

Nama : Nazwa Fitriyani Zahra

### Soal Test QA engineer

1. Apa itu QA (Quality Assurance) dan apa peran utama seorang QA Engineer dalam siklus pengembangan perangkat lunak?
2. Jelaskan perbedaan antara pengujian fungsional dan pengujian non-fungsional dalam konteks QA.
3. Bagaimana Anda akan merencanakan dan melakukan pengujian manual untuk aplikasi web e-commerce? Berikan langkah-langkah utama dalam proses ini.
4. Apa yang dimaksud dengan uji regresi? Mengapa uji regresi penting dalam pengujian perangkat lunak?
5. Apa perbedaan antara pengujian otomatis dan pengujian manual? Kapan Anda harus menggunakan pengujian otomatis daripada pengujian manual?
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?
8. Apa yang dimaksud dengan "bug tracking system" dan sebutkan beberapa alat umum yang digunakan untuk melacak dan mengelola bug dalam perangkat lunak.
9. Berikan contoh skenario uji fungsional untuk aplikasi pemesanan tiket pesawat secara online.
10. Anda sedang menguji aplikasi perbankan online. Bagaimana Anda akan menguji keamanan aplikasi ini? Berikan beberapa contoh uji keamanan yang akan Anda lakukan.
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.

## Jawaban

1. Quality Assurance adalah proses untuk memastikan bahwa produk perangkat lunak memenuhi standar sebelum perangkat lunak di rilis dan digunakan oleh end-user. Standar tersebut maksudnya perangkat lunak bebas dari bug, aman digunakan dan sesuai dengan requirement dari klien. Peran dari seorang QA antara lain membuat test planning, membuat skenario test case, test execution, dan bug report.
2. Pengujian fungsional lebih ke menguji fungsionalitas / fitur yang ada di aplikasi apakah sesuai dengan yang diharapkan atau tidak, termasuk seperti input outputnya apakah sudah sesuai dengan requirement atau belum. Sedangkan untuk pengujian non-fungsional lebih ke menguji kualitas dari aplikasi tersebut seperti kecepatan aplikasi, performance, atau keamanan. Contohnya menguji seberapa baik aplikasi digunakan apabila terdapat banyak user.
3. Berikut adalah perencanaan yang dilakukan :
  - Pahami dulu requirement dari aplikasi serta spesifikasi dari aplikasi.
  - Buat planning untuk aplikasi web yang akan di test. Tentukan jenis test apa yang akan dilakukan. Misalnya :
    - a. Untuk aplikasi web e-commerce akan dilakukan Unit Test, integration test, dan UI testing.
    - b. Tentukan scope dari pengujian : fitur apa saja yang akan di test dan tidak
    - c. Tentukan akan menggunakan pendekatan apa. Misal aplikasi akan dites menggunakan black box.
  - Tentukan schenario test untuk aplikasi misalnya :
    - a. Untuk Unit test : identifikasi modul apa saja yang akan diujikan. Misalnya pengujian pada modul checkout, modul pembayaran, modul produk.
    - b. Untuk Integration test : identifikasi proses apa yang akan diujikan, modul yang terlibat dan bagaimana antar modul saling berinteraksi. Misalnya pengujian proses pembelian, maka modul yang terlibat adalah modul checkout, modul pembayaran, modul produk. Pengujian harus memastikan bahwa interaksi antara modul-modul ini berjalan lancar dan tidak ada kesalahan saat data berpindah di antara mereka.
    - c. Untuk UI test : identifikasi page apa saja yang ada untuk menguji tampilan aplikasi bekerja dalam berbagai dimensi layar yang berbeda. Hal ini dilakukan mengingat aplikasi yang dites jenisnya adalah website. Misalnya page produk akan diujikan dalam dimensi 1920x1080 (Desktop), 768x1024(iPad), dan 375x667(Smartphone)
  - Tentukan test case untuk setiap schenario test aplikasi berdasarkan requirement dan bussiness rule aplikasi. Ujikan semua kondisinya termasuk negative casenya, misalnya :
    - Requirement : Pengguna dapat melakukan pembayaran dengan sukses.
    - Bussiness Rules : Pembayaran harus diverifikasi dalam waktu 5 menit dan Jika transaksi gagal, sistem harus mengembalikan pengguna ke halaman checkout dengan pesan error.
    - Test Schenario : Checkout dan pembayaran
    - Test Case : Checkout dengan Pembayaran Sukses

