

Soal dan Jawaban Test QA Engineer.

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

Jawaban:

QA (Quality Assurance) adalah proses sistematis untuk memastikan bahwa produk perangkat lunak memenuhi standar kualitas yang ditetapkan dan bebas dari bug atau masalah sebelum dirilis. Peran utama seorang QA Engineer dalam siklus pengembangan perangkat lunak meliputi:

- Merancang rencana pengujian berdasarkan persyaratan perangkat lunak.
- Menentukan metodologi dan alat pengujian yang akan digunakan.
- Menulis dan mengembangkan kasus uji yang mencakup skenario fungsional dan non-fungsional.
- Melakukan pengujian manual dan otomatisasi pada perangkat lunak.
- Mengidentifikasi dan mendokumentasikan bug atau masalah.
- Menyusun laporan hasil pengujian dan analisis.
- Bekerja sama dengan tim pengembang untuk memperbaiki bug yang ditemukan.
- Memastikan bahwa perangkat lunak yang diuji memenuhi standar kualitas dan persyaratan yang ditetapkan.
- Memverifikasi bahwa perbaikan bug telah berhasil dilakukan tanpa menimbulkan masalah baru.

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

Jawaban:

- a. **Pengujian Fungsional** dilakukan untuk memeriksa apakah fitur dan fungsi perangkat lunak bekerja sesuai dengan spesifikasi yang telah ditetapkan yang dimana berfokus pada apa yang dilakukan sistem.
Contoh: Sanity Testing, Regression Testing, Smoke Testing.
- b. **Pengujian Non-Fungsional** dilakukan untuk Memeriksa aspek kualitas perangkat lunak yang tidak terkait dengan fungsi spesifik, seperti kinerja, keandalan, dan keamanan yang dimana berfokus pada bagaimana sistem bekerja.
Contoh: Performance Testing, Security Testing, Load Testing.

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

Jawaban:

Berikut adalah langkah-langkah utama dalam proses pengujian manual untuk Aplikasi Web E-commerce:

1. Perencanaan Pengujian

- Membuat rencana pengujian yang mencakup tujuan pengujian, lingkup, strategi, jadwal, sumber daya, dan alat yang akan digunakan.
- Mengidentifikasi bagian-bagian aplikasi yang akan diuji seperti halaman produk, keranjang belanja, pembayaran, akun pengguna, dll.

2. Pengembangan Kasus Uji

- Menulis kasus uji berdasarkan persyaratan fungsional. Setiap kasus uji harus memiliki langkah-langkah yang jelas, input, dan hasil yang diharapkan.

- Menyiapkan data uji yang diperlukan seperti akun pengguna, produk, dan detail pembayaran.

3. Pelaksanaan Pengujian

- Melaksanakan kasus uji satu per satu dan mencatat hasilnya lalu membandingkan dengan hasil yang diharapkan.
- Mendokumentasikan bug atau masalah yang ditemukan dengan detail, termasuk langkah-langkah untuk mereproduksi masalah, tangkapan layar, dan informasi terkait lainnya.

4. Pelaporan dan Tindak Lanjut

- Menyusun Laporan yang merangkum hasil pengujian, termasuk kasus uji yang lulus, gagal, dan yang memerlukan perhatian lebih lanjut.
- Menyampaikan temuan kepada tim pengembang dan bekerja sama untuk memperbaiki bug yang ditemukan.
- Menjalankan kembali kasus uji untuk memastikan perbaikan berhasil dan tidak menyebabkan masalah baru.

5. Pengujian Regresi

- Melakukan pengujian regresi untuk memastikan perubahan baru tidak merusak fungsionalitas yang ada.
- Memperbaharui kasus uji jika diperlukan berdasarkan perubahan atau penambahan fitur.

6. Penutupan Pengujian

- Mereview proses pengujian untuk mengidentifikasi area yang dapat ditingkatkan.
- Menyimpan semua dokumentasi pengujian untuk referensi di masa depan dan sebagai bagian dari dokumentasi proyek.

