

Soal Test QA engineer

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

➔ QA adalah profesi yang bertugas untuk menjamin kualitas dari sebuah produk ataupun website. Mencari celah bug, mencari kesalahan atau error system dari sistem yang dibuat atau fitur yang dibuat.

➔ Peran utama seorang QA Engineer :

- Merancang dan menjalankan pengujian perangkat lunak atau aplikasi
- Menganalisis hasil pengujian dan membuat laporan tentang hasilnya
- Memastikan bahwa standar kualitas dan fungsionalitas dipenuhi
- Memastikan bahwa aplikasi siap digunakan oleh pengguna akhir

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

➔ Pengujian fungsional menentukan bahwa pengoperasian program atau aplikasi berjalan sesuai dengan persyaratan teknis dan bisnis serta produk bebas dari bug. Sedangkan pengujian non-fungsional yaitu menguji semua aspek yang tidak tercakup dalam pengujian fungsional seperti kinerja, ketersediaan, dan keandalan perangkat lunak. Serta memverifikasi bagaimana harapan pelanggan terpenuhi.

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

➔ Sebagai seorang QA, merencanakan dan melakukan pengujian manual untuk aplikasi web e-commerce butuh pendekatan yang sistematis untuk memastikan bahwa semua fungsi dan fitur aplikasi berjalan dengan baik.

Berikut langkah-langkah utama dalam proses ini :

- 1) Memahami persyaratannya.
 - meninjau dan memahami spesifikasi bisnis. Identifikasi modul dan fitur utama seperti pendaftaran, login, katalog produk, keranjang belanja, pembayaran, pengiriman serta manajemen akun pengguna.
 - membuat daftar lengkap use case yang mewakili skenario pengguna nyata seperti alur pembelian, pencarian produk dan filter produk.
- 2) Menyusun rencana pengujian
 - menentukan ruang lingkup pengujian, bagian bagian aplikasi yang akan diuji
 - beri prioritas untuk pengujian fitur yang paling sering digunakan
 - tetapkan kriteria lulus (pass/fail) untuk tiap kasus pengujian
- 3) Membuat test case
 - membuat test case manual berdasarkan skenarip yang berbeda. Menyertakan langkah2, input data dan hasil yang diharapkan
 - membuat skenario pengujian positif dan pengujian negatif
- 4) Persiapan pengujian
 - pastikan lingkungan pengujian siap untuk simulasi termasuk database server dan semua integrasi seperti payment gateway dan pengiriman.
 - lakukan diberbagai browser dan semua perangkat untuk memastikan bahwa aplikasi responsif dan lintas platform
- 5) Pelaksanaan pengujian manual
 - lakukan test case secara manual dengan langkah langkah yang telah disusun. Periksa hasil yang diharapkan untuk melihat apakah ada perbedaan atau ada bug.
 - melakukan pengujian fungsionalitas
 - pastikan bahwa semua input pengguna seperti form pendaftaran, login dan pengiriman memiliki validitas yang tepat

→ melakukan pengujian UI/UX, memverifikasi apakah user interface mudah digunakan dan berfungsi dengan baik. Termasuk tata letak, navigasi dan interaksi user.

6) Pengujian non fungsional

→ menguji kinerja aplikasi (uji kecepatan respon aplikasi)

→ menguji aspek keamanan (akses kontrol pengguna, perlindungan data pribadi dan transaksi yang aman)

7) Pelaporan bug dan pengujian ulang

→ jika menemukan bug atau masalah, laporkan dengan detail yang lengkap

→ setelah perbaikan dilakukan oleh tim pengembang, lakukan pengujian ulang untuk memastikan apakah bug tersebut telah diperbaiki dan tidak mempengaruhi bagian lain dari aplikasi

8) Pengujian akhir dan validasi

→ pastikan tidak ada fitur atau fungsi lain yang rusak setelah perbaikan bug atau penambahan fitur baru

→ koordinasikan dengan tim untuk melakukan pengujian penerimaan pengguna, dimana pengguna akhir akan menguji aplikasi dan memberikan masukan

