

Nama : Rindu Tegar Senjawati

Software Quality Assurance

1. Apa itu QA (Quality Assurance) dan apa peran utama seorang QA Engineer dalam siklus pengembangan perangkat lunak?

Quality Assurance (QA) adalah proses sistematis yang memastikan bahwa perangkat lunak yang dikembangkan memenuhi standar yang telah ditetapkan sebelum rilis ke pengguna. Fokusnya pada *defect prevention* dengan penerapan proses, metode dan standar yang tepat dalam siklus pengembangan perangkat lunak.

Peran utama QA Engineer dalam siklus pengembangan perangkat lunak:

- 1). Analisis kebutuhan (terkait kebutuhan bisnis dan teknis produk yang akan dikembangkan)
- 2). Menyusun *Test Planning* (strategi pengujian berdasarkan spesifikasi dan kebutuhan aplikasi, menentukan cakupan pengujian, jenis pengujian dan lingkungan pengujian)
- 3). Membuat *test case* berdasarkan persyaratan fungsional dan non-fungsional
- 4). Melakukan pengujian perangkat lunak seperti *functional testing*, *regression testing*, *performance testing* dan *security testing* (manual maupun *automation*)
- 5). Melakukan pelaporan dan mendokumentasikan *bug* (bisa menggunakan *management tools* seperti Jira, Trello dan Asana)
- 6). Melakukan validasi dan verifikasi terhadap produk yang dikembangkan sesuai sebelum rilis

2. Jelaskan perbedaan antara pengujian fungsional dan pengujian non-fungsional dalam konteks QA

Pengujian Fungsional: Menguji perangkat lunak berfungsi sesuai dengan persyaratan yang telah ditentukan dan bisa dilakukan secara manual maupun otomatis dengan automation tools.

Contohnya: Unit Testing, User Acceptance Testing

Pengujian Non-Fungsional: Menguji aspek kualitas yang tidak terkait langsung dengan fungsi utama aplikasi seperti keamanan dan kinerja.

Contohnya: Performance Testing, Stress Testing

3. Bagaimana Anda akan merencanakan dan melakukan pengujian manual untuk aplikasi web e-commerce? Berikan langkah-langkah utama dalam proses ini.

- 1). Test Planning
Menentukan cakupan pengujian dan strategi yang akan digunakan. Dengan Langkah-langkah berikut:
 - Memahami alur dan persyaratan bisnis dan teknis aplikasi web e-commerce
 - Menentukan fitur utama
 - Memilih jenis pengujian yang diperlukan
 - Menyiapkan lingkungan pengujian (Browser, perangkat dan koneksi internet)
 - Identifikasi resiko yang mungkin terjadi
- 2). Membuat Test Case
Test case untuk aplikasi web e-commerce dapat dilihat pada tautan berikut:
https://docs.google.com/spreadsheets/d/168WgAM264LFam_6NXRiMeQX7NYlxnvfcJw_eDNvxMwRk/edit?usp=sharing
- 3). Eksekusi Pengujian

- Menjalankan test case sesuai dengan skenario yang telah dibuat.
 - Menguji aplikasi di berbagai browser dan perangkat
 - Mencatat hasil pengujian: Lulus (Passed) atau Gagal (Failed).
- 4). Melaporkan bug dan dokumentasi
- Merekam langkah-langkah reproduksi bug (misalnya: login gagal meskipun data valid).
 - Melampirkan screenshot atau video untuk memperjelas bug.
 - Menggunakan tools seperti JIRA, Trello, atau Asana
 - Mengelompokkan bug berdasarkan priority dan severity
- 5). Pengujian ulang dan regression testing
- Mengeksekusi kembali test case yang sebelumnya gagal.
 - Melakukan Regression Testing untuk memastikan perubahan kode tidak merusak fitur lain

4. Apa yang dimaksud dengan uji regresi? Mengapa uji regresi penting dalam pengujian perangkat lunak?

Uji regresi (regression testing) adalah pengujian yang memastikan bahwa perubahan atau perbaikan kode tidak menyebabkan masalah atau merusak fungsionalitas yang sudah ada. Regression testing penting untuk mencegah bug baru akibat perubahan kode, mengurangi kemungkinan adanya bug besar yang harus diperbaiki setelah aplikasi rilis sehingga mengurangi biaya perbaikan bug.

