#05
Masalah-masalah Strategis dan Pengujian Unit

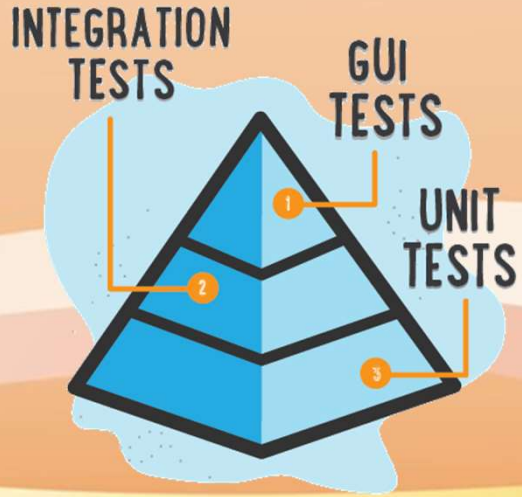
AIK21361 (3 sks)

UJI PERANGKAT LUNAK

Nurdin Bahtiar, S.Si., M.T.

Materi

1. Masalah-masalah Strategis
2. Pengujian Unit
3. Skema Pengujian Unit
   a. Interface
   b. Struktur Data Lokal
   c. Jalur Independen
   d. Jalur Penanganan Kesalahan
   e. Kondisi Batas
4. Prosedur Pengujian Unit



1. Masalah-masalah Strategis

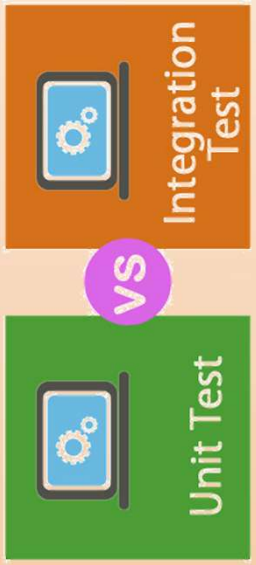
Beberapa hal yang harus diselesaikan agar Pengujian Perangkat Lunak dilaksanakan dengan sukses (Tom Gilb):

- Tentukan persyaratan produk dalam suatu cara yang dapat dikuantifikasi jauh sebelum pengujian dimulai
- Nyatakan sasaran pengujian secara eksplisit
- Pahami para pemakai perangkat lunak dan kembangkan profil bagi masing-masing kategori pemakai
- Kembangkan rencana pengujian yang menekankan “pengujian siklus cepat”

1. Masalah-masalah Strategis

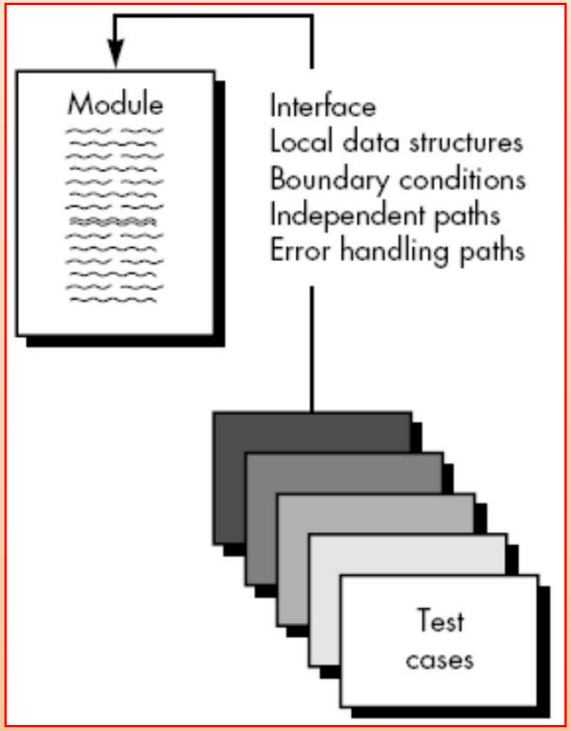
- Bangun perangkat lunak "*robust*" yang didisain untuk menguji dirinya sendiri (mampu mendiagnosis kelas kesalahan)
- Gunakan kajian teknis formal sebagai sebuah filter sebelum pengujian
- Lakukan kajian teknis formal untuk memperkirakan strategi pengujian dan melakukan *test case* terhadap dirinya sendiri
- Kembangkan pendekatan pengembangan yang kontinyu untuk proses pengujian.

2. Pengujian Unit (Satuan)



- Pengujian unit berfokus pada usaha verifikasi pada inti terkecil dari desain perangkat lunak, yaitu **modul** (komponen PL).
- Jalur kontrol yang penting diuji untuk mengungkap kesalahan di dalam batas dari modul tersebut.
- Pengujian unit biasanya berorientasi pada *white-box*, dan langkahnya dapat dilakukan secara paralel untuk modul bertingkat.