- Precondition : Pengguna sudah login ke akun, Pengguna memiliki produk dalam keranjang belanja, dan Sistem pembayaran terintegrasi dengan metode pembayaran yang valid (misalnya, kartu kredit, transfer bank, atau e-wallet).
- Step :
  - a) Akses halaman keranjang belanja.
  - b) Klik tombol "Checkout".
  - c) Pilih alamat pengiriman yang valid.
  - d) Pilih metode pembayaran yang tersedia.
  - e) Masukkan detail pembayaran (nomor kartu kredit, kode CVV, dll.).
  - f) Klik tombol "Bayar" atau "Konfirmasi Pembayaran".
  - g) Tunggu hingga sistem memproses pembayaran.
- Expected Result : Pembayaran berhasil, dan pengguna diarahkan ke halaman konfirmasi pesanan.
- Lakukan test execution untuk test case yang telah dibuat
- Buat bug report untuk mencatat bug yang ada pada aplikasi.

**Catatan :** Untuk mempermudah dalam mencatat test case bisa menggunakan aplikasi seperti spreadsheet untuk mencatat butir uji dan untuk bug report bisa menggunakan aplikasi seperti Jira atau Trello untuk assign ke programmer yang menangani bug tersebut.

4. Uji Regresi adalah pengujian untuk memastikan update atau perubahan pada aplikasi tidak mempengaruhi fungsi yang lain. Perubahan yang dimaksud bisa dari perubahan akibat bug fixing, ada penambahan fitur yang terkait dengan fitur lain, atau modifikasi program. Pengujian regresi ini penting untuk mencegah ada bug baru atau malfunction aplikasi akibat dari hasil modifikasi.
5. Perbedaannya untuk pengujian otomatis dilakukan menggunakan alat atau software untuk menjalankan testnya, sedangkan untuk manual itu proses menjalankan skenarionya dilakukan oleh manusia. Menggunakan pengujian otomatis lebih cocok apabila ada pengujian yang membutuhkan pengulangan, banyak kombinasi data, butuh pengujian multi user atau pada aplikasi yang skalanya besar agar lebih efektif.
6. Berikut adalah perencanaan yang dilakukan :
  - Pahami requirement dan fitur dari aplikasi yang akan diujikan.
  - Tentukan Test Schenario dan Test case dari aplikasi yang akan diujikan
  - Identifikasi test case yang perlu diautomasi. Test case yang cocok untuk di automasi adalah test case yang di eksekusi dengan bermacam macam data set, test case yang di eksekusi pada envirotnment yang berbeda, dan test case yang harus diuji oleh jenis user yang berbeda (dengan berbagai otoritas). Misalnya pengujian untuk fitur login pengguna yang membutuhkan kombinasi data yang berbeda dan jenis user yang berbeda
  - Buat logika untuk setiap test case yang diautomasi berupa step step yang dilakukan dan catat modul apa atau page apa yang terlibat untuk step tersebut. Misalnya :

Test Case	Step to Execute	Page / Modul yang terlibat
Berhasil	- Click Item Produk	- ProductPage
Checkout	- Click Button	- CheckoutPage

Produk	“Tambah Produk” - Click Icon “Cart” - Click Button Checkout	
--------	---	--

- Kelompokkan setiap step menjadi sebuah modul / page. Hal ini dilakukan agar pengujian lebih efektif karena setiap modul bisa reusable atau digunakan kembali. Misalnya:

Page / Modul	Steps	Methode
Product Page	Click Item Produk	chooseProduct()
	Click Tambah Produk	addProducttocart()
	Click Icon “Cart”	showCart()
Checkout Page	Click button Checkout	checkoutProduct()

- Persiapan Project Testing:
    - Membuat kerangka selenium: Digunakan untuk mempermudah pembuatan program untuk automation testing terdiri dari 4 yaitu :
      - tests : berisi test case yang di eksekusi
      - page : berisi class page dan methode yang ada pada setiap page agar dapat reusable.
      - utils : berisi konfigurasi untuk program
      - report : berisi file untuk reporting hasil test, biasanya berupa html code.
    - Buat masing masing file pada folder “page” untuk setiap Page / Modul yang telah di buat.
    - Buat methode untuk setiap modul / page
  - Konversi test case menjadi program
    - Buat file untuk masing masing schenario test pada folder “tests”.
    - Buat methode. Methode disini isinya adalah test case yang telah dibuat.
  - Eksekusi test.
7. Kerangka kerja pengujian (testing framework) adalah alat yang digunakan untuk membuat dan menjalankan skenario test otomatis. Kerangka kerja ini dapat membantu dalam mengeksekusi skenario test dengan lebih konsisten dan mempermudah dalam pembuatan script testing karena sudah tersedia library / modul modulnya.
8. Bug tracking system adalah alat atau aplikasi yang digunakan untuk mencatat, dan mengelola bug yang ditemukan sehingga dapat dikomunikasikan kepada anggota tim lainnya agar bug dapat ditangani. Alat yang sering digunakan antara lain Jira, Trello dan Taiga