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

Jawaban:

Uji regresi adalah jenis pengujian perangkat lunak yang dilakukan untuk memastikan bahwa perubahan terbaru, seperti bug fix atau penambahan fitur baru, tidak menyebabkan kerusakan atau gangguan pada fungsionalitas yang sudah ada sebelumnya. Uji regresi penting dilakukan karena beberapa faktor di bawah ini:

- Memastikan bahwa fungsi-fungsi yang sudah ada tetap berfungsi dengan benar setelah perubahan.
- Mengidentifikasi bug yang mungkin muncul akibat perubahan kode.
- Menjamin kualitas dan keandalan perangkat lunak sebelum dirilis ke pengguna.
- Mengurangi risiko kegagalan sistem saat perangkat lunak di-deploy ke produksi.

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

Jawaban:

a. Perbedaan antara Pengujian Otomatis dan Pengujian Manual.

- Pengujian Manual
 - Definisi: Pengujian yang dilakukan secara langsung oleh penguji manusia tanpa bantuan alat otomatis.
 - Kelebihan:
 - Fleksibel dan dapat menyesuaikan dengan perubahan mendadak.

- Kekurangan:
 - Memakan waktu dan tenaga.
 - Rentan terhadap kesalahan manusia.
 - Kurang efisien untuk pengujian yang harus dilakukan berulang kali.
- Pengujian Otomatis
- Definisi: Pengujian yang dilakukan menggunakan alat atau skrip otomatis untuk menjalankan tes dan memeriksa hasil.
- Kelebihan:
 - Cepat dan efisien untuk pengujian yang dilakukan berulang kali.
 - Mengurangi kemungkinan kesalahan manusia.
 - Dapat menjalankan tes yang kompleks dan berulang dengan konsisten.
- Kekurangan:
 - Memerlukan investasi awal dalam pengembangan skrip dan alat.
 - Kurang fleksibel dalam menangani perubahan mendadak.
 - Memerlukan pemeliharaan untuk memastikan skrip tetap relevan.

b. Waktu Penggunaan Pengujian Otomatis daripada Pengujian Manual.

- Pengujian Berulang
- Tes yang Kompleks dan Menyeluruh
- Frekuensi Tinggi
- Efisiensi Waktu

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

Jawaban:

Berikut proses yang dapat dilakukan untuk mengotomatisasi pengujian menggunakan tools Selenium:

- Instal Selenium WebDriver untuk bahasa pemrograman yang dipilih.
- Siapkan Integrated Development Environment (IDE) seperti Visual Studio Code.
- Unduh dan konfigurasi driver browser yang diperlukan.
- Buat proyek baru dalam IDE dan tambahkan pustaka Selenium.
- Tulis skrip uji menggunakan Selenium WebDriver. Skrip ini harus mencakup:
 - Meluncurkan browser dan membuka URL aplikasi.
 - Mengidentifikasi elemen pada halaman menggunakan metode seperti findElement dan berinteraksi dengan elemen tersebut (misalnya, klik, input teks).
 - Memeriksa hasil pengujian dengan menggunakan asersi untuk memastikan bahwa aplikasi berfungsi seperti yang diharapkan.
 - Mengimplementasikan penanganan kesalahan untuk situasi seperti elemen yang tidak ditemukan atau kesalahan pemuatan halaman.
- Eksekusi skrip uji untuk memeriksa apakah tes berjalan seperti yang diharapkan.
- Identifikasi dan perbaiki bug atau masalah yang muncul selama eksekusi skrip.
- Integrasikan skrip uji dengan sistem Continuous Integration/Continuous Deployment (CI/CD) seperti Jenkins atau GitLab CI. Konfigurasi build otomatis untuk menjalankan pengujian saat kode di-commit atau saat proses build dijalankan.
- Atur jadwal pengujian untuk menjalankan skrip secara berkala atau pada

waktu tertentu.

