Pengujian Perangkat Lunak



Disusun Oleh:

Bobby Hanjaya 2012730040

Devin Lubianto 2012730075

Teknik Informatika

Fakultas Teknik Informasi dan Sains

Universitas Katolik Parahyangan

**1. Pendahuluan**

**1.1 Latar Belakang**

Pemilihan algoritma yang tepat sangat mempengaruhi performance aplikasi yang akan dibuat. Penilaian performa dapat dilihat dari berbagai hal, diantaranya dilihat berdasarkan waktu. Dalam pembuatan diperlukan beberapa proses yaitu (design, implementasi, dan pengujian), proses tersebut dapat menilai baik atau buruknya sebuah aplikasi / sistem. Pengujian merupakan proses yang sangat penting dalam pembangunan aplikasi / sistem. Sistem penjadwalan pertemuan adalah sebuah aplikasi yang memungkinkan sebuah agenda pertemuan yang melibatkan banyak peserta (participant) dibuat dan dijadwalkan. Jadwal pertemuan disimpulkan secara otomatis dengan menggunakan sebuah algoritma pencarian jadwal berdasarkan ketersediaan waktu para participant. Aplikasi ini dibangun menggunakan bahasa Java dengan data-data yang disimpan dalam format JSON. Data yang disimpan di dalam JSON tersebut digunakan sebagai acuan terhadap pengujian aplikasi ini.

**1.2 Software Test Plan**

Source code dan document dapat dilihat melalui link github yaitu https://github.com/dev1nce/TA\_PengujianPerangkatLunak

**Rencana Kerja**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Rencana Kerja** | **Environment** | **Tools** |
| 1 | Reverse Engineering | Intel inside core i5 4570 | draw.io, Microsoft Word |
| 2 | Software Description | Intel inside core i5 4570 | Microsoft Word |
| 3 | Manual Testing | Intel inside core i5 4570 | Net-Beans 8.2, Microsoft Word,Paint |
| 4 | Performance Testing | Intel inside core i5 4570 | Net-Beans 8.2, Microsoft Word |
| 5 | Boundary Testing | Intel inside core i5 4570 | Net-Beans 8.2, Microsoft Word |

