



PENGUKURAN THROUGHPUT LOAD TESTING MENGGUNAKAN TEST CASE SAMPLING GORILLA TESTING

Desy Intan Permatasari¹⁾, Budi Santoso²⁾, Nadia Ningtias³⁾, M. Halim Y.R⁴⁾, Rafidah Atika⁵⁾, Nadia Widad⁶⁾, Ikbar Maulana⁷⁾, Aditya Abdurrahman R⁸⁾

^{1,2,3,4,5,6,7,8)}Departemen Teknik Informatika dan Komputer, Politeknik Elektronika Negeri Surabaya

¹⁾desy@pens.ac.id, ²⁾budisantosodeveloper@gmail.com, ³⁾nadia.ningtias20@gmail.com,

⁴⁾mhalimyr@gmail.com, ⁵⁾rafidahatika97@gmail.com, ⁶⁾widadnadia@gmail.com,

⁷⁾ikbarmaulana77@gmail.com, ⁸⁾rafifabdurrahman12@gmail.com

Abstrak

Load Testing adalah teknik *performance testing* dengan mengukur respon sistem dalam berbagai *load condition*. Penelitian ini membantu menentukan bagaimana software berperilaku ketika beberapa user mengakses software secara bersamaan. Pengujian menggunakan metode *Black Box Testing* dan *Gorilla Testing* sebagai *Test Case Sampling*. *Black Box Testing* adalah metode pengujian fungsional software tanpa mengetahui struktur internal program (*no knowledge*). *Gorilla Testing* adalah metode pengujian pada modul program yang dilakukan berulang kali untuk memastikan bahwa modul tersebut berfungsi dengan benar dan tidak ada bug. Modul dapat diuji lebih dari seratus kali dan dengan cara yang sama. *Gorilla Testing* juga dikenal sebagai "Frustrating Testing". Tujuan pengujian *Gorilla testing* adalah untuk menguji robustness. Pada penelitian ini target pengujian aplikasi adalah *throughput* aplikasi lebih dari 100.000/menit dan data dapat masuk kedalam fungsi. Hasil pengujian *load testing* menunjukkan bahwa dengan menggunakan Apache JMeter, target pengujian berupa *loading time* dan *process memory* sudah terpenuhi.

Kata kunci : furniture, sistem informasi, UML (*Unified Modelling Language*)

Abstract

Load Testing is a performance testing technique by measuring the response system in various load conditions. This research helps determine how the software behaves compilation of several users accessing the software simultaneously. Testing uses Black Box Testing and Gorilla Testing methods as Test Case Sampling. Black Box Testing is a method of testing software without learning the internal program structure (no knowledge). Gorilla Testing is a method of testing the module program which is done many times to ensure the module is made correctly and there are no bugs. Modules can quote more than one hundred times and in the same way. Gorilla Testing is also known as "Frustrating Testing". The purpose of Gorilla testing is testing for endurance. In this study the application testing target is an application throughput of more than 100,000 / minute and data can enter functions. The test results include testing using Apache JMeter, the test target consists of loading time and memory processes that have been met.

Keywords: *Software Testing, Load Testing, Performance Testing, Apache Jmeter*

I. PENDAHULUAN

Aplikasi Print-Kita adalah aplikasi pencetakan secara *online* yang memiliki fungsi mencetak secara *online* tanpa perlu download file. Aplikasi ini memudahkan pengguna untuk mencetak file secara

online tanpa bergantung dengan server printer tertentu.

Pada paper ini, nilai throughput akan diukur berdasarkan load testing dengan menggunakan test case sampling gorilla testing. Semakin tinggi nilai throughput, maka semakin baik kinerja server.



Pengujian adalah suatu proses pelaksanaan suatu program dengan tujuan menemukan suatu kesalahan. Suatu kasus test yang baik adalah apabila test tersebut mempunyai kemungkinan menemukan sebuah kesalahan yang tidak terungkap. Suatu test yang sukses adalah bila test tersebut membongkar suatu kesalahan yang awalnya tidak ditemukan. Tujuan utama dari pengujian adalah untuk mendesain test yang secara sistematis membongkar jenis kesalahan dengan usaha dan waktu minimum.

Load Testing adalah bagian dari *performance testing* yang mana respon sistem diukur dalam berbagai load condition.

