

FUNCTIONAL TEST PADA APLIKASI BERBASIS WEB MENGGUNAKAN SELENIUM

Hadi Zeftin¹, Dana Sulisty Kusumo², Yanuar Firdaus A.w.³

¹Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Software testing harus dapat dengan efektif menemukan cacat yang ada, dan banyak metode yang dapat digunakan untuk merancang test-case. Tidak hanya efektif, software testing juga harus dengan efisien melakukan test secepat dan semurah mungkin, untuk itu terdapat tool yang dapat meng-automasi testing.

Software testing yang ter-automasi dapat dengan signifikan mengurangi usaha yang dibutuhkan dalam testing yang menyeluruh, atau dengan signifikan meningkatkan jumlah testing yang dapat dilakukan dalam waktu yang terbatas. Terdapat beberapa tool untuk meng-automasi test khususnya aplikasi berbasis web yang dapat dijalankan dalam browser, salah satunya adalah Selenium.

Buku ini menghadirkan analisis untuk dua metode yang dapat diaplikasikan pada level functional test menggunakan tool Selenium, Cause-Effect graphing dan State-Transition analysis, mulai dari permasalahan implementasi dan ongkos uji.

Dari percobaan, walaupun masing-masing metode memiliki keunggulan untuk fungsionalitas tertentu, namun kedua metode tersebut dapat memodelkan spesifikasi yang sama tanpa kehilangan cakupan uji, sehingga tiap metode dapat mendeteksi semua cacat yang ada.

Kata Kunci : functional test, Selenium, Cause-Effect, State-Transition.

Abstract

Software testing needs to be effective at finding any defects which are there, and there are many method that can be use to design test case. Not only effective, it should also be efficient, performing the tests as quickly and cheaply as possible, therefore, exist testing tools to automate testing.

Automating software testing can significantly reduce the effort required for adequate testing, or significantly increase the testing which can be done in limited time. Tests can be run in minutes that would take hours to run manually. There are tool to automate testing spesifically web based application, that can be run in a browser.

This book present analysis for two method aplicable for functional test level with Selenium tool, Cause-Effect graphing and State-Transition analysis, from implementation issue, and cost.

From the experiment, although each method has its own advantage over certain functionality, yet those methods able to model the same spesification without coverage loss, thus each can detect every defect which are there.

Keywords : functional test, Selenium, Cause-Effect, State-Transition

1. Pendahuluan

1.1 Latar belakang

Dalam mengembangkan suatu aplikasi, *acceptance test* menjadi parameter final dalam menentukan apakah aplikasi tersebut layak dan sesuai dengan *requirement stakeholder*. Dalam prakteknya *acceptance test* dilakukan oleh pihak pengguna, berdasarkan skenario yang telah disepakati dalam *requirement specification*, umumnya dikenal sebagai proses *User Acceptance Testing* (UAT).

Seperti telah disebut diatas, subjek dari *acceptance test* adalah skenario penggunaan aplikasi, sebelum menuju *acceptance test* skenario ini juga dapat dites dipihak developer melalui berbagai metode *testing*. Beberapa metode *testing* skenario yang paling populer adalah *system* dan *functional test*, namun terdapat perbedaan tipis dari dua metode tersebut.

System test berusaha untuk memverifikasi sebuah sistem dari sudut pandang end-user, dengan cara menyerupai *end-user*, umumnya konsentrasi terbesar pada *system test* adalah bagaimana integrasi dari komponen-komponen sistem bekerja sesuai dengan proses bisnis yang ada.

Berbeda dengan *system test*, *functional test* tidak menyerupai end-user, tapi pada dasarnya *functional test* adalah *end-user* itu sendiri. Sebuah *functional test* mengemudikan/menjalankan aplikasi melalui antarmuka pengguna seperti seorang pengguna yang akan menggunakan aplikasi tersebut. Hal ini meletakkan posisi *functional test* diatas *system test*, *functional test* adalah bentuk *acceptance test* yang dilakukan oleh pihak developer.

Ada beberapa *framework testing* yang dapat digunakan pada aplikasi web, namun banyak developer web berbasis PHP yang telah membuktikan kehandalan dari Selenium. Selenium adalah sebuah *test-tool* yang digunakan untuk melakukan *functional test* pada aplikasi web, aplikasi web yang dapat dites tidak harus dibangun menggunakan bahasa pemrograman PHP, melainkan dapat juga dibangun dengan bahasa lain seperti Ruby,C# dan lain-lain. Selain itu Selenium merupakan sebuah *test-tool* yang lintas browser dan platform, dan dapat dijalankan melalui browser-browser populer, seperti IE pada sistem operasi Windows, Firefox pada Windows dan Linux, serta Safari pada mac.

1.2 Perumusan Masalah

Permasalahan yang dirumuskan adalah sebagai berikut:

- a. Implementasi *functional test* pada aplikasi berbasis web menggunakan Selenium.
- b. Perancangan *test-case* yang dapat diimplementasikan pada *functional test* menggunakan Selenium.

Untuk menghindari terlalu meluasnya pembahasan, penelitian ini memiliki batasan sebagai berikut:

1. Aplikasi yang dibangun, merupakan aplikasi lomba geMasTIK.

2. Aplikasi dibangun adalah web lomba geMasTIK dengan menggunakan PHP.
3. *Test-case* akan dibangun menggunakan beberapa fungsionalitas (tidak semua) yang memiliki kompleksitas rendah.
4. Metode *testing* yang digunakan dibatasi untuk dua metode yaitu *Cause-Effect* dan *State-Transition*.

