SOFTWARE TESTING AUTOMATION MENGGUNAKAN RATIONAL SOFTWARE

Adi Pratomo, ST, M.Kom
Dosen STMIK ASIA Malang

ABSTRAKSI

Tujuan dari setiap pengembang perangkat lunak adalah menghasilkan suatu perangkat lunak yang berkualitas, tepat waktu, dan sesuai anggaran. Untuk mendapatkan kualitas perangkat lunak yang baik maka diperlukan suatu proses pengujian yang dapat menjaga kualitas.

Untuk melakukan pengujian dengan baik, maka diperlukan suatu metodologi yang tepat untuk menerapkan proses pengujian dalam pengembangan perangkat lunak. Salah satu metodologi yang banyak digunakan adalah Rational Unified Process, dengan dukungan tools yang lengkap.

Dalam penelitian ini disajikan pemanfaatan metodologi Rational Unified Process untuk melakukan pengujian terhadap perangkat lunak sesuai tahapan-tahapan dalam workflow testing. Kemudian dilanjutkan dengan melakukan pengujian terhadap aplikasi berbasis Web untuk memperoleh pemahaman mengenai tahapan pengujian perangkat lunak menggunakan alat bantu Rational TestManager dan Rational Robot dari tahap persiapan, perencanaan, desain, implementasi, eksekusi, dan evaluasi pengujian.

PENDAHULUAN

1. LATAR BELAKANG

Pesatnya perkembangan teknologi saat ini meyebabkan kemajuan dalam pengembangan perangkat lunak. Dengan adanya kemajuan dalam bidang teknologi informasi, maka pengembang perangkat lunak dituntut untuk dapat menghasilkan suatu produk perangkat lunak yang berkualitas. Produk perangkat lunak yang dihasilkan haruslah bebas dari *defect* dan dapat berjalan dengan baik.

Software testing merupakan bagian dari proses pengembangan perangkat lunak yang memastikan suatu program, *subsystem* atau aplikasi berfungsi sesuai dengan desain. Agar dapat melakukan software testing dengan baik, maka diperlukan suatu pemahaman terhadap *workflow test*. Pada metodologi RUP, *workflow testing* merupakan suatu tahapan dalam pengembangan perangkat lunak yang berhubungan dengan tahapan sebelumnya dan juga didukung oleh ketersediaan tools untuk memudahkan pekerjaan. Tools yang tersedia untuk mendukung langkah-langkah testing diantaranya Rational TestManager dan Rational Robot.

2. PERUMUSAN MASALAH

Ada beberapa pertanyaan yang muncul dari peneliti dalam merencanakan penelitian ini, diantaranya :

- Bagaimana melakukan pengujian perangkat lunak yang efektif dalam menemukan *defect*, sehingga mampu meningkatkan kualitas perangkat lunak ?
- Bagaimana mengurangi tingkat kesalahan yang dilakukan oleh seorang *tester* (penguji) dalam melakukan pengujian perangkat lunak yang dikarenakan melakukan hal yang sama secara berulang-ulang secara manual ?
- Seberapa besar manfaat *tools* (alat bantu) dalam melakukan pengujian perangkat lunak sehingga memperoleh hasil pengujian yang maksimal ?

3. RUANG LINGKUP DAN BATASAN

Dalam penelitian ini akan dilakukan pembahasan mengenai metodologi RUP (*Rational Unified Process*) dan workflow test dalam metodologi RUP. Langkah-langkah yang diambil dalam penelitian ini memanfaatkan alat bantu pengujian perangkat lunak yang tersedia dalam RUP, yaitu Rational Robot dan Rational TestManager. Untuk memberikan gambaran yang lebih jelas dalam proses pengujian, akan dipergunakan suatu aplikasi berbasiskan client server. Aplikasi yang dipilih adalah Webmail UI, yaitu suatu aplikasi yang menyediakan fasilitas email untuk lingkungan Universitas Indonesia.

4. TUJUAN DAN MANFAAT

Tujuan dari penelitian ini adalah untuk memberikan gambaran mengenai penggunaan alat bantu dalam melakukan pengujian perangkat lunak menggunakan Rational TestManager dan Rational Robot.

Manfaat yang dapat diambil dari pengerjaan penelitian ini diharapkan dapat memberikan suatu alternatif dalam melakukan pengujian perangkat lunak.

5. METODE PENELITIAN

Dalam penelitian ini digunakan metode studi kasus dengan mengambil kasus pengujian aplikasi Webmail di Universitas Indonesia. Penelitian dengan pendekatan ini nantinya akan berusaha memotret situasi sebagaimana adanya, sedetail mungkin. Selanjutnya hasil penelitian dianalisis dan disimpulkan.

TINJAUAN PUSTAKA

1. METODOLOGI

Dalam bidang pengembangan perangkat lunak dikenal istilah metodologi. Metodologi merupakan suatu kumpulan langkah-langkah yang telah terbukti dan diaplikasikan dalam suatu rangkaian pekerjaan untuk mendapatkan hasil yang baik.

Dalam bidang pengembangan perangkat lunak, banyak metodologi yang dapat diterapkan. Pada dasarnya langkah-langkah dalam pengembangan perangkat lunak adalah:

