

# **EFISHERY QA ENGINEER TEST**

**Noviadi Suryo - Yogyakarta**

**(Q) Bug Cycle (pre-production dan post-production) praktikal yang kamu pakai untuk mempersiapkan test buatmu dan orang lain bagaimana? Lalu apa saja Artifacts yang kamu pakai?**

**(A)** Bug life cycle pada fase pre-production / staging env dan production phase tidak memiliki banyak perbedaan.

Pada umumnya, bug life cycle meliputi:

- NEW : Kondisi dimana bug baru saja dibuat
- ASSIGNED : Bug di assign as a task ke programmer / developer oleh QA
- INPROGRESS : Bug sedang dalam proses fixing oleh developer
- READY FOR TEST : Bug telah di fix oleh developer dan siap untuk retest
- RETEST : Fase dimana bug dilakukan tes ulang oleh QA pasca fixing
- REOPEN : Kondisi dimana bug kembali di kirim ke dev karena issue masih persist
- VERIFIED / CONFIRMED : Fix bug telah terkonfirmasi oleh QA, dan issue sudah tidak terjadi lagi
- CLOSED : Bug ditutup dan masuk ke bug history / legacy

Pada fase pre-production, umumnya QA melakukan testing berdasarkan design document dari product manager dan designer, dimana design document ini akan direview bersama oleh tim setiap kali melakukan product screening di awal sprint, atau pada saat sprint planning.

Di fase ini, adalah peluang bagi QA untuk memahami seluk beluk product design, API contract, dan membuat sebuah TEST PLAN dan TEST STRATEGY berdasarkan dari observasi yang dilakukan pada saat product screening.

QA dapat menentukan dan merencanakan, untuk melakukan testing menggunakan testing technique yang beragam, berdasarkan fitur yang dibuat.

Misalkan, product manager sedang merencanakan melakukan REVAMP pada user login feature, QA dapat merencanakan testing dengan membuat TEST PLAN yang berisi:

- Fitur apa saja yang akan di test yang kemungkinan tersenggol revamp
- Komponen-komponen apa saja dalam fitur tersebut yang harus dites
- Perlu tidaknya melakukan compatibility testing untuk UI testing
- Merencanakan testing technique yang akan digunakan seperti:

- Boundary Value Analysis
- Equivalence Partitioning
- Decision Table
- DII jika diperlukan
- Melakukan koordinasi dengan developer untuk akses swagger dari fitur yang dibuat untuk merencanakan API testing yang akan dilakukan.
- Perlu tidaknya dilakukan AUTOMATION testing di fase ini baik untuk UI atau API
- Planning waktu yang dibutuhkan untuk testing
- Planning resource QA yang dibutuhkan berapa banyak
- Kapan tenggat waktu feature harus selesai dan siap LIVE
- Apakah ada hard deadline atau tidak untuk feature ini
- Apakah ada marketing campaign dari tim sales pasca release ke production
- DII

Semua hal diatas sangat membantu dan menentukan planning serta efektifitas dari kinerja QA kedepannya.

Dalam pembuatan test case, difokuskan untuk membuat test case based on business perspective, yang mana akan sangat membantu pembuatan testcase lebih terarah, mengcover semua critical path dari setiap user flow, dan lebih time efisien.

Artifact yang digunakan pada fase ini:

- Product Screening: Design Document, Design image, API contract, tech documentation
- Bug Creating: Video attachment, image attachment, error log, crash log, QA input

**(Q) Unit Testing vs Integration Testing, ada opini terkait ini? Menurutmu sebaiknya bagaimana dalam praktikal untuk Produk/Fitur baru?**

**(A)** Unit testing merupakan sebuah testing yang dilakukan pada level komponen atau unit dari sebuah software. Unit tersebut bisa berbentuk fungsi, kode, metode, prosedur, modul ataupun objek itu sendiri.

Unit testing umumnya dilakukan oleh developer pada saat selesai membuat sebuah script atau fungsi, karena mereka lebih paham dengan script dan codingan yang mereka buat.

Ada pula unit testing yang dilakukan oleh QA tapi hal ini jarang dan justru akan membuat fase development menjadi lebih lama.

Sementara, integration testing merupakan testing yang dilakukan pada unit atau modul dan berinteraksi dengan modul lain dalam sebuah group.

Misalnya, integration testing pada module FORM dan DATABASE, dimana user form terdiri dari beberapa unit yang berinteraksi menjadi sebuah feature, dan perlu integrasi dengan database dimana nanti user input akan disimpan sebagai data disana.

Umumnya, untuk best practice, unit testing dilakukan oleh developer, dan integration testing dilakukan oleh QA.

Untuk fitur baru, feature HARUS sudah PASSED unit testing oleh developer sebelum feature bisa dites oleh QA team, .

**(Q) Anda diminta untuk eksplorasi halaman <https://music.youtube.com/>**

**Buat Test Cases menggunakan gherkin syntax sesuai dengan template yang menurutmu paling nyaman**

- 1. Test Case diutamakan untuk fitur-fitur utama Youtube Music tersebut. Tentukan sendiri apa saja yang ingin dibuat Test Case-nya. Tidak perlu semua fitur, pilih fitur yang menurutmu krusial dan cover sebagian besar common cases.**
- 2. Tuliskan alasan memilih template Test Case yang digunakan, serta tuliskan juga alasan memilih Test Cases yang dibuat untuk halaman/aplikasi tersebut.**
- 3. Outputnya: Spreadsheet Test Cases, Alasan pemilihan template Test Case dan Test Cases-nya**

**(A) Hasil attached, atau bisa di download disini**

**(Q) Buat sebuah script sederhana (dengan/tanpa framework testing) untuk menguji Test Cases yang sudah dibuat**

- 1. Tidak perlu semua Test Cases, pilih Test Cases yang menurutmu krusial untuk di-cover dengan automation ini. Tuliskan alasannya.**
- 2. Script dapat dibuat dengan NodeJS/Python atau Selenium**
- 3. Output task automation ini disimpan di repo Github beserta README.md menjalankan testnya**