Validasi dan Verifikasi Software

OLEH : RAHMAT ROBI WALIYANSYAH, M.KOM.

Verifikasi vs. Validasi

- Verifikasi: “Are we building the product right “
  - Software seharusnya sesuai dengan spesifikasinya
- Validasion:"Are we building the right product”.
  - Software seharusnya melakukan apa yang benar-benar disyaratkan oleh user.

Proses Verifikasi & Validasi

- Menemukan kekurangan dalam sebuah sistem.
- Memperkirakan apakah sistem berguna dan dapat digunakan atau tidak dalam situasi operasional.

TUJUAN VERIFIKASI & VALIDASI

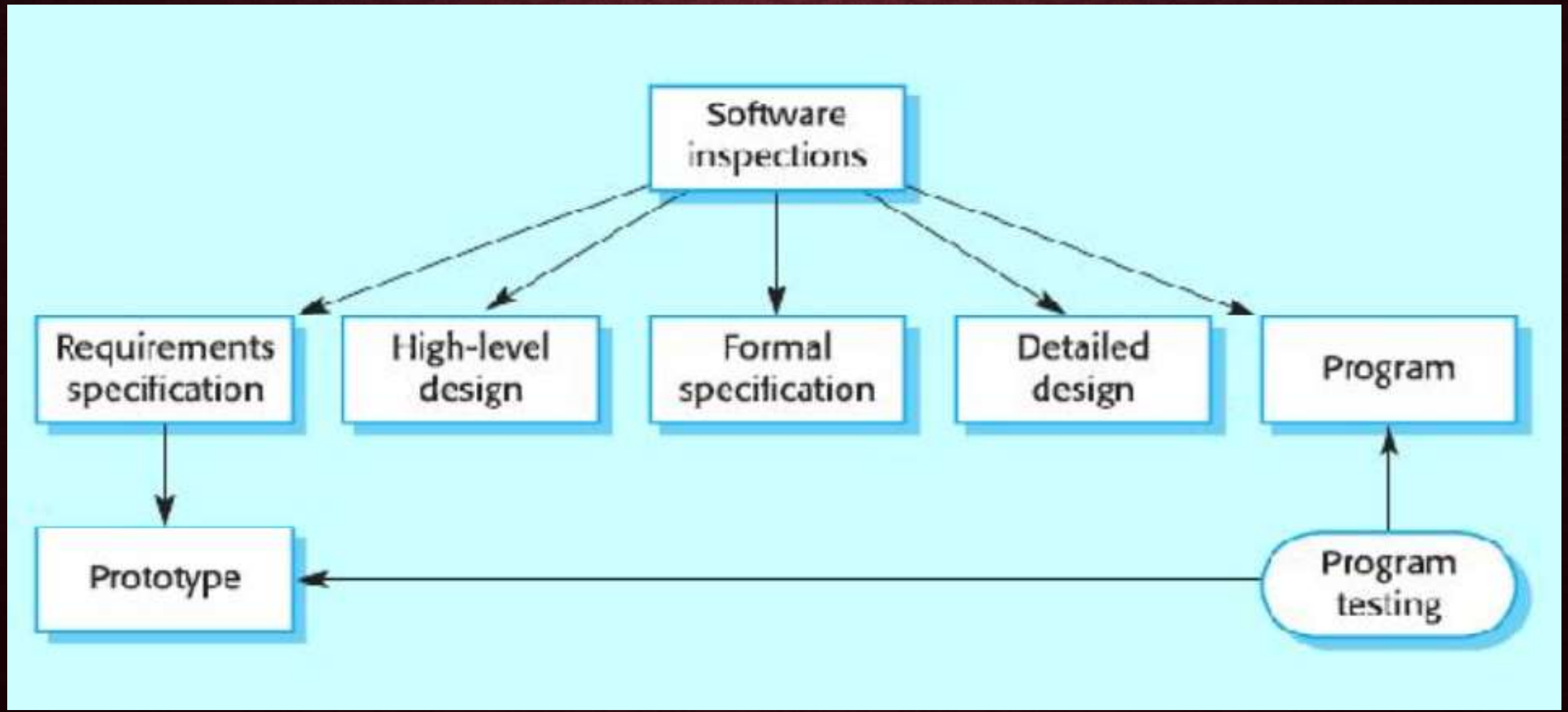
- Verifikasi dan validasi harus memastikan bahwa software sesuai dengan tujuannya.