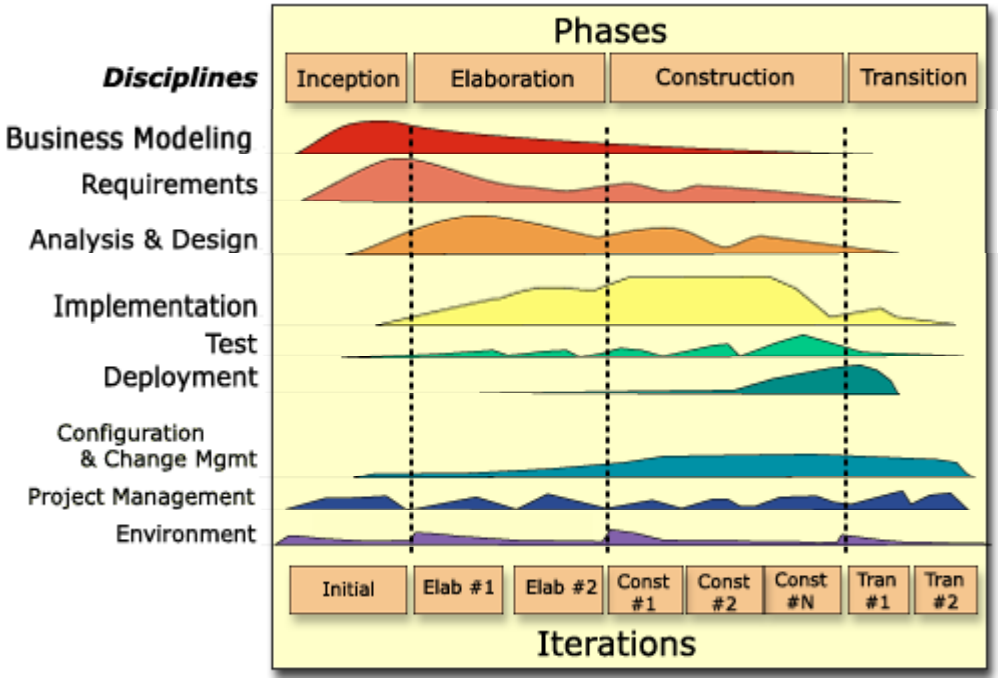
1. *Analizing problems and requirements*
2. *Design*
3. *Implementation and development*
4. *Testing*
5. *Integration*
6. *Maintenance*

Proses-proses tersebut dapat dikerjakan secara berulang-ulang (*iterasi*) dalam suatu daur proses (*life cycle*) untuk mendapatkan hasil yang paling baik. Setiap tahapan pada pengembangan perangkat lunak harus dikerjakan dengan baik, sebab akan sangat berpengaruh pada tahapan-tahapan lainnya. Apabila salah satu tahapan tidak berjalan dengan baik, maka bisa jadi produk yang dihasilkan akan tidak sesuai dengan harapan. Metodologi sangat dibutuhkan karena akan membuat proses pengembangan menjadi teratur.

2. RATIONAL UNIFIED PROCESS

Metodologi RUP adalah suatu proses rekayasa perangkat lunak yang dikembangkan oleh Rational Software Corporation, dengan mengumpulkan beberapa *best practices* di industri pengembangan perangkat lunak. Proses dalam RUP menggunakan use-case driven dan pendekatan iteratif untuk siklus pengembangan perangkat lunak. Tujuan dari metodologi RUP adalah untuk

memastikan proses pengembangan perangkat lunak dapat menghasilkan produk yang berkualitas, tepat waktu dan sesuai dengan anggaran.



Gambar Asitektur Rational Unified Process
(sumber : Rational Unified Process 2003)

3. TESTING DALAM METODOLOGI RUP

Test discipline bertindak sebagai suatu jasa layanan bagi displin yang lain dalam metode RUP. proses *testing* fokus pada penilaian dan evaluasi kualitas dari produk. Pada tahap pengujian dapat diketahui dimana koreksi dapat dilakukan pada setiap iterasi, selain itu juga dapat diketahui kualitas saat ini dari sistem yang sedang dikembangkan Kegiatan pada pengujian perangkat lunak diantaranya berkisar pada menemukan dan mendokumentasikan *defect* pada software quality, melakukan validasi dan pembuktian terhadap asumsi yang dibuat pada tahap desain dan spesifikasi kebutuhan melalui suatu demonstrasi yang konkrit, memastikan bahwa produk perangkat lunak telah berjalan sesuai dengan desain yang telah dibuat dan memastikan bahwa requirement telah diimplementasi dengan benar.

a. Rational TestManager

TestManager adalah suatu framework yang menggabungkan beberapa alat bantu (tools), aset dan data yang berhubungan dengan usaha pengujian. Rational TestManager merupakan tempat untuk mengatur semua aktifitas pengujian termasuk perencanaan, desain, implementasi, pelaksanaan dan analisa. Dengan menggunakan TestManager sebagai suatu framework, semua yang bekerja dalam pegujian dapat mendefinisikan dan menyaring kualitas pengujian. Selain itu pada framework ini tim pengembang dapat mendefinisikan rencana implementasi serta menentukan kondisi sistem. .

b. Rational Robot

Rational Robot adalah suatu gabungan dari beberapa komponen yang dipergunakan untuk mengotomatisasi testing terhadap Microsoft Windows client/server dan aplikasi Internet yang berjalan pada Windows NT 4.0, Windows XP, Windows 2000, Windows 98, and Windows Me.

Rational Robot dipergunakan untuk mengembangkan tiga macam script, yaitu GUI script untuk functional testing dan virtual user (VU) dan VB script untuk performance testing. Teknologi Object-Oriented Recording pada Rational Robot dapat menghasilkan *scripts* secara cepat dengan menjalankan dan menggunakan application-under-test. Robot menggunakan perekaman berorientasi objek untuk mengidentifikasi objek pada aplikasi yang akan diuji. Teknologi pengujian objek pada Rational Robot mampu menguji objek pada aplikasi yang akan di uji, termasuk objek properties dan data. Selain itu Rational Robot dapat menguji standar windows objects dan IDE-spesific object, baik dalam keadaan visible pada interface maupun tersembunyi.

4. Jenis Pengujian

Rational Robot merupakan sebuah alat bantu yang sangat bermanfaat untuk melakukan perencanaan, pengembangan, pelaksanaan dan analisa functional test. Apabila penggunaan Rational Robot digabungkan dengan Rational TestManager, Rational Robot dapat membantu melakukan perencanaan dan mengembangkan performance test. Berikut adalah jenis-jenis pengujian yang dapat dilakukan dengan menggunakan Rational Robot.

1. Functional Test
   Functional test dapat membantu menentukan apakah sistem berjalan sesuai dengan yang diharapkan. Functional test biasanya hanya melibatkan pengguna GUI. Pada functional test, dapat dilakukan pengujian terhadap operasi dan tampilan dari obyek GUI.
2. Performance Test
   Performance test dapat membantu untuk menentukan apakah *multi client system* telah berjalan sesuai dengan definisi standar user dibawah beban yang bervariasi. Performance server dijalankan terhadap database servers, TUXEDO servers dan Web Servers untuk mendapatkan suatu titik dimana sistem tidak dapat menghadapi beban atau tekanan tertentu. Performance test biasanya membebankan server dengan beberapa virtual tester.