1.3 Tujuan

Berikut adalah tujuan yang ingin dicapai oleh penelitian ini:

1. Mendesain *test-case* menggunakan metode *cause-effect* dan *state-transition*.
2. Mengimplementasikan *test-case* yang dihasilkan menggunakan Selenium.
3. Menganalisis penggunaan metode desain *test-case*.

1.4 Metodologi Penyelesaian Masalah

- a) Studi literatur, pada tahap ini dilakukan pencarian dan pengumpulan literatur-literatur tentang metode design, dan implementasi *functional test* dari buku-buku referensi dan situs-situs menunjang.
- b) Pembangunan studi kasus untuk dijadikan subjek *functional test*.
- c) Pada tahap ini meliputi analisis dan desain aplikasi serta implementasi menggunakan PHP.
- d) Mendeskripsikan beberapa spesifikasi fungsionalitas dari subjek yang akan diuji.
- e) Memodelkan tiap spesifikasi yang ingin diuji menggunakan metode *Cause-Effect* dan *State-Transition*.
- f) Menganalisa model, untuk kemudian diturunkan menjadi *test-case*.
- g) Mengimplementasikan *test-case* menggunakan selenium.
- h) Terakhir, akan dilakukan proses review dan analisis terhadap tiap metode *testing*, beserta hasilnya.

Telkom
University

5. Kesimpulan dan Saran

5.1 Kesimpulan

Beberapa kesimpulan yang dapat diambil dari penelitian ini antara lain:

1. *Cause-Effect* modelling berkonsentrasi pada kombinasi dari *cause-cause* yang terjadi dalam memunculkan *effect*, namun sulit untuk menggambarkan urutan kejadian. Secara *nature Cause-Effect graph* tidak memiliki notasi yang dapat menggambarkan urutan terjadinya *cause* atau *effect*. Sehingga *Cause-Effect* lebih mudah diimplementasikan fungsionalitas atau spesifikasi yang bersifat *single-step*.
2. *State-Transition modelling* menggambarkan status-status dari sistem dan bagaimana sistem bertransisi antar status tersebut, permasalahan yang sering muncul adalah pada saat menentukan *guard* pada tiap transisi, *tester* harus mempertimbangkan semua kemungkinan kombinasi dari kondisi yang mungkin terjadi, ketidak-menyeluruhan pertimbangan kombinasi kondisi dapat berarti *test-case suite* tidak lengkap atau komprehensif, atau paling buruk adalah kesalahan analisis.
3. Jika dibandingkan dua metode tersebut untuk fungsionalitas yang memiliki kompleksitas rendah, *State-Transition* menghasilkan *test-case* yang relatif lebih sedikit (jika tidak sama) dibandingkan menggunakan metode *Cause-Effect*. Dikarenakan *state-transition* tidak secara eksplisit menjelajahi kombinasi dari element-element yang ada untuk fungsionalitas yang diuji.
4. Kemampuan *test-case* dari 2 metode tersebut dalam mendeteksi *defect* juga pada dasarnya tidak berbeda, Karena spesifikasi fungsionalitas dapat ditranslasikan kedalam 2 model yang berbeda tanpa kehilangan informasi yang berarti, hal tersebut sangat tergantung pada kejelasan dan kelengkapan spesifikasi.

5.2 Saran

Untuk dapat mengembangkan penelitian ini, adapun saran-saran yang bisa diberikan adalah sebagai berikut:

1. Untuk dapat lebih meningkatkan akurasi pengukuran, metode-metode *testing* sebaiknya diimplementasikan untuk spesifikasi yang lebih kompleks.
2. Perancang *test-case* atau *tester* sebaiknya bukan orang yang membangun aplikasi, sehingga hasil penelitian penggunaan metode-metode *testing* diatas semakin objektif.

Daftar Pustaka

- [1] Burnstein, Ilene. 2003 *Practical software testing : a process-oriented approach* / Ilene Burnstein. p. cm. Springer-Verlag New York, Inc.
- [2] Dustin, Elfriede. 2003 *Effective software testing : 50 specific ways to improve your testing* / Elfriede Dustin. Pearson Education, Inc.
- [3] Gao, Jerry. 2003, *Testing and quality assurance for component-based software* / Jerry Zeyu Gao, H. -S. Jacob Tsao, Ye Wu. Artech House computing library
- [4] Godbole, Nina S. 2004, *Software Quality Assurance: Principles and practice*, Alpha Science International.
- [5] Hutcheson, M. L., 2003, *Software Testing fundamental-methods and metrics*, Indianapolis, Wiley Publishing, Inc.
- [6] Lewis, William E. *Software testing and continuous quality improvement* / William E. Lewis ; Gunasekaran Veerapillai, technical contributor.--2nd ed.
- [7] Li Kanglin, Wu Menqi, 2004, *Effective Software Test Automation: Developing an Automated Software Testing Tool* Sybex.
- [8] McGregor, John D, 2001 *A practical guide to testing object-oriented software* / John D. McGregor, David A. Sykes.
- [9] Myers, Glenford J. 2004 *The art of software testing—2nd ed* / Glenford J. Myers, Wiley Publishing, Inc.
- [10] Patton, R. 2005, *Software Testing*, Sams Publishing.
- [11] *Selenium Official Site*,
<http://Selenium.openqa.org/>.
Didownload pada tanggal 10 maret 2008.
- [12] *The Fuzz on Developer Testing* ,
<http://thedisoblog.com/2006/02/07/developer-testing-defined/>.
Didownload pada tanggal 10 maret 2008.
- [13] *Functional Testing Research* ,
http://www.openplans.org/projects/openplans_testing/project-home.
Didownload pada tanggal 12 maret 2008
- [14] *Functional tests à la Selenium* ,
<http://www.testearly.com/2006/08/07/functional-tests-la-Selenium/>.
Didownload pada tanggal 11 maret 2008