

PERTEMUAN 3

KAPITA SELEKTA



1204011 – WILDAN AZRIL ARVANY

**PROGRAM STUDI D IV TEKNIK INFORMATIKA
UNIVERSITAS LOGISTIK DAN BISNIS INTERNATIONAL
2022**

3 Alasan dunia tidak butuh software testing

by Haris Dermawan

ALASAN 1 Waktu dan Biaya Pengembangan Menjadi Lebih Lama

7 Prinsip Pengujian (Seven Testing Principles)

(1) Pengujian Menunjukkan Adanya Cacat

- ❖ Tetapi tidak dapat membuktikan bahwa tidak ada cacat.
- ❖ Pengujian harus dirancang untuk menemukan sebanyak mungkin cacat Mungkin

(2) Pengujian Lengkap Tidak Mungkin

- ❖ Menguji semuanya (semua kombinasi input dan prasyarat) tidak layak kecuali untuk kasus sepele.
- ❖ Alih-alih pengujian menyeluruh, analisis risiko dan prioritas harus digunakan untuk memfokuskan upaya pengujian.

(3) Pengujian Awal

- ❖ Untuk menemukan cacat lebih awal, kegiatan pengujian harus dimulai sedini mungkin dalam siklus hidup pengembangan perangkat lunak atau sistem, dan harus terfokus atau menetapkan tujuan.

(4) Pengelompokan Cacat

- ❖ Sekitar 80 persen cacat berasal dari 20 persen modul (Pareto–Zipf) [L24]
- ❖ Sejumlah kecil modul biasanya berisi sebagian besar cacat yang ditemukan selama pengujian prailis, atau bertanggung jawab atas sebagian besar kegagalan operasional.
- ❖ Prinsip Pareto 80/20

(5) Paradoks Pestisida

- ❖ Jika pengujian yang sama dilakukan berulang-ulang, pada akhirnya kumpulan kasus pengujian yang sama tidak akan lagi menemukan cacat baru.
- ❖ Uji kasus perlu ditinjau dan direvisi secara berkala

(6) Pengujian Bergantung pada Konteks

- ❖ Menguji id dilakukan secara berbeda dalam konteks yang berbeda
- ❖ Risiko dapat menjadi faktor besar dalam menentukan jenis pengujian yang diperlukan

(7) Ketiadaan kesalahan-kekeliruan

- ❖ Menemukan dan memperbaiki cacat tidak membantu jika sistem yang dibangun tidak dapat digunakan dan tidak memenuhi kebutuhan dan harapan pengguna.

ALASAN 2 Keyakinan Berlebihan pada Pengalaman proyek sebelumnya

Solusi Keterbatasan Sumber Daya yang Dimiliki :

- Menentukan Prioritas: Fokus pada pengujian pada area yang paling penting dan vital dalam perangkat lunak. Tentukan fitur atau fungsi mana yang paling mempengaruhi keseluruhan kinerja sistem dan fokus pengujian pada area tersebut.
- Automatisasi Pengujian: Otomatisasi pengujian perangkat lunak dapat membantu perusahaan menghemat waktu dan sumber daya yang diperlukan. Beberapa jenis pengujian, seperti pengujian regresi, dapat diotomatisasi dengan alat pengujian perangkat lunak.
- Outsourcing Pengujian: Mempekerjakan layanan pengujian perangkat lunak pihak ketiga atau mengontrak pengujian ke perusahaan yang spesialis dalam pengujian perangkat lunak dapat membantu perusahaan menghemat waktu dan sumber daya yang diperlukan.
- Menggunakan Metode Pengujian yang Efisien: Memilih metode pengujian yang efisien dan efektif dapat membantu perusahaan mengoptimalkan penggunaan sumber daya yang tersedia. Misalnya, pengujian exploratory dapat membantu perusahaan menemukan kesalahan dengan cepat dan efisien.
- Menggunakan Tools Gratis atau Open-Source: Terdapat beberapa alat pengujian perangkat lunak gratis atau open-source yang dapat membantu perusahaan melakukan pengujian dengan biaya yang rendah.