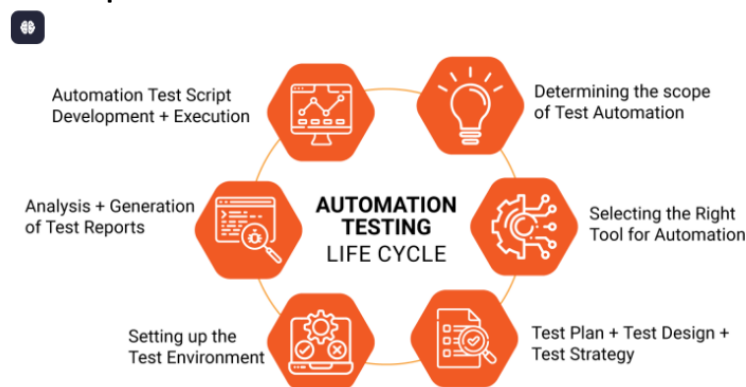
5. Apa perbedaan antara pengujian otomatis dan pengujian manual? Kapan Anda harus menggunakan pengujian otomatis daripada pengujian manual?

Pengujian otomatis: dilakukan menggunakan skrip atau automation tools seperti JMeter, Selenium. Cocok untuk pengujian yang dapat diulang dengan scenario tetap seperti stress testing, performance testing.

Pengujian manual: dilakukan langsung oleh QA tanpa bantuan skrip atau automation tools dan fleksibel untuk pengujian eksploratif dan kasus yang tidak dapat di prediksi. Cocok untuk UAT.

Pengujian otomatis dilakukan untuk pengujian yang dilakukan berulang, pengujian performa, pengujian skala besar.

6. Gambarkan aliran kerja atau proses yang akan Anda ikuti untuk mengotomatisasi pengujian menggunakan alat seperti Selenium.



Gambar 1. Workflow Automation Testing

- 1). Menentukan Cakupan Otomatisasi Pengujian
 - Identifikasi fitur dan skenario yang akan diuji otomatis.
 - Prioritaskan pengujian regresi, uji fungsional, dan uji beban.
 - Pastikan pengujian otomatis memberikan manfaat dibanding pengujian manual.
- 2). Memilih Alat Otomasi yang Tepat
 - Contohnya menggunakan Selenium WebDriver untuk mengotomatisasi browser.
 - Pilih framework sesuai kebutuhan.
- 3). Perencanaan Pengujian + Desain Pengujian + Strategi Pengujian
 - Buat rencana pengujian otomatisasi yang mencakup cakupan dan jadwal eksekusi.
 - Rancang skrip uji berbasis Page Object Model (POM) agar mudah dipelihara.
 - Gunakan pendekatan Data-Driven Testing untuk berbagai skenario input.
- 4). Menyiapkan Lingkungan Pengujian
 - Instal browser driver seperti ChromeDriver atau GeckoDriver.
 - Konfigurasi framework pengujian dengan dependensi yang diperlukan.
- 5). Pengembangan dan Eksekusi Skrip Pengujian Otomatis
 - Tulis skrip Selenium untuk mengotomatisasi pengujian UI aplikasi.
 - Jalankan pengujian secara lokal.
- 6). Analisis dan Pembuatan Laporan Hasil Pengujian
 - Evaluasi hasil pengujian dan catat bug menggunakan tools seperti JIRA.

7. Apa itu kerangka kerja pengujian (testing framework) dalam pengujian otomatis, dan bagaimana kerangka kerja ini membantu dalam pengujian perangkat lunak?

Testing framework adalah sekumpulan aturan, tools dan praktik yang digunakan untuk Menyusun dan menjalankan pengujian otomatis. Framework membantu dalam menulis, mengelola, dan eksekusi pengujian secara sistematis.

