Test QA Engineer – Fadhil Muhammad - FaMuh5f/Skill-Test-KNITTO (github.com)

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

   Quality Assurance (QA) adalah proses yang bertujuan untuk memastikan kualitas suatu produk, dalam konteks ini perangkat lunak, agar sesuai dengan standar yang telah ditetapkan sebelum dirilis ke pengguna akhir. QA berfokus pada pencegahan masalah atau cacat sejak tahap awal pengembangan, bukan hanya sekadar menemukan bug saat pengujian akhir.

   Peran utama seorang QA Engineer dalam siklus pengembangan perangkat lunak adalah menjaga agar produk yang dikembangkan berjalan dengan baik dan memenuhi kebutuhan serta harapan pengguna. QA Engineer terlibat dalam seluruh siklus pengembangan, mulai dari analisis kebutuhan (requirements analysis), desain, pengembangan, pengujian, hingga rilis produk.

   Pada tahap awal, QA Engineer bekerja sama dengan tim pengembang dan pemangku kepentingan (stakeholders) untuk memahami kebutuhan bisnis dan pengguna. Di sini, mereka memastikan bahwa spesifikasi dan persyaratan yang ditetapkan realistis, dapat dicapai, dan selaras dengan tujuan produk. Dalam fase pengembangan, mereka memantau proses untuk mencegah potensi penyimpangan dari rencana yang telah ditetapkan.

   Selama fase pengujian, QA Engineer bertugas merancang, menyiapkan, dan melaksanakan berbagai jenis pengujian (seperti pengujian fungsional, non-fungsional, keamanan, dan performa) untuk memastikan bahwa perangkat lunak bebas dari kesalahan dan cacat, serta siap untuk digunakan oleh pengguna akhir.

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

   Pengujian fungsional adalah proses pengujian yang bertujuan untuk memastikan bahwa setiap fitur atau fungsi dari perangkat lunak telah bekerja sesuai dengan spesifikasi yang telah ditentukan. Pengujian ini berfokus pada aspek operasional dari aplikasi dan memastikan bahwa setiap persyaratan fungsional yang diberikan oleh pelanggan atau pemangku kepentingan sudah terpenuhi. Pengujian fungsional melibatkan verifikasi input dan output sesuai harapan, interaksi antarmuka pengguna, dan alur kerja logis dari aplikasi. Contoh pengujian fungsional mencakup

pengujian formulir login, pengiriman data, dan operasi CRUD (Create, Read, Update, Delete) pada basis data.

Pengujian non-fungsional bertujuan untuk memeriksa aspek-aspek yang tidak terkait langsung dengan fungsi dasar aplikasi, tetapi lebih ke karakteristik kualitas sistem secara keseluruhan. Pengujian ini meliputi performa, keamanan, kegunaan (usability), skalabilitas, reliabilitas, dan kompatibilitas sistem. Fokus dari pengujian non-fungsional adalah untuk memastikan bahwa aplikasi mampu beroperasi secara efektif dan efisien di bawah berbagai kondisi. Misalnya, pengujian beban (load testing) dilakukan untuk melihat bagaimana aplikasi berperforma saat menghadapi banyak pengguna secara bersamaan, sementara pengujian keamanan (security testing) bertujuan untuk mengidentifikasi celah keamanan yang bisa dieksploitasi.

3. Bagaimana Anda akan merencanakan dan melakukan pengujian manual untuk aplikasi web e-commerce? Berikan langkah-langkah utama dalam proses ini.
    1. Perancangan Pengujian
        a. Memahami spesifikasi aplikasi e-commerce, memahami persyaratan bisnis dan teknis untuk dapat mengerti bagaimana aplikasi web e-commerce seharusnya berfungsi.
        b. Identifikasi kasus pengujian (test case), Contohnya :
            i. Registrasi pengguna dan login
            ii. Pencarian dan pemfilteran produk
            iii. Penambahan produk ke keranjang belanja
            iv. Proses checkout dan pembayaran
            v. Notifikasi email atau SMS terkait pesanan
            vi. Pengelolaan akun pengguna (alamat pengiriman, riwayat pembelian)
            vii. Keamanan, seperti perlindungan terhadap serangan SQL Injection dan XSS
    2. Mempersiapkan lingkungan pengujian
        a. Setup lingkungan pengujian, untuk mencerminkan kondisi produksi yang sebenarnya sehingga diharapkan mendapatkan hasil pengujian yang lebih komprehensif.
        b. Mengumpulkan data uji, data uji yang realistis seperti akun pengguna fiktif, produk, dan detail kartu credit dummy (untuk

pengujian pembayaran)