3. Skema Pengujian Unit



- Interface memastikan bahwa informasi secara tepat mengalir masuk dan keluar dari inti program yang diuji.
- Struktur data lokal memastikan bahwa data yang tersimpan secara temporal dapat tetap menjaga integritasnya selama semua langkah di dalam suatu algoritma dieksekusi.
- Kondisi batas uji memastikan bahwa modul beroperasi dengan tepat pada batas yang ditentukan untuk membatasi pemrosesan.
- Jalur independen (jalur dasar) yang melalui struktur kontrol dipastikan dipakai sedikitnya satu kali.
- Terakhir, semua jalur penanganan kesalahan dilakukan.

3. Skema Pengujian Unit

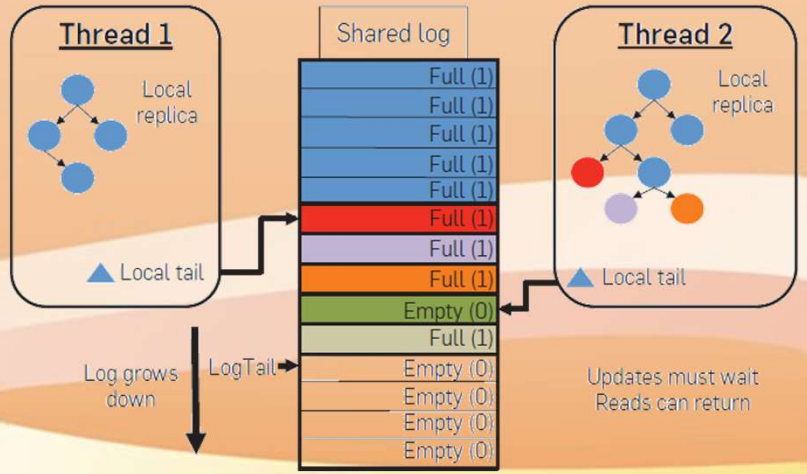
a. Interface

- Menurut Mayers, beberapa usulan checklist untuk pengujian interface:
  - ✓ Apakah jumlah parameter input sama dengan jumlah argumen?
  - ✓ Apakah atribut parameter dan argumen sudah cocok?
  - ✓ Apakah sistem satuan parameter dan argumen sudah cocok?
  - ✓ Apakah jumlah argumen yang ditransmisikan ke modul yang dipanggil sama dengan jumlah parameter?
  - ✓ Apakah atribut argumen yang ditransmisikan ke modul yang dipanggil sama dengan atribut parameter?

3. Skema Pengujian Unit

b. Struktur Data Lokal

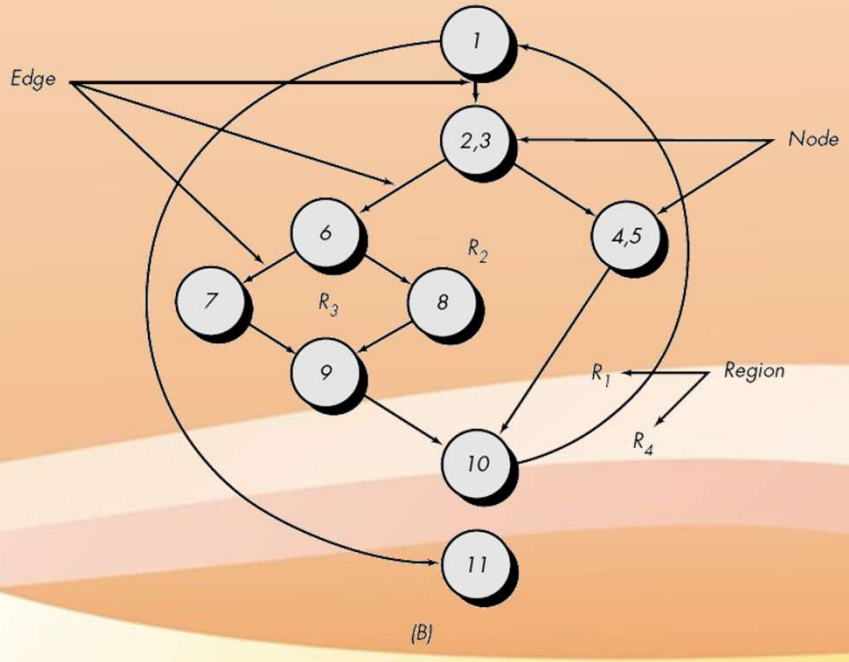
- Struktur data lokal untuk suatu modul adalah sumber umum kesalahan.
- Test case harus didisain utk mengungkap kesalahan dg kategori:
  - ✓ Pengetikan yang tidak teratur dan tidak konsisten
  - ✓ Inisialisasi yang salah atau nilai-nilai default
  - ✓ Nama variabel yang salah (salah eja atau terpotong)
  - ✓ Tipe data yang tidak konsisten



3. Skema Pengujian Unit

c. Jalur Independen

- Test case harus didisain untuk mengungkap kesalahan yang berhubungan dengan:
  - ✓ Komputasi yang salah
  - ✓ Perbandingan yang tidak benar
  - ✓ Aliran kontrol yang tidak tepat.



