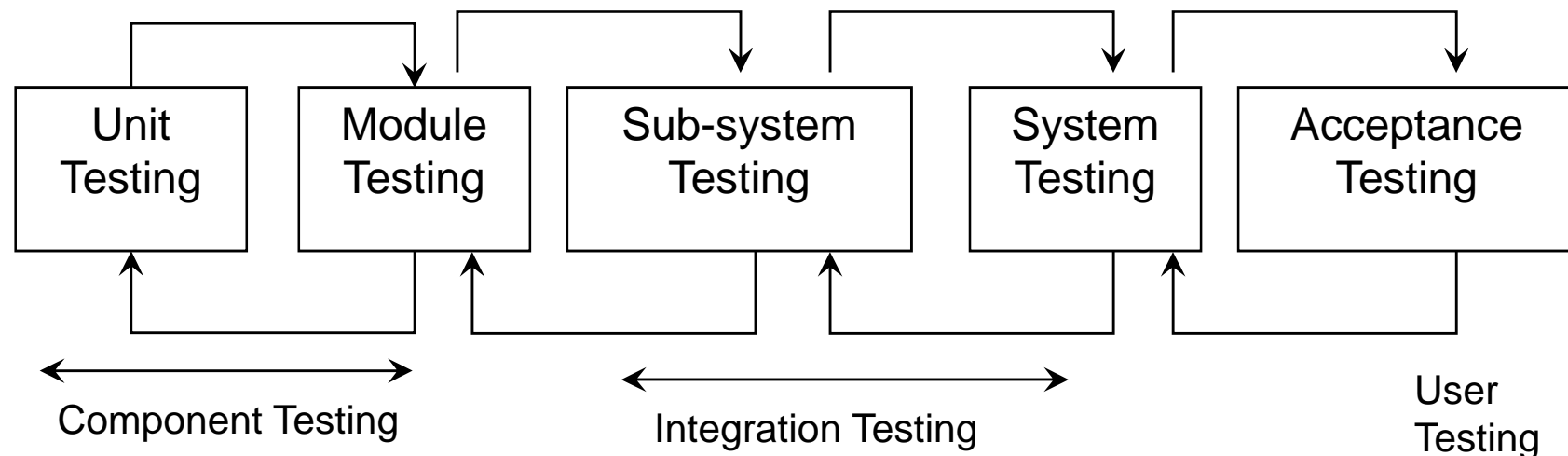


PERTEMUAN 13

STRATEGI PENGUJIAN PERANGKAT LUNAK

Strategi uji coba perangkat lunak dilakukan untuk memudahkan para perancang untuk menentukan keberhasilan system yang telah dikerjakan

Proses Testing



Proses Testing

❖ Unit testing

Pengujian masing-masing unit komponen program untuk meyakinkan bahwa sudah beroperasi secara benar

❖ Module Testing

Pengujian terhadap koleksi unit-unit komponen yang saling berhubungan.

❖ Sub-system Testing

Pengujian terhadap koleksi module-module yang membentuk suatu sub-system (aplikasi)

Proses Testing

❖ System Testing

Pengujian terhadap integrasi sub-system, yaitu keterhubungan antar sub-system

❖ Acceptance Testing

- ☐ Pengujian terakhir sebelum sistem dipakai oleh user.
- ☐ Melibatkan pengujian dengan data dari pengguna sistem.
- ☐ Biasa dikenal sebagai “alpha test” (“beta test” untuk software komersial, dimana pengujian dilakukan oleh potensial customer)

Rencana Pengujian

- ❖ Proses testing
 - Deskripsi fase-fase utama dalam pengujian
- ❖ Pelacakan Kebutuhan
 - Semua kebutuhan user diuji secara individu
- ❖ Item yg diuji
 - Menspesifikasi komponen sistem yang diuji
- ❖ Jadwal Testing
- ❖ Prosedur Pencatatan Hasil dan Prosedur
- ❖ Kebutuhan akan Hardware dan Software
- ❖ Kendala-kendala
 - Mis: kekurangan staff, alat, waktu dll.