SOFTWARE TESTING

Testing merupakan bagian yang tidak lepas dari proyek pengembangan perangkat lunak. Salah satu kunci untuk menghasilkan aplikasi yang berkualitas adalah dengan melakukan perencanaan dan eksekusi pengujian secara pararel terhadap pengembangan aplikasi. Dalam RUP, pendekatan yang dipergunakan dalam melakukan testing tidak lepas dari penggunaan iterative process dan tools.

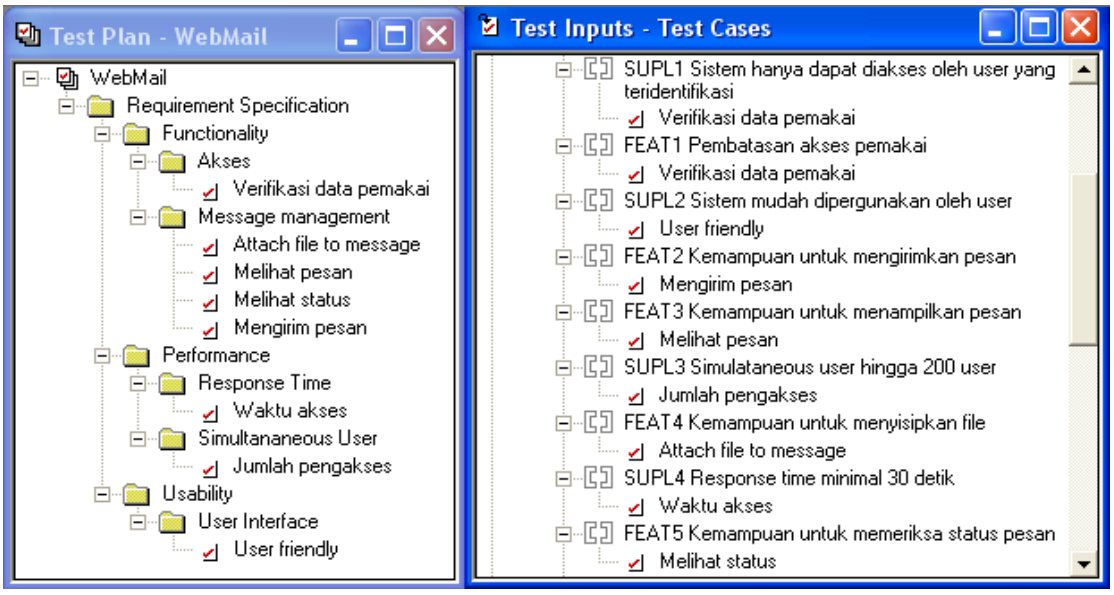
1. TAHAP PERSIAPAN

Sebelum melakukan perencanaan dalam melakukan software testing menggunakan Rational, hal yang harus dilakukan sebelumnya adalah membuat test project. Test project dibuat untuk menyimpan semua test asset. Yang termasuk test asset diantaranya test case, test result dan lain lainnya.

2. TAHAP PERENCANAAN

Test plan adalah suatu mekanisme yang dipergunakan di Rational TestManager untuk mengelola dan membuat test case untuk project. Tahapan ini termasuk dalam tahapan perencanaan test. Didalam worflow testing pada Rational Unified Process, tahapan pembuatan test plan termasuk

di dalam workflow detail mendefinisikan misi evaluasi (*define evaluation mission*), meningkatkan aset pengujian (*improve test assets*) dan verifikasi pendekatan test (*verify test approach*). Aktifitas yang terlibat dalam pembuatan test plan diantaranya *define test approach*, *identify test motivator*, *identify target of test* dan *define test environment configuration*.



Gambar Test Case

3. TAHAP DESAIN

Setelah tahap perencanaan selesai, tahap selanjutnya adalah tahap desain. Pada tahap desain ditentukan bagaimana testing akan dilakukan. Pada tahap desain test dilakukan identifikasi terhadap :

- Langkah dasar yang dibutuhkan untuk melakukan pengujian.
- Bagaimana memastikan bahwa item-item atau feature yang diuji berjalan dengan baik.
- Kondisi awal (pre condition) dari test case, atau bagaimana melakukan set-up terhadap aplikasi dan sistem agar test case dapat berjalan.
- Kondisi akhir (post condition) dari test case, atau bagaimana melaksanakan clean-up setelah test case dijalankan.
- Kriteria yang dapat diterima, atau bagaimana menentukan kriteria test case yang dapat diterima.

Tahap desain test dilakukan berdasarkan test input seperti deskripsi feature dan software specification (requirement). Hasil dari tahap desain merupakan acuan bagi tahap implementasi. Secara garis besar, tahap desain test terdiri dari dua langkah utama, yaitu :

- Menentukan spesifikasi dari langkah dasar dan verifikasi point.
- Menentukan spesifikasi dari *precondition*, *post-condition* dan *acceptance criteria*.

4. TAHAP IMPLEMENTASI

Pada tahapan implementasi test case, dibuat test script atau test suite yang kemudian diasosiasikan dengan test case. Pada Rational TestManager terdapat built-in yang mendukung implementasi test script yang dihasilkan melalui otomatisasi menggunakan Rational Robot.