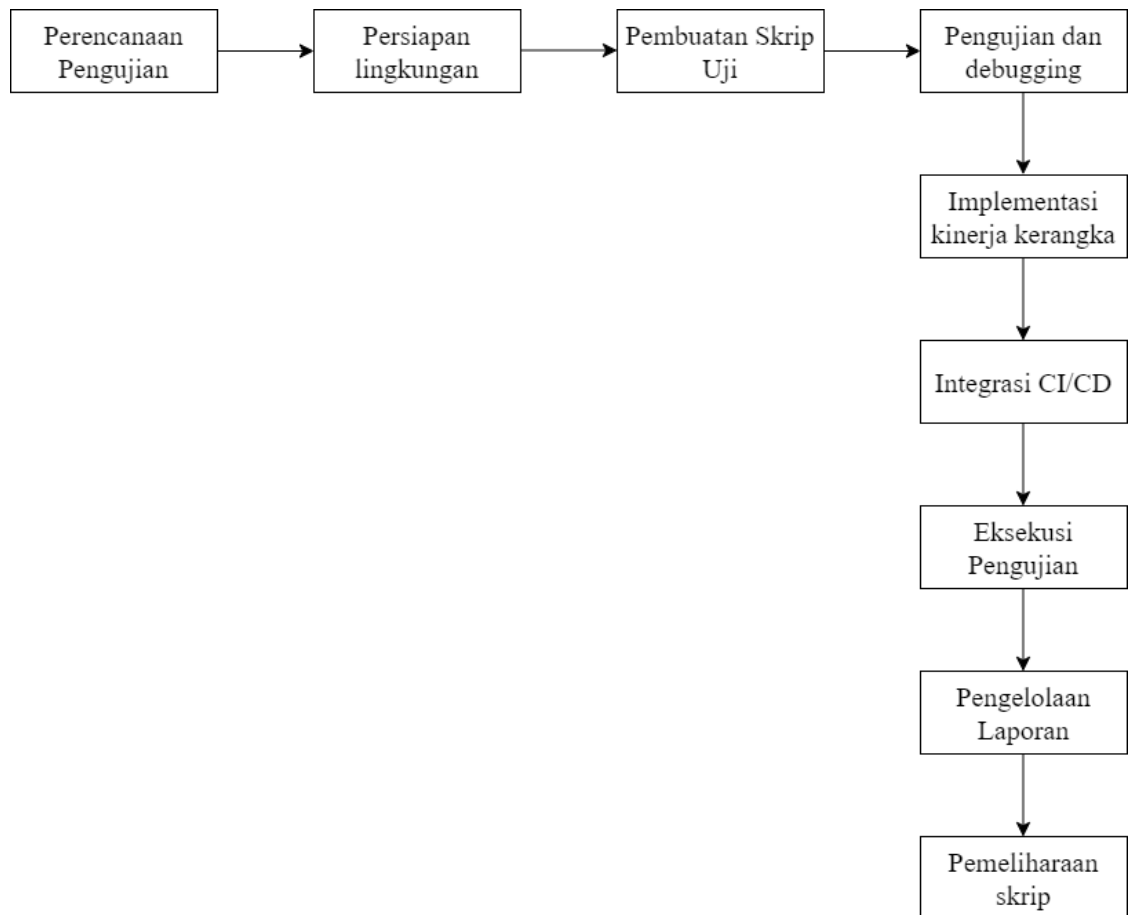
9) Dokumentasi dan kesimpulan

→ buat laporan lengkap mengenai hasil pengujian, termasuk jumlah test case yang gagal dan berhasil serta rekomendasi untuk perbaikan atau penyempurnaan aplikasi

→ berdasarkan hasil pengujian manual, buatlah kesimpulan apakah aplikasi ini sudah siap untuk diluncurkan atau masih memerlukan perbaikan lebih lanjut

4. Apa yang dimaksud dengan uji regresi? Mengapa uji regresi penting dalam pengujian perangkat lunak?
- ➔ Uji regresi mengacu pada proses pemeriksaan fitur baru dalam software apakah telah merusak atau menurunkan fungsionalitas yang sudah ada. Tujuannya adalah memastikan bahwa aplikasi tetap berjalan dengan baik setelah perubahan terakhir.
5. Apa perbedaan antara pengujian otomatis dan pengujian manual? Kapan Anda harus menggunakan pengujian otomatis daripada pengujian manual?
- ➔ Perbedaan antara pengujian manual dan pengujian otomatis adalah bahwa dalam pengujian manual, akan melakukan pengujian langkah demi langkah tanpa bantuan alat apa pun, sedangkan dalam pengujian otomatis, pengujian dijalankan secara otomatis menggunakan alat & kerangka kerja otomatisasi seperti katalon, selenium, appium, dsb.
- ➔ Pengujian otomatis lebih tepat digunakan dalam situasi-situasi tertentu seperti :
- Uji regresi
 - Pengujian yang memerlukan volume data besar
 - Pengujian kinerja (apakah sebuah situs web tetap responsif saat diakses oleh ratusan bahkan ribuan pengguna secara bersamaan?)