9. Skenario uji fungsional untuk aplikasi pemesanan tiket pesawat online yang dapat dilakukan antara lain pengujian untuk Pemesanan Tiket, Pembayaran Tiket, Cek Jadwal Penerbangan, Cancel Pemesanan Tiket, dan Reschedule Jadwal.

Contoh test casenya :

Bussiness Rules : Pengguna harus mendaftar atau login untuk melakukan pemesanan, Tiket yang dipesan tidak dapat dibatalkan setelah 24 jam dari waktu penerbangan.

Test Schenario : Cancel Pemesanan Tiket

Test Case	Precondition	Steps	Expected Result
Pembatalan Tiket Berhasil	Pengguna sudah login dan memiliki tiket yang sudah dipesan.	1. Akses menu "Pemesanan". 2. Pilih tiket yang ingin dibatalkan. 3. Klik tombol "Batalkan Tiket". 4. Konfirmasi pembatalan.	Tiket Berhasil dibatalkan dan pengguna menerima notifikasi pada aplikasi
Pembatalan Tiket Gagal	Pengguna sudah login dan tiket keberangkatan dalam 24 jam.	1. Akses menu "Pemesanan". 2. Pilih tiket yang ingin dibatalkan. 3. Klik tombol "Batalkan Tiket". 4. Konfirmasi pembatalan.	Sistem menampilkan pesan bahwa tiket tidak bisa dibatalkan karena sudah mendekati waktu keberangkatan.

10. Pengujian yang dapat dilakukan untuk aplikasi perbankan online meliputi verification dan validation misalnya :

- Pengujian Autentikasi: Memastikan sistem otentikasi aman, mencakup proses login, pendaftaran, pemulihan kata sandi, dan logout.
- Menguji Validasi Data dan Validasi Upload File: Mencegah data berbahaya diterima oleh aplikasi.
- Menguji Keamanan API: Memastikan API tidak mengekspos data sensitif.

Pengujian ini bisa dilakukan melalui penetration testing (brute-force, SQL injection, XSS) dan vulnerability assessment menggunakan tools, untuk menemukan kerentanan sistem.

11. Test Schenario : Pengujian Fitur Update Pembayaran

Precondition : Pengguna telah login aplikasi shopee dan Pesanan telah ada ada di tab belum bayar

