



Perancangan Pengujian -Desain dan Analisis Sistem -

NISA'UL HAFIDHOH, M.T.

PRODI DIII – TEKNOLOGI INFORMASI

Pengujian

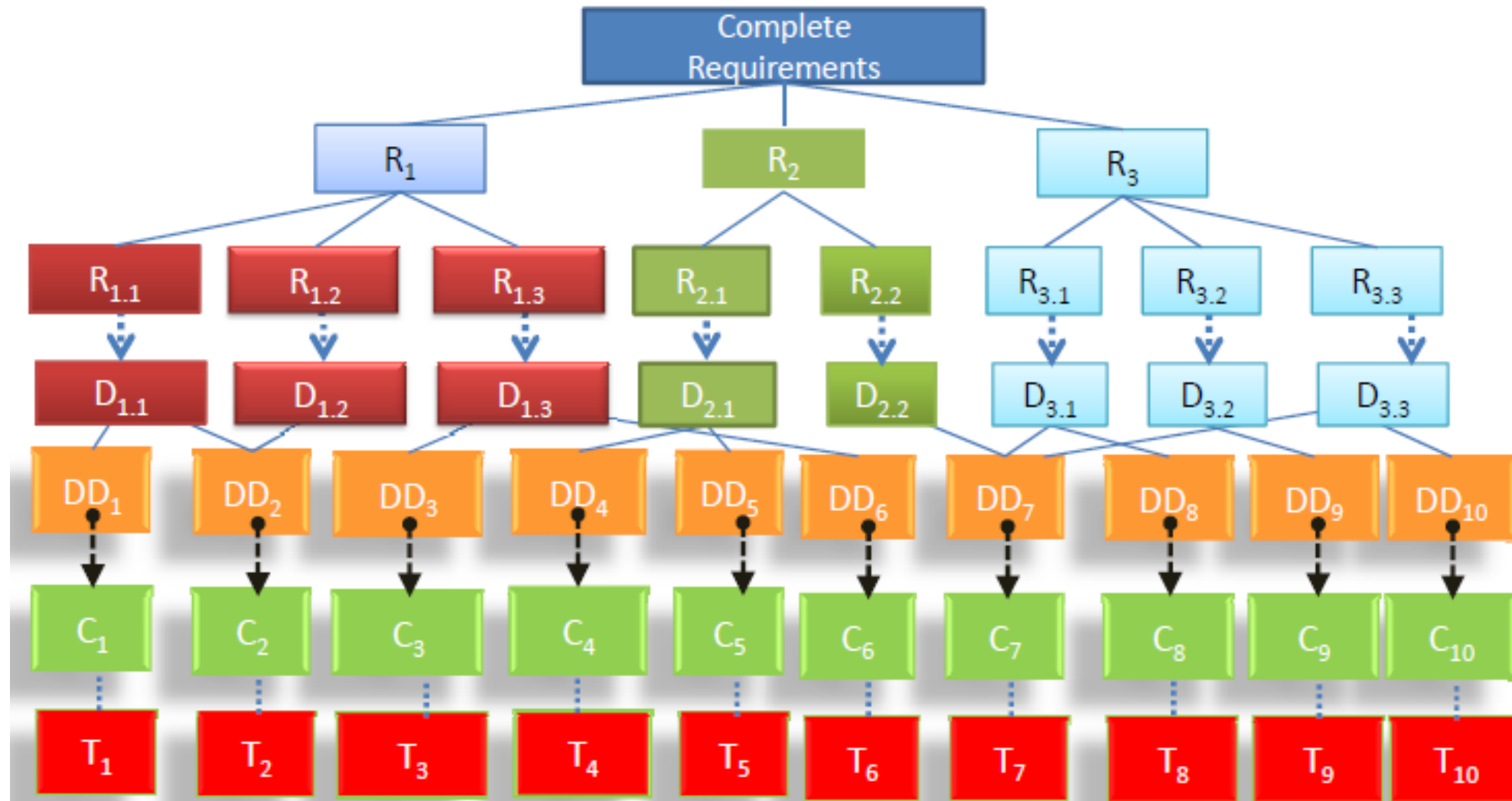
Testing adalah proses menganalisa suatu entitas software untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang tidak diinginkan (*defect/errors/bugs*) dan mengevaluasi fitur-fitur dari entitas software