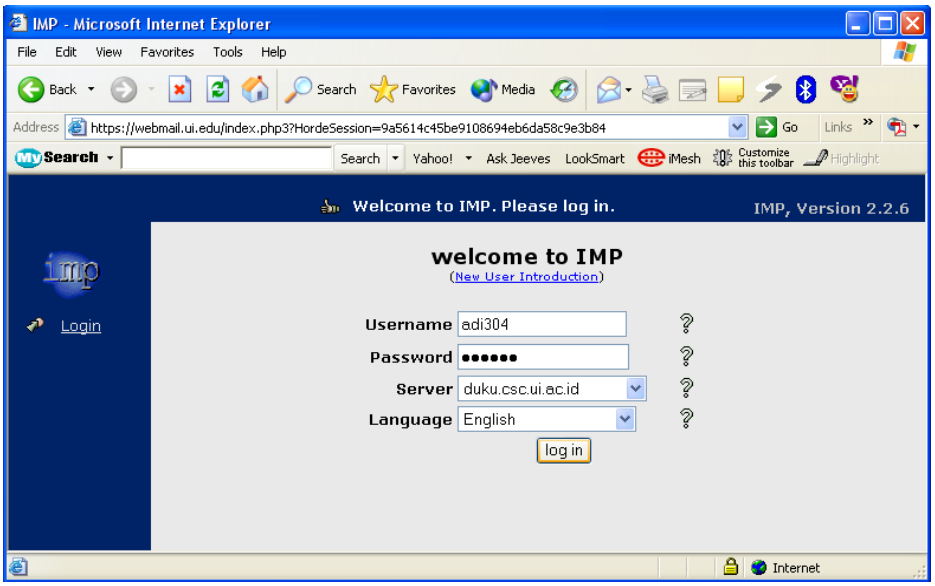
Pada saat perekaman sedang berjalan, sejauh ini sudah direkam beberapa aktifitas yang terjadi di aplikasi. Untuk menguji aplikasi, harus ditentukan beberapa *verification point (VP)*.

Tampilan, input, dan beberapa hal lainnya yang telah direkam pada aplikasi merupakan beberapa fungsi yang akan di verifikasi pada pengujian.

Verification point menyediakan kemampuan untuk melakukan verifikasi kondisi dan data dari aplikasi yang diuji. Pada dasarnya verification point adalah untuk menangkap suatu kondisi dari aplikasi sebagai suatu *baseline*. Baseline akan dibandingkan dengan kondisi sebenarnya dari aplikasi pada saat dilakukan *playback*. Apabila antara kondisi tersebut sama dengan baseline, maka verification point dianggap lolos (*passes*) jika tidak sama maka akan dianggap gagal (*fails*).

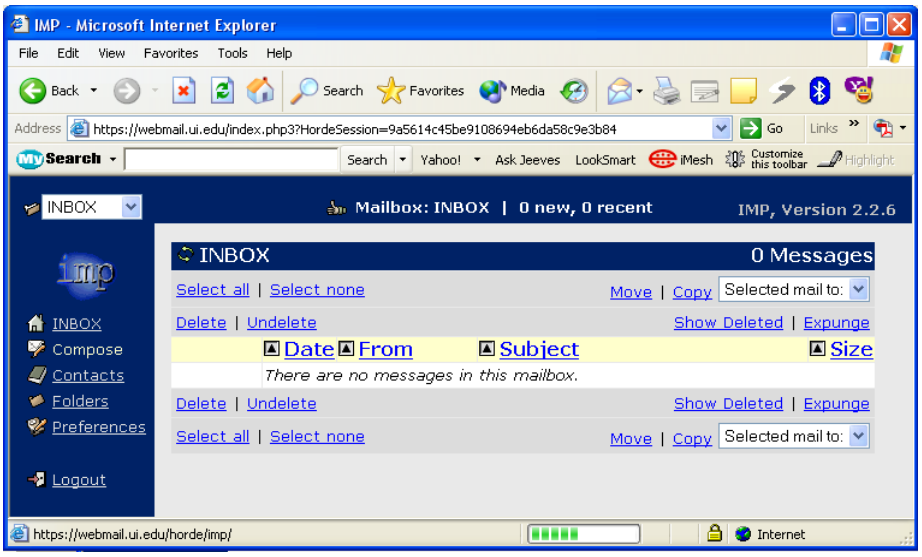
a. Aktifitas yang direkam

Berikut ini merupakan contoh studi kasus, dimana aplikasi yang dipergunakan adalah WebMail Universitas Indonesia. Aktifitas yang direkam adalah aktifitas untuk mengirimkan pesan, dimana sebelumnya user harus melakukan login.



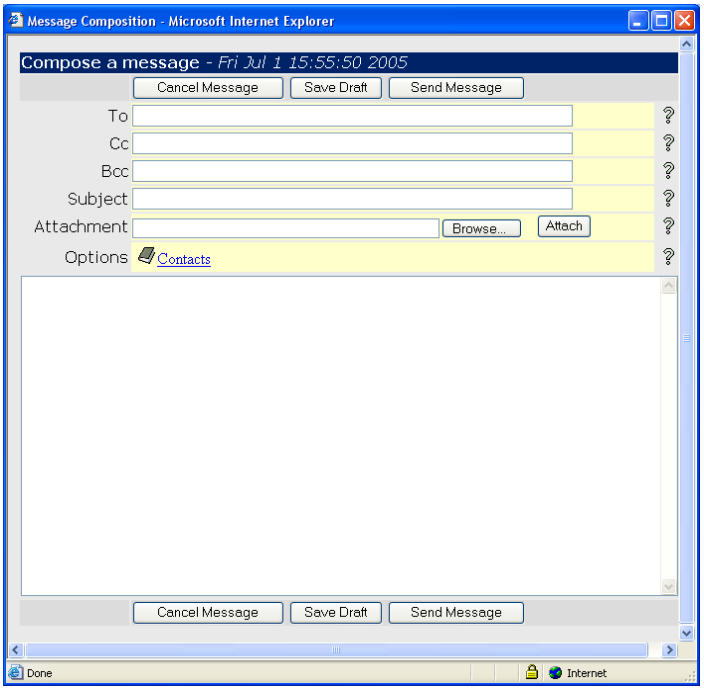
Gambar Login Page

Pada saat user akan melakukan log in, user harus memasukan username, password dan memilih server. Selanjutnya user menekan tombol Log in. Rational Robot akan merekam semua aktifitas yang terjadi, mulai dari klik mouse, memasukan username dan password, memilih server dan menekan tombol Log in.