Paper ini membantu menentukan bagaimana software berperilaku ketika beberapa user mengakses software secara bersamaan. Pengujian menggunakan metode *Black Box Testing* dan *Gorilla Testing* sebagai *test case sampling*.

Black Box Testing adalah pengujian fungsional software tanpa mengetahui struktur internal program (*no knowledge*).

Gorilla Testing adalah metode pengujian pada modul program yang dilakukan berulang kali untuk memastikan bahwa modul tersebut berfungsi dengan benar dan tidak ada bug. Modul dapat diuji lebih dari seratus kali, dan dengan cara yang sama. *Gorilla Testing* juga dikenal sebagai "*Frustrating Testing*". Tujuannya adalah untuk menguji robustness

II. KAJIAN LITERATUR

Pengujian Software

Pengujian *software* sangat diperlukan untuk memastikan *software/aplikasi* yang sudah/sedang dibuat dapat berjalan sesuai dengan fungsionalitas yang diharapkan. Pengembang atau penguji *software* harus menyiapkan sesi khusus untuk menguji program yang sudah dibuat agar kesalahan ataupun kekurangan dapat dideteksi sejak awal dan dikoreksi secepatnya. Pengujian merupakan elemen kritis dari jaminan kualitas perangkat lunak dan merupakan bagian yang tidak terpisah dari siklus hidup

pengembangan *software* seperti halnya analisis, desain, dan pengkodean. (Shi, 2010)

Pengujian software harus dilakukan dalam proses rekayasa perangkat lunak atau *software engineering*. Sejumlah strategi pengujian *software* telah diusulkan dalam literatur. Semuanya menyediakan template untuk pengujian bagi pembuat *software*. Dalam hal ini, semuanya harus memiliki karakteristik umum berupa (Bhat and Quadri, 2015).

Performance Testing

Performance testing adalah jenis pengujian untuk memastikan perangkat lunak akan bekerja dengan baik di bawah beban kerja yang diharapkan. Tujuan utamanya bukan untuk mencari bug, tapi untuk mengeliminasi *performance bottleneck*.

Fokus dari Performance Testing, yaitu :

- *Speed* - menentukan apakah aplikasi merespon dengan cepat
- *Scalability* - menentukan apakah jumlah maksimum user load dapat ditangani
- *Stability* - menentukan apakah aplikasi stabil dengan berbagai beban

Load Testing

Load testing adalah teknik *performance testing* yang mana respon sistem diukur dalam berbagai load condition. Pengujian ini membantu menentukan bagaimana software berperilaku ketika beberapa user mengakses software secara bersamaan. *Load testing* diperlukan untuk membuat simulasi akses aplikasi website secara simultan. Cara ini lebih baik dibandingkan dengan harus mengundang sekian belas, atau puluh orang sekaligus untuk mengakses sebuah website.

Langkah-langkah pengujian software ada 4 yaitu:

1. *Unit testing*-testing per unit yaitu mencoba alur yang spesifik pada struktur modul kontrol untuk memastikan pelengkapan secara penuh dan pendeteksian error secara maksimum

2. *Integration testing* yaitu testing per penggabungan unit yaitu pengalamatan dari isu-isu yang diasosiasikan dengan masalah ganda pada verifikasi dan konstruksi program
3. *High-order test* yaitu terjadi ketika software telah selesai diintegrasikan atau dibangun menjadi satu –tidak terpisah-pisah
4. *Validation test* yaitu menyediakan jaminan akhir bahwa software memenuhi semua kebutuhan fungsional, kepribadian dan performa.

Apache JMeter

Aplikasi Apache JMeter adalah perangkat lunak open source, 100% aplikasi Java murni dirancang untuk memuat tes perilaku fungsional dan mengukur kinerja. Ini pada awalnya dirancang untuk pengujian Aplikasi Web tetapi sejak diperluas untuk fungsi tes lainnya. [7]

APACHE JMETER adalah aplikasi open source berbasis Java yang dapat dipergunakan untuk performance test. Bagi seorang QA Engineer jMeter bisa digunakan untuk melakukan load/stress testing Web Application, FTP Application dan Database server test.