(standar ANSI/IEEE 1059)

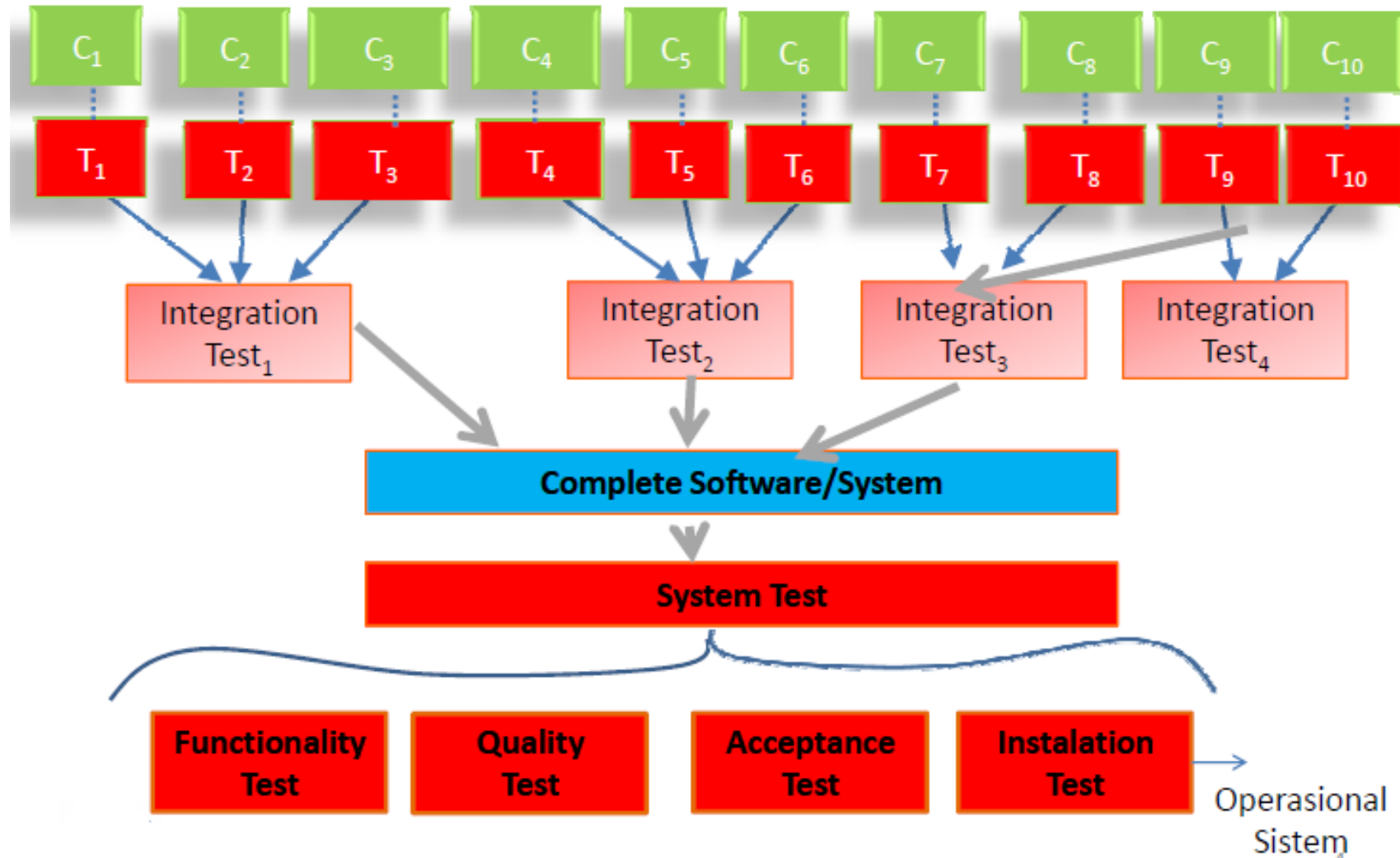
Tujuan pengujian :

- Menemukan sebanyak mungkin masalah (error)
- Tujuan dari menemukan masalah adalah memperbaikinya
- Sebuah pengujian yang sukses adalah menemukan kesalahan yang belum ditemukan.