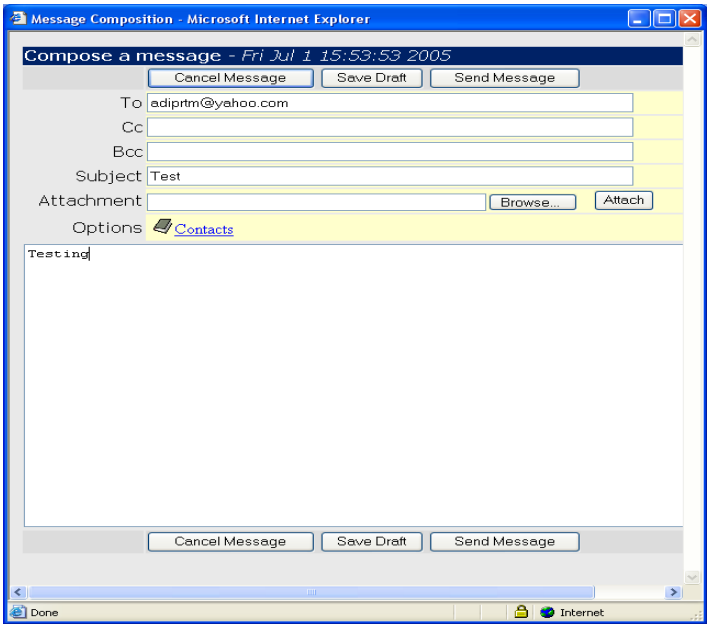
Gambar Halaman Utama WebMail Universitas Indonesia

Setelah menekan tombol Log in, akan ditampilkan halaman utama seperti yang terlihat pada gambar 4. Selanjutnya untuk mengirimkan pesan, user memilih *Compose* yang akan menampilkan *Compose Composition page* seperti terlihat pada gambar 5.



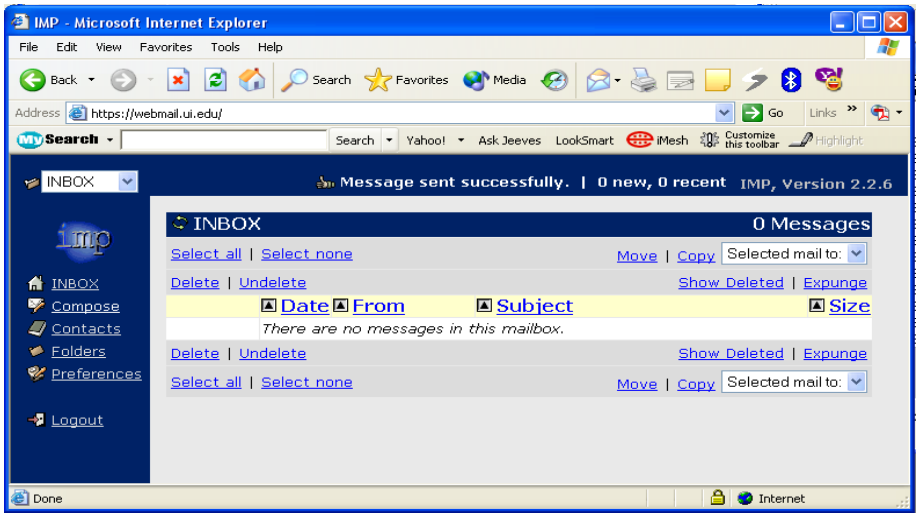
Gambar Compose Composition page

Nantinya pada compose composition page akan ditambahkan feature seperti verification point. Aktifitas selanjutnya pada aplikasi adalah memasukan alamat penerima pesan, subject, dan isi pesan.



Gambar Input Message

Setelah semua informasi dan isi pesan telah dimasukan, selanjutnya user menekan tombol *send message* untuk mengirimkan pesan. Secara otomatis window message composition akan menutup, dan akan kembali ke halaman utama. Pada halaman utama, akan dimunculkan informasi mengenai pesan yang dikirim. Hal ini dapat dilihat pada gambar 7.



Gambar Send Message Confirmation

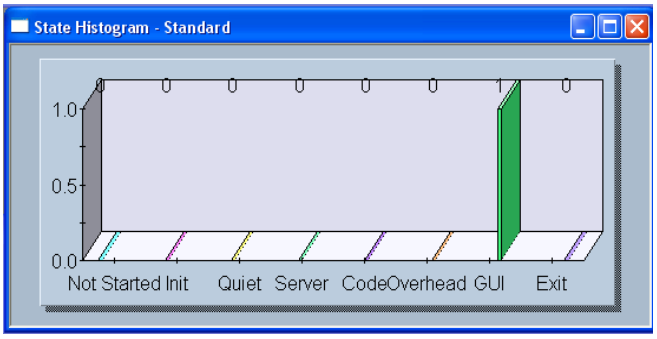
Setelah perekaman dihentikan, maka hasil yang diperoleh adalah suatu test script dari aktifitas yang telah dilakukan pada aplikasi.

5. TAHAP EKSEKUSI TEST

Aktifitas dalam tahap eksekusi test pada dasarnya adalah menjalankan implementasi dari setiap test case untuk melakukan validasi terhadap tingkah laku yang spesifik dari test case. Pelaksanaan pengujian di dalam metodologi RUP adalah dimana seorang tester melaksanakan *integration testing* selama beberapa kali sampai dapat ditentukan bahwa seluruh sistim telah berhasil diintegrasikan.

a. Functional Test

Gambar overall progress view menggambarkan *progress* dari eksekusi test. Apabila dalam eksekusi test terdapat lebih dari satu script, maka pada window ini akan dapat terlihat proses pelaksanaan test per script. Dalam studi kasus ini hanya dipergunakan satu script.

