Strategi Pengujian Perangkat Lunak

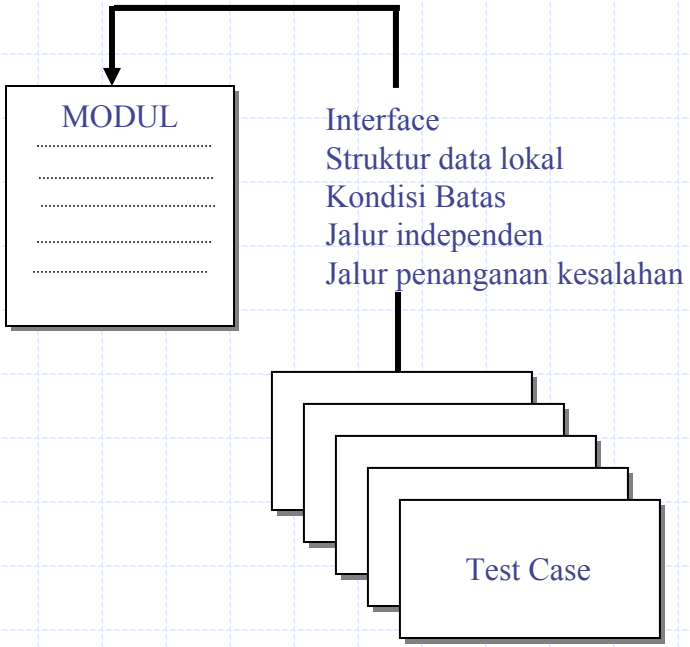
Minggu ke 8

Pendekatan Strategis ke pengujian perangkat lunak

- Pengujian Unit
- Pengujian Integrasi
- Pengujian Validasi
- Pengujian Sistem

Pengujian Unit

- Berfokus pada inti terkecil dari desain perangkat lunak yaitu modul
- Biasanya berorientasi pada white box



Pengujian Unit

- Checklist untuk pengujian interface
  - *Apakah jumlah parameter input sama dengan jumlah argumen?*
  - *Apakah antara atribut dan parameter argumen sudah cocok?*
  - *Apakah antara sistem satuan parameter dan argumen sudah cocok?*
  - *Apakah jumlah argumen yang ditransmisikan ke modul yang dipanggil sama dengan atribut parameter?*

Pengujian Unit

- *Apakah atribut dari argumen yang ditransmisikan ke modul yang dipanggil sama dengan atribut parameter?*
- *Apakah sistem unit dari argumen yang ditransmisikan ke modul yang dipanggil sama dengan sistem satuan parameter?*
- *Apakah jumlah atribut dan urutan argumen ke fungsi-fungsi built-in sudah benar?*
- *Adakah referensi ke parameter yang tidak sesuai dengan poin entri yang ada?*
- *Apakah argumen input only diubah?*

Pengujian Unit

- *Apakah definisi variabel global konsisten dengan modul ?*
- *Apakah batasan yang dilalui merupakan argumen?*

Test case harus didesain untuk mengungkap kesalahan dalam kategori

- pengetikan yang tidak teratur dan tidak konsisten
- inisialisasi yang salah atau nilai-nilai default
- Nama variabel yang tidak benar
- Tipe data yang tidak konsisten
- Underflow, overflow dan pengecualian pengalamatan