Kepastian Verifikasi & Validasi

- Tergantung pada tujuan sistem atau fungsi software yaitu Tingkat kritikal software terhadap sebuah organisasi.
- Tergantung pada Harapan User, user mungkin mempunyai harapan yang rendah terhadap software yang ada.
- Tergantung pada lingkungan pemasaran, lebih awal memasarkan sebuah produk ke pasar lebih penting daripada menemukan kekurangan dalam program.

Verifikasi Statik & Dinamik

- **Software inspection**. Berhubungan dengan analisis sistem statik untuk menemukan masalah (verifikasi statik). Dapat menjadi tambahan dari tool-based document dan code analysis.
- **Software testing**. Berhubungan dengan pelaksanaan dan perilaku produk (dinamik verifikasi). Sistem dijalankan dengan data tes dan proses operasionalnya diperhatikan.

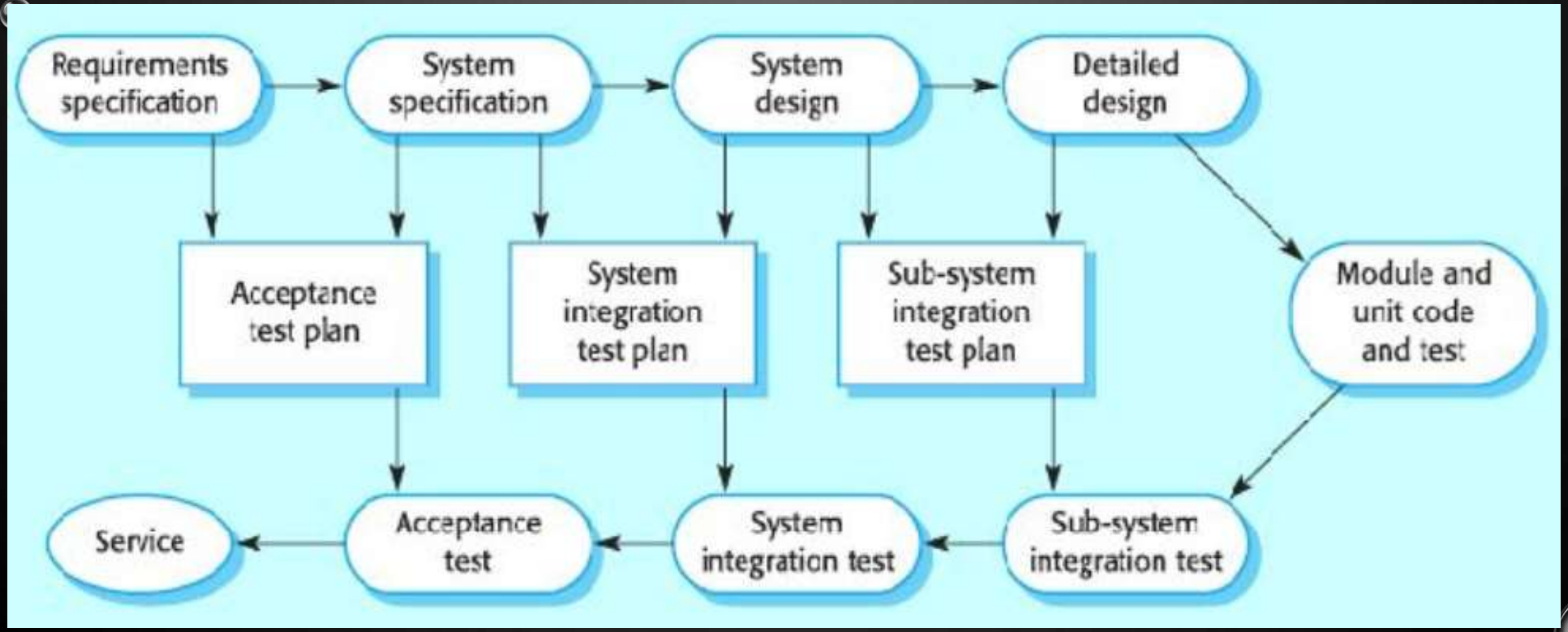

Software inspections
Requirements specification
High-level design
Formal specification
Detailed design
Program
Prototype
Program testing

PENGUJIAN PROGRAM

- Pengujian Program dapat mengungkapkan adanya kesalahan bukan mengecek tidak adanya kesalahan.
- Teknik validasi untuk persyaratan non-functional, sebagai sebuah software dapat dijalankan untuk melihat bagaimana perilakunya.

