

**Mitrais *User Interface Test Automation Framework* dan *Performance Test* Terhadap Aplikasi Rekam Medis**

**PT MITRAIS**

**Jalan Sidobali, Umbulharjo, Kota Yogyakarta, Daerah Istimewa Yogyakarta**

**Periode: 22 Juni 2015 – 21 Agustus 2015**

Oleh:

|  |  |
| --- | --- |
| Hamdi Ahmadi Muzakkiy | 5112100091 |
|  |  |

Pembimbing Jurusan

Prof. Ir.Supeno Djanali, M.Sc.,Ph.D

Pembimbing Lapangan

Achmad Afriadi Alamsyah ST. MTI.

JURUSAN TEKNIK INFORMATIKA

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya 2015

KERJA PRAKTIK – KI141330

*[Halaman ini sengaja dikosongkan]*

**

**Mitrais *User Interface Test Automation Framework* dan *Performance Test* Terhadap Aplikasi Rekam Medis**

**PT MITRAIS**

**Jalan Sidobali, Umbulharjo, Kota Yogyakarta, Daerah Istimewa Yogyakarta**

**Periode: 22 Juni 2015 – 21 Agustus 2015**

Oleh:

|  |  |
| --- | --- |
| Hamdi Ahmadi Muzakkiy | 5112100091 |
|  |  |

Pembimbing Jurusan

Prof. Ir.Supeno Djanali, M.Sc.,Ph.D

Pembimbing Lapangan

Achmad Afriadi Alamsyah ST. MTI.

JURUSAN TEKNIK INFORMATIKA

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya 2015

KERJA PRAKTIK – KI141330

*[Halaman ini sengaja dikosongkan]*

# LEMBAR PENGESAHAN

**KERJA PRAKTIK**

**Mitrais *User Interface Test Automation Framework* dan *Performance Test* Terhadap Aplikasi Rekam Medis**

Oleh:

|  |  |
| --- | --- |
| **Hamdi Ahmadi Muzakkiy** | 5112100091 |
|  |  |

Disetujui oleh Pembimbing Kerja Praktik:

|  |  |
| --- | --- |
| 1. Prof. Ir.Supeno Djanali, M.Sc.,Ph.D NIP. 130368610 | ................................ Prof. Ir.Supeno Djanali, M.Sc.,Ph.D |
|  |  |
| 1. Achmad Afriadi Alamsyah ST. MTI. NIP. | ................................ Achmad Afriadi Alamsyah ST. MTI. |

**SURABAYA**

**<<Agustus, 2015>>**

*[Halaman ini sengaja dikosongkan]*

**Mitrais *User Interface Test Automation Framework* dan *Performance test* terhadap aplikasi rekam medis**

**Nama Mahasiswa : Hamdi Ahmadi Muzakkiy**

**NRP : 5112100091**

**Jurusan : Teknik Informatika FTIf-ITS**

**Pembimbing Jurusan : Prof. Ir.Supeno Djanali, M.Sc.,Ph.D**

**Dosen Pembimbing II : Achmad Afriadi Alamsyah ST. MTI.**

# Abstrak

Mitrais *user interface test automation framework* adalah sebuah *framework* yang digunakan oleh PT Mitrais dalam melakukan kegiatan *interface testing* terhadap aplikasi web yang akan di berikan kepada *client*. Mitrais *user interface test automation framework* ini dibagi kedalam beberapa bagian, yaitu *framework* dan *text editor*. Untuk *framework* sendiri berfugsi untuk pengujian terhadap aplikasi web yang akan diuji, sedangkan *text editor* berfungsi untuk memudahkan dalam penulisan perintah-perintah yang akan digunakan sebagai *input* dari *framework*. *Performance test* adalah sebuah kegiatan dimana aplikasi yang dikembangkan akan diuji dari sisi kinerja aplikasi tersebut, apakah kinerja aplikasi sudah memenuhi *standard* yang diberikan atau tidak, jika belum memenuhi atau ada beberapa yang tidak sesuai, maka aplikasi tersebut akan diberikan kepada *developer* untuk dilakukan optimasi.

Dalam implementasi mitrais *user interface test automation framework,* *developer* menggunakan bahasa pemrograman java sebagai dasar dari pembuatan *framework* dan beberapa *dependency* seperti selenium dan cucumber. Sedangkan untuk *text editor*, diimplementasikan sebagai aplikasi web, menggunakan bahasa pemrograman HTML, javascript, dan CSS. *Performance test* dilakukan menggunakan jmeter dan dengan bantuan beberapa *tools* seperti sql server, sql profiler dan java mission control. Berdasarkan hasil uji coba dan evaluasi menunjukkan bahwa mitrais *user interface test automation framework* berhasil memenuhi semua kebutuhan yang telah ditetapkan, dan *performance test* terhadap aplikasi rekam medis telah selesai dilaksanakan.