Apache JMeter dapat digunakan untuk menguji kinerja baik pada sumber daya statis dan dinamis (web services (SOAP / REST), Web bahasa dinamis - PHP, Java, ASP.NET, File, dll -, Jawa Objects, Pangkalan Data dan Pertanyaan, FTP Server dll). Hal ini dapat digunakan untuk mensimulasikan beban berat pada server, sekelompok server, jaringan atau objek untuk menguji kekuatan atau untuk menganalisa kinerja secara keseluruhan di bawah jenis beban yang berbeda.

Rancangan sistem pengujian pada paper ini adalah sebagai berikut:

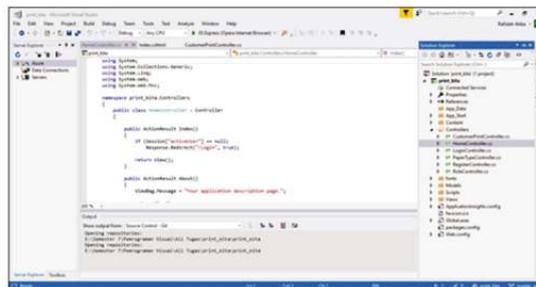
| LOAD | |
|------------------|--------------------|
| threshold/target | method |
| | Black Box |
| | test case sampling |
| | Gorilla |
| | tester's signature |
| result | |



Gambar 1. Rancangan sistem pengujian

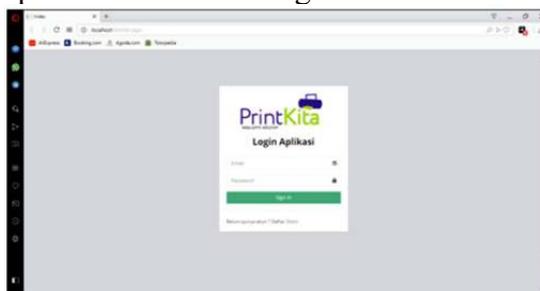
III. PENGUJIAN APLIKASI DENGAN LOAD TESTING

Langkah pertama adalah mempersiapkan *environment* pengujian:



Gambar 2. Environment Pengujian

Tampilan utama halaman login dari aplikasi PrintKita sebagai berikut:



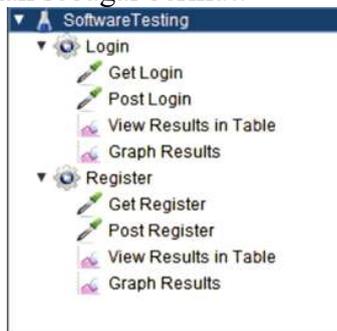
Gambar 3. Tampilan utama aplikasi PrintKita

Menentukan Test Plan Dan Target Pengujian

Langkah pertama adalah membuat test plan dan selanjutnya dilakukan testing terhadap aplikasi. Skenario pengujian aplikasi adalah:

- a. Thread Login
- b. Thread Register

Rancangan skenario testing yang akan dilakukan sebagai berikut:



Gambar 4. Rancangan skenario testing

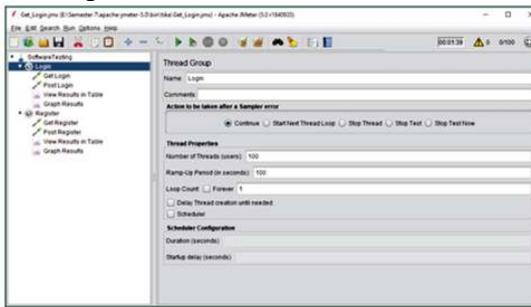
Pengujian thread Login

Thread login dibagi menjadi 2 yaitu konfigurasi Set konfigurasi dan percobaan request HTTP.

- Set Konfigurasi Tahapan

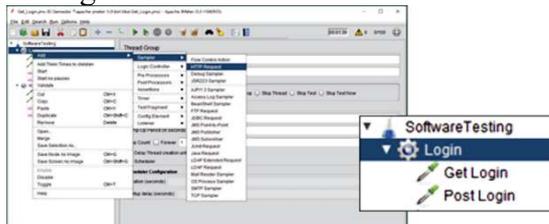
Untuk set konfigurasi tahapannya sebagai berikut:

- Set Number of Thread (user) / jumlah user yang akan mengakses modul/fitur tersebut.
- Set Ramp-Uo Period (in seconds) / jumlah detik yang dibutuhkan untuk semua user yang telah di set sebelumnya. Misalkan pada contoh diatas, jumlah user diset sebesar 100 dan jumlah detik untuk semua user adalah 100 detik. Maka dapat dihitung bahwa setiap user mengakses fitur tersebut dalam setiap 1 detik.