3. Membuat dan menjalankan kasus pengujian
    a. Buat kasus pengujian (test case) yang detail, tuliskan langkah-langkah pengujian manual dalam bentuk dokumen kasus uji. Kasus ini mencakupi kasus skenario positif, dimana fungsi berjalan dengan benar dan kasus skenario negatif, dimana fungsi berjalan dengan salah demi melakukan pengujian bagaimana aplikasi dapat mengatasi error (*error handling*)
    b. Eksekusi kasus pengujian, pengujian dilakukan sesuai dengan kasus pengujian yang telah disiapkan, pencatatan cacat (bug) dilakukan bila ditemukan saat melakukan eksekusi.
    c. Pengujian fungsional, pengujian yang fokus pada fitur-fitur utama e-commerce, contohnya :
        i. Pengujian UI/UX
        ii. Pengujian formulir dan validasi
        iii. Pengujian proses pembelian
        iv. dll
    d. Pengujian non-fungsional, pengujian yang fokus pada aspek-aspek kegunaan aplikasi e-commerce, contohnya :
        i. Pengujian kinerja
        ii. Pengujian keamanan
        iii. Pengujian kompatibilitas
        iv. dll
4. Pelaporan dan tindak lanjut
    a. Dokumentasi hasil pengujian, catatan yang berisi hasil dari semua eksekusi kasus pengujian maupun memberikan hasil positif dan negatif.
    b. Laporkan bug, laporkan hasil eksekusi menggunakan sistem pelaporan bug (seperti JIRA dan Bugzilla) untuk melacak bug yang ditemukan dan komunikasi dengan tim pengembangan dan tim penguji yang lain.
    c. Tidak lanjut dan retesting, setelah tim pengembang melakukan perbaikan pada bug yang ada pada sistem pelaporan bug maka akan dilakukan testing kembali atas bug tersebut untuk memastikan

bahwa bug tidak ditemukan lagi.

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

   Uji regresi adalah proses pengujian ulang pada fitur-fitur yang sudah ada dalam perangkat lunak setelah dilakukan perubahan, seperti penambahan fitur baru, perbaikan bug, atau peningkatan performa. Tujuan utama dari uji regresi adalah memastikan bahwa perubahan tersebut tidak mempengaruhi fungsi yang sudah berjalan dengan baik sebelumnya. Pengujian ini sangat penting untuk menjaga stabilitas dan kualitas perangkat lunak, karena perubahan yang tidak diantisipasi dapat menyebabkan malfungsi atau kerusakan pada bagian lain dari sistem.

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

   Perbedaan antara pengujian otomatis dan pengujian manual terletak pada cara pelaksanaan dan alat yang digunakan. Pengujian manual dilakukan oleh tester yang secara langsung menjalankan tes pada aplikasi atau sistem, melibatkan pemeriksaan elemen antarmuka pengguna, proses fungsional, dan hasil keluaran untuk memastikan bahwa sistem berjalan sesuai dengan yang diharapkan. Pengujian ini sering kali membutuhkan intuisi dan perhatian detail dari manusia untuk mendeteksi bug yang mungkin tidak terduga.

   Di sisi lain, pengujian otomatis dilakukan dengan bantuan skrip atau alat khusus yang mengeksekusi serangkaian tes secara otomatis. Tester menulis skrip yang mereplikasi interaksi pengguna dengan sistem dan memeriksa hasil secara otomatis tanpa campur tangan manusia. Pengujian otomatis sangat efektif untuk menguji skenario yang berulang, regresi, dan pengujian volume besar yang akan terlalu memakan waktu jika dilakukan secara manual.

   Pengujian manual umumnya digunakan dalam situasi di mana pengujian yang dilakukan memerlukan pemahaman intuitif tentang perilaku sistem, misalnya, pada pengujian antarmuka pengguna (UI), pengalaman pengguna (UX), atau ketika fitur baru pertama kali diperkenalkan. Dalam kasus-kasus seperti ini, tester dapat mengidentifikasi bug yang lebih halus dan masalah pengalaman pengguna yang mungkin tidak terdeteksi oleh skrip otomatis.

   Pengujian otomatis sebaiknya digunakan ketika:

- Ada banyak kasus uji yang berulang yang harus dijalankan berkali-kali, seperti pengujian regresi.
- Sistem memerlukan pengujian pada berbagai konfigurasi atau perangkat yang berbeda secara simultan.
- Pengujian perlu dilakukan dengan cepat dan efisien, terutama ketika batas waktu ketat.
- Validasi rutin perlu dilakukan setelah pembaruan atau perubahan kecil dalam kode.