- Secara berkala perbarui skrip uji untuk mencerminkan perubahan dalam aplikasi (misalnya, perubahan antarmuka pengguna, fitur baru).
- Tinjau hasil pengujian secara rutin dan optimalkan skrip untuk meningkatkan efisiensi dan efektivitas pengujian.
- Kumpulkan dan analisis hasil pengujian. Gunakan alat pelaporan untuk menghasilkan laporan yang jelas dan informatif tentang hasil pengujian.
- Berikan umpan balik kepada tim pengembang tentang masalah yang ditemukan dan status pengujian.

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

Jawaban:

Kerangka kerja pengujian (testing framework) dalam pengujian otomatis adalah kumpulan alat, pustaka, dan aturan yang menyediakan struktur untuk menulis, menjalankan, dan mengelola tes otomatis. Kerangka kerja ini mendefinisikan cara tes diatur, bagaimana hasilnya dilaporkan, dan bagaimana tes berinteraksi dengan aplikasi yang diuji. kerangka kerja ini membantu dalam pengujian perangkat lunak dalam hal berikut:

- Menyediakan struktur yang konsisten untuk menulis dan mengelola tes, memudahkan pemeliharaan dan pemahaman.
- Mempermudah penulisan skrip uji dengan menyediakan pustaka.
- Mendukung pengujian otomatis yang dapat dijalankan berulang kali tanpa campur tangan manual, serta memungkinkan penggunaan kembali kode pengujian.
- Menghasilkan laporan hasil pengujian yang jelas dan terperinci, membantu dalam menganalisis hasil tes dan mengidentifikasi masalah.
- Memungkinkan integrasi dengan alat lain seperti CI/CD, memfasilitasi proses otomatisasi dan continuous testing.

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

Jawaban:

Bug Tracking System (Sistem Pelacakan Bug) adalah alat atau aplikasi yang digunakan untuk melacak, mengelola, dan memantau bug atau masalah dalam perangkat lunak selama siklus hidup pengembangan. Sistem ini membantu tim pengembangan dalam mengidentifikasi, merekam, dan menyelesaikan bug dengan cara yang terorganisir. Berikut beberapa alat umum yang digunakan untuk melacak dan mengelola bug dalam perangkat lunak:

- JIRA: Alat populer yang digunakan untuk manajemen proyek, pelacakan bug, dan pelacakan masalah.
- Bugzilla: Sistem pelacakan bug open-source yang banyak digunakan.

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

Jawaban:

a. Registrasi dan Login Pengguna

No.	Skenario	Test Cases	Data	Hasil Yang Diharapkan	Hasil Pengujian (OK/NOK)	Ket. Pengujian
1	Pengguna mendaftar dan login ke akun yang telah didaftarkan	<ul style="list-style-type: none">Buka aplikasi pemesanan tiket pesawat.Klik tombol "Daftar" dan masukkan informasi yang diperlukan (nama, email, password).Klik tombol "Daftar" untuk menyelesaikan pendaftaran.Keluar dari aplikasi dan klik tombol "Login".Masukkan email dan password, lalu klik "Login".	<ul style="list-style-type: none">Informasi Pengguna	<ul style="list-style-type: none">Pengguna dapat mendaftar dan login ke akun dengan sukses.	OK	Pengujian berhasil dilakukan tanpa ditemukan adanya Bug.

b. Pencarian Penerbangan

No.	Skenario	Test Cases	Data	Hasil Yang Diharapkan	Hasil Pengujian (OK/NOK)	Ket. Pengujian
2	Pengguna mencari penerbangan dengan kriteria tertentu.	<ul style="list-style-type: none">Buka aplikasi pemesanan tiket pesawat.Masukkan kota keberangkatan dan kota tujuan.Pilih tanggal keberangkatan dan tanggal kembali (jika berlaku).Gunakan filter tambahan seperti waktu keberangkatan, harga, atau maskapai.Klik tombol "Cari".	<ul style="list-style-type: none">Informasi Penerbangan	<ul style="list-style-type: none">Aplikasi menampilkan daftar penerbangan yang sesuai dengan kriteria pencarian.Aplikasi menampilkan informasi penerbangan termasuk jadwal, harga, dan maskapai.	OK	Pengujian berhasil dilakukan tanpa ditemukan adanya Bug.