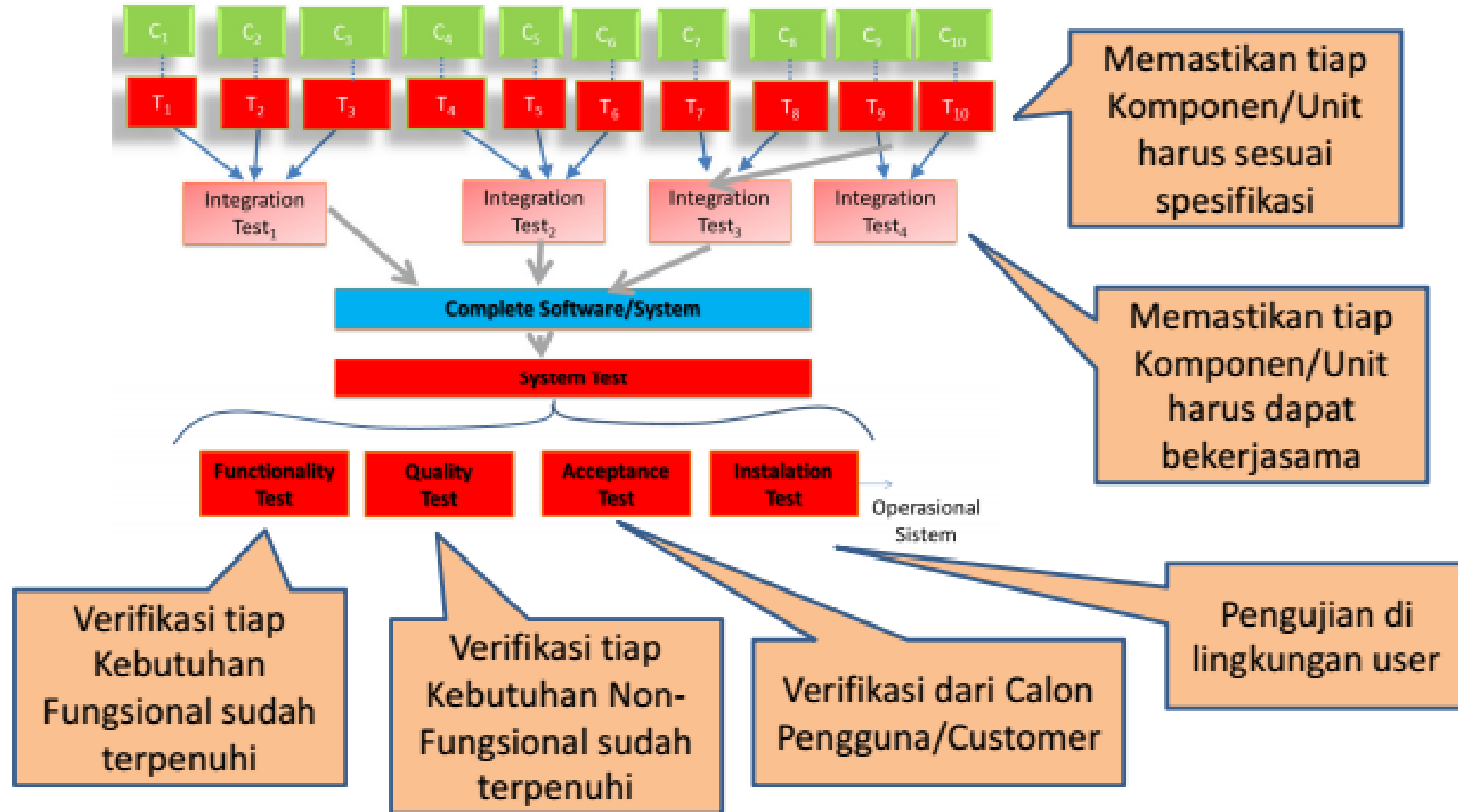
Aktivitas Pengembangan PL



Aktivitas Pengujian



Peran tiap Proses Pengujian



Tester?