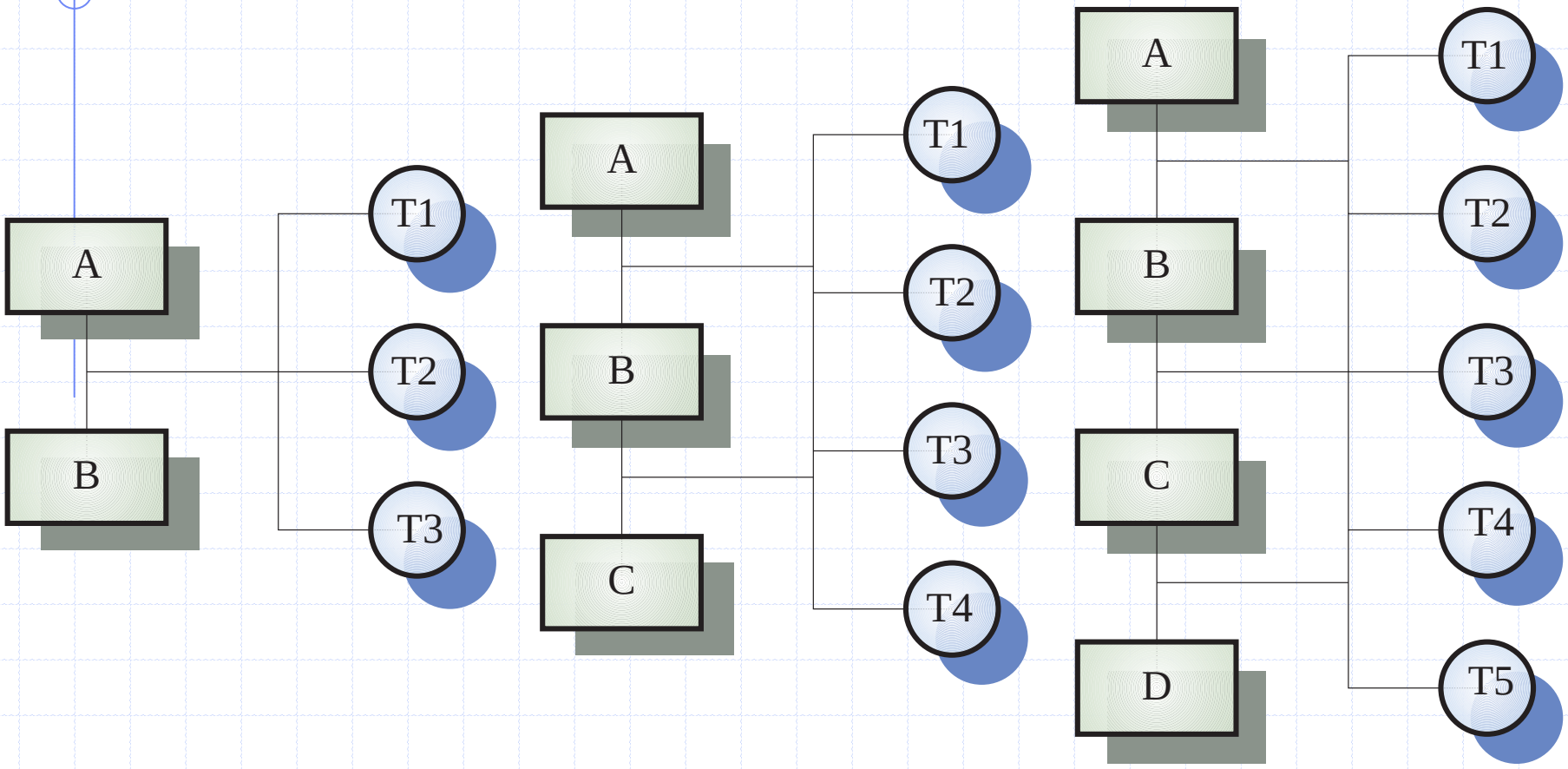
Seberapa baik sistem yang sudah dibangun ?

- Dua Aspek yang dipertimbangkan:
  - Apakah implementasi sudah sesuai dengan spesifikasi ?
  - Apakah spesifikasi sesuai dengan kebutuhan user ?
- Validasi
  - "Apakah sistem yang dikembangkan sudah benar?"
  - Pengujian dimana sistem ketika diimplementasikan sesuai dengan yang iharapkan
- Verifikasi
  - "Apakah sistem dikembangkan dengan cara yang benar ?"
  - Pengujian apakah sistem sudah sesuai dengan spesifikasi

Integration testing

- Pengujian keseluruhan system atau sub-system yang terdiri dr komponen yg terintegrasi.
- Test integrasi menggunakan black-box dengan test case ditentukan dari spesifikasi.
- Kesulitannya adalah menemukan/melokasikan
- Penggunaan Incremental integration testing dapat mengurangi masalah tersebut.