Failure and Faults

- ❑ Failure: output yang tidak benar/tidak sesuai ketika sistem dijalankan
- ❑ Fault: kesalahan dalam source code yang mungkin menimbulkan failure ketika code yang fault tersebut dijalankan

Failure Class	Deskripsi
Transient	Muncul untuk input tertentu
Permanent	Muncul untuk semua input
Recoverable	Sistem dapat memperbaiki secara otomatis
Unrecoverable	Sistem tidak dapat memperbaiki secara otomatis
Non-corrupting	Failure tidak merusak data
Corrupting	Failure yang merusak sistem data

Prioritas Testing

- ❖ Hanya test yang lengkap yang dapat meyakinkan sistem terbebas dari kesalahan, tetapi hal ini sangat sulit dilakukan.
- ❖ Prioritas dilakukan terhadap pengujian kemampuan sistem, bukan masing-masing komponennya.
- ❖ Pengujian untuk situasi yang tipikal lebih penting dibandingkan pengujian terhadap nilai batas.

Test Data Dan Kasus Test

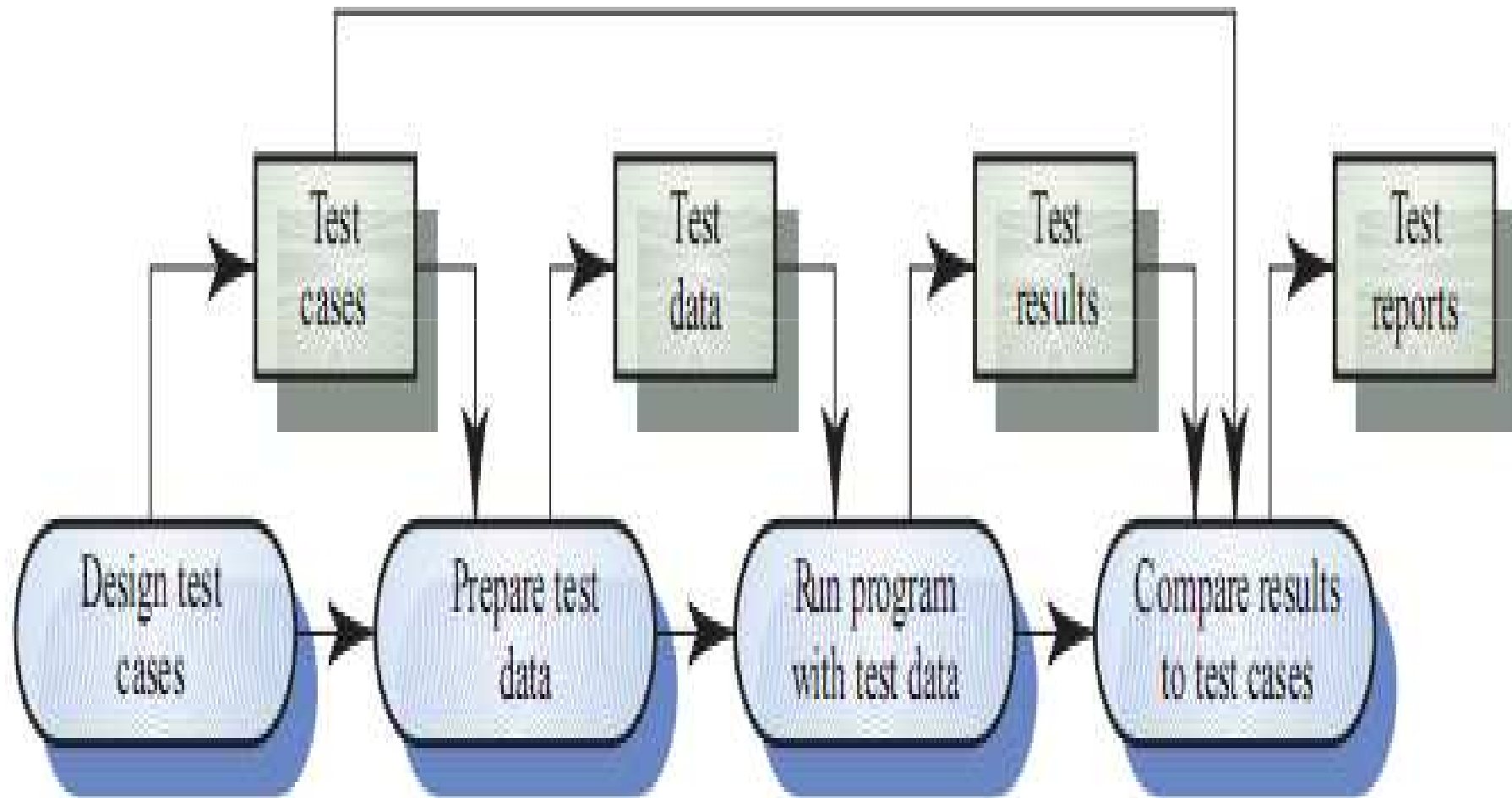
☐ Test Data

Input yang direncanakan digunakan oleh sistem.