SOFTWARE DEVELOPER



Mengerti system

Menguji untuk membuktikan bahwa programnya sudah benar

INDEPENDENCE TESTING GROUP



Harus belajar sistem

Menguji untuk membuktikan bahwa program masih ada yang salah

Siklus Pengujian

Perencanaan Pengujian

Perancangan Pengujian

Eksekusi Pengujian

Laporan Pengujian

Siklus Pengujian

Perencanaan Pengujian

Perancangan Pengujian

Eksekusi Pengujian

Laporan Pengujian

- ✓ Perencanaan **apa** saja yang akan diuji
- ✓ Perencanaan **waktu**
- ✓ Perencanaan **sumberdaya** yang diperlukan
 - Orang
 - PL pendukung
 - Alat/perangkat keras
 - Lingkungan pengujian
 - dan lain-lain
- ✓ Perencanaan prosedur umum pengujian

Siklus Pengujian

Perencanaan Pengujian

Perancangan Pengujian

Eksekusi Pengujian

Laporan Pengujian

- Untuk apapun yang akan diuji, maka perlu dirancang **Kasus Pengujian (Test Case)**
- Test Case berisi:
 - **Masukan yang diuji**
 - **Harapan Pengujian**
 - **Cara menilai hasil eksekusi pengujian**

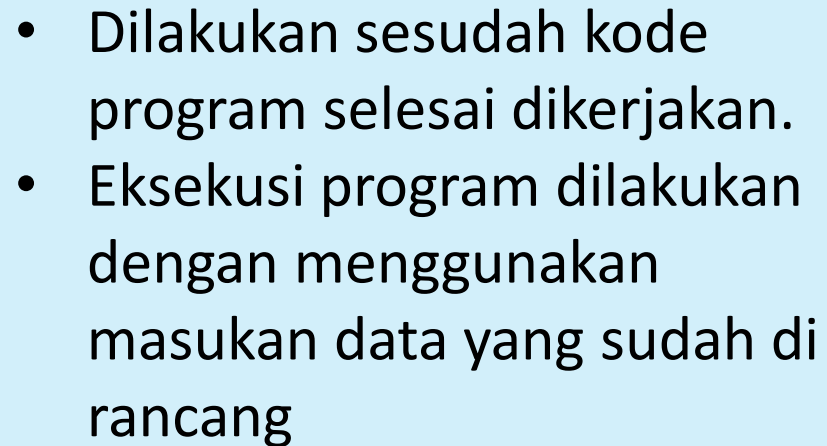
Siklus Pengujian

Perencanaan Pengujian

Perancangan Pengujian

Eksekusi Pengujian

Laporan Pengujian

- 
- A diagram showing the testing cycle. On the left, four steps are listed vertically: 'Perencanaan Pengujian', 'Perancangan Pengujian', 'Eksekusi Pengujian', and 'Laporan Pengujian'. A blue line connects 'Eksekusi Pengujian' to a light blue rectangular callout box on the right. This box contains two bullet points describing the execution step.
- Dilakukan sesudah kode program selesai dikerjakan.
 - Eksekusi program dilakukan dengan menggunakan masukan data yang sudah di rancang