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

   Dijawab pada folder “No 6. Aliran kerja selenium” di repository github

7. Apa itu kerangka kerja pengujian (testing framework) dalam pengujian otomatis, dan bagaimana kerangka kerja ini membantu dalam pengujian perangkat lunak? (sumber: Test Automation Frameworks | SmartBear)

   Kerangka kerja pengujian adalah seperangkat pedoman atau aturan yang digunakan untuk membuat dan merancang kasus pengujian. Kerangka kerja terdiri dari kombinasi praktik dan alat yang dirancang untuk membantu profesional QA melakukan pengujian dengan lebih efisien.

   Pedoman ini dapat mencakup standar pengkodean, metode penanganan data pengujian, repositori objek, proses penyimpanan hasil pengujian, atau informasi tentang cara mengakses sumber daya eksternal.

   Kerangka kerja pengujian memiliki 6 tipe yang umum digunakan yaitu,

   - Linear Automation Framework
   - Modular Based Testing Framework
   - Library Architecture Testing Framework
   - Data-Driven Framework
   - Keyword-Driven Framework
   - Hybrid Testing Framework

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 sebuah sistem yang dirancang khusus untuk memonitor dan melacak bug atau masalah yang ditemukan dalam perangkat lunak. Sistem ini memungkinkan tim *Quality Assurance* (QA) untuk mencatat bug baru yang ditemukan selama proses pengujian, serta memantau status perbaikan yang dilakukan oleh tim pengembang. Dengan menggunakan *bug tracking system*, tim pengembang dapat mengakses laporan bug secara terstruktur, memperbaiki masalah, dan kemudian mengembalikannya ke tim QA untuk diuji ulang guna memastikan bahwa bug tersebut telah diselesaikan.

Bug tracking system sangat penting dalam proses pengembangan perangkat lunak, karena memungkinkan komunikasi yang efisien antara tim QA dan tim pengembang. Selain itu, sistem ini juga membantu menjaga catatan riwayat bug, memprioritaskan perbaikan berdasarkan tingkat keparahan, serta memastikan bahwa tidak ada bug yang terlewatkan atau tidak teratasi.

Beberapa alat *bug tracking* yang umum digunakan dalam industri pengembangan perangkat lunak termasuk **JIRA**, yang merupakan platform populer untuk manajemen proyek dan pelacakan bug, serta **Bugzilla**, alat *open source* yang sering digunakan untuk melacak bug dan masalah pada perangkat lunak. Alat-alat ini tidak hanya mendukung pelacakan bug, tetapi juga integrasi dengan berbagai alat pengembangan lainnya, seperti sistem kontrol versi dan alat manajemen tugas, untuk menciptakan alur kerja yang lebih terorganisir dan kolaboratif.

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

- Registrasi Pengguna dan Login

Skenario 1	Pengguna dapat berhasil melakukan registrasi dengan data yang valid.
Langkah	Masukkan nama, email, password, dan konfirmasi password, kemudian klik "Daftar".
Hasil yang diharapkan	Pengguna mendapatkan notifikasi bahwa akun telah berhasil dibuat, dan email verifikasi dikirim ke email terdaftar.

Skenario 2	Pengguna dapat login dengan kredensial yang benar.
Langkah	Masukkan email dan password yang telah terdaftar, kemudian klik "Login".

Hasil yang diharapkan	Pengguna diarahkan ke dashboard utama aplikasi.

Skenario 3	Kesalahan ditampilkan jika pengguna memasukkan password yang salah.
Langkah	Masukkan email yang terdaftar dan password yang salah, klik "Login".
Hasil yang diharapkan	Muncul pesan error yang menyatakan bahwa kombinasi email dan password tidak sesuai.

- Pencarian dan Pemfilteran Tiket Pesawat

Skenario 1	Pengguna dapat mencari tiket pesawat berdasarkan kota asal dan tujuan.
Langkah	Masukkan kota asal dan tujuan, tanggal keberangkatan, dan jumlah penumpang, kemudian klik "Cari".
Hasil yang diharapkan	Hasil pencarian tiket pesawat yang tersedia muncul di layar.

Skenario 2	Pengguna dapat memfilter tiket berdasarkan harga, maskapai, atau waktu keberangkatan.
Langkah	Setelah hasil pencarian muncul, pilih filter berdasarkan harga terendah hingga tertinggi atau waktu keberangkatan paling awal.
Hasil yang diharapkan	Daftar tiket disaring sesuai dengan kriteria filter yang dipilih.

Skenario 3	Pencarian menampilkan hasil yang relevan untuk tiket pulang-pergi.
Langkah	Pilih opsi "Pulang-Pergi", masukkan kota asal dan tujuan, tanggal keberangkatan, dan tanggal pulang.
Hasil yang diharapkan	Daftar tiket untuk perjalanan pulang-pergi muncul di layar.