Tipe Pengujian

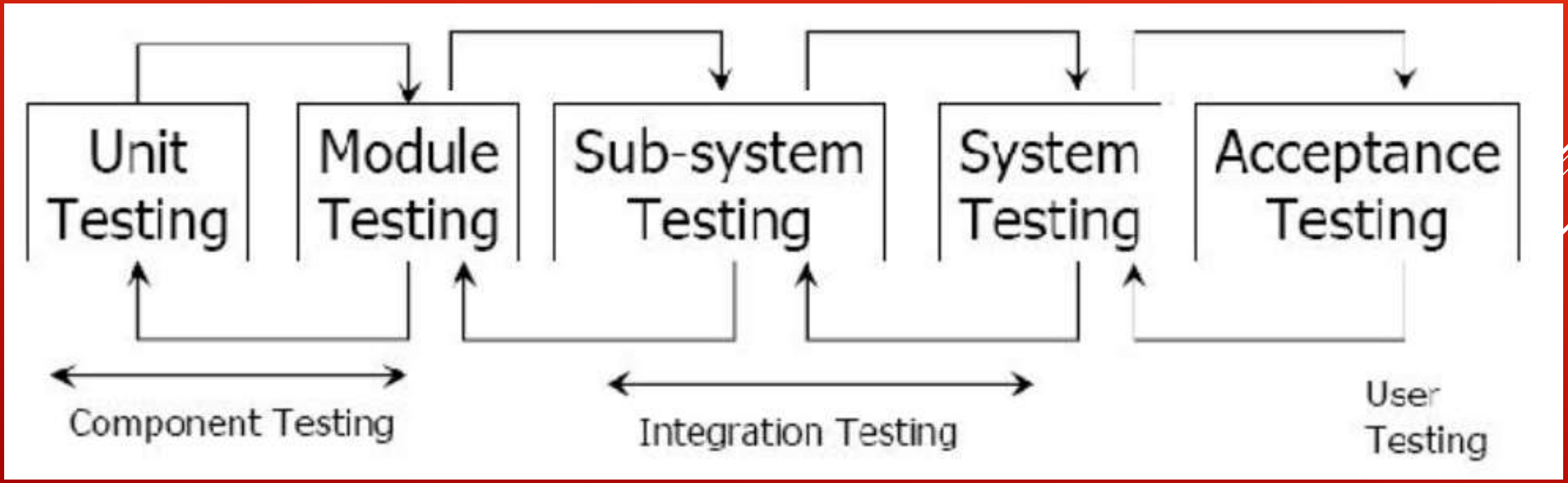
- **Pengujian Kekurangan**, dalam pengujian kekurangan test dirancang untuk menemukan kekurangan sistem. Uji kekurangan yang berhasil salah satunya adalah menunjukkan keberadaan kekurangan dalam sebuah sistem.
- **Pengujian Validasi**, ditujukan untuk memperlihatkan bahwa software sesuai dengan persyaratannya. Tes yang berhasil adalah salah satu yang menunjukkan bahwa persyaratan telah diterapkan secara tepat.


Requirements specification
System specification
System design
Detailed design
Acceptance test plan
System integration test plan
Sub-system integration test plan
Module and unit code and test
Service
Acceptance test
System integration test
Sub-system integration test

Proses Testing

- **System Testing**, pengujian terhadap integrasi sub-system, yaitu keterhubungan antar sub-system
- **Acceptance Testing** adalah 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)

1. **Component testing** adalah pengujian komponen-komponen program dan biasanya dilakukan oleh component developer (kecuali untuk system kritis).
2. **Integration testing** merupakan pengujian kelompok komponen-komponen yang terintegrasi untuk membentuk sub-system ataupun system, dilakukan oleh tim penguji yang independent. Pengujian berdasarkan spesifikasi sistem.