Test Case	Step to Execute	Data Uji	Expected Result
Pengguna dapat Mengubah Metode Pembayaran Menjadi COD	<ol style="list-style-type: none"> <li>1. Masuk ke aplikasi shopee</li> <li>2. Pilih menu Pesanan Saya dan pilih tab Belum bayar</li> <li>3. Periksa button “Ubah Pembayaran”</li> <li>4. Klik Button “Ubah Pembayaran”</li> <li>5. Pilih metode pembayaran menjadi “COD”</li> </ol>	{ “order_id” : ORD001 “updated_payment”: COD }	Sistem berhasil mengubah metode pembayaran COD dan Pesanan tidak tampil di Tab Belum Bayar.
Pengguna mengubah Metode Pembayaran menjadi ShopeePay dan melakukan konfirmasi pembayaran	<ol style="list-style-type: none"> <li>1. Masuk ke aplikasi shopee</li> <li>2. Pilih menu Pesanan Saya dan pilih tab Belum bayar</li> <li>3. Periksa button “Ubah Pembayaran”</li> <li>4. Klik Button “Ubah Pembayaran”</li> <li>5. Pilih metode pembayaran “ShopeePay”</li> <li>6. Konfirmasi pembayaran menggunakan shopeepay (Input PIN atau sidik jari)</li> </ol>	{ “order_id” : ORD002 “updated_payment”: ShopeePay }	Sistem berhasil mengubah metode pembayaran ShopeePay dan Pesanan tidak tampil di Tab Belum Bayar.
Pengguna mengubah metode pembayaran menjadi ShopeePay namun tidak melakukan konfirmasi pembayaran	<ol style="list-style-type: none"> <li>1. Masuk ke aplikasi shopee</li> <li>2. Pilih menu Pesanan Saya dan pilih tab Belum bayar</li> <li>3. Periksa button “Ubah Pembayaran”</li> <li>4. Klik Button “Ubah Pembayaran”</li> <li>5. Pilih metode pembayaran</li> <li>6. Klik kembali</li> </ol>	{ “order_id” : ORD003 “updated_payment”: ShopeePay }	Sistem berhasil mengubah metode pembayaran ShopeePay dan Pesanan tetap ditampilkan pada Tab Belum Bayar
Pengguna ingin mengubah metode pembayaran menjadi ShopeePay namun nominal shopeepay kurang	<ol style="list-style-type: none"> <li>1. Masuk ke aplikasi shopee</li> <li>2. Pilih menu Pesanan Saya dan pilih tab Belum bayar</li> <li>3. Periksa button “Ubah Pembayaran”</li> <li>4. Klik Button “Ubah Pembayaran”</li> </ol>	-	Update pembayaran gagal, tampilan pilihan pembayaran shopeepay menjadi disable / tidak dapat dipilih
Pengguna ingin mengubah metode pembayaran menjadi shopeepaylater	<ol style="list-style-type: none"> <li>1. Masuk ke aplikasi shopee</li> <li>2. Pilih menu Pesanan Saya dan pilih tab Belum bayar</li> <li>3. Periksa button “Ubah Pembayaran”</li> <li>4. Klik Button “Ubah Pembayaran”</li> <li>5. Pilih metode pembayaran menjadi “ShopeePayLater”</li> </ol>	{ “order_id” : ORD004 “updated_payment”: ShopeePayLater }	Sistem berhasil mengubah metode pembayaran ShopeePayLater dan Pesanan tidak tampil lagi di Tab Belum Bayar.
Pengguna dapat mengubah metode pembayaran menjadi pembayaran dengan mitra Shopee (Indomaret, Alfamart, dll)	<ol style="list-style-type: none"> <li>1. Masuk ke aplikasi shopee</li> <li>2. Pilih menu Pesanan Saya dan pilih tab Belum bayar</li> <li>3. Periksa button “Ubah Pembayaran”</li> <li>4. Klik Button “Ubah Pembayaran”</li> <li>5. Pilih metode pembayaran</li> </ol>	{ “order_id” : ORD005 “updated_payment”: Indomaret }	Sistem menampilkan kode pembayaran pesanan, Sistem berhasil mengubah metode pembayaran Indomaret dan Pesanan tetap tampil pada tab belum bayar

Pengguna dapat mengubah metode pembayaran menjadi menggunakan Virtual account	<ol style="list-style-type: none"> <li>1. Masuk ke aplikasi shopee</li> <li>2. Pilih menu Pesanan Saya dan pilih tab Belum bayar</li> <li>3. Periksa button “Ubah Pembayaran”</li> <li>4. Klik Button “Ubah Pembayaran”</li> <li>5. Pilih metode pembayaran</li> </ol>	<pre>{   "order_id" : ORD005   "updated_payment": BRI   VA }</pre>	Sistem menampilkan nomor rekening virtual account, Sistem berhasil mengubah metode pembayaran BRI VA dan Pesanan tetap tampil pada tab belum bayar
Pengguna Gagal Mengubah Metode Pembayaran Menjadi COD Ketika Pesanan Tidak Memenuhi Syarat	<ol style="list-style-type: none"> <li>1. Masuk ke aplikasi shopee</li> <li>2. Pilih menu Pesanan Saya dan pilih tab Belum bayar</li> <li>3. Periksa button “Ubah Pembayaran”</li> <li>4. Klik Button “Ubah Pembayaran”</li> </ol>	-	Update pembayaran gagal, tampilan pilihan pembayaran COD menjadi disable / tidak dapat dipilih

**Catatan :**

- Pada Aplikasi Mobile Shopee pembayaran dengan metode COD dan ShopeePayLater walaupun pengguna belum membayar langsung, status pesanan tidak dianggap sebagai pesanan dengan status belum bayar.
- Penyebab tidak memenuhi syarat pembayaran COD antara lain karena alamat pengiriman diluar cakupan, atau produk yang dipesan tidak menyediakan pembayaran COD