3. Skema Pengujian Unit

d. Jalur Penanganan Kesalahan

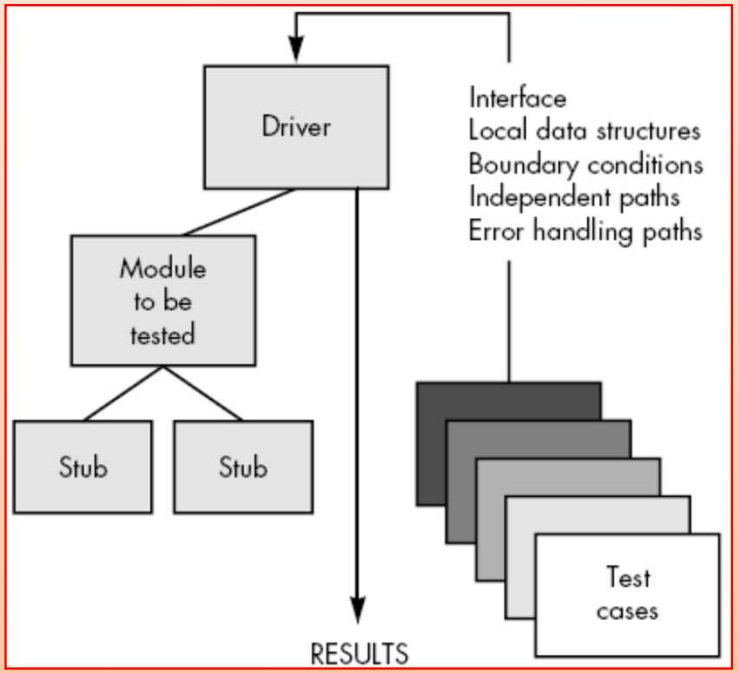
- Beberapa kesalahan potensial yang harus diuji:
  - ✓ Deskripsi kesalahan yang tidak dapat dipahami
  - ✓ Kesalahan yg dicatat tidak sesuai dengan kesalahan yg terjadi
  - ✓ Kondisi kesalahan yang menyebabkan intervensi sistem sebelum penanganan kesalahan

3. Skema Pengujian Unit

e. Kondisi Batas

- Pengujian batasan merupakan tugas yang terakhir (mungkin paling penting) dari langkah pengujian unit.
- Sering terjadi kesalahan pada pada saat elemen ke-*n* dari array dimensi *n* diproses, atau pada saat pengulangan ke-*i* dari sebuah loop dengan *i* perulangan.
- Test case yang menggunakan struktur data, aliran kontrol, dan nilai data tepat di bawah, pada, atau tepat di atas maksimum dan minimum, sangat mungkin untuk menemukan kesalahan.

4. Prosedur Pengujian Unit



- Karena modul bukanlah program yang berdiri sendiri, perangkat lunak **driver** dan **stub** harus dikembangkan bagi masing-masing pengujian unit.
- Pada sebagian besar aplikasi, driver berperan seperti "program utama" yang menerima data test case, lalu melewatkan data tersebut ke modul (untuk diuji).
- Sedangkan stub berfungsi untuk menggantikan modul yang merupakan subordinat dari modul yang akan diuji.

Diskusi

- Berikan contoh driver dan stub untuk pengujian program ini:

```
Program UTAMA
variabel   a, b, c : float
           nama, nim : string

input n, nama, nim
input a
LABEL
KALI
output b, c

Subprogram LABEL
variabel i : integer
   output nama, nim
   for i <- 1 to n output '='

Subprogram KALI
   b <- 2 * exp (a)
   c <- b * b - 4 * a
```

End of File

Latihan

1. Pilih salah satu aplikasi umum apa saja yang pernah Anda gunakan (misalnya Notepad, Calculator, dsb).
2. Tentukan 5 (lima) sub fungsi dari aplikasi tersebut.
3. Tentukan 5 (lima) pengujian apa saja yang dapat dilakukan dengan *test case* yang mungkin pada setiap sub fungsi tersebut.
4. ~~Kumpulkan!~~