Rencana Pengujian

- **Proses testing**

  Deskripsi fase-fase utama dalam pengujian

- **Pelacakan Kebutuhan**

  Semua kebutuhan user diuji secara individu

- **Item yang 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.

Failures & Faults

1. **Failure** : output yang tidak benar/tidak sesuai ketika sistem dijalankan.
2. **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

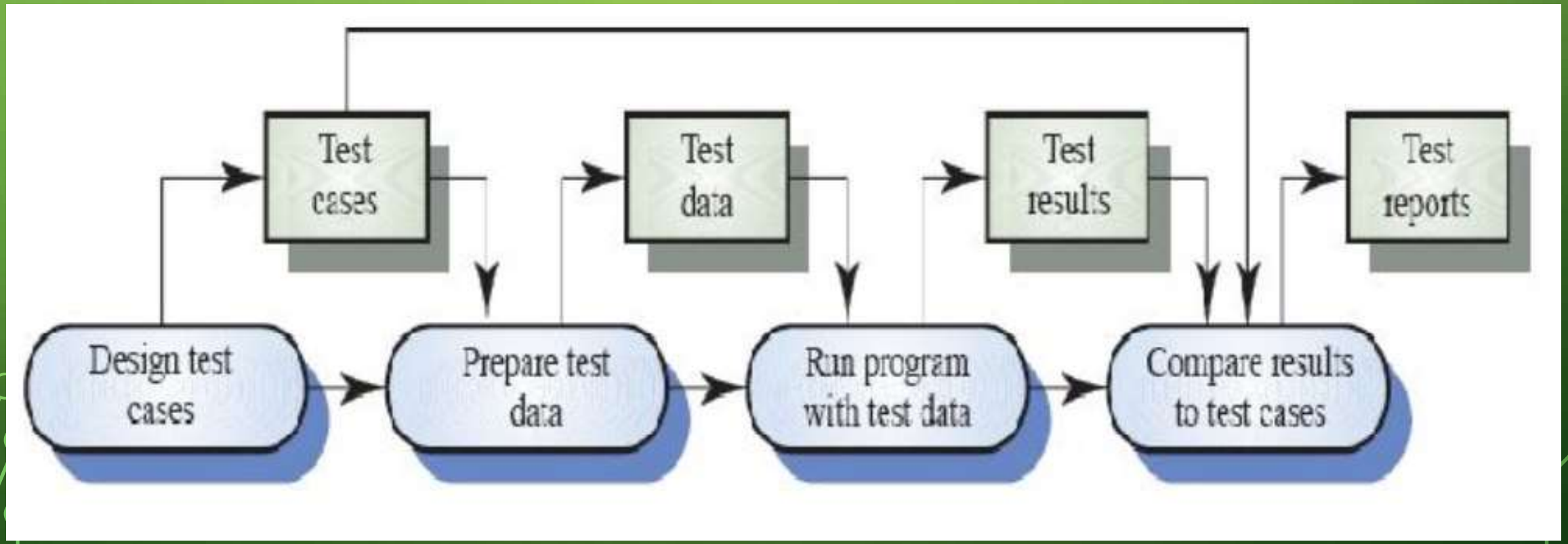
PRIORITAS TESTING

- Hanya test yang lengkap yang dapat memastikan sistem terbebas dari kesalahan, tetap 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.

