

1. Role job Quality Assurance.
2. Tools, rules dan aktifitas project yg biasa dilakukan.
3. Portofolio.

A. Quality Assurance

QA yaitu Quality Assurance dengan arti jaminan kualitas, jaminan kualitas pada hal ini yang dimaksud jaminan perangkat lunak yang berkualitas sesuai dengan keinginan perusahaan atau pun customer. Banyak istilah pada pekerjaan ini salah satunya seperti Tester, Software Quality Assurance dan sebagainya. Begitu juga level-level pada profesi yang telah ditempuh, seperti saya entry level (pemula), Associate, Mid Level, Senior, dan sebagainya.

B. Role model Quality Assurance

Quality Assurance memiliki prosedur atau kemampuan masing-masing baik personal maupun secara company, berikut hal-hal yang biasa saya lakukan dalam pengerjaan Software Quality Assurance

1. Gambaran

- Sebagai bagian dari proyek maka saya akan menguji beberapa fungsi dari web atau aplikasi.
- Menulis dokumen sebagai perancangan pengujian tingkat tinggi dengan rincian:
 - Lingkup Proyek
 - Strategi Pengujian
 - Jadwal Pengujian, dan
 - *Resource requirements*
 - *test deliverables* dan *schedule*

2. Lingkup

Testing ruang lingkup proyek mencakup pengujian fitur – fitur berikut:

- Register
- Login & Logout
- Forgot Password
- Search
- Product
- Product Display Page
- Add to Cart
- Wish List
- Shopping Cart
- Home Page
- Checkout Page
- My Account Page
- Order History Page
- Downloads Page
- Contact Us Page
- Menu Options
- Footer Options
- Category Pages

3. *Test Environment*

- *Windows 10 – Chrome, Firefox and Edge*

4. *Pengecualian*

- *Test Automation*

5. *Test Strategy*

Sebagai bagian dari pengujian fungsional , saya akan mengikuti pendekatan pengujian berikut ini

Langkah pertama:

Pembuatan scenario uji dan Kasus uji untuk berbagai fitur pada

Scope

- Saya akan menerapkan beberapa tehnik desain tes saat membuat kasus uji
- Saya juga menerapkan keahlian dalam membuat kasus uji dengan menerapkan
- Saya memprioritaskan kasus uji

Langkah kedua:

Proses pengujian saya,

- Pertama, kami akan melakukan ***smoke testing*** untuk memeriksa apakah perbedaan dan fungsi penting dari aplikasi bekerja.
- Saya akan menolak *build*, jika smoke testing gagal dan akan menunggu *stable* sebelum melakukan pengujian mendalam dari fungsionalitas aplikasi.
- Setelah menerima *build* yang stabil, yang lulus smoke testing akan dilakukan pengujian secara mendalam menggunakan ***test case*** yang dibuat.
- Beberapa *test resources* akan menguji aplikasi yang sama dibeberapa lingkungan yang didukung secara bersamaan.
- Saya kemudian akan melaporkan ***bug*** di alat pelacak *bug* dan mengirimkannya ke *dev.management* jika cacat ditemukan pada saat itu
- Sebagai bagian dari pengujian maka saya akan melakukan jenis pengujian
 - *Smoke Testing and Sanity Testing*
 - *Regression Testing and Retesting*
 - *Usability Testing, Functionality & UI Testing*
- Saya akan mengulangi siklus uji coba sampai mendapatkan produk yang berkualitas

Langkah ketiga:

Saya akan mengikuti praktik untuk menjadikan Pengujian yang baik:

- *Context Driving Testing*, Saya akan melakukan pengujian sesuai dengan konteksnya dari aplikasi yang diberikan
- *Shift Left Testing*, Saya akan memulai pengujian tersebut dari tahap awal pengembangan itu sendiri alih – alih menunggu *stable build*.
- *Exploratory Testing*, Saya akan melakukan eksploratory pengujian , selain dari mengeksekusi kasus uji normal
- *End to end flow testing*, Saya akan menguji scenario end to end yang melibatkan beberapa fungsi untuk mensimulasikan *end users flow*.