Incremental integration testing



Test sequence 1

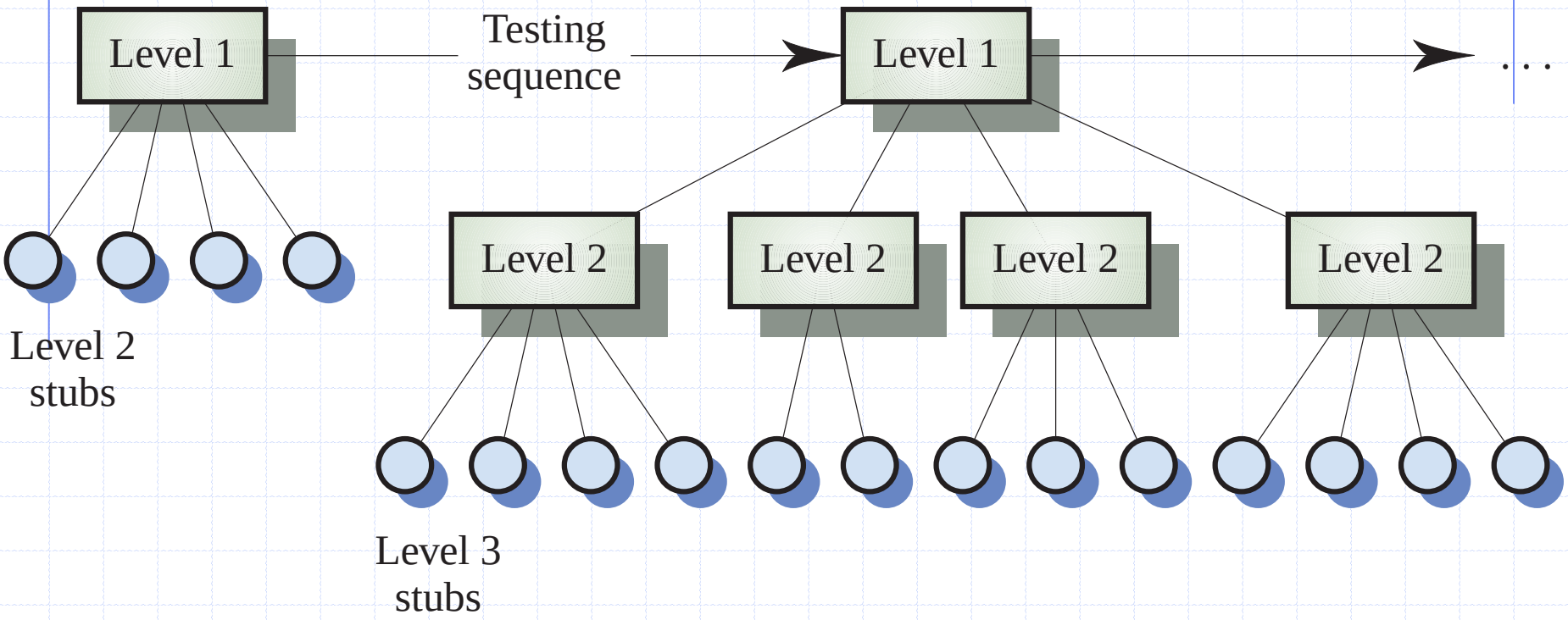
Test sequence 2

Test sequence 3

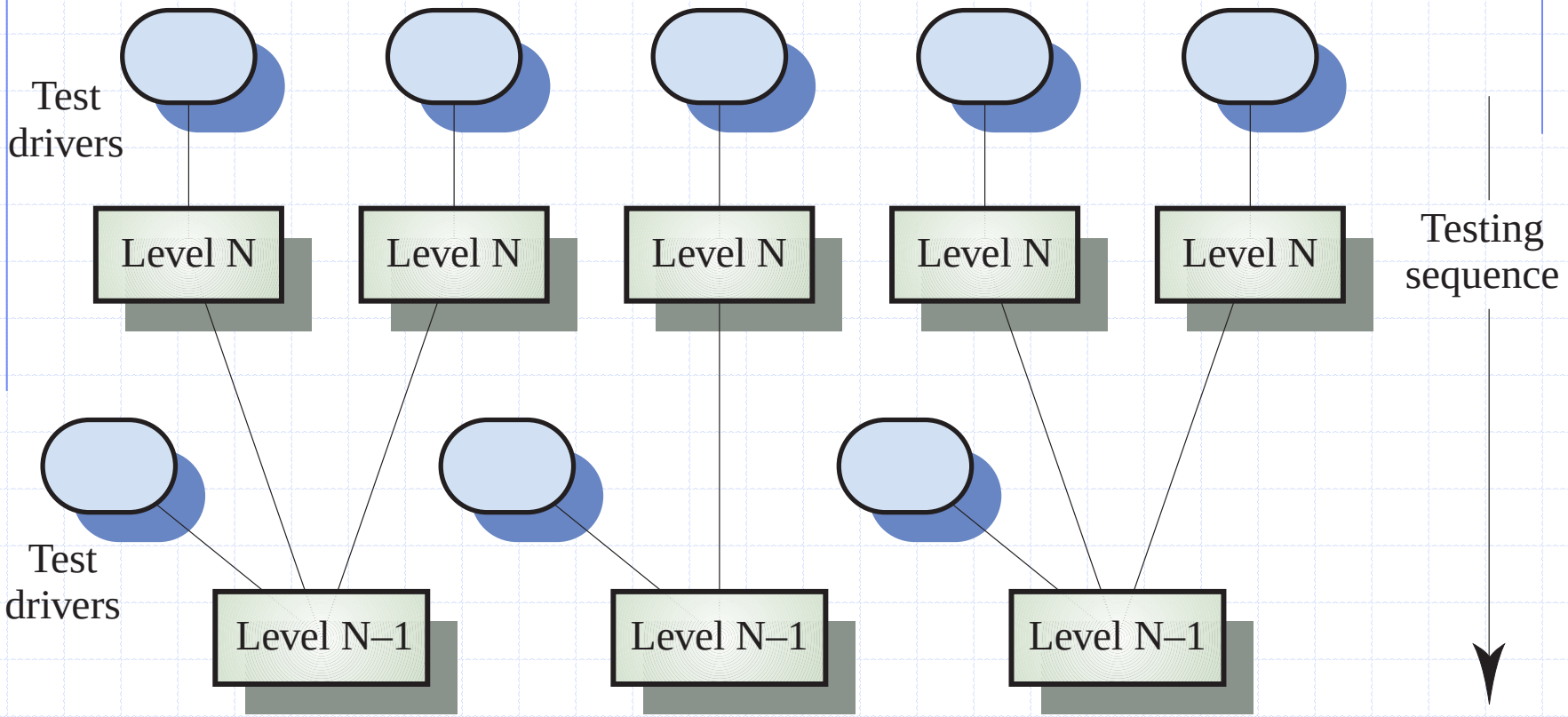
Pendekatan integration testing

- Top-down testing
  - Berawal dari level-atas system dan terintegrasi dengan mengganti masing-masing komponen secara top-down dengan suatu stub (program pendek yg mengenerate input ke sub-system yg diuji).
- Bottom-up testing
  - Integrasi components di level hingga sistem lengkap sudah teruji.
- Pada prakteknya, kebanyakan test integrasi menggunakan kombinasi kedua strategi pengujian tsb.

Top-down testing



Bottom-up testing



Pendekatan Testing

- Architectural validation
  - Top-down integration testing lebih baik digunakan dalam menemukan error dalam sistem arsitektur.
- System demonstration
  - Top-down integration testing hanya membatasi pengujian pada awal tahap pengembangan system.
- Test implementation
  - Seringkali lebih mudah dengan menggunakan bottom-up integration testing

Interface testing

- Dilakukan kalau module-module dan sub-system terintegrasi dan membentuk sistem yang lebih besar
- Tujuannya untuk medeteksi fault terhadap kesalahan interface atau asumsi yg tidak valid terntang interface tsb.
- Sangat penting untuk pengujian terhadap pengembangan sistem dgn menggunakan pendekatan object-oriented yg didefinisikan oleh object-objectnya