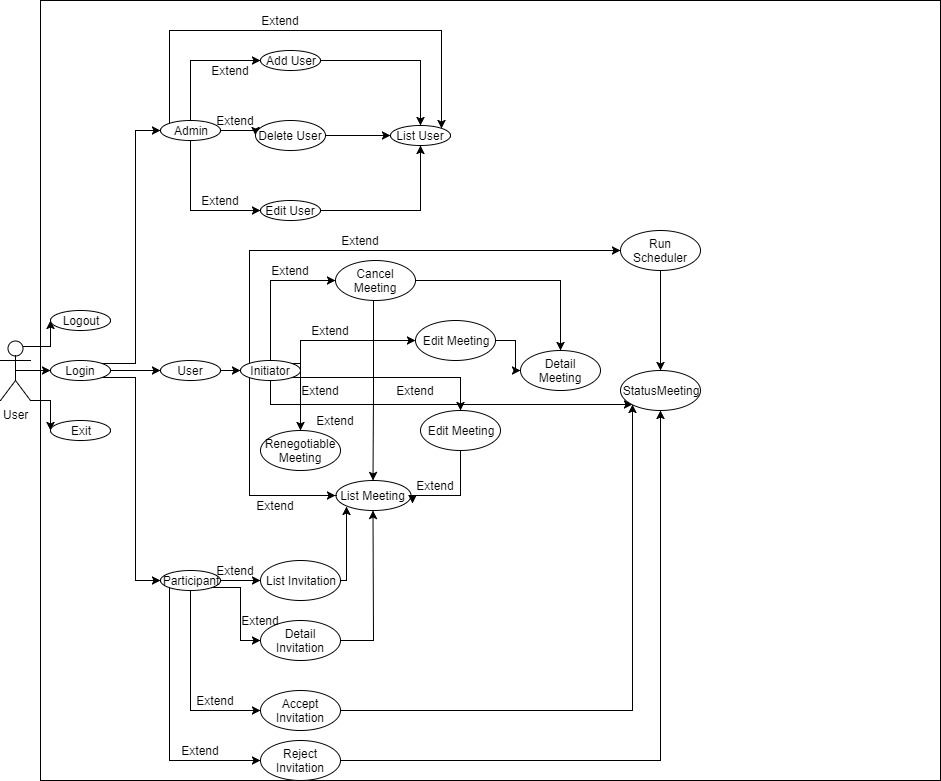
**2.Pengujian**

**2.1 Reverse Engineering**

sebuah proses untuk mencari dan menemukan system teknologi, fungsi dan operasi yang bekerja di balik suatu desain, komponen atau objek melalui sebuah proses analisa yang mendalam pada setiap komponen struktur dari desain atau objek yang diteliti.

* Use Case

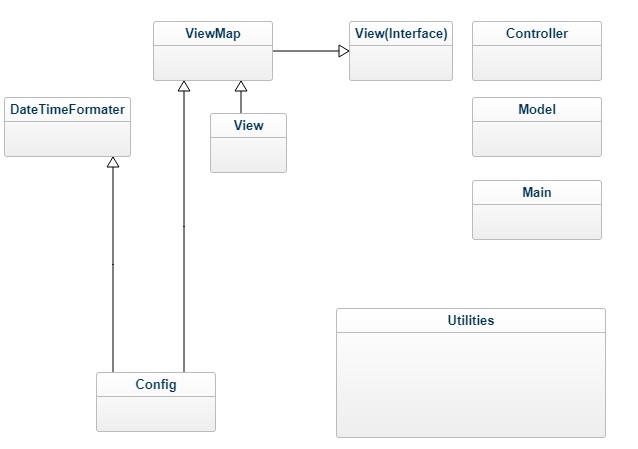
merupakan pemodelan untuk menggambarkan kelakuan (behavior) sistem yang akan dibuat. **Diagram use case** mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat.



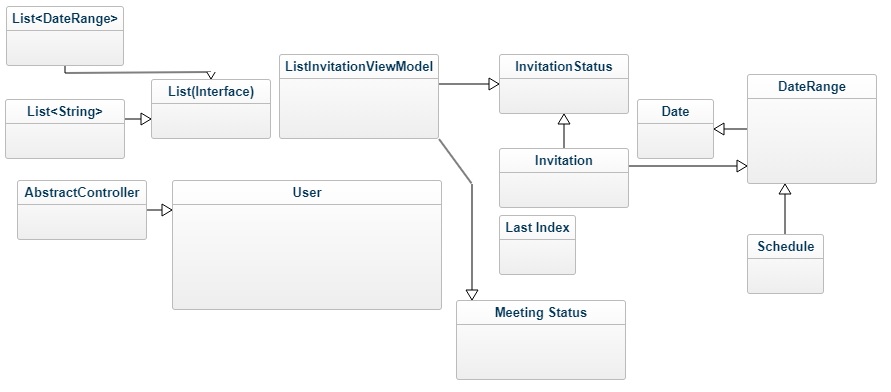
Gambar Use Case Diagram

* Class Diagram

model statis yang menggambarkan struktur dan deskripsi **class** serta hubungannya antara **class**. **Class diagram** mirip ER-**Diagram** pada perancangan database, bedanya pada ER-**diagram** tdk terdapat operasi/methode tapi hanya atribut. **Class** terdiri dari nama kelas, atribut dan operasi/methode



Gambar Class Diagram



**Gambar Class Diagram**

**2.2 Software Test Description**

Software Test Description mendeksripsikan preparasi tes, test case dan prosedur tes yang digunakan untuk melakukan tes kualitas dari applikasi / sistem.