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



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

- ➔ Sebuah kerangka kerja pengujian digunakan sebagai panduan aturan dan praktik terbaik yang harus diikuti untuk mencapai hasil terbaik selama proyek otomatisasi. Ini memungkinkan penguji menjadi lebih efisien dalam membuat dan merancang kasus uji.
- ➔ Kerangka kerja membantu memisahkan logika pengujian dari data uji dan logika aplikasi. Kerangka kerja memungkinkan pengembang dan penguji

untuk menulis kode uji yang dapat digunakan kembali di berbagai skenario. Dengan mengikuti kerangka kerja, proses pengujian menjadi lebih terorganisir dan pengujian otomatisasi lebih terukur.

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

➔ Merupakan sistem, program atau aplikasi yang memudahkan proses melacak kesalahan pada sistem sehingga memungkinkan anggota tim proyek untuk melaporkan, mengelola, dan menganalisa laporan kesalahan (Bug report).

➔ Beberapa alat pelacak bug umum digunakan yaitu Jira, Bugzilla, Senin.com.

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

Skenario	Deskripsi	Langkah-langkah	Hasil Yang Diharapkan
Pemesanan tiket	User melakukan pemesanan tiket setelah memilih penerbangan	1. Pilih penerbangan dari hasil pencarian 2. Masukkan informasi penumpang (nama, nomor identitas, tanggal lahir) 3. Pilih kursi dan	User dapat melanjutkan ke halaman pembayaran setelah mengisi semua informasi yang diperlukan

Skenario	Deskripsi	Langkah-langkah	Hasil Yang Diharapkan
		layanan tabahan(apabila ada) 4. Klik tombol “Lanjutkan ke pembayaran”	

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

➔ Beberapa contoh uji keamanan yang dapat dilakukan :

Contoh Uji Keamanan	Deskripsi	Langkah-langkah	Hasil Yang Diharapkan
Uji autentifikasi dan otorisasi	Memastikan hanya user yang terdaftar yang dapat mengakses akun mereka	<ul style="list-style-type: none"> - Uji login dengan kombinasi username dan password yang valid dan tidak valid. - Uji mekanisme two-factor authentication (2FA). 	Hanya pengguna yang berwenang yang dapat mengakses akun dan fitur sesuai dengan hak akses mereka.

Contoh Uji Keamanan	Deskripsi	Langkah-langkah	Hasil Yang Diharapkan
		<ul style="list-style-type: none"> - Coba akses halaman yang seharusnya hanya bisa diakses oleh pengguna lain tanpa otentikasi yang tepat (misalnya, akses ke akun orang lain menggunakan link langsung). 	
Uji ketahanan	Memastikan bahwa aplikasi tahan terhadap serangan brute-force di mana pelaku mencoba menebak kata sandi dengan mencoba berbagai kombinasi.	<ul style="list-style-type: none"> - Coba beberapa upaya login dengan password yang salah secara berurutan. - Pastikan sistem mengunci 	Sistem harus mengunci akun setelah beberapa upaya gagal dan mengirim notifikasi kepada user.

Contoh Uji Keamanan	Deskripsi	Langkah-langkah	Hasil Yang Diharapkan
		akun atau membatasi percobaan login setelah beberapa kali gagal.	
Uji penyimpanan kata sandi	Memastikan bahwa aplikasi tidak menyimpan kata sandi pengguna dalam format teks biasa dan menggunakan metode hashing yang aman.	Verifikasi penyimpanan kata sandi di server dengan melihat apakah password menggunakan algoritma hashing yang kuat seperti bcrypt atau lainnya	Kata sandi disimpan dalam bentuk hash yang aman, tidak dalam teks biasa.
Uji log aktivitas user	Memastikan bahwa aplikasi mencatat semua aktivitas penting yang dilakukan oleh user atau administrator, sehingga tindakan mencurigakan dapat dilacak.	<ul style="list-style-type: none"> - Periksa apakah aplikasi mencatat aktivitas login, logout, perubahan data akun, dan transaksi keuangan. - Menguji apakah 	Semua aktivitas sensitif tercatat dalam log yang aman, dan ada notifikasi jika ada anomali.

Contoh Uji Keamanan	Deskripsi	Langkah-langkah	Hasil Yang Diharapkan
		sistem memberi peringatan kepada user atau administrator jika ada aktivitas mencurigakan.	

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 :
- Jika Order tsb belum dibayar
 - 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.

