

Cara Membuat Skenario Testing Untuk Melakukan Pengujian Aplikasi

By autodika - Dec 2, 2019 - 1 Comment

Autodika.com - Pengujian Aplikasi merupakan salah satu hal yang paling penting dalam proses pembuatan aplikasi atau dalam bahasa inggrisnya "Application Development", aplikasi memiliki pengertian yang sangat luas dalam implementasinya. Aplikasi juga dibagi dalam banyak jenis seperti aplikasi web, aplikasi mobile dan aplikasi desktop.

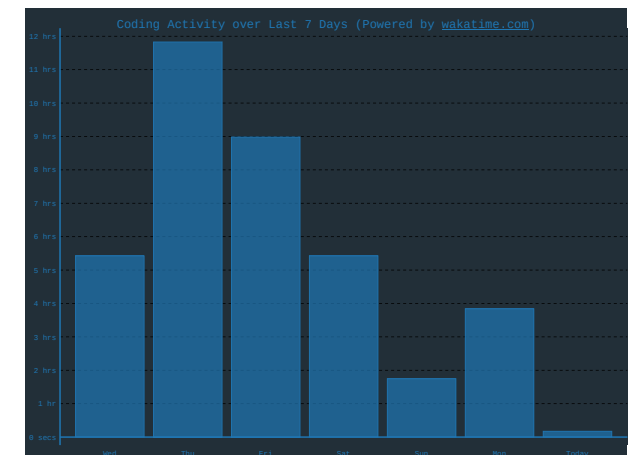


Pengujian Aplikasi digunakan untuk memastikan bahwa aplikasi yang dibuat telah sesuai dengan kebutuhan dan telah lolos dari bug atau error. Sebelum melakukan pengujian aplikasi

COVID-19



CODING ACTIVITIES



kita perlu membuat "Testcases" atau skenario testing yang digunakan sebagai acuan dalam pengujian aplikasi.

Testcase adalah sekumpulan skenario yang disusun oleh tester agar sistem yang akan dites dapat memenuhi ketentuan, standar tertentu serta dapat berfungsi dengan baik. Penyusunan test case dapat membantu tester untuk menemukan masalah pada dokumen ketentuan dan pada desain dari aplikasi itu sendiri.

Baca Juga

- [Documentation Automation Api Testing Using Cucumber Ruby Httparty](#)
- [Cara Membuat Pengujian Web Secara Otomatis - Tahap 2](#)
- [Cara Membuat Pengujian Web Secara Otomatis - Tahap 1](#)

Test case creation dibutuhkan untuk memudahkan seorang QA untuk melakukan testing menurut perspektif end user terhadap aplikasi yang sedang dikembangkan agar sesuai dengan kebutuhan. Test creation tentunya membuat QA melakukan investasi waktu dalam pengujian aplikasi agar lebih efektif. Berikut adalah yang perlu disiapkan oleh QA untuk membuat test case yang baik:

Requirement's description

Tester harus paham bagaimana cara aplikasi berjalan, paham dengan business goal dari pembuatan aplikasi tersebut sehingga pembuatan test case tidak berseberangan dengan requirement yang telah ditetapkan sebelumnya.

BLOG ARCHIVE

Blog Archive



BERLANGGANAN

▼  Posts

▼  Comments

System explanation

Tester dapat menjelaskan ke developer mengenai aplikasi terkait, karena pada umumnya pekerjaan developer dibagi menjadi beberapa bagian dan bagian lainnya dikerjakan oleh orang lain.

Crystal clear and compact test cases

Dikarenakan pembaca test case bukan hanya tester, penggunaan bahasa dan langkah-langkah pengetesan harus jelas. Langkah pengetesan diusahakan tidak rumit dan kompleks.

Inputs and outputs or actions (positive-negative scenarios) and expected results

Tester harus memahami hasil yang tampil setelah mengeksekusi langkah-langkah yang dituliskan dalam langkah pengetesan.

Proofs

Untuk kemudahan tracking dan dokumentasi, ada baiknya tester merekam atau menyertakan bukti output dari test case yang gagal dieksekusi. Ini juga memudahkan developer dalam memperbaiki sebab kegagalan dari sistem yang sedang diuji.

Updates

Setiap tester menemukan skenario baru, ada baiknya untuk terus memperbarui test case yang sudah ada. Sehingga kemungkinan test case dapat mencakup seluruh aspek dari aplikasi, semakin besar.

Naming

Konsisten dalam penamaan pembuatan test case berguna untuk kemudahan dalam tracking. Contoh : setelah menggunakan istilah 'Login' kemudian muncul istilah 'Sign In'.

Reviewing Test Case

Berikut point-point yang perlu diperhatikan untuk membuat test case yang baik:

Simple and transparent

Semua langkah pelaksanaanya pun juga harus jelas dan tidak ada yang terlewat. Contoh : untuk melakukan order, harus 'Login' terlebih dahulu. Apabila langkah sudah terlalu banyak, login bisa dicantumkan di pre-condition.