- Percobaan request HTTP
 Setelah mengeset, langkah berikutnya adalah:

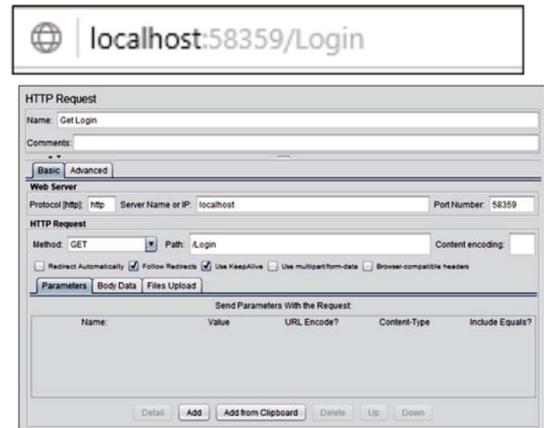
- Membuat 2 request HTTP dengan cara klik kanan Login kemudian pilih Add -> Sample -> HTTP Request. Pada percobaan ini, 2 request tersebut adalah Get Login dan Post Login:



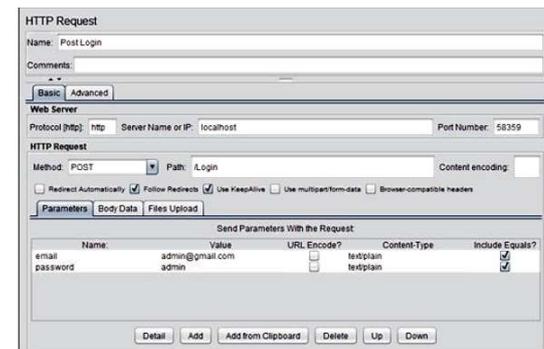
Gambar 5. Request HTTP untuk Login

- Isi data pada bagian web server sesuai dengan aplikasi yang digunakan. Dan pada pengujian ini, protokol yang digunakan adalah http, untuk server namanya adalah localhost dengan port number 58359, untuk methodnya adalah GET, dan untuk Pathnya adalah Login.
- Isi data pada bagian web server sesuai dengan aplikasi yang

digunakan. Dan pada pengujian ini, protokol yang digunakan adalah http, untuk server namanya adalah localhost dengan port number 58359, untuk methodnya adalah POST, dan untuk Pathnya adalah Login.



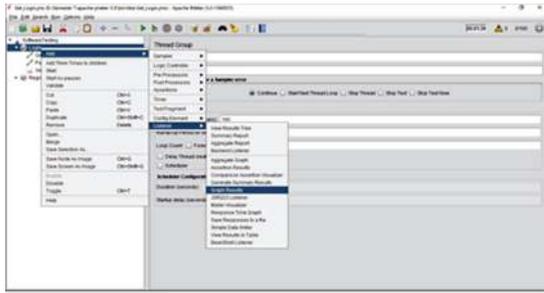
Gambar 6. Konfigurasi HTTP Post



Gambar 7. Konfigurasi Path Login

Kemudian tambahkan parameter sesuai dengan inputan pada login, yaitu email dan password. Isi value pada bagian-bagian tersebut sesuai dengan data yang ada pada database. Pada gambar 6 dan 7, value yang diisikan untuk email adalah admin@gmail.com dan value untuk password adalah admin.

- Menampilkan hasil process memory
 Untuk menampilkan hasil process memory berupa tabel, dapat dilakukan dengan cara klik kanan pada HTTP Request, pilih Add -> Listener -> View Results in Table. Kemudian lakukan proses:

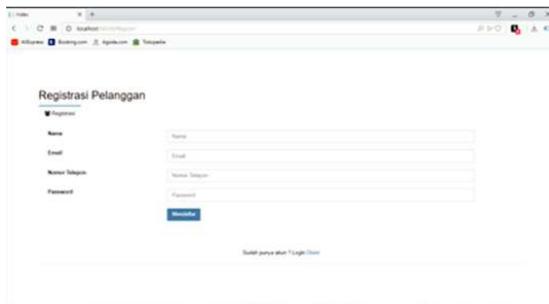


Gambar 8. Konfigurasi menampilkan hasil process memory

Untuk menampilkan hasil process memory berupa grafik, klik kanan pada HTTP Request, pilih Add -> Listener -> Graph Result. Kemudian lakukan proses penyimpanan file tersebut. Dari pengujian diatas, dapat diambil kesimpulan bahwa dalam 100 user dengan 1 request tiap 1 second didapat sample time maximum sebesar 915ms (Get Login) dan dalam 100 user dengan 1 request tiap 1 second didapat sample time maximum sebesar 193ms (Post Login)

Pengujian tread group register

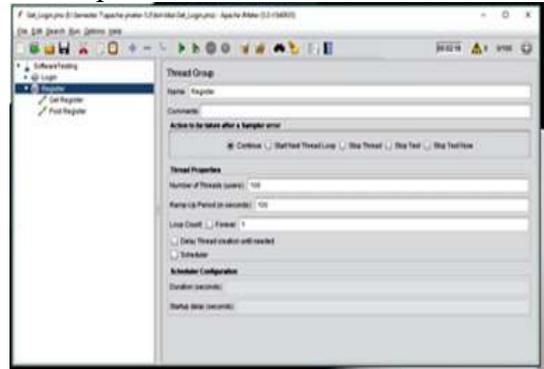
Berikut tampilan register pelanggan pada website Print Kita:



Gambar 9. Register pelanggan di website Thread group register dibagi menjadi 2 yaitu konfigurasi Set konfigurasi dan percobaan request HTTP.

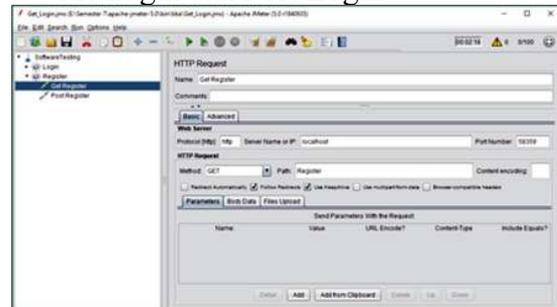
- Set Konfigurasi Tahapan Untuk set konfigurasi tahapannya sebagai berikut:
 - a. Set Number of Thread (user) / jumlah user yang akan mengakses modul/fitur tersebut.
 - b. Set Ramp-Uo Period (in seconds) / jumlah detik yang dibutuhkan untuk semua user yang telah di set sebelumnya. Misalkan pada contoh diatas, jumlah user diset sebesar 100 dan jumlah detik untuk semua user

adalah 100 detik. Maka dapat dihitung bahwa setiap user mengakses fitur tersebut dalam setiap 1 detik.



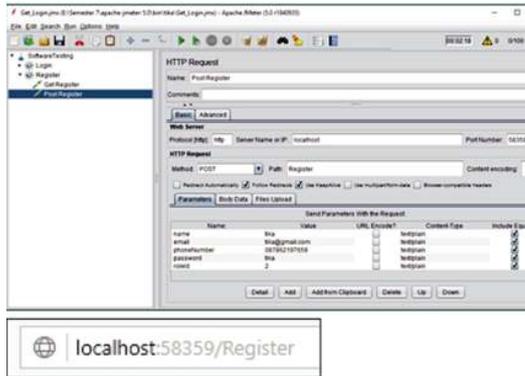
Gambar 10. Tahapan konfigurasi register

- Percobaan request HTTP Setelah mengeset, langkah berikutnya adalah:
 - a. Membuat 2 request HTTP dengan cara klik kanan Login kemudian pilih Add -> Sample -> HTTP Request. Pada percobaan ini, 2 request tersebut adalah Get Login dan Post Login



Gambar 11. Konfigurasi Request HTTP

- b. Isi data pada bagian web server sesuai dengan aplikasi yang digunakan. Pada pengujian ini, protokol yang digunakan adalah http, untuk server namanya adalah localhost dengan port number 58359, untuk methodnya adalah GET, dan untuk Pathnya adalah Register.