c. Pemesanan Tiket

No.	Skenario	Test Cases	Data	Hasil Yang Diharapkan	Hasil Pengujian (OK/NOK)	Ket. Pengujian
3	Pengguna memesan tiket untuk penerbangan yang dipilih.	<ul style="list-style-type: none">Pilih penerbangan dari daftar hasil pencarian.Pilih kelas tiket (Ekonomi, Bisnis, dll.).Masukkan detail penumpang (nama, kontak, dll.).Pilih metode pembayaran dan masukkan detail pembayaran.Klik tombol "Pesan Sekarang".	<ul style="list-style-type: none">Tiket	<ul style="list-style-type: none">Aplikasi memproses pembayaran dan menampilkan konfirmasi pemesanan.Pengguna menerima email atau notifikasi dengan detail pemesanan dan tiket.	OK	Pengujian berhasil dilakukan tanpa ditemukan adanya Bug.

d. Pembaruan Detail Pemesan

No.	Skenario	Test Cases	Data	Hasil Yang Diharapkan	Hasil Pengujian (OK/NOK)	Ket. Pengujian
4	Pengguna memperbarui detail penumpang untuk pemesanan yang ada.	<ul style="list-style-type: none">Masuk ke akun pengguna.Akses riwayat pemesanan.Pilih tiket yang detailnya ingin diperbarui.Edit detail penumpang (misalnya, nama atau kontak).Simpan perubahan.	<ul style="list-style-type: none">Informasi Penumpang	<ul style="list-style-type: none">Aplikasi menyimpan dan memperbarui detail penumpang.Pengguna menerima konfirmasi pembaruan melalui email atau notifikasi.	OK	Pengujian berhasil dilakukan tanpa ditemukan adanya Bug.

e. Pembatalan Tiket

No.	Skenario	Test Cases	Data	Hasil Yang Diharapkan	Hasil Pengujian (OK/NOK)	Ket. Pengujian
5	Pengguna membatalkan tiket yang telah dipesan.	<ul style="list-style-type: none">Masuk ke akun pengguna.Akses riwayat pemesanan.Pilih tiket yang akan dibatalkan.Klik tombol "Batalkan Pemesanan".Konfirmasi pembatalan.	<ul style="list-style-type: none">Tiket	<ul style="list-style-type: none">Aplikasi mengonfirmasi pembatalan tiket.Pengguna menerima email atau notifikasi tentang status pembatalan dan pengembalian dana jika berlaku.	OK	Pengujian berhasil dilakukan tanpa ditemukan adanya Bug.

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

Jawaban:

Berikut adalah beberapa contoh uji keamanan yang dapat dilakukan untuk aplikasi perbankan online:

- Uji Kekuatan Kata Sandi:
 - Verifikasi bahwa aplikasi memerlukan kata sandi yang kuat dengan kriteria kompleksitas (panjang, kombinasi karakter).
- Uji Pengelolaan Sesi:
 - Periksa apakah sesi pengguna berakhir secara otomatis setelah periode tidak aktif.
 - Uji keamanan sesi untuk memastikan bahwa sesi pengguna tidak dapat disalahgunakan oleh pihak ketiga.
- Uji Otentikasi Multi-Faktor:
 - Verifikasi bahwa otentikasi multi-faktor (MFA) diterapkan dan berfungsi dengan baik untuk akses akun.
- Uji Pengendalian Akses Berbasis Peran:
 - Verifikasi bahwa pengguna hanya dapat mengakses data dan fungsi yang sesuai dengan peran dan izin mereka.
- Uji Privilege Escalation:
 - Coba untuk memperoleh hak akses lebih tinggi atau akses ke fitur yang tidak seharusnya tersedia untuk pengguna dengan hak istimewa rendah.