Siklus Pengujian

Perencanaan Pengujian

Perancangan Pengujian

Eksekusi Pengujian

Laporan Pengujian

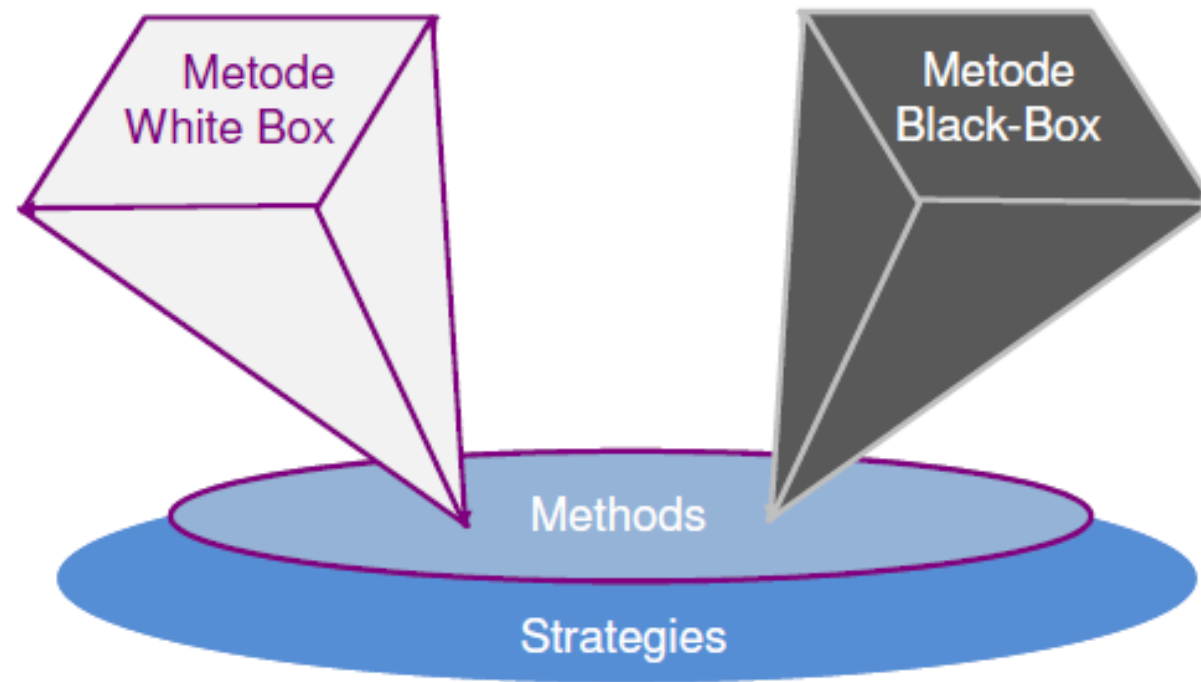


Laporan pengujian berisi laporan untuk setiap hasil eksekusi Test

Case:

- **Lulus pengujian** jika hasil eksekusi program memberikan hasil yang sama dengan harapan hasil
- **Tidak lulus pengujian** jika hasil eksekusi memberikan hasil yang berbeda
- Kadang dapat disertai dengan catatan khusus

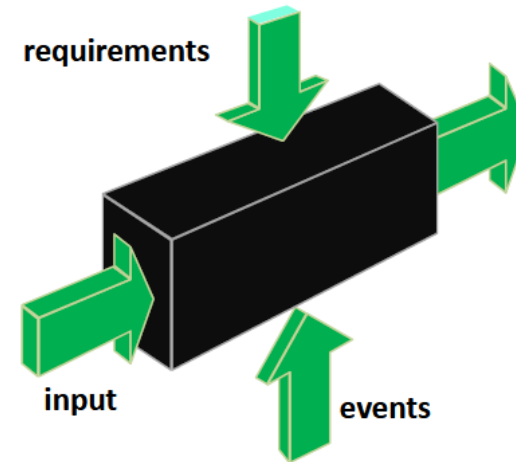
Metode Pengujian



Blackbox VS Whitebox

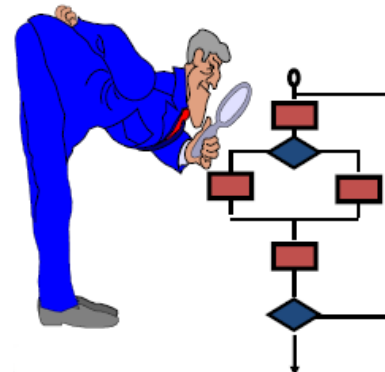
Blackbox testing

- Tidak perlu mengetahui struktur software
- Juga disebut dengan specification-based atau functional testing



Whitebox testing

- Harus mengetahui struktur dan implementasi dari software



Whitebox Testing

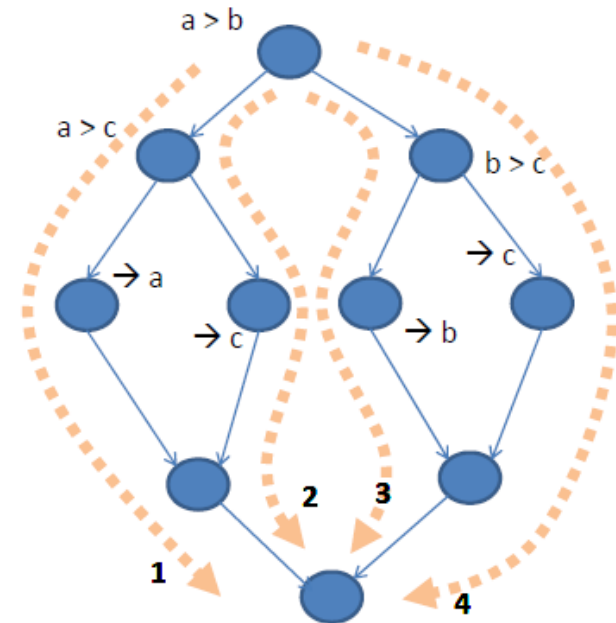
Meyakinkan bahwa semua statement / kondisi dieksekusi minimum satu kali