Testing framework membantu hal-hal berikut dalam pengujian perangkat lunak:

- Meningkatkan efisiensi dan reusability
- Mempermudah manajemen pengujian
- Meningkatkan akurasi dan stabilitas pengujian

8. Apa yang dimaksud dengan "bug tracking system" dan sebutkan beberapa alat umum yang digunakan untuk melacak dan mengelola bug dalam perangkat lunak.

Bug tracking system adalah system atau alat yang digunakan untuk mencatat, melacak dan mengelola bug dalam perangkat lunak selama siklus pengembangan. Didalamnya bisa mendokumentasikan bug, menentukan prioritas dan tingkat keparahan bug, mengalokasikan bug, melacak status bug dan menganalisis pola bug untuk mencegah kesalahan serupa di masa depan.

Jenis-jenis bug tracking system tools:

- Jira
- Trello
- Asana

9. Berikan contoh skenario uji fungsional untuk aplikasi pemesanan tiket pesawat secara online.

Skenario uji fungsionalitas untuk aplikasi pemesanan tiket pesawat online dapat dilihat pada tautan berikut:

https://docs.google.com/spreadsheets/d/1cKsgynOev_xcAx8zb531n7uQMdH2iDCZGSmPOsypfH0/edit?usp=sharing

10. Anda sedang menguji aplikasi perbankan online. Bagaimana Anda akan menguji keamanan aplikasi ini? Berikan beberapa contoh uji keamanan yang akan Anda lakukan

- 1). Authentication & Authorization Testing
Memastikan hanya pengguna yang sah yang dapat mengakses akun dan fitur tertentu.
Contoh pengujian:
 - Uji apakah sistem menolak login dengan password yang salah.
 - Pastikan sistem mengunci akun setelah beberapa kali gagal login
- 2). Cross-Site Scripting (XSS) Testing
Mencegah injeksi skrip berbahaya yang dapat mencuri informasi pengguna.
Contoh pengujian:
 - Masukkan skrip <script>alert('XSS')</script> di kolom input (seperti nama pengguna atau chat).
- 3). Cross-Site Request Forgery (CSRF) Testing
Mencegah serangan CSRF di mana pengguna tanpa sadar melakukan tindakan di aplikasi perbankan
Contoh pengujian:
 - Akses halaman transaksi melalui link pihak ketiga untuk melihat apakah transaksi bisa terjadi tanpa otorisasi pengguna
- 4). Session Management Testing
Memastikan sesi pengguna diamankan dan tidak bisa diambil alih oleh pihak lain
Contoh pengujian:
 - Cek apakah sesi berakhir secara otomatis setelah periode idle tertentu.
 - Uji apakah sesi tetap aktif setelah logout atau berpindah perangkat.
 - Coba gunakan token sesi pengguna lain untuk melihat apakah akses bisa direplikasi
- 5). File Upload & Input Validation Testing
Mencegah pengguna mengunggah file berbahaya atau memasukkan data yang tidak valid
Contoh pengujian:
 - Unggah file dengan ekstensi .exe atau .php untuk melihat apakah sistem memblokirnya

11. Pada Aplikasi Mobile Shopee, pada Fitur Pesanan Saya - Tab Belum Bayar terdapat penambahan aturan terkait Edit Jenis Pembayaran hanya bisa dilakukan pada kondisi order

: a) Jika Order tsb belum dibayar

b) Jenis Pembayaran yg sudah dipilih memiliki status 'gagal-bayar'

Jika salah satu syarat tsb terpenuhi maka pada Fitur Pesanan Saya- Tab Belum Bayar pada baris order nya terdapat button 'Ubah Pembayaran'.

Coba Buatkan Test Case terkait update fitur tsb dengan menggunakan Test Variable serta contoh Test Data berdasarkan ketentuan diatas

Test Case dan Test Data aplikasi mobile shopee dapat diakses melalui tautan berikut:

<https://docs.google.com/spreadsheets/d/1gC8lQ0evJMQG9iUNUWEmm1y9lmBFNoSy9UklwpSA-9Q/edit?usp=sharing>