☐ Test Cases

Input yang digunakan untuk menguji sistem dan memprediksi output dari input jika sistem beroperasi sesuai dengan spesifikasi.

Proses defect testing



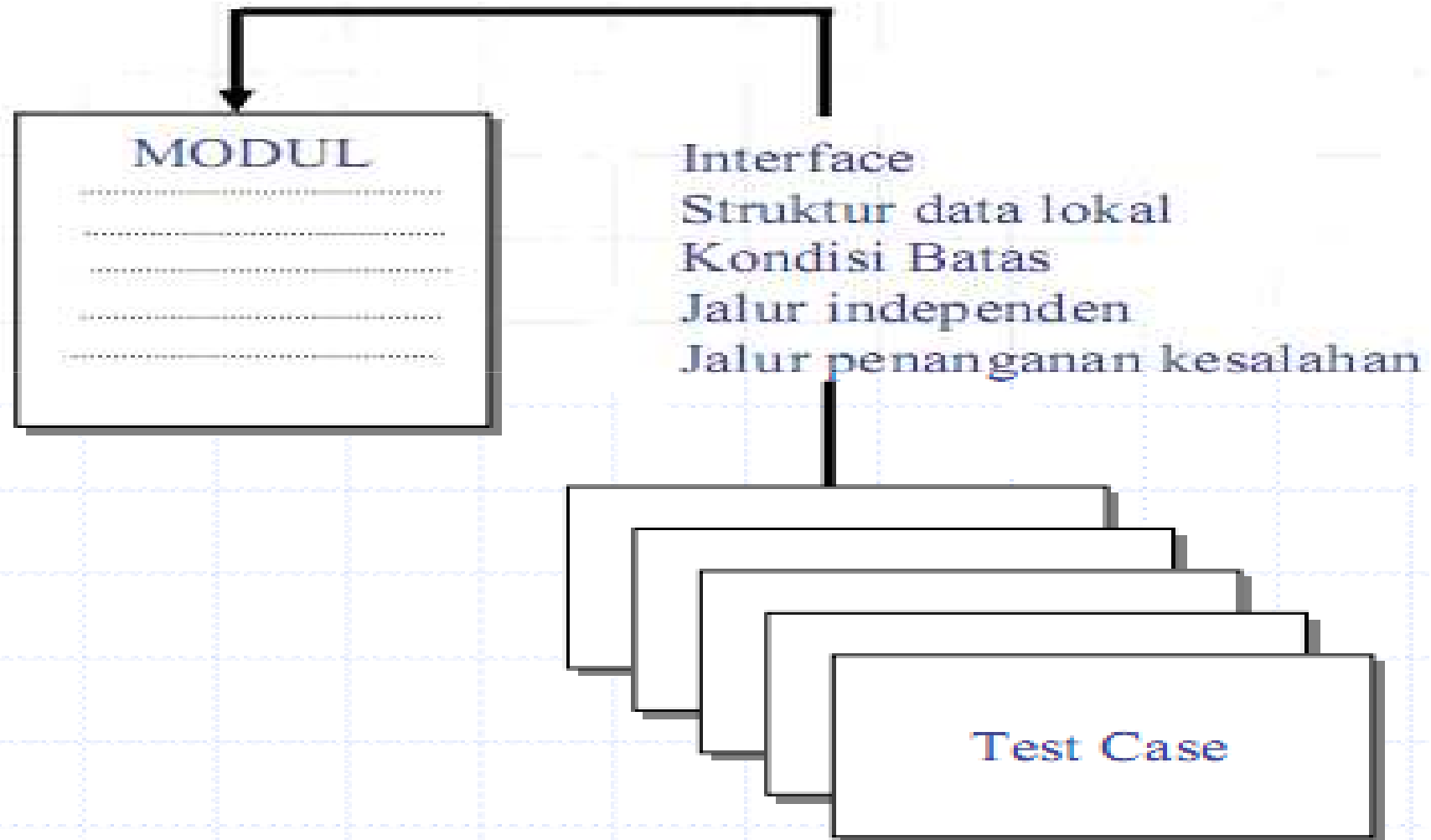
Pendekatan Strategis Pengujian Perangkat Lunak

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

Pengujian Unit

- ☐ Berfokus pada inti terkecil dari desain perangkat lunak yaitu modul
- ☐ Uji coba unit selalu berorientasi pada white box testing
- ☐ Dapat dikerjakan paralel atau beruntun dengan modul lainnya.