6. **Prosedur Pelaporan Cacat**

Selama eksekusi tes

- Setiap penyimpangan dari perilaku yang diharapkan oleh aplikasi akan dicatat, jika hal tersebut tidak dapat dilaporkan sebagai cacat, maka akan dilaporkan sebagai pengamatan/masalah atau diajukan sebagai pertanyaan
- Setiap masalah kegunaan juga akan dilaporkan
- Setelah ditemukan cacat, selanjutnya akan diuji ulang untuk memverifikasi reproduktifitas dari cacat. Tangkapan layar dengan langkah-langkah mereproduksi akan didokumentasikan.
- Diakhir pelaksanaan tes, cacat yang ditemui akan dikirim bersama dengan observasi.

** Catatan: - Cacat akan didokumentasikan di excel

: - Skenario uji dan Kasus uji akan didokumentasikan dalam dokumen excel

Responsibilities

- Escalations
- Create the Test Plan and get the client signoffs
- Interact with the application, create and execute the test cases
- Report defects
- Coordinate the test execution. Verify validity of the defects being reported.
- Interact with the application
- Create and Execute the Test cases.
- Report defects

7. Jadwal Tes

Jadwal tes yang direncanakan adalah sebagai berikut

| Task | Time Duration |
|-----------------------------------|-------------------------------|
| <i>Creating Test Plan</i> | <i>Start Date to End Date</i> |
| <i>Test case creation</i> | <i>Start Date to End Date</i> |
| <i>Test Case Execution</i> | <i>Start Date to End Date</i> |
| <i>Summary Reports Submission</i> | <i>Date</i> |

8. Hasil Uji

Berikut ini harus disampaikan kepada klien:

| Pengujian | Deskripsi | Tanggal penyelesaian target |
|-----------|--|-----------------------------|
| Test Plan | Details on the scope of the project, test strategy, test schedule, resource requirements, test deliverables and schedule | Date |

| | | |
|-----------------------|--|------|
| Functional test cases | Test cases created for the scope defined | Date |
| Defect reports | Detailed description of the defects indentified along with screenshots and steps to reproduce on a daily basis | NA |
| Summary reports | Summary reports— Bugs by bug#, Bugs by functional area and Bugs by priority | Date |

9. Pricing

NA

10. Entry and Exit Criteria

Berikut ini adalah kriteria masuk dan keluar untuk setiap fase software testing life cycle:

Requirement Analysis

Kriteria masukan : Setelah tim penguji menerima dokumen persyaratan atau detail tentang proyek

Kriteria keluaran : Daftar persyaratan dieksplorasi dan dipahami oleh tim penguji, menghilangkan keraguan

Test Planning

Kriteria masukan : Persyaratan yang dapat diuji berasal dari dokumen persyaratan yang diberikan atau detail proyek,

Kriteria keluaran : Dokumentasi rencana uji (termasuk strategi uji) ditandatangani oleh klien

Test Designing

Kriteria masukan : Dokumentasi rencana uji ditandatangani oleh klien

Kriteria keluaran : Dokumentasi test scenario dan test cases ditandatangani oleh klien

Test Execution

Kriteria masukan : Dokumentasi test scenario dan test cases ditandatangani oleh klien, aplikasi siap diuji

Kriteria keluaran : Laporan kasus uji, laporan cacat telah siap

Test Closure

Kriteria masukan : Test case reports, Laporan cacat telah selesai

Kriteria keluaran : Laporan ringkasan tes

11. Tools

Berikut adalah daftar alat yang saya gunakan dalam mengerjakan proyek ini:

- Visual Studio Code (learning automated testing)
- Cypress (learning automated testing)
- Snipping screenshot tool
- Word and excel documents

12.Risiko dan Mitigasi

Berikut adalah daftar risiko yang mungkin terjadi dan cara untuk menguranginya
Risk : Tidak tersedianya sumber daya

Mitigasi: Perencanaan sumber daya cadangan

Risk : Build URL is not working

Mitigasi: Sumber daya tidak akan bekerja pada task lainnya

Risk : Lebih sedikit waktu untuk pengujian

Mitigasi: Meningkatkan sumber daya berdasarkan kebutuhan klien secara dinamis

13.Persetujuan

Saya akan mengirimkan berbagai jenis dokumen untuk persetujuan seperti,

- Test Plan
- Test Scenarios
- Test Cases
- Reports

C. Portofolio

Disini saya memiliki beberapa project yang saya letakkan pada repository github..
Pembelajaran tested project saya berupa manual tested maupun automation tested dan memiliki status selesai dan belum. Berikut portofolio project tested saya:

<https://github.com/prasetiohk?tab=repositories>