#1 Langkah awal berkarir SQA oleh HaloTech Academy

1. Pengenalan SQA
2. Skill SQA
3. Tipe – tipe pengujian
4. Test case
5. Bug report
6. Siklus Hidup perangkat lunak / SDLC (Metode Agile)
7. Pengertian Software Quality Assurance

Quality Assurance (QA):

* Merupakan sebuah kegiatan yang berfokus pada penyediaan jaminan bahwa kualitas yang diminta akan tercapai dengan memastikan keseluruhan proses pembuatan produk sesuai dengan spesifikasi yang sudah ditetapkan.

Contoh: pada bidang manufaktur, khususnya produk berupa barang.

Software Quality Assurance (SQA)

* Merupakan kegiatan yang berfokus untuk memastikan keseluruhan proses pengembangan software sesuai dengan spesifikasi atau perminataan yang sudah ditetapkan.

Contoh: ketika pada saat pembuatan sebuah website

* Software quality assurance berkontribusi pada keseluruhan pengembangan software, software quality assurance sendiri dapat melakukan kontribusi pada tahap pengembangan desain karena dapat memberikan masukan atau inisiasi agar dapat berjalan dengan baik.
* Pada saat diserahkan pada developer, SQA dapat membuat skenario pengujian dari sebuah fitur yang digunakan.
* Ketika software telah selesai di develop, seorang SQA melakukan proses pengujian berdasarkan skenario yang dibuat sebelumnya

Pentingnya Software Quality Assurance

* Mengidentifikasi error sedini mungkin.
* Memastikan software sesuai dengan spesifikasi.

Sehingga meingkatkan kepercayaan pengguna.

#2 QA Manual dan QA Automation.

Macam – macam jenis pekerjaan di Software Quality Assurance

* Quality Assurance engineer
* Software Tester
* Software Development Engineer in Test (SDET)
* Automation Engineer
* Quality Analyst
* Quality Assurance Automation
* Software Engineer in Test (SEIT)
* Automation Software Engineer in Test (ASET)
* Quality Coach
* Test Architect
* Software Tester
* Quality Assurance Manual
* Test engineer

Software Quality Assurance dibagi menjadi 3 secara umum:

* QA Analyst
* QA Manual
* QA Engineer

1. QA Analyst

QA Analyst Bertugas membuat dokumen pengujian yang berfungsi sebagai panduan saat melakukan pengujian. Seperti membuat test case saat pengujian.

A diagram of a software quality assurance

AI-generated content may be incorrect.

1. QA Manual

Job scope QA manual adalah melakukan pengujian secara manual dengan menjalankan software sama seperti pengguna pada umumnya.

Contoh: Pada login, QA manual melakukan pengujian berpedoman pada dokumen pengujian yang telah dibuat.

1. QA engineer

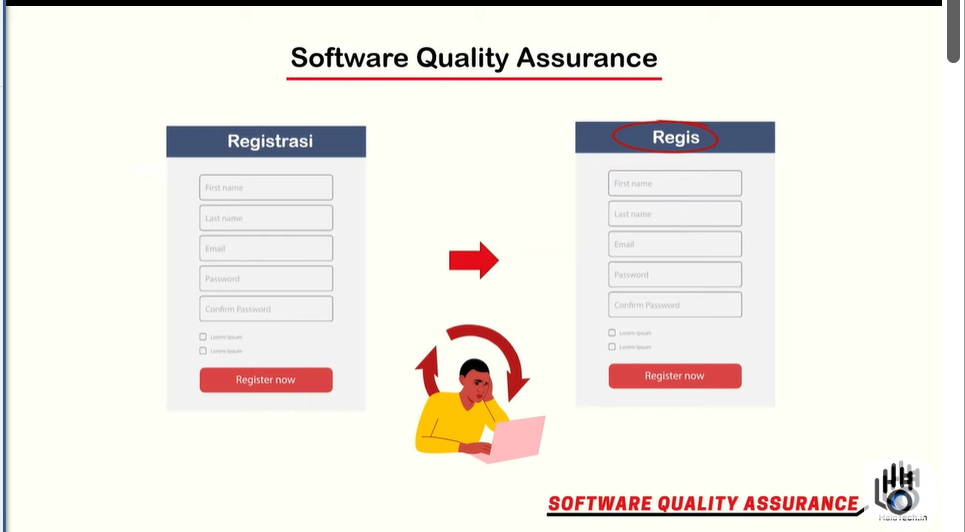
QA engineer bertugas melakukan pengujian secara otomatis dengan membuat kode yang berfungsi agar testing sebuah software berjalan dengan sendirinya / otomatis.

