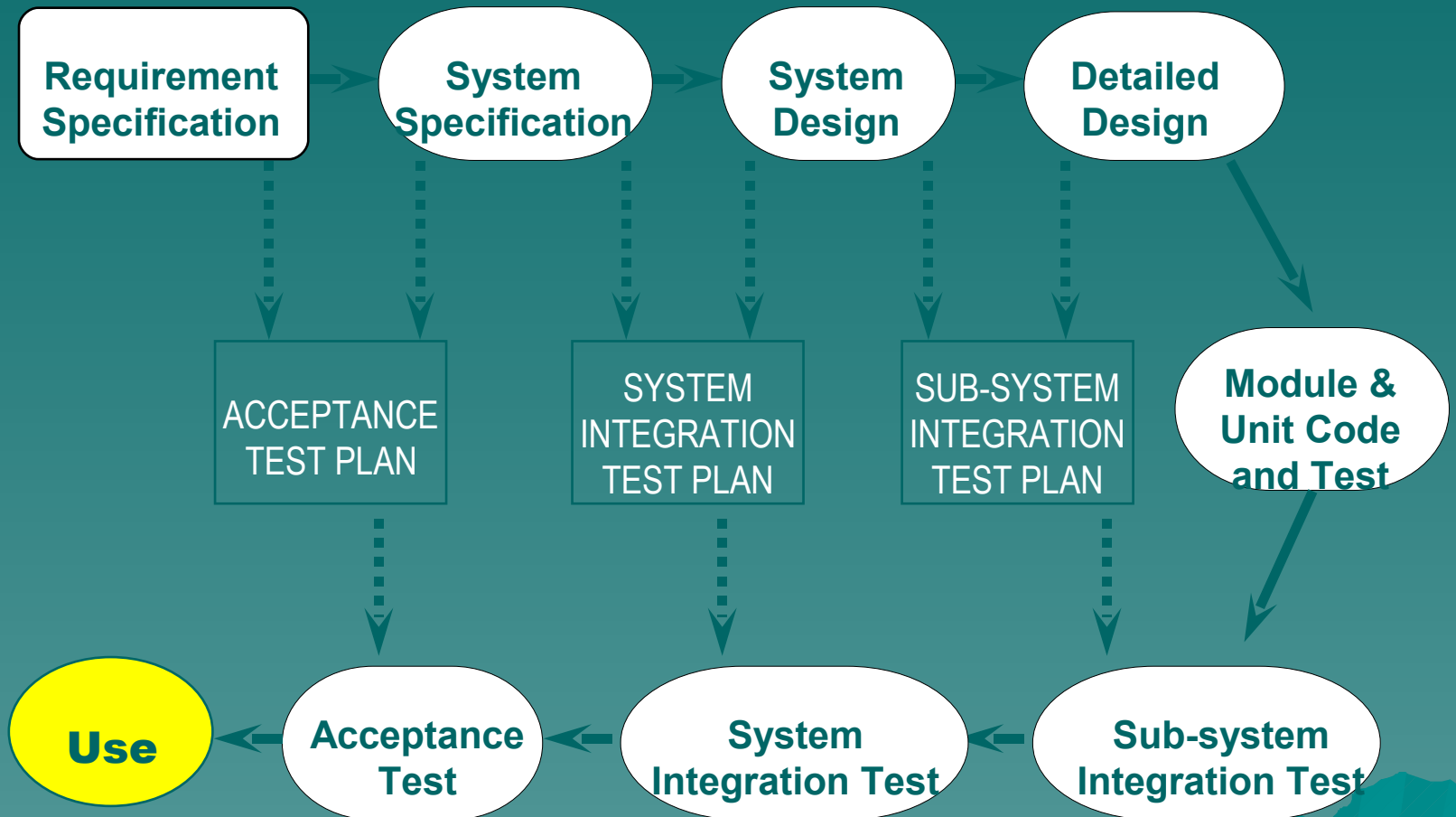
### a. Pengujian Tahap Akhir

### b. Pengujian User (acceptance testing)

- diuji dengan data sebenarnya
- pengujian terhadap fasilitas yang tersedia
- menilai kinerja (performance)



## 14.3.2 Perencanaan Pengujian



\*\*\*