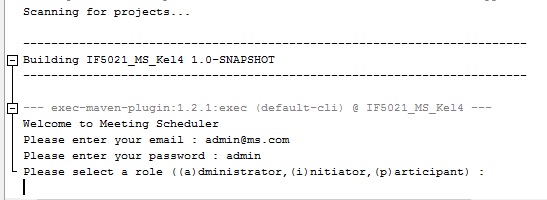
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Case** | **Tindakan** | **Keterangan** | **Test Case** | **Hasil** |
| **1** | Login | User memasukkan id & password sesuai yang tersedia di JSON | Tidak ada bug | Email benar , password salah dan email salah | Sesuai |
| **2** | Select Role | User memilih antara administrator, initiator, atau participant | Tidak ada bug | Role yang terdaftar adalah (a,i,p), sisanya tidak | Sesuai |
| **3** | Melihat command Admin | User dapat melihat command yang tertera | Tidak ada bug | “help” | Sesuai |
| **4** | Add user | menambahkan user dengan cara mengetikkan kata “add-user”. | Tidak ada bug | “add-user” | Sesuai |
| **5** | List user | melihat list user dengan cara mengetikkan kata “list-user”. | Tidak ada bug | “list-user” | Sesuai |
| **6** | Delete user | dapat menghapus user dengan cara mengetikkan kata “delete-user” dan diikuti sebuah email yang ingin dihapus. | Tidak ada bug | “delete-user user@ms.com” | Sesuai |
| **7** | Edit user | mengubah user dengan cara mengetikkan kata “edit-user” dan diikuti sebuah email yang ingin diubah. | Tidak ada bug | “edit-user user@ms.com” | Sesuai |
| **8** | Edit all user | mengubah user dengan cara mengetikkan kata “edit-user” dan diikuti sebuah email yang ingin diubah, selanjutnya mengetikkan kata “all”. | Tidak ada bug | edit-user admin@ms.com(enter) all | Sesuai |
| **9** | Edit username | mengubah user dengan cara mengetikkan kata “edit-user” dan diikuti sebuah email yang ingin diubah, selanjutnya mengetikkan kata “name” lalu diikuti dengan nama terbaru. | Tidak ada bug | edit-user admin@ms.com(enter) name joko | Sesuai |
| **10** | Edit user password | mengubah user dengan cara mengetikkan kata “edit-user” dan diikuti sebuah email yang ingin diubah, selanjutnya mengetikkan kata “password” lalu diikuti dengan nama password terbaru diketikkan secara dua kali. | Tidak ada bug | edit-user admin@ms.com(enter) password medog | Sesuai |

**2.3 Manual Testing Method**

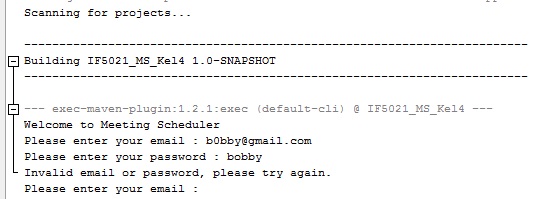
Manual Testing Method merupakan proses pengujian perangkat lunak secara manual untuk mencari kecacatan dari sebuah program. Hal ini membutuhkan penguji untuk memainkan peran sebagai pengguna dimana merekalah yang menggunakan sebagian besar fitur aplikasi untuk memastikan program berjalan dengan benar. Dibawah ini kami laporkan hasil-hasil yang berhubungan dengan manual testing dari sistem penjadwalan pertemuan (meeting scheduler).

1. **Login**

Fitur login merupakan fitur paling awal yang akan ditemui saat menjalankan program. Pada fitur login data yang ada pada JSON akan diperiksa kecocokannya. Pada gambar di bawah ini akan ditunjukkan gambar login yang valid (Gambar 1.1) dan gambar login yang tidak valid (Gambar 1.2).



**Gambar 1.1 Login Valid**



**Gambar 1.2 Login Invalid**

**2. Administrator**

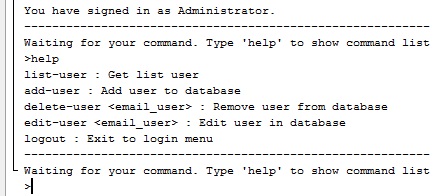
Fitur administrasi dapat berjalan jika menginput e-mail dan password yang sesuai dengan administrator. Ada tiga pilihan user yaitu a untuk administrator i untuk initiator dan p untuk participant. Gambar di bawah akan menunjukkan login sebagai administrator (Gambar 2.1).



**Gambar 2.1 Login as Administrator**

**3. Fitur dari Administrator**

Administrator dapat mengakses semua role. Gambar di bawah menunjukan contoh fitur-fitur dari administrator (Gambar 3.1).