Test case yang telah dibuat dijalankan secara manual dibuatkan programnya agar dapat dilakukan secara otomatis.

Test dilakukan secara berulang jadi jika membuat fitur baru seperti login, fitur register harus dilakukan pengecekan ulang supaya tidak ada isu pada masalah register tersebut.

Pengujian secara manual dapat membosankan karena sering dilakukkan secara berulang dengan menggunakan pengujian otomatis dapat menghemat waktu pengujian dan mencegah human error.

Contoh:



Tujuannya test secara otomatis adalah untuk mengoptimalkan pengujian tapi tidak menggantikan manual testing.

Ada perusahaan yang membagi menjadi 2 dan atau hanya 1 saja seperti

QA analyst dan QA manual dengan QA engineer. Adapun ketiganya digabung.

QA analyst dan QA manual dapat belajar SQA dasar.

QA engineer dapat belajar SQA Automation.

#3 Skill yang Dibutuhkan untuk Berkarir Sebagai SQA

Kegagalan pada kualitas perangkat lunak:

* Kerugian secara finansial
* Rusaknya reputasi
* Membahayakan keselamatan

Skill yang dibutuhkan Software Quality Assurance:

1. Pemahaman produk

* Pemahaman mengenai produk yang dikembangkan seperti spesifikasi produk, tujuan, dan calon pengguna sangat penting untuk dapat memahami kebutuhan pengguna dengan baik.

Contoh: pada bagian pembayaran E-commerce

Tujuan utama: membuat customer dapat melakukan pembayaran secara aman dan efisien. Serta kita juga harus memastikan bahwa dapat melakukan pembayaran, memilih metode pembayaran, dan memastikan bahwa pembayaran telah dilakukan hingga sukses.

Fitur pembayaran ini melibatkan pihak ketiga jadi penting juga untuk menguji integrasi antara aplikasi E-commerce dengan pihak ketiga tersebut.

Jadi seperti apakah pihak ketiga mendapatkan data yang sesuai, serta verifikasi terhadap data yang dikirmkan, tidak hanya itu pihak ketiga juga dapat memberikan informasi mengenai pembayaran seperti gagal atau valid.

Tidak hanya itu pengujian juga dilakukan mengenai privasi pengguna.

1. Teliti, Berorientasi Detail, dan Terstruktur.

* Hal ini diperlukan agar tidak ada rencana pengujian yang terleewatkan dan dapat memastikan hasil pengujian benar dan telah sesuai seperti yang diharapkan.

1. Kreatif dan Inovatif

* QA harus menjadi kreatif dan inovatif agar dapat menganalisis kemungkinan kecacatan pada produk dari berbagai skenario dan dapat terus berinovasi mengembangkan pengetahuan untuk mengimplementasikan tools atau metode terbaik.

1. Komunikasi dan kerja Tim

* Pekerjaan QA yang banyak berhubungan dengan berbagai pihak menjadikan kemampuan berkomunikasi yang baik dan semangat bekerja dalam tim sangat dibutuhkan.

Umumnya QA akan bekerja dengan QA lainnya, developer, Project manager, product owner, UI/UX designer, Business development dan lainnya.

Misalnya:

* 1. QA dengan Developer

Seorang QA akan banyak berdiskusi mengenai suatu fitur.