Pengujian Unit (2)



Pengujian Unit (3)

- ☐ 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?
- ☐ 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?

Pengujian Unit (4)

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

Pengujian Unit (5)

- ❑ Test case harus didesain untuk mengungkap kesalahan dalam kategori
 1. Pengetikan yang tidak teratur dan tidak konsisten inisialisasi yang salah atau nilai-nilai default
 2. Nama variabel yang tidak benar
 3. Tipe data yang tidak konsisten
 4. Underflow, overflow dan pengecualian pengalamatan

Seberapa Baik Sistem Yang Sudah di Bangun

❖ Dua Aspek yang dipertimbangkan:

- ☐ Apakah implementasi sudah sesuai dengan spesifikasi ?
- ☐ Apakah spesifikasi sesuai dengan kebutuhan user ?

❖ Validasi

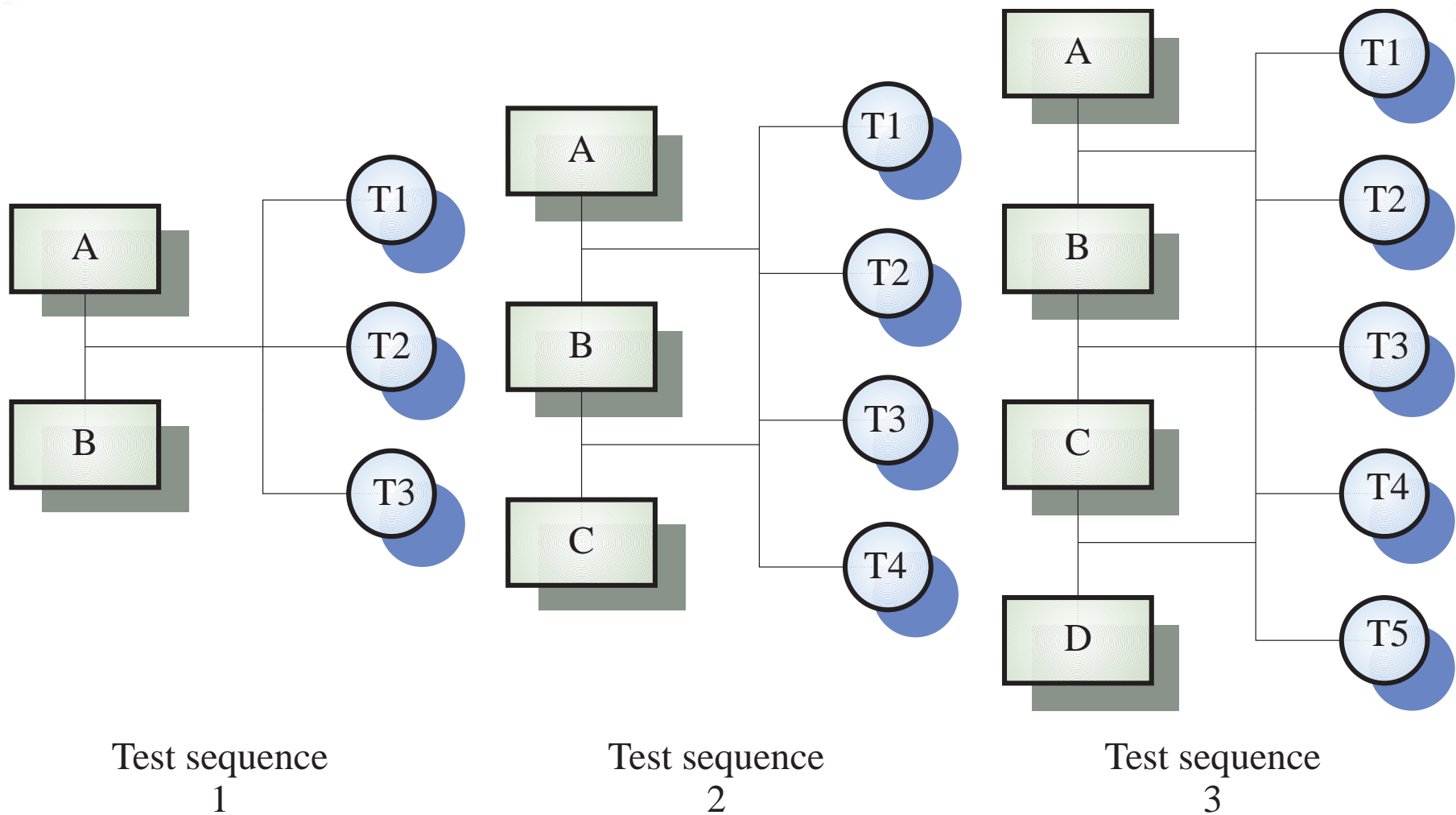
- ☐ Apakah sistem yang dikembangkan sudah benar?
- ☐ Pengujian dimana sistem ketika diimplementasikan sesuai dengan yang diharapkan

