

Tugas Rangkuman
Ditujukan sebagai salah satu syarat
Untuk memperoleh nilai pada Mata Kuliah Kapita Selekt
Program Studi DIV Teknik Informatika



ULBI

Universitas Logistik & Bisnis Internasional

Oleh
Jose Chasey Pratama (1204004)

Program Studi
Diploma IV Teknik Informatika
Politeknik Pos Indonesia
BANDUNG 2023

3 Alasan dunia tidak butuh software testing

by Haris Dermawan

Mengapa Pengujian Diperlukan?

- Karena perangkat lunak cenderung memiliki kesalahan
- Untuk mengetahui tentang keandalan perangkat lunak
- Karena pengujian termasuk dalam rencana proyek
- Karena kegagalan dapat sangat mahal
- Untuk menghindari gugatan dari pelanggan
- Untuk tetap berada dalam bisnis

1. Alasan 1 waktu dan biaya pengembangan menjadi lebih lama

- 7 Prinsip Pengujian (Seven Testing Principles)
 - Testing shows presence of defects : Tujuan dari pengujian perangkat lunak adalah untuk menemukan cacat yang ada. Pengujian harus dirancang untuk menemukan cacat sebanyak mungkin. Contoh: Pengujian pada aplikasi e-commerce dilakukan untuk menemukan cacat seperti bug, error, atau ketidakcocokan pada antarmuka pengguna.
 - Exhaustive testing is impossible : Pengujian yang lengkap dan menyeluruh terhadap semua kondisi masukan dan prasyarat tidak mungkin dilakukan, kecuali untuk kasus yang sangat sederhana. Karena itu, analisis risiko dan prioritas harus digunakan untuk memusatkan usaha pengujian pada hal-hal yang paling penting. Contoh: Pengujian pada aplikasi perbankan harus difokuskan pada transaksi penting seperti transfer dana, pembayaran tagihan, dan penarikan uang.
 - Early testing : Pengujian harus dimulai sedini mungkin dalam siklus hidup pengembangan perangkat lunak atau sistem, dan harus terfokus atau menetapkan tujuan. Contoh: Pengujian unit harus dilakukan sebelum mengintegrasikan modul atau komponen menjadi sistem yang lebih besar.
 - Defect clustering : Sejumlah kecil modul biasanya bertanggung jawab atas sebagian besar cacat yang ditemukan selama pengujian pre-release atau bertanggung jawab atas sebagian besar kegagalan operasional. Prinsip Pareto 80/20. Contoh: Modul manajemen inventaris pada aplikasi toko online bertanggung jawab atas sebagian besar cacat yang ditemukan selama pengujian.
 - Pesticide paradox : Jika pengujian yang sama diulang terus menerus, pada akhirnya kumpulan kasus pengujian yang sama tidak akan lagi menemukan cacat baru. Kasus pengujian perlu ditinjau dan direvisi secara berkala.

Contoh: Kasus pengujian pada aplikasi media sosial harus diperbarui secara berkala untuk memastikan tidak ada cacat baru yang muncul.