* 1. QA dengan Product Owner

QA akan berdiskusi mengenai suatu fitur yang akan digunakan.

Diatas semua merupakan skill non-teknis mengenai QA, adapun juga skill teknis yaitu:

1. Pemahaman Tools QA

* Penting untuk seorang QA menguasai tools yang biasa digunakan, seperti tools untuk manajemen projek, manajemen test, ataupun tools untuk melakukan pengujian. Seperti manajemen project contohnya Jira, Manajemen tes contohnya Qase.io, tools untuk melakukan pengujian API yaitu postman.

1. Basis data dan pemograman dasar

* Pengetahuan tentang basis data diperlukan untuk melakukan pengujian database atau API(application programming interface). Sedangkan pengetahuan pemograman diperlukan untuk melakukkan pengujian secara otomatis.

A computer screen shot of a person with a book

AI-generated content may be incorrect.

#4 Membuat Test Case Part 1

Pengertian:

* Test case
* Test scenario
* Test Suit

1. Test case adalah serangkaian tindakan berurutan yang dijalankan untuk memverifikasi fitur atau fungsionalitas tertentu dari sebuah software.

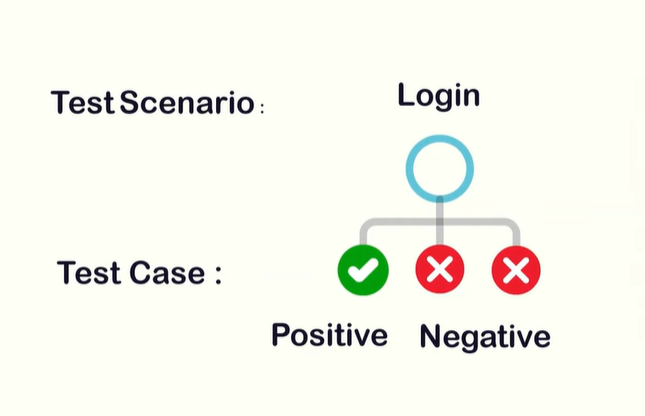
* Langkah – langkah pengujian
* Data yang diperlukan
* Hasil yang diharapkan

1. Test scenario adalah fungsionalitas yang diuji.

Perbedaan test scenario dan test case yaitu:

Test case bagaimana melakukan pengujian sedangkan Test scenario yaitu apa yang harus diuji.

Contoh:



Test scenario seperti fitur yang diuji contohnya diatas adalah login sedangkan testcase merupakan pengujian yang dilakukan di dalam scenario tersebut.

1. Test Suit adalah satu set pengujian atau prosedur pengujian yang akan dijalankan dalam uji coba tertentu.

Contohnya:

A diagram of a test

AI-generated content may be incorrect.

Test suit login berupa isian dari test case yang berada di test scenario login.

Dibagi menjadi 2 yaitu:

* 1. Test suit negative yang berisikan test case test case yang berisi test case negatif yang berada didalam test scenario login.
  2. Test suit positive case yaitu berisikan test case yang berisikan test case positive dari keseluruhan aplikasi.

Format test case

* ID Test Case: Login001
* Judul Test Case: Sukses login dengan valid username dan password.
* Prasyarat: User sudah melakukan registrasi.
* Langkah uji:

1. Akses laman login
2. Input valid username
3. Input valid password
4. Klik tombol login

* Data:

1. Username = usertesting
2. Password = testing123

* Hasil yang diharapkan:

User diarahkan ke halaman home

* Hasil yang didapatkan:Expected
* Status tes – Lulus / Gagal : Lulus

Jika hasil yang diharapkan = Hasil yang didapat maka Lulus

Jika hasil yang diharapkan ≠ dengan yang didapatkan maka gagal.

