

KAPITA SELEKTA



Disusun oleh :

1204007 Wulan Nur Annisah

**Universitas Logistik Bisnis & Internasional
Bandung 2023**

7 PRINSIP DASAR TESTING

1. **Testing shows presence of defects**, tujuan utama dari proses testing adalah menemukan cacat/bug pada suatu software. Pengujian dapat menunjukkan cacat yang hadir, tetapi tidak dapat membuktikan bahwa tidak ada cacat. Bahkan setelah pengujian aplikasi atau produk secara menyeluruh kita tidak bisa mengatakan bahwa produk tersebut adalah 100% bebas cacat. Pengujian selalu mengurangi jumlah cacat yang belum ditemukan yang tersisa dalam perangkat lunak tetapi bahkan jika tidak ada cacat yang ditemukan, itu bukan bukti kebenaran.
2. **Exhausting testing is impossible**, Menguji segala sesuatu termasuk semua kombinasi input dan prasyarat tidak mungkin. Jadi, bukannya melakukan pengujian lengkap kita dapat menggunakan risiko dan prioritas untuk memfokuskan upaya pengujian. Jadi, mengakses dan mengelola risiko adalah salah satu kegiatan yang paling penting dan alasan untuk pengujian dalam setiap proyek.
3. **Early testing**, Dalam *Software Development Life Cycle* (SDLC) kegiatan pengujian harus dimulai sedini mungkin dan harus difokuskan pada tujuan yang telah ditetapkan.
4. **Defect clustering**, Sejumlah kecil modul berisi sebagian besar cacat ditemukan selama pengujian pra-rilis atau menunjukkan kegagalan yang paling operasional. Hal ini dapat memudahkan *tester* dalam mengidentifikasi.
5. **Pesticide paradox**, Jika jenis yang sama dari tes diulang lagi dan lagi, akhirnya set yang sama dari kasus uji tidak akan lagi dapat menemukan bug baru. Untuk mengatasi hal ini “Pesticide Paradox”, itu benar-benar sangat penting untuk meninjau kasus uji secara teratur dan tes baru dan berbeda harus ditulis untuk latihan bagian yang berbeda dari perangkat lunak atau sistem untuk berpotensi menemukan lebih banyak cacat.
6. **Testing is context different**, Pengujian pada dasarnya tergantung konteks. Berbagai jenis situs yang diuji berbeda. Misalnya, *Testing software* yang berkaitan erat dengan keamanan tentu berbeda dari sebuah situs e-commerce.

7. **Absence of errors fallacy**, Jika sistem yang dibangun tidak dapat digunakan dan tidak memenuhi kebutuhan dan harapan pengguna kemudian ditemukan dan memperbaiki cacat/bugs. hal ini tidak akan berguna sama sekali.

FUNCTIONAL TESTING

- kebutuhan tentang fungsi software secara menyeluruh
- pemodelan dengan UML, ataupun penjelasan fitur-fitur dalam bentuk problem statements, adalah termasuk dalam functional requirement (UI Test, API Test, Sanity Test)
 - diagrams : use case diagrams, activity diagrams
 - problem statements : harus mencari inventaris, harus melakukan perhitungan, harus menghasilkan laporan tertentu

NON-FUNCTIONAL TESTING

- Operational – Physical/technical environment
- Performance – Speed and reliability
- Security – Who can use the system
- Cultural & Political – Company policies, legal issues

Contoh : Security testing, stress test, load test

STRUCTURAL TESTING

- Cakupan merupakan sejauh mana suatu struktur telah dilaksanakan oleh pengujian, dinyatakan sebagai persentase item yang dicakup
- Jika cakupan tidak 100%, maka lebih banyak tes dapat dirancang untuk menguji item yang terlewatkan untuk meningkatkan cakupan

Contoh : component testing, unit testing, database schema testing

TESTING RELATED TO CHANGE

- Pengujian konfirmasi atau pengujian ulang, setelah cacat terdeteksi dan diperbaiki, perangkat lunak harus diuji ulang untuk memastikan bahwa cacat asli telah berhasil dihilangkan

- Pengujian regresi, adalah pengujian berulang dari program yang sudah diuji, setelah modifikasi, untuk menemukan kesalahan cacat diperkenalkan atau terungkap sebagai akibat dari perubahan
- Contoh : regression testing, confirmation testin