Karakteristik Black Box Testing :

- Test condition, test case, dan data uji berasal dari test basis yang dapat mencakup persyaratan perangkat lunak, spesifikasi, kasus penggunaan, dan user story
- Test case dapat digunakan untuk mendeteksi kesenjangan antara persyaratan dan implementasi persyaratan, serta penyimpangan dari persyaratan
- Cakupan diukur berdasarkan item yang diuji dalam dasar pengujian dan teknik yang diterapkan pada dasar pengujian

ALASAN 3 Keterbatasan Sumber Daya yang Dimiliki

Jadi Mengapa Pengujian Diperlukan?

- Karena perangkat lunak cenderung memiliki kesalahan
- Untuk mempelajari tentang keandalan perangkat lunak
- Untuk mengisi waktu antara pengiriman perangkat lunak dan tanggal rilis
- Untuk membuktikan bahwa perangkat lunak tidak memiliki kesalahan
- Karena pengujian termasuk dalam rencana proyek
- Karena kegagalan bisa sangat mahal
- Untuk menghindari dituntut oleh pelanggan
- Untuk bertahan dalam bisnis

Test Type :

• **Functional testing**

- Kebutuhan tentang fungsi software secara menyeluruh
- Pemodelan dengan UML, ataupun penjelasan fitur-fitur dalam bentuk pernyataan masalah, adalah termasuk dalam Persyaratan Fungsional
- Diagram:
 - Gunakan Diagram Kasus
 - Diagram Aktivitas
- Pernyataan Masalah:
 - Harus mencari inventaris
 - Harus melakukan perhitungan ini
 - Harus menghasilkan laporan khusus

- Functional testing adalah jenis pengujian perangkat lunak yang menguji apakah fungsionalitas atau fitur-fitur dari perangkat lunak sudah berjalan dengan baik dan sesuai dengan spesifikasi. Contoh dari jenis ini antara lain:
 - Pengujian fungsionalitas tombol "submit" di halaman web.
 - Pengujian fungsionalitas pencarian produk di situs e-commerce.
 - Pengujian fungsionalitas login pada sebuah aplikasi.

- **Non-functional testing**

- Operational – Physical/technical environment
- Performance – Speed and reliability
- Security – Who can use the system
- Cultural & Political – Company policies, legal issues
- Non-Functional testing adalah jenis pengujian perangkat lunak yang menguji aspek-aspek non-fungsional dari perangkat lunak, seperti performa, keamanan, dan skalabilitas. Contoh dari jenis ini antara lain:
 - Pengujian performa situs web saat ada banyak pengunjung yang membukanya.
 - Pengujian keamanan aplikasi web untuk memastikan tidak ada celah keamanan yang dapat dimanfaatkan oleh pihak yang tidak bertanggung jawab.
 - Pengujian skalabilitas sebuah sistem untuk memastikan sistem tersebut dapat menangani jumlah pengguna yang besar.

- **Structural testing**

- Cakupan adalah sejauh mana struktur telah dilaksanakan oleh pengujian, dinyatakan sebagai persentase dari item yang tercakup
- Jika cakupannya tidak 100%, maka pengujian lebih lanjut dapat dirancang untuk menguji item-item yang terlewat untuk meningkatkan cakupan
- Non-Functional testing adalah jenis pengujian perangkat lunak yang menguji aspek-aspek non-fungsional dari perangkat lunak, seperti performa, keamanan, dan skalabilitas. Contoh dari jenis ini antara lain:
 - Pengujian performa situs web saat ada banyak pengunjung yang membukanya.

- Pengujian keamanan aplikasi web untuk memastikan tidak ada celah keamanan yang dapat dimanfaatkan oleh pihak yang tidak bertanggung jawab.
- Pengujian skalabilitas sebuah sistem untuk memastikan sistem tersebut dapat menangani jumlah pengguna yang besar.

- **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 dimodifikasi, untuk menemukan cacat yang diperkenalkan atau ditemukan sebagai akibat dari perubahan
- Testing related to change adalah jenis pengujian perangkat lunak yang dilakukan saat terjadi perubahan pada kode program atau konfigurasi sistem. Contoh dari jenis ini antara lain:
 - Pengujian regresi setelah perubahan kode program untuk memastikan perubahan tersebut tidak merusak fungsionalitas yang sudah ada sebelumnya.
 - Pengujian integrasi setelah perubahan konfigurasi sistem untuk memastikan bahwa semua sistem terintegrasi dengan baik setelah perubahan tersebut.
 - Pengujian smoke testing setelah perubahan kecil untuk memastikan perubahan tersebut tidak mengganggu fungsionalitas utama dari perangkat lunak.