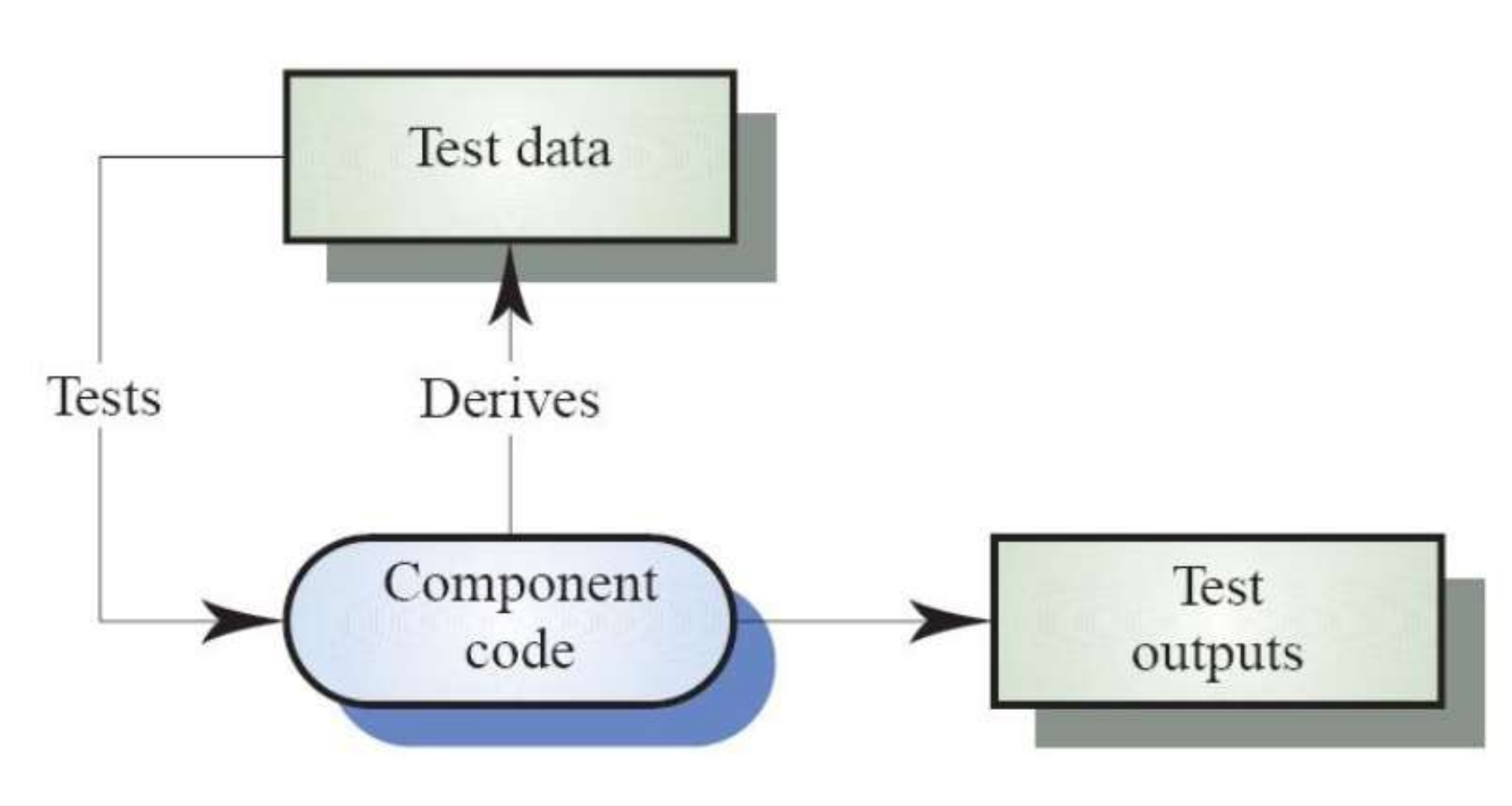
Test Data & Test Kasus

- Test data : Input yang 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



STRUKTURAL TESTING

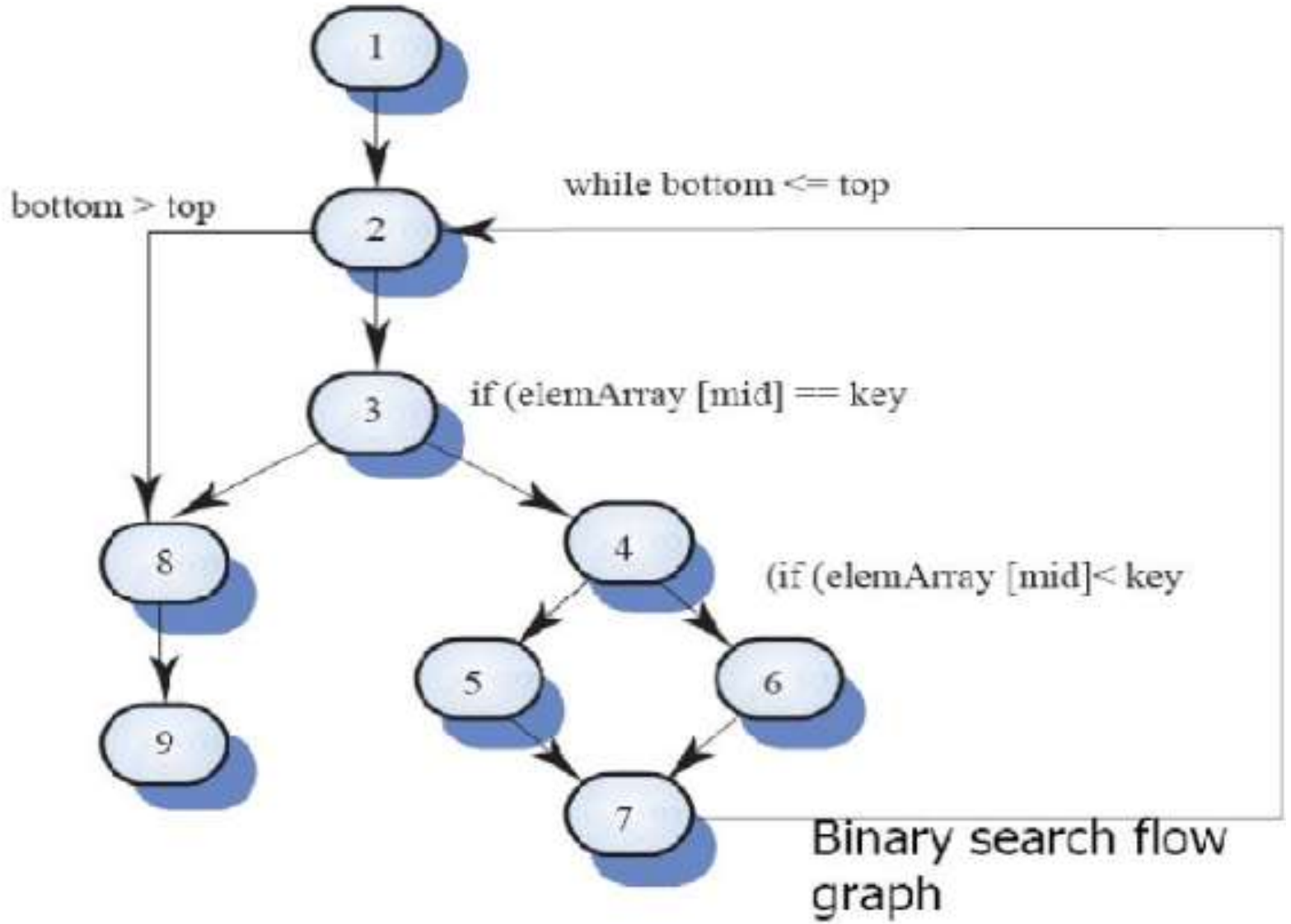


PATH TESTING

- Tujuannya memastikan bahwa himpunan test case akan menguji setiap path pada suatu program paling sedikit satu kali.
- Titik awal untuk path testing adalah suatu program flow graph yang menunjukkan node-node yang menyatakan program decisions (mis.: If-then-else condition) dan busur menyatakan alur control.
- Statements dengan conditions adalah node-node dalam flow graf.

Program Flow Graph

- Menggambarkan alur kontrol. Setiap cabang ditunjukkan oleh path yang terpisah dan loop ditunjukkan oleh arrows looping kembali ke loop kondisi node. Digunakan sebagai basis untuk menghitung cyclomatic complexity.
- Cyclomatic complexity = (Jumlah edges –Jumlah Node) + 2.
- Cyclomatic complexity menyatakan jumlah test untuk menguji control statements