**Gambar 3.1 Fitur dari Administrator**

**4. Add User**

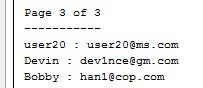
Fitur ini berfungsi untuk menambahkan user ke dalam JSON. Terdapat beberapa input yang harus diisi ke dalam JSON. Gambar ini akan ditunjukkan beberapa inputan yang harus diisi(Gambar 4.1).



**Gambar 4.1 Add User**

**5. Fitur dari List User**

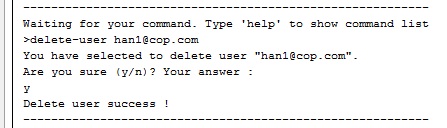
Fitur ini berfungsi untuk melihat semua user yang ada di dalam data JSON. Gambar di bawah ini menunjukan user-user tersebut, termasuk user yang sudah diinput di Gambar 4.1. (Gambar5.1).



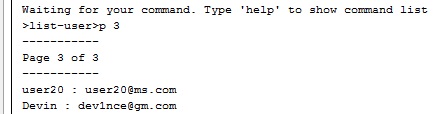
**Gambar 5.1 List User**

**6. Delete User**

Delete user merupakan fitur yang berfungsi untuk menghapus user dari data di JSON. Untuk menghapus user, yang dipakai parameternya hanya emailnya saja. Gambar 6.1 menunjukan proses penghapusan, dan Gambar 6.2 menunjukan list user setelah dihapus.



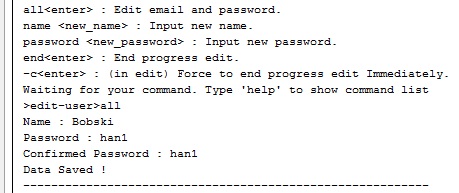
**Gambar 6.1 Delete User**



**Gambar 6.2 List User sesudah di delete**

**7. Edit user**

Edit User merupakan fitur yang berfungsi untuk mengubah data-data yang telah terdapat di dalam JSON. Pada fitur ini terdapat sebuah parameter yang ingin dirubah yaitu e-mail. Perubahan yang dilakukan bisa untuk 1 user atau untuk semua user. (Gambar 7.1) menunjukan perubahan email, dan (Gambar 7.2) merupakan hasil setelah dilakukan pengeditan.



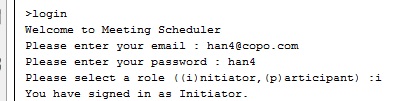
**Gambar** 7.1



**Gambar 7.2**

**8. Initiator**

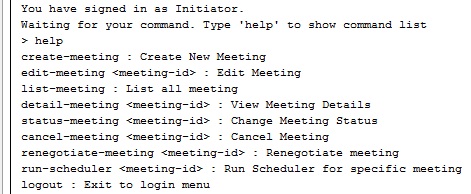
Fitur Initiator hanya dapat berjalan jika menggunakan e-mail yang sudah terdaftar dalam data JSON. (Gambar 8.1) menunjukan login sebagai initiator.



**Gambar 8.1 Login Initiator**

**9. Fitur dalam Initiator**

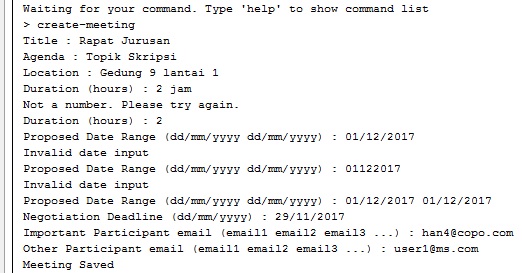
Fitur dalam initiator ini hanya dapat dijalankan jika user ingin membuat jadwal meeting, dan user yang sebagai initiatornya. (Gambar 9.1) menunjukan fitur user sebagai initiator.



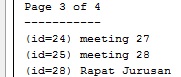
**Gambar 9.1 Fitur User sebagai Initiator**

**10. Create Meeting**

Create Meeting merupakan fitur yang berguna untuk membuat sebuah pertemuan, dimana data nya dimasukan ke dalam data JSON. (Gambar 10.1) menunjukan fitur Create Meeting dan (Gambar 10.2) menunjukan hasil pembuatan meeting.