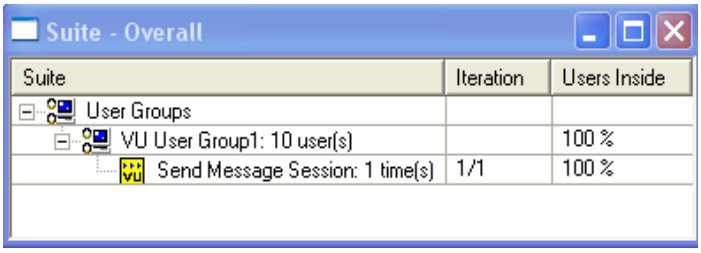


Gambar Functional Test State Histogram

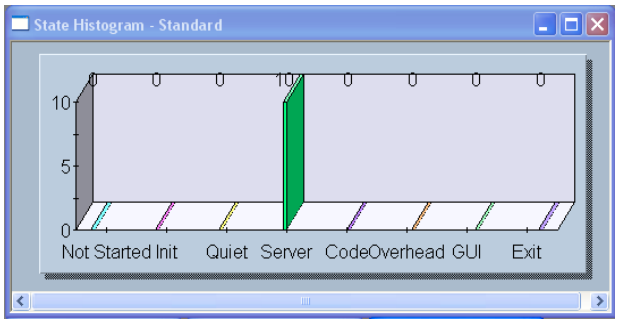
State histogram merupakan suatu grafik yang menggambarkan apa yang sedang terjadi pada komputer yang menjalankan pengujian. Pada studi kasus ini hanya dipergunakan satu komputer untuk melaksanakan test, sehingga pada grafik hanya menunjukkan satu y-*axis*.

b. Performance Test

Virtual tester juga dipergunakan untuk mengukur client response time untuk mendapatkan end-to-end response. Hal ini akan lebih menunjukkan pengalaman sesungguhnya dari user. Pada pengujian biasanya dilibatkan beberapa test script dan beberapa komputer, dimana pada runtime test script plaback dikoordinasikan oleh suite yang telah didesain sebelumnya. Test suites pada pengujian memberikan beban kerja ke server. Performance test yang dilakukan menggunakan 10 virtual user. Jumlah user dapat diubah sesuai keperluan, namun jumlah maksimum user sesuai dengan jumlah yang telah ditentukan sebelumnya pada Run Properties window.



Gambar Performance Test Overall Progress View



Gambar Performance Test State Histogram

State histogram merupakan suatu grafik yang menggambarkan apa yang sedang terjadi pada komputer yang menjalankan pengujian. Pada studi kasus ini terdapat 10 tester, sehingga pada *y-axis* bernilai 10.

User View - Compact

	Groups		Script	State	Time
	Suite	Computer			
1	VU User Group1[01]	Local computer[01]	Send Message Session	Waiting for Response	00:00:07
2	VU User Group1[02]	Local computer[02]	Send Message Session	Waiting for Response	00:00:12
3	VU User Group1[03]	Local computer[03]	Send Message Session	Waiting for Response	00:00:06
4	VU User Group1[04]	Local computer[04]	Send Message Session	Waiting for Response	00:00:13
5	VU User Group1[05]	Local computer[05]	Send Message Session	Waiting for Response	00:00:02
6	VU User Group1[06]	Local computer[06]	Send Message Session	Waiting for Response	00:00:03
7	VU User Group1[07]	Local computer[07]	Send Message Session	Waiting for Response	00:00:12
8	VU User Group1[08]	Local computer[08]	Send Message Session	Waiting for Response	00:00:00
9	VU User Group1[09]	Local computer[09]	Send Message Session	Waiting for Response	00:00:10
10	VU User Group1[10]	Local computer[10]	Send Message Session	Waiting for Response	00:00:11

Gambar Performance Test Computer View

6. TAHAP EVALUASI

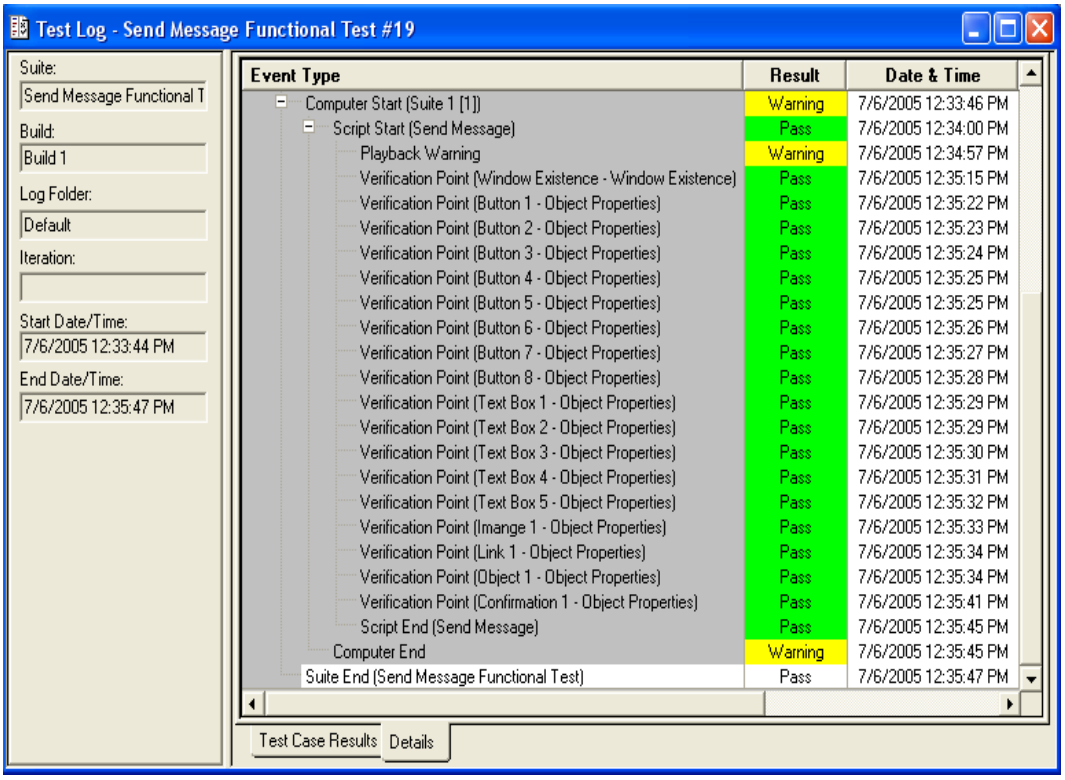
Tahap evaluasi test pada metodologi RUP adalah dimana test designer melakukan review dan pengukuran terhadap kualitas dari test target dan proses pengujian secara keluruhan. Test evaluation summary dihasilkan setelah test designer melakukan review dan pengukuran terhadap hasil pengujian, mengidentifikasi dan merekam change request, serta memperkirakan key measure dari test.

a. Evaluasi Test Log

Setelah menjalankan suite, test case atau test script, TestManager memberikan hasil pengujian dalam test log. Test log window dari TestManager dipergunakan untuk melihat test log yang dibuat oleh TestManager setelah eksekusi. Dengan melihat hasil dari pengujian pada test log, akan dapat diketahui apakah pengujian yang dilakukan berhasil atau tidak. Selain itu test log juga dapat memberikan informasi apakah penyebab kegagalan pengujian merupakan cacat (defect) atau diakibatkan oleh perubahan design.

i. Functional Test Log.