Contoh White-box testing

- Control Flow Testing
- Data Flow Testing
- Mutation Testing

Path	Input	Expected Result
1	(10, 3, 2)	10
2	(25, 15, 40)	40
3	(10, 15, 6)	15
4	(10, 15, 19)	19

```
if a > b then  
  if a > c then  
    → a  
  else  
    → c  
else  
  if b > c then  
    → b  
  else  
    → c
```



Blackbox Testing

Pengujian Black-Box dilakukan karena isi source code tidak bisa dilihat, hanya spesifikasinya saja

Disebut juga *Functional Testing*, karena kita hanya melihat fungsionalitas dari unit yang diuji

Spesifikasi mendefinisikan apa yang dilakukan, Initial State (IS), dan juga Final State(FS)

- Apa saja input-nya
- Apa saja output-nya
- Apa saja requirement/event yang merubah state

Strategi Pengujian Integrasi

Pengujian integrasi akan mendeteksi fault yang belum terdeteksi saat pengujian unit

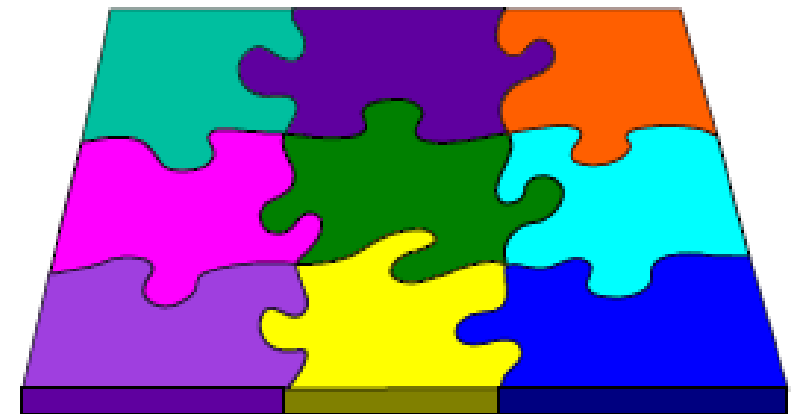
- Fokusnya lebih ke sekumpulan komponen/unit

Dua atau lebih komponen akan diintegrasikan dan di uji

- Setelah tidak ditemukan kesalahan, maka komponen baru dapat digabungkan.
- Urutan integrasi dapat menentukan usaha keseluruhan untuk melakukan pengujian integrasi
 - Kesalahan urutan integrasi dapat menambah biaya, sumber daya yang diperlukan.

Strategi Pengujian Integrasi :

- Pendekatan 'BIG BANG'
- Pendekatan incremental
 - Top Down testing
 - Bottom Up testing
 - Sandwich Testing



System Testing

Pengujian validasi: Fokus pada kebutuhan software

Pengujian sistem: Fokus pada integrasi sistem

Pengujian alpha / Beta: Fokus pada pengguna (bagaimana mereka menggunakan)

Recovery Testing: Software dibuat 'gagal' dan dilihat apakah proses pemulihan sudah dapat dilakukan. Contoh: software yang memanfaatkan internet melihat perilaku pemulihan ketika internet tiba-tiba turun.

Pengujian keamanan: Pastikan apakah mekanisme perlindungan sudah melindungi 'penetrasi'

Stress testing: Sistem diuji bagaimana ketika dihadapkan dengan permintaan penggunaan sumber daya yang melebihi jumlah normal, atau frekuensi atau volume yang normal

Pengujian kinerja: Menguji kinerja software ketika dijalankan pada konteks tertentu