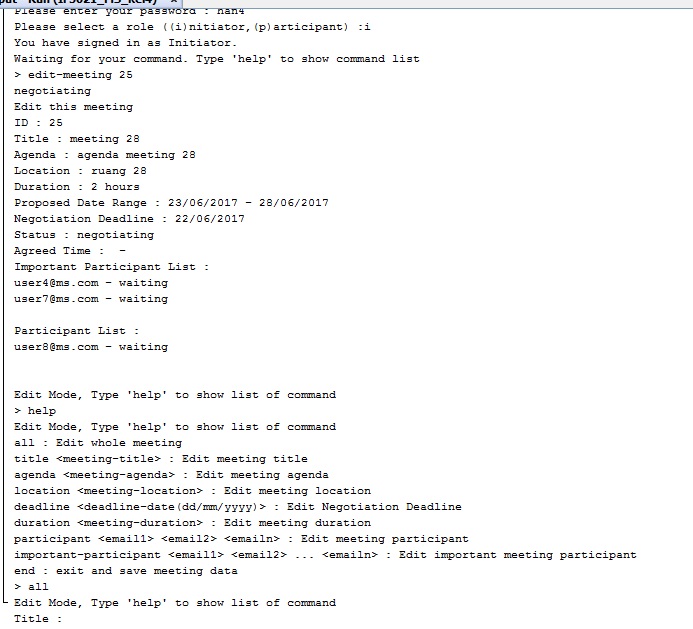
**Gambar 10.1 Create Meeting**



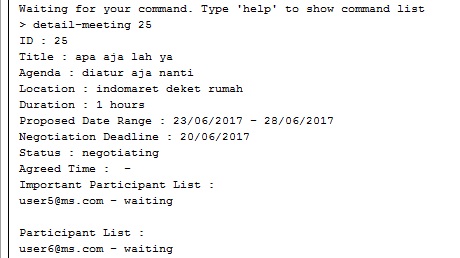
**Gambar 10.2 Hasil Pembuatan Meeting**

**11. Edit Meeting**

Edit meeting merupakan fitur yang berguna untuk mengubah seluruh atau sebagian data pertemuan dalam data di JSON. Dalam fitur ini membutuhkan sebuah parameter yang menandakan pertemuan yang akan dirubah. (Gambar 11.1) menunjukan fitur Edit Meeting dan (Gambar 11.2) menunjukan hasil setelah dirubah.



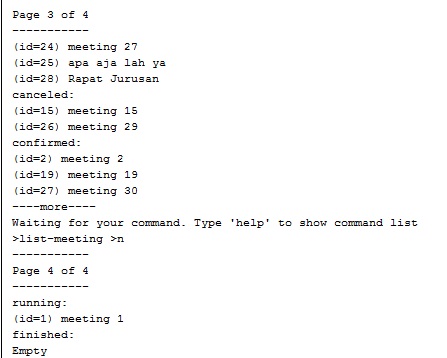
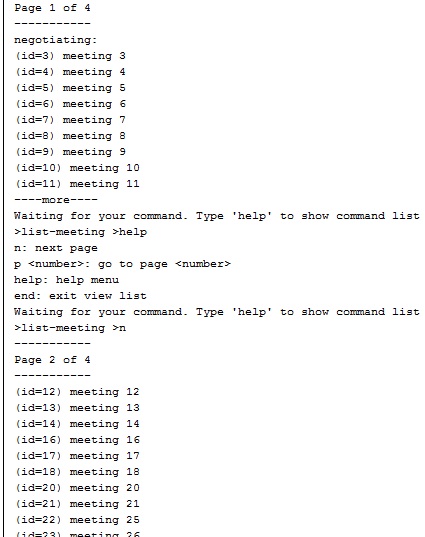
**Gambar 11.1 Edit Meeting**



**Gambar 11.2 Hasil Edit**

**12. List Meeting**

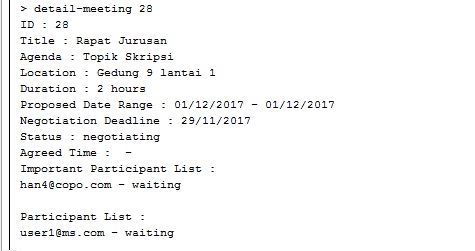
List Meeting merupakan fitur yang berguna untuk melihat semua isi meeting yang ada di dalam data JSON. Dapat melihat jenis-jenis meeting yang diterima, dinegosiasikan, dan dibatalkan. Gambar 12.1 menunjukan list-list meeting yang ada.