Setelah eksekusi test selesai dilaksanakan, secara otomatis akan ditampilkan log file. Log file dari pelaksanaan automated functional test dalam tutorial ini dapat dilihat pada gambar 12.



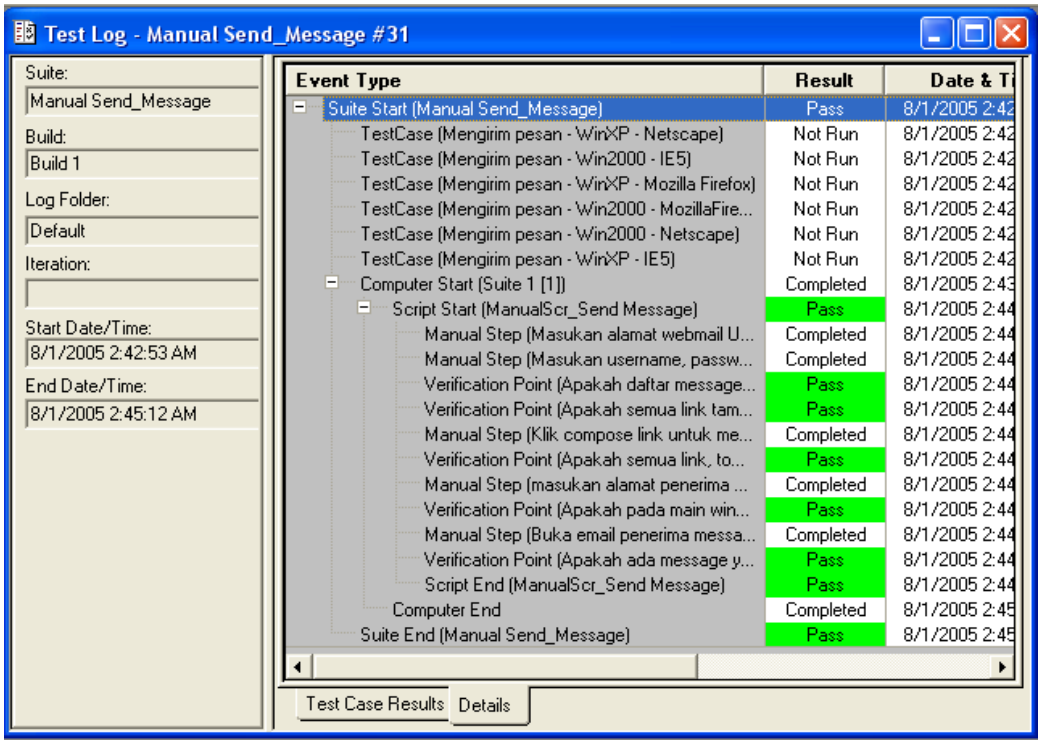
Gambar Automated Functional Test Log

Pada gambar diatas terlihat bahwa semua event memiliki status *pass*, yang berarti pengujian yang dilakukan berjalan dengan baik. Bila pada *test log* terdapat event yang memiliki status *fail*, langkah yang harus diambil adalah menelusuri kesalahan yang terjadi pada saat pengujian. Untuk menelusuri kesalahan yang terjadi dapat dilakukan melalui properties dari *verification point* yang gagal atau membandingkan *baseline* dengan *actual*.

Pada *detail tab* di test log window dapat dilihat penyebab kegagalan. Untuk pengujian secara manual, penyebab kegagalan dapat diketahui dengan cepat oleh tester yang menjalankan test. Untuk pengujian secara otomatis, untuk mengetahui penyebab kegagalan dapat diketahui melalui detail pada test log untuk menentukan penyebab kegagalan dan tindakan lebih lanjut yang diperlukan. Apabila kegagalan merupakan akibat dari eksekusi pengujian secara otomatis, pembanding otomatis akan ditampilkan untuk setiap verification point dan menandai perbedaan antara nilai yang diharapkan dan nilai aktual dari aplikasi.

Berbeda dengan langkah-langkah pengujian yang dilaksanakan sesuai dengan manual test script. Apabila ada kegagalan, maka penguji menelusuri kegagalan bukan berdasarkan detail pada test script yang dihasilkan secara otomatis, melainkan pada langkah-langkah yang dilakukan oleh penguji. Verification point pada pengujian secara manual harus ditentukan sendiri oleh penguji sehingga akan memakan waktu. Pada pengujian secara otomatis, penguji akan menentukan verification point secara otomatis. Pengujian secara manual akan memakan waktu lebih lama dibandingkan pengujian otomatis, hal ini disebabkan karena tester akan menjalankan langkah-