❖ 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 dari komponen yang terintegrasi.
- ❑ Test integrasi menggunakan black-box dengan test case ditentukan dari spesifikasi.
- ❑ Kesulitannya adalah menemukan/melokasikan
- ❑ Penggunaan Incremental integration testing dapat mengurangi masalah tersebut.



Pendekatan Integration Testing

❖ Top-down Testing

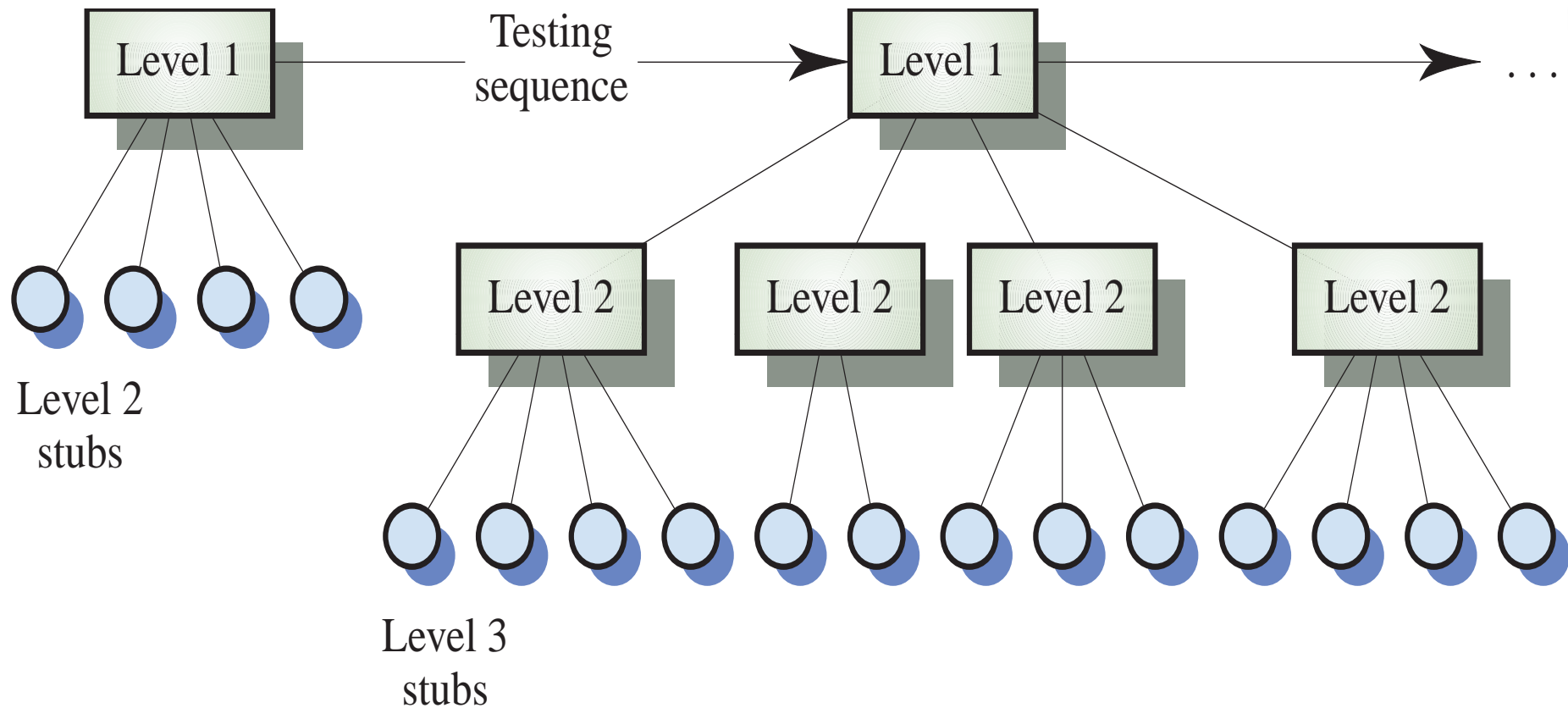
Berawal dari level-atas system dan terintegrasi dengan mengganti masing-masing komponen secara top-down dengan suatu stub (program pendek yg generate input ke sub-system yang diuji).