- Proses Checkout dan Pembayaran Tiket Pesawat

Skenario 1	Pengguna dapat melakukan pemesanan tiket hingga proses checkout dengan lancar.
Langkah	Pilih tiket yang diinginkan dari hasil pencarian, masukkan informasi penumpang, dan lanjutkan ke proses pembayaran.
Hasil yang diharapkan	Pengguna dapat melihat rincian harga tiket dan mengonfirmasi pemesanan.

Skenario 2	Pengguna dapat memilih metode pembayaran dan menyelesaikan transaksi.
Langkah	Pilih metode pembayaran (kartu kredit, transfer bank, e-wallet), masukkan detail pembayaran, dan klik "Bayar Sekarang".
Hasil yang diharapkan	Pengguna diarahkan ke halaman konfirmasi pembayaran, dan pembayaran berhasil diproses.

Skenario 3	Sistem menampilkan pesan error jika pembayaran gagal.
Langkah	Lakukan pembayaran menggunakan kartu kredit yang tidak valid atau saldo yang tidak mencukupi.
Hasil yang diharapkan	Muncul pesan error yang menyatakan pembayaran gagal dan memberikan opsi untuk mencoba metode pembayaran lain.

- Notifikasi Email atau SMS Terkait Pesanan

Skenario 1	Pengguna menerima email atau SMS konfirmasi setelah pembayaran berhasil.
Langkah	Selesaikan pembayaran tiket.
Hasil yang diharapkan	Pengguna menerima email atau SMS yang berisi detail pemesanan dan konfirmasi pembayaran.

Skenario 2	Pengguna menerima email atau SMS jika terjadi pembatalan tiket.
Langkah	Lakukan pembatalan tiket melalui aplikasi.
Hasil yang diharapkan	Pengguna menerima email atau SMS konfirmasi bahwa tiket telah dibatalkan.

- Pengelolaan Akun Pengguna

Skenario 1	Pengguna dapat melihat dan mengelola informasi pribadi pada akun.
Langkah	Masuk ke akun, buka bagian pengaturan akun, dan edit nama, email, atau password.
Hasil yang diharapkan	Informasi pribadi berhasil diperbarui, dan sistem menampilkan pesan sukses.

Skenario 2	Pengguna dapat melihat riwayat pembelian tiket pesawat.
Langkah	Masuk ke akun, buka tab "Riwayat Pemesanan".
Hasil yang diharapkan	Riwayat tiket yang pernah dibeli muncul di layar, lengkap dengan detail perjalanan.

Skenario 3	Pengguna dapat menghapus akun atau menonaktifkan akun sementara.
Langkah	Masuk ke pengaturan akun, pilih opsi "Hapus Akun" atau "Nonaktifkan Akun".
Hasil yang diharapkan	Pengguna mendapatkan notifikasi bahwa akun berhasil dihapus atau dinonaktifkan.

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

    Dalam pengujian keamanan aplikasi perbankan online, terdapat beberapa langkah penting yang harus dilakukan untuk memastikan bahwa aplikasi tersebut terlindungi dari berbagai ancaman siber dikarenakan data nasabah yang sangat sensitif. Berikut adalah beberapa contoh uji keamanan yang dapat dilakukan:

    1. Pengujian SQL Injection

       Uji ini bertujuan untuk mengidentifikasi apakah aplikasi rentan terhadap serangan injeksi SQL, yang dapat dimanfaatkan oleh penyerang untuk mendapatkan akses tidak sah ke data nasabah atau melakukan manipulasi pada basis data. Pengujian dilakukan dengan memasukkan input yang berpotensi mengeksekusi perintah SQL berbahaya untuk menguji respons aplikasi.

    2. Pengujian Cross-Site Scripting (XSS)

       Uji XSS dilakukan untuk melihat apakah aplikasi rentan terhadap skrip berbahaya yang disuntikkan oleh pengguna jahat ke dalam halaman web. Jika berhasil, penyerang dapat mencuri informasi sensitif seperti kredensial login pengguna atau melakukan aksi atas nama pengguna yang sah.

    3. Pengujian DDoS (Distributed Denial of Service)

       Uji DDos akan melakukan simulasi serangan DDoS untuk mengevaluasi seberapa baik aplikasi dapat menahan beban lalu lintas yang sangat tinggi secara tiba-tiba. Hal ini bertujuan untuk memastikan bahwa aplikasi tetap dapat diakses dan berfungsi normal meskipun ada upaya penyerangan untuk membuat layanan tidak tersedia.