PERTEMUAN 3

KAPITA SELEKTA

1204021 – MAYKE ANDANI ROHMANIAR

PROGRAM STUDI D IV TEKNIK INFORMATIKA UNIVERSITAS LOGISTIK DAN BISNIS INTERNATIONAL

2023

3 Alasan dunia tidak butuh software testing

by Haris Dermawan

Kegagalan Project Software

50% lebih project teknologi informasi gagal (42% - Standish Group, 53% - General Accounting Office) • Dibatalkan sebelum selesai

• Selesai tapi tidak pernah dipakai

• Tidak bermanfaat bagi pengguna

• Tidak sesuai dengan keinginan pengguna

Sistem Kritis Keselamatan

• Kesalahan perangkat lunak dapat menyebabkan kematian atau cedera:

• Perawatan radiasi membunuh pasien (Therac-25)

• Pengemudi kereta tewas

• Kecelakaan pesawat (Airbus & Korean Airlines)

• Surat cerukan sistem bank menyebabkan bunuh diri

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 cacatMungkin

(2) Pengujian Lengkap Tidak Mungkin

❖ Menguji semuanya (semua kombinasi input dan prasyarat) tidak layakkecuali untuk kasus sepele.

❖ Alih-alih pengujian menyeluruh, analisis risiko dan prioritas harusdigunakan untuk memfokuskan upaya pengujian.

(3) Pengujian Awal

(4) Pengelompokan Cacat

Sekitar 80 persen cacat berasal dari 20 persen modul (Pareto–Zipf) [L24]

❖ Sejumlah kecil modul biasanya berisi sebagian besar cacat yangditemukan

selama pengujian prarilis, 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 cacatbaru.

❖ 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 pengujianyang diperlukan

(7) Ketiadaan kesalahan-kekeliruan

❖ Menemukan dan memperbaiki cacat tidak membantu jika sistem yangdibangun 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 palingpenting dan vital dalam perangkat lunak. Tentukan fitur atau fungsimana 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 dapatmembantu perusahaan menghemat waktu dan sumber daya yang diperlukan.
- Menggunakan Metode Pengujian yang Efisien: Memilih metodepengujian

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 beberapaalat pengujian perangkat lunak gratis atau open-source yang dapatmembantu perusahaan melakukan pengujian dengan biaya yang rendah.

Karakteristik Black Box Testing :

- Test condition, test case, dan data uji berasal dari test basis yangdapat 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 pengujiandan 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 tanggalrilis
- 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 fiturfitur dalam bentukpernyataan 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

- Non-functional testing

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

- 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 meningkatkancakupan

- Testing related to Change

- Pengujian konfirmasi atau pengujian ulang, setelah cacat terdeteksi dan diperbaiki, perangkat lunak harus diuji ulang untuk memastikanbahwa cacat asli telah berhasil dihilangkan
- Pengujian regresi, adalah pengujian berulang dari program yang sudahdiuji, setelah dimodifikasi, untuk menemukan cacat yang diperkenalkan atau ditemukan sebagai akibat dari perubahan