❖ Bottom-up Testing

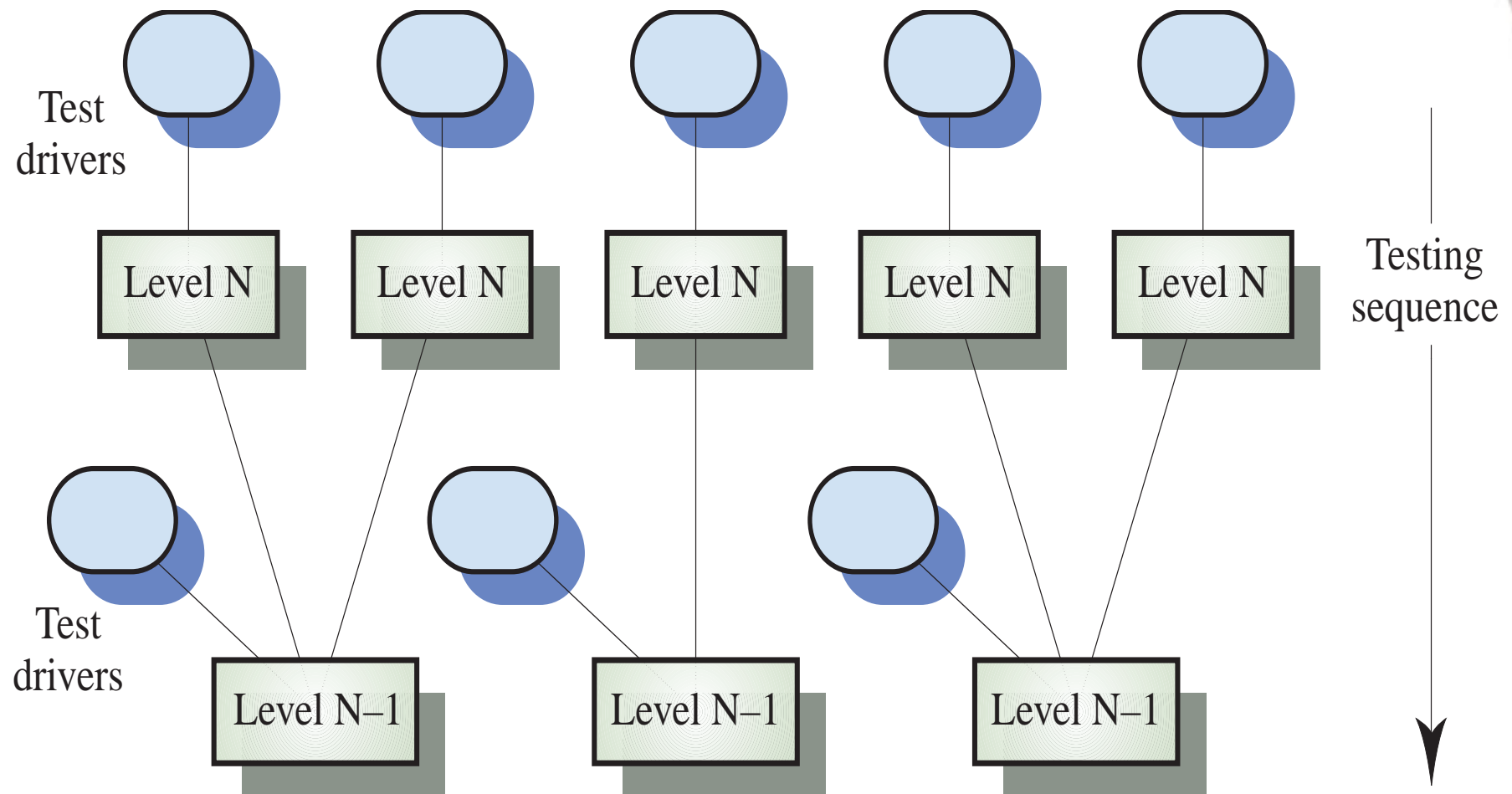
Integrasi components di level hingga sistem lengkap sudah teruji.