No	Test Case	Deskripsi	Precondition	Test Step	Expected Result	Test Variable	Contoh Test Data
1	Verifikasi tombol 'Ubah Pembayaran' muncul jika	Memastikan bahwa tombol Ubah Pembayaran muncul pada	Ada pesanan dengan status belum dibayar di	1) Buka aplikasi Shopee 2) Buka Pesanan	Tombol Ubah Pembayaran muncul pada baris pesanan	-Status pesanan -Status pembayarannya	- Status pesanan: belum dibayar - Status

No	Test Case	Deskripsi	Precondition	Test Step	Expected Result	Test Variable	Contoh Test Data
	pesanan belum dibayar	pesanan yang belum dibayar.	Tab Belum Bayar.	Saya - Tab Belum Bayar. 3) Cek apakah tombol Ubah Pembayaran muncul di baris pesanan	jika status pesanan adalah Belum Dibayar.		pembayaran: belum dilakukan
2	Verifikasi tombol 'Ubah Pembayaran' muncul jika pembayaran gagal	Memastikan bahwa tombol Ubah Pembayaran muncul jika metode pembayaran gagal (status 'gagal-bayar').	Ada pesanan dengan status pembayaran gagal di Tab Belum Bayar.	1) Buka aplikasi Shopee. 2) Buka Pesanan Saya - Tab Belum Bayar. 3) Cek apakah tombol Ubah	Tombol Ubah Pembayaran muncul pada baris pesanan jika status pembayaran adalah gagal-bayar.	- Status pesanan - Status pembayaran	- Status pesanan: belum dibayar - Status pembayaran : gagal

No	Test Case	Deskripsi	Precondition	Test Step	Expected Result	Test Variable	Contoh Test Data
				Pembayaran muncul di baris pesanan.			
3	Verifikasi tombol 'Ubah Pembayaran' tidak muncul jika pesanan dibayar	Memastikan bahwa tombol Ubah Pembayaran tidak muncul jika pesanan sudah dibayar.	Ada pesanan dengan status sudah dibayar di Tab Belum Bayar.	1) Buka aplikasi Shopee. 2) Buka Pesanan Saya - Tab Belum Bayar. 3) Cek apakah tombol Ubah Pembayaran muncul di baris pesanan.	Tombol Ubah Pembayaran tidak muncul jika status pesanan adalah sudah dibayar.	- Status pesanan - Status pembayaran	- Status pesanan: sudah dibayar - Status pembayaran : berhasil
4	Verifikasi tombol 'Ubah Pembayaran' muncul	Memastikan bahwa tombol Ubah Pembayaran hanya	Ada pesanan dengan status yang berbeda-	1) Buka aplikasi Shopee. 2) Buka Pesanan	Tombol Ubah Pembayaran hanya muncul pada	- Status pesanan - Status pembayaran	- Status pesanan: belum dibayar/ gagal

No	Test Case	Deskripsi	Precondition	Test Step	Expected Result	Test Variable	Contoh Test Data
	hanya jika ada syarat terpenuhi	muncul jika pesanan belum dibayar atau metode pembayaran gagal.	beda (belum dibayar, sudah dibayar, pembayaran gagal).	Saya - Tab Belum Bayar. 3) Cek tombol Ubah Pembayaran pada semua pesanan dengan status berbeda.	pesanan yang statusnya belum dibayar atau pembayaran gagal.		- Status pembayaran : gagal
5	Verifikasi tindakan setelah mengklik tombol 'Ubah Pembayaran'	Memastikan bahwa ketika tombol Ubah Pembayaran diklik, pengguna diarahkan ke halaman ubah metode pembayaran.	Ada pesanan dengan status belum dibayar atau gagal-bayar.	1) Buka aplikasi Shopee. 2) Buka Pesanan Saya - Tab Belum Bayar. 3) Klik tombol Ubah	Pengguna diarahkan ke halaman ubah metode pembayaran setelah mengklik tombol Ubah Pembayaran.	- Status pesanan - Status pembayaran - Navigasi	- Status pesanan: belum dibayar - Status pembayaran : gagal atau belum dilakukan

No	Test Case	Deskripsi	Precondition	Test Step	Expected Result	Test Variable	Contoh Test Data
				Pembayaran.			
6	Verifikasi perilaku jika tidak ada syarat yang terpenuhi	Memastikan bahwa tombol Ubah Pembayaran tidak muncul jika syarat tidak terpenuhi (pesanan sudah dibayar dan pembayaran berhasil).	Ada pesanan dengan status sudah dibayar dan pembayaran berhasil di Tab Belum Bayar.	1) Buka aplikasi Shopee. 2) Buka Pesanan Saya - Tab Belum Bayar. 3) Cek apakah tombol Ubah Pembayaran muncul pada pesanan.	Tombol Ubah Pembayaran tidak muncul jika pesanan sudah dibayar dan status pembayaran berhasil.	- Status pesanan - Status pembayaran	- Status pesanan: sudah dibayar - Status pembayaran : berhasil