End-user mind

Tester bermindset bahwa ia adalah end-user. Untuk itu, kemudahan dalam penggunaan aplikasi juga perlu diperhatikan karena kepuasan konsumen adalah saat aplikasi tidak memiliki kegagalan sistem yang bisa menghambat berjalannya sistem tersebut.

Don't assume

Tester diharapkan tidak berasumsi bahwa end-user mengerti keseluruhan dari aplikasi. Jadi usahakan susun test case walaupun hal yang termudah.

100% Coverage

Kemungkinan test case bisa mengcover 100% dari semua aspek dari aplikasi tidaklah mudah. Tetapi apabila tester terus berfokus pada kualitas testing, bukan tidak mungkin 100% coverage bisa tercapai.

Easy to understand

Pemilihan bahasa sewaktu menyusun test case juga harus diperhatikan karena pengguna test case bukan hanya tester saja. Bahasa yang sederhana dan mudah dipahami lebih disukai karena bisa dimengerti bahkan oleh end-user yang mungkin awam.

Repeatable

Artinya test case harus mempunyai kondisi hasil yang sama di seluruh platform yang sedang dites. Contoh : apabila di OS Android setelah login, maka akan tampil halaman my profile maka di iOS pun akan demikian.

Peer review

Meminta tolong pada tester lain atau orang lain untuk me-review test case yang kita susun. Apabila pemilih bahasa yang dipilih kurang baik dan menimbulkan banyak pertanyaan, tester perlu melakukan perubahan agar test case lebih mudah untuk dipahami.

Skenario Testing dibagi menjadi dua skenario, skenario positive dan skenario negative, Berikut contoh Testcase simple untuk Login.

| Case | Type | Description | Step | Expected Result | Actual Result | Status |
|--|----------|-------------|------------------------------------|--|---------------|--------------------------|
| Sukses Login By Email | Positive | | 1 Klik "Log in" on Homepage Header | Direct to "Homepage" Condition "Login" | | <input type="checkbox"/> |
| | | | 2 On page "Login Form" | | | |
| | | | 3 Input valid "Email" | | | |
| | | | 4 Input valid "password" | | | |
| | | | 5 Klik "Button Login" | | | |
| Failed Login By Email (Registered Email) | Negative | | 1 Klik "Log in" on Homepage Header | Alert "Please Try Again Password Is Incorrect" | | <input type="checkbox"/> |
| | | | 2 On page "Login Form" | | | |
| | | | 3 Input valid "Email" | | | |
| | | | 4 Input invalid "Password" | | | |
| | | | 5 Klik "Button Login" | | | |
| Failed Login By Email | Negative | | 1 Klik "Log in" on Homepage Header | Alert "Please Try Again, Email Is Incorrect" | | <input type="checkbox"/> |
| | | | 2 On page "Login Form" | | | |
| | | | 3 Input invalid "Email" | | | |
| | | | 4 Input Password | | | |
| | | | 5 Klik "Button Login" | | | |
| Failed Login By Email (Email Not Registered Email) | Negative | | 1 Klik "Log in" on Homepage Header | Alert "Invalid, User Has Not Been Registered" | | <input type="checkbox"/> |
| | | | 2 On page "Login Form" | | | |
| | | | 3 Input "Not Registered Email" | | | |
| | | | 4 Input Password | | | |
| | | | 5 Klik "Button Login" | | | |

Autodika.com - Segitu dulu yang bisa saya bagikan, jika ada yang ditanyakan silahkan komen dibawah. Semoga Bermanfaat.



autodika ✓

Autodika.com menyediakan jawaban serta materi seputar tugas kuliah Manajemen Informatika, Sistem Informasi, Teknik Informatika dan berbagai macam tugas kuliah yang berkaitan tentang teknologi.

SHARE THIS POST



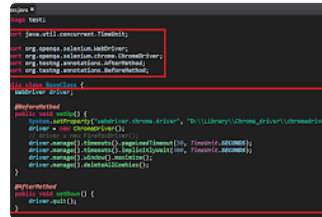
RELATED POSTS



Documentation
Automation Api Testing
Using Cucumber Ruby
Httparty



Cara Membuat Skenarion
Testing Untuk
Melakukan Pengujian
Aplikasi



Cara Membuat Pengujian
Web Secara Otomatis -
Tahap 2



Cara Membuat Pengujian
Web Secara Otomatis -
Tahap 1

1 RESPONSE TO "CARA MEMBUAT SKENARION TESTING UNTUK MELAKUKAN PENGUJIAN APLIKASI"

Anonymous December 20, 2021 at 12:23 PM

Thank You Infonya bang...

Reply

Reply

Add comment

Load more...

To leave a comment, click the button below to sign in with Google.

SIGN IN WITH GOOGLE



Newer Post

Older Post

Autodika.com menyediakan jawaban serta materi seputar tugas kuliah Manajemen Informatika, Sistem Informasi, Teknik Informatika dan berbagai macam tugas kuliah yang berkaitan tentang teknologi

Copyright 2022 Autodika