**Gambar 12.1 List Meeting**

**13. Detail Meeting**

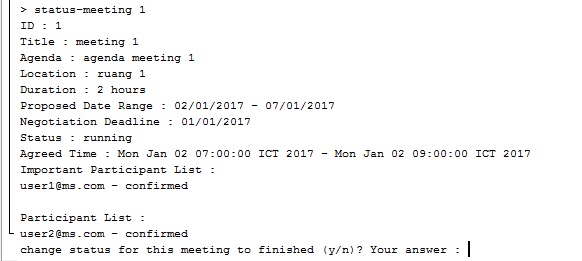
Detail Meeting merupakan fitur yang berguna untuk melihat detail meeting dari data dalam JSON, yang menggunakan parameter berupa id. (Gambar 13.1) merupakan contoh detail meeting dari meeting id 25.



**Gambar 13.1 Detail Meeting**

**14. Status Meeting**

Status Meeting merupakan fitur yang berguna untuk melihat status meeting dan merubah status meeting. Membutuhkan parameter berupa id. Untuk mengetahui status meeting ini diperlukan konfirmasi terlebih dahulu atau menjalankan meeting tersebut terlebih dahulu. Gambar 14.1 menunjukan status meeting.



**Gambar 14.1 Status Meeting**

**15. Cancel Meeting**

Cancel Meeting merupakan fitur yang berguna untuk membatalkan meeting. Fitur ini sendiri membutuhkan parameter berupa id untuk membatalkan meeting. Gambar 15.1 merupakan proses pembatalan meeting.



**Gambar 15.1 Cancel Meeting**

**16. Renegotiate meeting**

Renegotiate meeting merupakan fitur yang berguna untuk negosiasi ulang meeting. Fitur ini memerlukan parameter berupa id. Berikut ini akan kami tunjukan hasilnya.

**17. Run Scheduler**

Run Scheduler merupakan fitur yang berguna untuk menjalankan meeting yang telah ada dalam data pada JSON sehingga apabila participant yang paling paling penting tidak dapat hadir/merespon maka run-scheduler akan membatalkan meeting tersebut. Pada fitur run scheduler memerlukan id sebagai parameter.

**18. Participant**

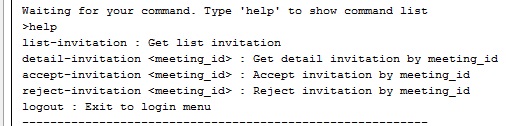
Participant ini merupakan fitur login biasa, yang menggunakan email dan password sebagai parameter.



**Gambar 18.1 Login Participant**

**19. Fitur Participant**

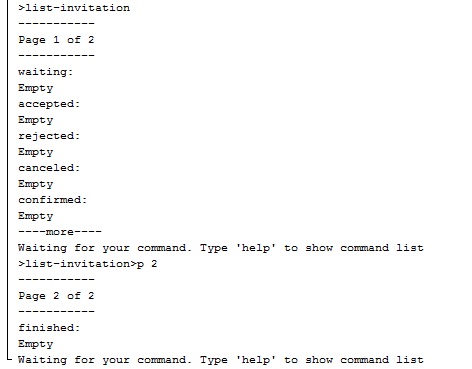
Dalam Participant sendiri terdapat fitur-fitur di dalamnya. Berikut fitur-fitur dari participant.



**Gambar 19.1 Fitur dari Participant**

**20. List Invitation**

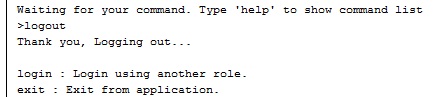
List Invitation sendiri merupakan fitur dari participant dimana dapat melihat participant-participant lain yang ada dalam data JSON. Gambar 20.1 menunjukan list invitation participant.



**Gambar 20.1 List Invitation**

**21. Logout**

Logout fitur yang semua pastinya sudah tahu, untuk keluar dari aplikasi kalau sudah selesai. Gambar 21.1 menunjukan fitur logout tersebut.