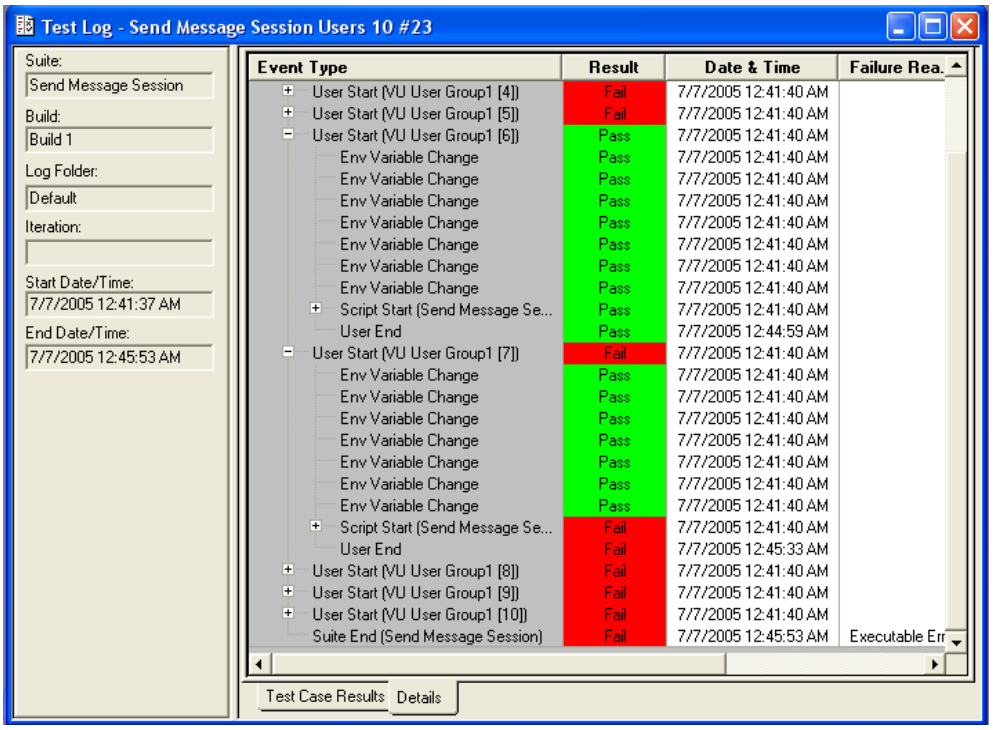
langkah pada manual script dan memeriksa veification point satu persatu. Keberhasilan pengujian secara manual ditentukan oleh ketelitian dari penguji.



Gambar Manual Functional Test Log

ii. Performance Test Log.

Pada sub bab sebelumnya telah dijelaskan bagaimana melaksanakan performance test. Setelah eksekusi performance test selesai, secara otomatis akan ditampilkan log file. Log file dari pelaksanaan performance test dalam tutorial ini dapat dilihat pada gambar 14.



Gambar Performance Testing Test Log

Pada test log diatas hanya satu user yang mempunyai status *pass* yaitu user 6, sedangkan user lainnya mempunyai status *fail*. Apabila ditelusuri dengan membuka user yang mempunyai status result fail, maka dapat dilihat dimana terjadi *fail*. Untuk melihat log event properties dapat dilakukan dengan klik kanan pada event tertentu dan memilih properties. Pada log event properties dapat dilihat properti dan nilai dari dari event, termasuk penyebab kegagalan.

Untuk memeriksa kegagalan dengan lebih detail dapat dilakukan dengan membuka performance testing report, dimana melalui performance testing report dapat ditelusuri kesalahan yang terjadi dengan melihat command ID pada script.

PENUTUP

1. KESIMPULAN

Salah satu kunci untuk menghasilkan aplikasi yang berkualitas adalah dengan melakukan perencanaan dan eksekusi pengujian secara pararel terhadap pengembangan aplikasi. Selain itu juga diperlukan pemahaman terhadap *workflow test*. Pada metodologi Rational Unified Process, *workflow testing* merupakan suatu tahapan dalam pengembangan perangkat lunak yang berhubungan dengan tahapan sebelumnya dan juga didukung oleh ketersediaan tools untuk memudahkan pekerjaan.

Untuk memudahkan aktifitas testing, dapat dipergunakan suatu alat bantu (tools). Dengan melakukan otomatisasi pengujian dengan menggunakan tools, maka dapat mengurangi kesalahan yang biasanya dilakukan oleh faktor manusia. Pengujian yang dilakukan dengan Rational TestManager dan Rational Robot dapat mengukur beban kerja dari lingkungan kerja yang sesungguhnya, dengan membuat model virtual user yang bervariasi. Hal ini akan dapat menghemat waktu, tenaga, dan biaya.

Pengujian yang dilakukan secara otomatis akan dapat menghemat waktu dibandingkan pengujian secara manual. Pada pengujian manual, tester harus melakukan pengujian berdasarkan instruksi pada manual script secara manual, sedangkan pada pengujian secara otomatis script akan berjalan secara otomatis dalam waktu yang lebih cepat.

2. SARAN

Dalam melakukan pengembangan perangkat lunak, akan lebih baik apabila menyertakan tahapan pengujian semenjak awal pengembangan perangkat lunak. Hal ini dilakukan dengan tujuan agar dapat diperoleh feedback sedini mungkin, sehingga proses pengembangan dapat berjalan dengan baik.

Dalam melakukan pengujian diperlukan suatu guideline yang jelas, agar tim pengembang dapat mengetahui apa yang diuji, apa yang menjadi tolak ukur pengujian. Selain itu informasi mengenai guideline harus disebarkan kepada seluruh anggota tim pengembang.

DAFTAR PUSTAKA

Fewster, Mark and Dorothy Graham, 1999, *Software Test Automation : Effective use of Test Execution Tools,*Addison-Wesley, New York.

Kelly, Mike, 2003, *Running your first functional regression test*, Diambil Februari 2005 dari http://www-136.ibm.com/developerworks/rational/

Kelly, Mike, 2004, *Using IBM Rational Robot to run your first performance test*, Diambil Februari 2005 dari http://www-136.ibm.com/developerworks/ rational/

Rational Developer Domain, *Functional Testing With Rational Robot*, Diambil Februari 2005 dari http://www-136.ibm.com/developerworks/rational/

Rational Developer Domain, *Managing Application Testing*, Diambil Februari 2005 dari http://www-136.ibm.com/developerworks/rational/

Rational Software, 2003, *Rational Robot, Version 2003.06.00.436.000,* Rational Software Corporation.

Rational Software, 2003, *Rational TestManagers, Version 2003.06.00.436.000,* Rational Software Corporation

Rational Software, 2003, *Rational Unified Process, Version 2003.06.00.65,* Rational Software Corporation

http://www-106.ibm.com/

http://www.rational.com/