* ID Test ase: Login002
* Judul Test Case: Gagal login dengan valid username dan invalid password
* Prasyarat : User sudah melakukan registrasi
* Langkah uji:

1. Akses laman login
2. Input valid username
3. Input invalid password
4. Klik tombol login

* Data:

1. Username : testing002
2. Password : 002testing

* Hasil yang diharapkan : Expected
* Status Tes – Lulus / Gagal : Lulus

Status pengujian

Passed : Hasil yang diharapkan = Hasil yang didapat

Failed : Hasil yang diharapkan ≠ Hasil yang didapat

Blocked : Tes case tidak bisa dieksekusi karena terhalang fitur sebelumnya/terkait.

Pending : Penundaan pengujian dikarenakan perubahan / perbaikan fitur ataupun perubahan priortias pengerjaan fitur.

#5 Membuat Test Case Part 2

Membuat test case login pada excel link ada pada github:  
<https://github.com/DreamlandSR/SQA/tree/main/folderku/Pengujian%20Manual>

#6 Membuat Bug Report

Perbedaan Error, Bug, dan Failure

1. Error/Mistake

Kesalahan yang dilakukan oleh manusia

* Terdapat acceptance criteria fitur yang kurang dari tim product
* Wording yang salah dari tim desain
* Client yang memberikan informasi yang tidak memadai

1. Bug/Defect/Fault

Hasil yang diharapkan tidak sama dengan hasil yang didapat.Biasanya ditemukan oleh Quality assurance bug itu mencerminkan ketidakmampuan memenuhi persyaratan atau kriteria yang sudah ditentukan sehingga mencegah software berjalan sesuai yang diharapkan.

1. Failure

Kesalahan perilaku sistem yang dapat teramati saat dilakukan uji coba. Failure itu merupakan perilaku sistem yang dapat teramati saat melakukan uji cob, failure itu juga merupakan konsukuensi dari bug.

Jadi, kesalahan error yang dilakukan oleh pengembang dapat mengakibatkan bug atau perbedaan hasil yang diharapkan dan hasil yang didapatkan lalu bug menyebabkan failure atau fungsi yang tidak dapat digunakan.

Ada 2 jenis bug yaitu:

1. Bug Severity  
   Ukuran dampak bug pada fungsi software
2. Low

Bug tidak mengakibatkan kerusakan pada sistem.

1. Minor

Terdapat perilaku sistem yang tidak sesuai ekspektasi, tapi tidak mengganggu fungsi sistem.

1. Major

Mengganggu sebagian besar sistem (kategori bug yang harus dibenahi segera)

1. Critical

Sistem tidak bisa digunakan.

1. Bug Priority

Urutan pengerjaan bug secara kepentingan bisnis

1. Low

Bug dapat diperbaiki dikemudian hari. Bug lain yang lebih serius diprioritaskan.

1. Medium

Bug dapat diperbaiki dalam proses pengembangan dan pengujian normal.

1. High

Bug harus diselesaikan paling awal karena berdampak buruk pada sistem dan membuatnya tidak dapat digunakan.

Bug severity seberapa parahnya masalah dan bug priority berkaitan dengan seberapa cepatnya masalah itu dapat diatasi berdasarkan nilai bisnis

A group of black text

AI-generated content may be incorrect.

Gambar 1. 1 Contoh bug priority dan severity

Penyebab terjadinya Bug:

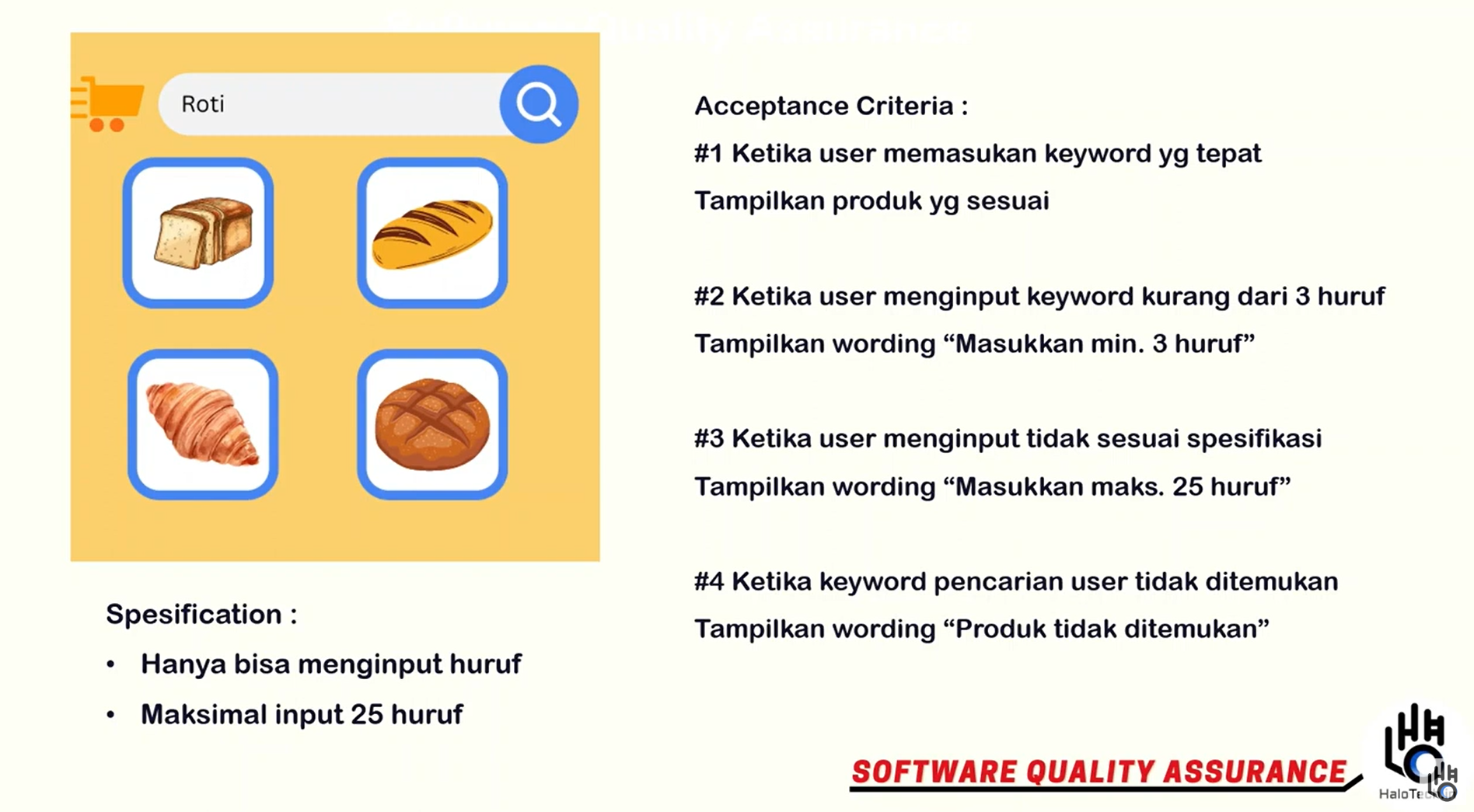
* Kurangnya komunikasi tim
* Deadline yang mepet
* Software yang kompleks misalnya banyak modulnya dan dikerjakan oleh banyak programmer
* Perubahan kode yang tidak didokumentasikan dengan baik
* Proses testing yang bermasalah

Cara menghindari bug

* Bangun komunikasi yang baik
* Gunakan tools kolaborasi yang efektif bisa gunakan Jira atau treelo
* Jangan mengerjakan dengan terburu – buru
* QA melakukan dengan baik dan menyeluruh

Format Bug Report

* Bug title
* Bug ID
* Step to reproduce
* Actual result
* Build number
* Priority
* Severity
* Assignee
* Reporter
* Reported on
* Testing on



Gambar 1. 2 Contoh bug report