Gambar 12. Konfigurasi Path Register

Analisa hasil pengujian

Pengujian yang dilakukan dengan Load Testing menggunakan metode Black Box dan Gorilla Testing, dapat diambil analisa sebagai berikut:

- Dalam 100 user dengan 1 request tiap 1 second didapat sample time maximum sebesar 1424ms (Get Register)
- Dalam 100 user dengan 1 request tiap 1 second didapat sample time maximum sebesar 252ms (Post Register)

IV. KESIMPULAN DAN SARAN

Throughput adalah parameter yang paling penting. Throughput mewakili kemampuan server untuk menangani beban berat. Semakin tinggi Throughput, semakin baik kinerja server.

Hasil pengujian menunjukkan throughput pada server aplikasi print kita (login) adalah 121.396/menit. Hal ini berarti, server aplikasi print kita dapat menangani 121.396 permintaan login per menit. Dan throughput pada server aplikasi print kita (register) adalah 121.272/menit. Sehingga server aplikasi print kita dapat menangani 121.396 permintaan login per menit. Nilai ini cukup tinggi, sehingga dapat disimpulkan bahwa server aplikasi print kita memiliki kinerja yang baik dalam fitur login dan register

REFERENSI

[1] Shi, Mingtao, 2010, Software Functional Testing from the Perspective of Business Practice Computer and Information Science, www.cssenet.org/cis .

[2] Bhat, A, and Quadri, S.M.K, 2015, Equivalence Class Partitioning and Boundary Value Analysis = A review, 2nd International Conference on Computing for Sustainable Global Development (INDIACom)

[3] Khan, Mohd Ehmer, 2011, Different Approach to Blackbox Testing Technique for Finding Error, International Journal of Software Engineering & Applications (IJSEA), Vol.2, No.4, October 2011

[4] Nidhra, Srinivas, and Dondeti, Jagruthi, 2012, Blackbox and Whitebox Testing Techniques - A Literature Review, International Journal of Embedded Systems and Applications (IJESA) Vol.2, No.2, June 2012

[5] Mustaqbal, Sidiq, and Firdaus, Roeri Fajar, 2019, Pengujian aplikasi menggunakan black box testing boundary value analysis, Jurnal Ilmiah Teknologi Informasi Terapan (JITTER)

[6] <https://jmeter.apache.org/>

[7] <https://lms.onnocenter.or.id/wiki/index.php/Jmeter>

[8] S. Sharmila¹, Dr. E. Ramadevi². Analysis of Performance Testing on Web Applications. International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 3, March 2014

[9] Pooja Ranout, Neha Thakur. Comparative Analysis of Load testing tool: LOAD Complete and WAPT pro. IJEDR Volume 5, Issue 2 2017.

[10] Narinder Kaur, Kailash Bahl. Performance Testing Of Insitute Website Using Jmeter. International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 4, April 2016.

[11] Shikha Dhiman Pratibha Sharma et al. Performance Testing: A Comparative Study and Analysis of Web Service Testing Tools. International Journal of Computer Science and Mobile Computing. Vol.5 Issue.6, June- 2016, pg. 507-512

[12] M. Sidi Mustaqbal, Roeri Fajri Firdaus, Hendra Rahmadi. PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS. Jurnal Ilmiah Teknologi Informasi Terapan Volume I, No 3, 10 Agustus 2015

[13] I. Bornstein, "Practical Software Testing: process oriented approach," Springer Professional Computing, 2003.

[14] M. E. Khan, "Different Forms of Software Testing Techniques for Finding Errors,"



International Journal of Software Engineering (IJSE), vol. 7, no. 3, 2010.

[15] IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 3, Issue 3, June-July, 2015

[16] International Journal of Technical Research and Applications e-ISSN: 2320-8163, www.ijtra.com Volume 2, Issue 2 (March-April 2014)

[17] International Journal of Current Engineering and Technology E-ISSN 2277 – 4106, P-ISSN 2347 - 5161 ©2014 INPRESSCO®, All Rights Reserved Available at

32TUhttp://inpressco.com/category/ijcetU32T

[18] International Journal of Advance Research in Computer Science and Management Studies Volume 3, Issue 10, October 2015

[19] F.Elberzhag, J.Munch and R.Eschbach."Reducing test effort: A Systematic Mapping study on Existing approaches," Information and Technology 54, 2012

[20] http://www.pcmag.com/encyclopedia_term/0,2542,t=White+box+testing&i=54432,00.asp, February 08, 2009.

[21] http://www.testingstandards.co.uk/living_glossary.htm#Testing, February 08, 2009