**Gambar 21.1 Logout**

**2.4 Performance Testing**

Praktek pengujian yang dilakukan untuk menentukan bagaimana sistem/software dapat melakukan hal respon dan stabilitas di bawah beban kerja tertentu. Pengujian ini dapat melibatkan tes kuantitatif yang dilakukan di laboratorium, seperti mengukur waktu respon atau jumlah MIPS (juta instruksi per detik) di mana fungsi sistem.Pengujian kinerja sering dilakukan bersamaan dengan stress testing. Hal ini berfungsi untuk menyelidiki, mengukur, memvalidasi atau memverifikasi terkait kualitas atribut dari sistem, seperti kehandalan , skalabilitas dan interoperabilitas.Pengujian kinerja dapat meninjau bahwa sistem memenuhi spesifikasi yang ditetapkan oleh vendor atau produsen. Proses ini dapat membandingkan dua atau lebih perangkat/ program dalam hal parameter seperti kecepatan transfer data , bandwidth, throughput, efisiensi atau keandalan. Pengujian kinerja yang efektif, dengan cepat dapat mengidentifikasi masalah kinerja dari perangkat lunak

|  |  |
| --- | --- |
| **Device** | **Specification** |
| Processor | Intel inside core i5 4570 |
| Memory | HDD 1TB |
| Operating System | Windows 10 Pro 64 bit |
| RAM | 8GB |

Dengan spesifikasi komputer di atas, dengan melakukan pengujian sebanyak 10 kali dan diambil rata-ratanya dengan performance testing menggunakan syntax yang dimiliki java yaitu **System.currentTimeMillis().** Didapatkan Hasil sebagai berikut ini

|  |  |  |
| --- | --- | --- |
| **No** | **Pengujian** | **Waktu** |
| 1 | Login | 0,6 detik |
| 2 | List User | 2,5 detik |
| 3 | Create User | 6,3 detik |
| 4 | Delete User | 0,8 detik |
| 5 | Edit all User | 5,1 detik |
| 6 | Edit username | 4,6 detik |
| 7 | Edit Password | 4,5 detik |

**2.5 Boundary Testing**

Testing dengan cara membagi domain-domain input dari suatu program ke dalam kelompok-kelompok data kemudianmelakukan pengujian hanya pada batas-batas domain input. Yang tujuannya adalah untuk konsentrasi upaya pengujian pada daerah rawan kesalahan dalam akurasi penentuan batasan-batasan kondisi (misalnya, programmer dapat menetapkan >, bila status persyaratan > atau =) .Berikut ini adalah tabel-tabel boundary testing dari sistem penjadwalan pertemuan(meeting scheduler).

Create Meeting Duration <=10

|  |  |
| --- | --- |
| **Sample Data** | **Result** |
| 80 | *Gagal* |
| 10 | Berhasil |
| 11 | *Gagal* |
| 3 | Berhasil |

Create-meeting Proposed Date Range 01-9999

|  |  |
| --- | --- |
| **Sample Data** | **Result** |
| 06/01/1990 | Berhasil |
| 04/01/2000 | Berhasil |
| 11/01/4000 | Berhasil |
| 01/01/9998 | Berhasil |

Add user name panjang minimal 6

|  |  |
| --- | --- |
| **Sample Data** | **Result** |
| asdfsaffsfsaf | Berhasil |
| asdfsfasfasfa | Berhasil |
| asd | *Gagal* |
| asdf | *Gagal* |

Add password panjang minimal 6

|  |  |
| --- | --- |
| **Sample Data** | **Result** |
| qwerasdf | Berhasil |
| qwerty | Berhasil |
| qwertyuiop | Berhasil |
| qw | *Gagal* |

**Usulan Perbaikan**

* Lebih baik setelah mengetik list-user, print command yang dapat dipakai
* Setelah membuat meeting, lebih baik memberikan ID meetingnya agar user dapat dengan jelas melihat
* Saat melakukan renegotiate meeting, jika jadwal meeting tidak ada program menampilkan null. Lebih baik apabila program memberikan pemberitahuan yang jelas dengan kalimat yang singkat dan padat.
* Pada fitur login sebaiknya langsung membaca email dan password yang dibaca agar user mengetahui role-nya (admin, initiator, atau participant).