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 mendeteksi fault terhadap kesalahan interface atau asumsi yang tidak valid tentang interface tersebut.
- ☐ Sangat penting untuk pengujian terhadap pengembangan sistem dengan menggunakan pendekatan object-oriented yang didefinisikan oleh object-objectnya

Pengujian Validasi

❖ Kajian Konfigurasi (audit)

- ☐ Elemen dari proses validasi
- ☐ Memastikan apakah semua elemen konfigurasi perangkat lunak telah dikembangkan dengan tepat

❖ Pengujian Alpha dan Beta

- ☐ Pengujian Alpha
 - ✓ Usability labs
 - ✓ Usability factors checklist
- ☐ Pengujian Beta

Pengujian Sistem :

- ☐ Pengujian Perbaikan
- ☐ Pengujian Keamanan
- ☐ Pengujian Stress
- ☐ Pengujian Kinerja

Pengujian Aplikasi Server :

- ☐ Volume Testing
- ☐ Stress Testing
- ☐ Performance Testing
- ☐ Data Recovery Testing
- ☐ Data Backup and Restore Testing
- ☐ Data Security Testing

Volume Testing

- ❑ Menemukan kelemahan sistem selama melakukan pemrosesan data dalam jumlah yang besar dalam periode waktu yang singkat.
- ❑ Tujuan: meyakinkan bahwa sistem tetap melakukan pemrosesan data antar batasan fisik dan batasan logik.
- ❑ Contoh:
Menguji proses antar server dan antar partisi hardisk pada satu server.

Stress Testing

- ❑ Tujuan: mengetahui kemampuan sistem dalam melakukan transaksi selama periode waktu puncak proses.
- ❑ Contoh periode puncak: ketika penolakan proses login on-line setelah sistem down atau pada kasus batch, pengiriman batch proses dalam jumlah yang besar dilakukan setelah sistem down.
- ❑ Contoh: Melakukan login ke server ketika sejumlah besar workstation melakukan proses menjalankan perintah sql database.

Performance Testing

- ☐ Dilakukan secara paralel dengan Volume dan Stress testing untuk mengetahui unjuk kerja sistem (waktu respon, throughput rate) pada beberapa kondisi proses dan konfigurasi.
- ☐ Dilakukan pada semua konfigurasi sistem perangkat keras dan lunak. Misal : pada aplikasi Client-Server diujikan pada kondisi korporate ataupun lingkungan sendiri (LAN vs. WAN, Laptop vs. Desktop)
- ☐ Menguji sistem dengan hubungannya ke sistem yang lain pada server yang sama.
- ☐ Load Balancing Monitor
- ☐ Network Monitor

Data Recovery Testing

- ☐ Investigasi dampak kehilangan data melalui proses recovery ketika terjadi kegagalan proses.
- ☐ Penting dilakukan karena data yg disimpan di server dapat dikonfigurasi dengan berbagai cara.
- ☐ Kehilangan Data terjadi akibat kegagalan sistem, hardisk rusak, penghapusan yang tidak sengaja, kecelakaan, virus dan pencuri.

Data Backup dan Restore Testing

- ☐ Dilakukan untuk melihat prosedur back-up dan recovery.
- ☐ Diakukan dengan mensimulasikan beberapa kesalahan untuk menguji proses backup dan recovery.
- ☐ Pengujian dilakukan terhadap strategi backup: frekuensi , medium, waktu, mekanisme backup (manual/ otomatis), personal, ? Berapa lama backup akan disimpan.
- ☐ Switching antara live dan backup server ketika terjadi kerusakan (load log transaction pada back-up kemudian melakukan recovery).

Data Security Testing

- ❑ *Privilege access* terhadap *database* diujikan pada beberapa user yang tidak memiliki privilege access ke database.
- ❑ *Shutdown database engine* melalui *operating system* (dengan beberapa perintah OS) yang dapat mematikan aplikasi *database*.

Debugging