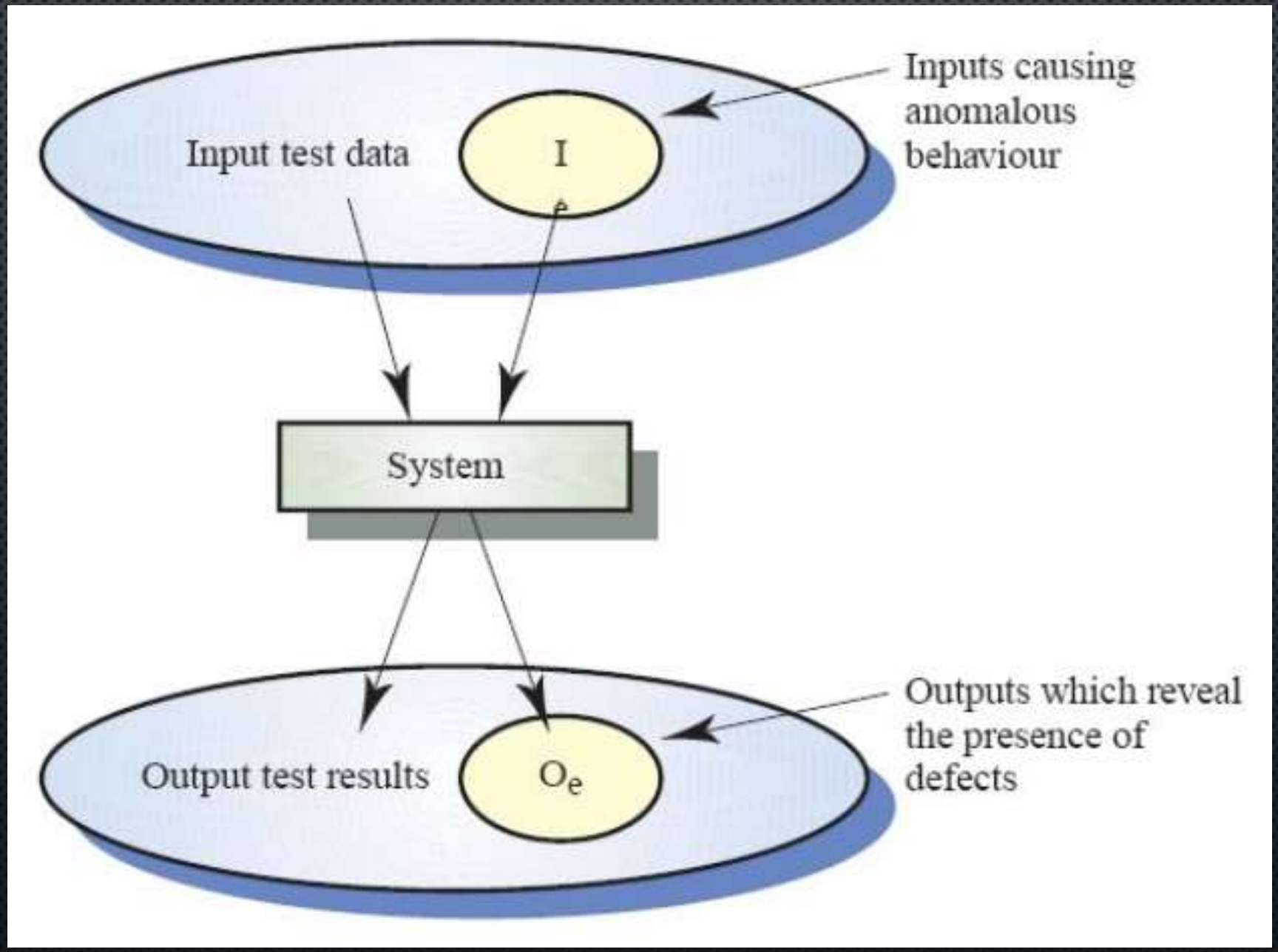
Binary search flow graph

Independent Path

- 1, 2, 3, 8, 9
- 1, 2, 3, 4, 6, 7, 2
- 1, 2, 3, 4, 5, 7, 2
- 1, 2, 3, 4, 6, 7, 2, 8, 9
- Test cases harus ditentukan sehingga semua path tersebut tereksekusi.

Black Box Testing

- Pendekatan pengujian dimana program dianggap sebagai suatu 'black-box' ('kotak hitam').
- Program test case berbasiskan spesifikasi.
- Test planning dapat dimulai sejak awal proses pengembangan sistem.


Input test data
I
e
Inputs causing anomalous behaviour
System
Output test results
Oe
Outputs which reveal the presence of defects

PENGUJIAN BLACK BOX BERUSAHA MENEMUKAN KESALAHAN DALAM KATEGORI:

- Fungsi-fungsi yang tidak benar atau hilang
- Kesalahan interface
- Kesalahan dalam struktur data atau akses database eksternal
- Kesalahan kinerja
- Inisialisasi dan kesalahan terminasi