- Testing is context dependent : Pengujian dilakukan secara berbeda dalam konteks yang berbeda. Risiko dapat menjadi faktor besar dalam menentukan jenis pengujian yang diperlukan. Contoh: Pengujian pada aplikasi kesehatan harus mempertimbangkan risiko terhadap kesehatan dan keselamatan pengguna.
- Absence-of-errors fallacy : Menemukan dan memperbaiki cacat tidak membantu jika sistem yang dibangun tidak dapat digunakan dan tidak memenuhi kebutuhan dan harapan pengguna. Contoh: Meskipun tidak ada cacat pada aplikasi karya seni digital, tetapi jika aplikasi tersebut tidak memenuhi harapan pengguna seperti fitur atau antarmuka yang tidak sesuai, maka aplikasi tersebut tidak akan sukses.
- Penentuan Risiko mulai dari dampak dari risiko tersebut, seperti Menimbulkan biaya kegagalan, Merilis perangkat lunak yang belum teruji atau kurang teruji, dan Kehilangan kredibilitas dan pangsa pasar. dari dampak tersebut, bisa untuk menentukan Apa yang paling banyak diuji, Seberapa menyeluruh menguji setiap item, Apa yang tidak boleh diuji, Alokasikan waktu yang tersedia untuk pengujian dengan memprioritaskan pengujian.
- Tahapan STLC (Software Testing Life Cycle)
 - Test planning : menetapkan tujuan dan pendekatan pengujian untuk memenuhi batasan yang ada. Contoh: membuat rencana pengujian untuk mengetes fitur baru dalam sebuah aplikasi.
 - Test monitoring and control : memantau kemajuan pengujian dan membandingkannya dengan rencana pengujian. Contoh: memastikan bahwa pengujian berjalan sesuai jadwal yang ditentukan dalam rencana pengujian.
 - Test analysis : menganalisis basis pengujian untuk mengidentifikasi fitur yang dapat diuji dan menentukan kondisi pengujian terkait. Contoh: menganalisis fitur-fitur yang harus diuji pada sebuah aplikasi e-commerce.
 - Test design : merancang dan memprioritaskan test case dan mengidentifikasi data pengujian yang diperlukan. Contoh: merancang kasus pengujian untuk memastikan bahwa fitur pencarian pada sebuah situs web berfungsi dengan baik.
 - Test implementation : membuat perangkat pengujian yang diperlukan dan mengurutkan kasus uji. Contoh: membuat perangkat lunak pengujian otomatis untuk mengecek apakah sebuah website responsif terhadap berbagai jenis perangkat.
 - Test execution : menjalankan test dan mencatat detail pengujian. Contoh: menjalankan test pada aplikasi mobile untuk memeriksa apakah fitur-fitur berjalan dengan baik pada berbagai jenis perangkat dan OS.
 - Test completion : memastikan bahwa semua laporan cacat telah ditutup dan semua item pengujian telah dicatat dengan benar. Contoh: memeriksa bahwa semua cacat pada sebuah situs web telah diperbaiki sebelum rilis ke publik.

2. Alasan 2 Keyakinan Berlebihan pada Pengalaman proyek sebelumnya

- Test development process :
 - Analisis: menentukan kondisi pengujian

- Desain: merancang kasus pengujian untuk memverifikasi kondisi atau tujuan pengujian
- Implementasi: menuliskan prosedur pengujian dan jadwal pelaksanaannya
- Eksekusi: menjalankan prosedur pengujian dan melaporkan ketidaksesuaian hasil test kepada pengembang jika ada.
- Black Box Testing :
 - Test blackbox menggunakan test basis, termasuk persyaratan, spesifikasi, kasus penggunaan, dan user story sebagai sumber kondisi, kasus uji, dan data uji
 - Test case black box dapat mengidentifikasi kesenjangan dan penyimpangan dari persyaratan dan implementasi persyaratan
 - Cakupan blackbox diukur berdasarkan item yang diuji dalam dasar pengujian dan teknik yang digunakan.
- White Box Testing :
 - White Box testing adalah pengujian berdasarkan struktur perangkat lunak atau sistem
 - Tujuan dari teknik white box adalah mengukur cakupan uji dan merancang kasus uji
 - Teknik ini digunakan untuk menilai jumlah pengujian dan merancang test tambahan untuk meningkatkan cakupan test.

3. Alasan 3 keterbatasan sumber daya yang dimiliki

- Solusi Keterbatasan Sumber Daya :
 - Memprioritaskan pengujian: Fokus pada area perangkat lunak yang paling penting dan vital untuk diuji, ditentukan oleh fitur atau fungsi apa yang paling mempengaruhi kinerja sistem secara keseluruhan.
 - Otomatisasi pengujian: Mengotomatiskan pengujian perangkat lunak dapat membantu menghemat waktu dan sumber daya, terutama untuk pengujian regresi, yang dapat diotomatiskan dengan alat pengujian perangkat lunak.
 - Pengujian outsourcing: Menyewa layanan pengujian perangkat lunak pihak ketiga atau mengontrakkan pengujian kepada perusahaan pengujian perangkat lunak khusus dapat menghemat waktu dan sumber daya.
 - Menggunakan metode pengujian yang efisien: Memilih metode pengujian yang efisien dan efektif dapat membantu mengoptimalkan penggunaan sumber daya yang tersedia, seperti menggunakan pengujian eksploratif untuk menemukan kesalahan dengan cepat dan efisien.
 - Menggunakan alat pengujian gratis atau open-source: Beberapa alat pengujian perangkat lunak gratis atau sumber terbuka dapat membantu perusahaan menguji perangkat lunak dengan biaya rendah.