***Kata kunci: User Interface Automation Test, Performance test, Text editor, User Interface Automation Test Framework, Text Editor***

*[Halaman ini sengaja dikosongkan]*

# KATA PENGANTAR

Puji dan syukur penulis hantarkan ke kehadirat Allah Subhanahu Wa Ta’alla atas rahmat dan karuniaNya, penulis dapat menyelesaikan rangkaian Kerja Praktek di PT. Mitrais yang dimulai pada tanggal 22 Juni 2015 hingga 21 Agustus 2015, serta dapat menyelesaikan laporan Kerja Praktek sesuai jadwal.

Kerja Praktek di Jurusan Teknik Informatika, Institut Teknologi Sepuluh Nopember Surabaya merupakan salah satu kegiatan wajib yang harus dilaksanakan oleh mahasiswa yang telah menempuh minimal 90 SKS. Penerapan kerja praktek ini sendiri pada hakikatnya adalah mengajarkan kepada kita untuk menghadapi situasi sebenarnya yang tidak beda jauh dengan kondisi disaat kita bekerja nantinya dan juga diharapkan dapat memberikan manfaat bagi pihak Mitrais.

Pelaksanaan kerja praktek ini dapat dilaksanakan sampai selesai berkat bimbingan dan saran dari berbagai pihak. Pada kesempatan ini, kami ingin menyampaikan terima kasih kepada :

1. Allah Subhanahu Wa Ta’ala, karena tanpa ijin dari-Nya, penulis tidak akan mampu menyelesaikan Kerja Praktek ini.
2. Kedua orang tua penulis yang telah memberikan doa dan dukungannya.
3. Ibu Nanik Suciati, S.Kom, M.Kom, Dr.Eng. selaku Ketua Jurusan Teknik Informatika ITS.
4. Dosen wali penulis, Bapak Ridho Rahman Hariadi, S.Kom, M.Sc atas dukungan selama berada di Teknik Informatika.
5. Dosen pembimbing penulis, Bapak Prof. Ir. Supeno Djanali, M.Sc, Ph.D atas dukungan dan motivasinya saat sebelum sampai berakhirnya proses kerja praktek kami.
6. Bapak Radityo Anggoro, S.Kom, M.Sc selaku koordinator kerja praktek Jurusan Teknik Informatika ITS atas segala bimbingan dan arahannya selama ini yang sangat membantu kami untuk segera menyelesaikan kerja praktek ini.
7. Bapak Achmad Afriandi , dan Bapak Widodo Nugroho sebagai pembimbing kerja praktik.
8. Bapak Argo Negoro dan seluruh kru Mitrais untuk pengalaman, atmosfir, dan kesempatan bekerja sama yang sangat berharga.
9. Bapak dan Ibu dosen jurusan Teknik Informatika ITS yang telah mencurahkan semua ilmu yang sangat besar manfaatnya selama proses belajar.
10. Karyawan dan staf administrasi Teknik Informatika yang membantu dalam proses pengerjaan dan administrasi kerja praktek.
11. Rekan-rekan Teknik Informatika ITS angkatan 2012 atas segala dorongan semangat dan ilmu yang diberikan.
12. Teman-teman warga Teknik Informatika ITS.
13. Pihak-pihak lain yang tidak sempat penulis sebutkan, yang telah membantu dalam kelancaran pengerjaan kerja praktek kami.

Apabila terdapat kesalahan dan kekurangan dalam penulisan laporan kerja praktek ini, penulis mengharapkan kritik dan saran yang membangun sebagai perbaikan di masa akan datang dan kami meminta maaf yang sebesar-besarnya. Semoga laporan kerja praktek ini dapat memberikan manfaat dan dapat digunakan sebagaimana mestinya.

Surabaya, September 2015

Hamdi Ahmadi Muzakkiy

*[Halaman ini sengaja dikosongkan]*

# DAFTAR ISI

[LEMBAR PENGESAHAN v](#_Toc430390248)

[Abstrak vii](#_Toc430390249)

[KATA PENGANTAR x](#_Toc430390250)

[DAFTAR ISI xiii](#_Toc430390251)

[DAFTAR GAMBAR xvi](#_Toc430390252)

[DAFTAR TABEL xix](#_Toc430390253)

[DAFTAR KODE SUMBER xxi](#_Toc430390254)

[1 BAB I PENDAHULUAN 23](#_Toc430390255)

[1.1. Latar Belakang 23](#_Toc430390256)

[1.1.1. Mitrais *User Interface Test Automation Framework* 23](#_Toc430390257)

[1.1.2. *Performance test terhadap aplikasi rekam medis* 23](#_Toc430390258)

[1.2. Tujuan 24](#_Toc430390259)

[1.3. Manfaat 24](#_Toc430390260)

[1.4. Rumusan Permasalahan 25](#_Toc430390261)

[1.5. Lokasi dan Waktu Kerja Praktik 25](#_Toc430390262)

[1.6. Metodologi Kerja Praktik 25](#_Toc430390263)

[1.7. Sistematika Laporan 27](#_Toc430390264)

[2 BAB II PROFIL PERUSAHAAN 30](#_Toc430390265)

[2.1. Sejarah Perusahaan 30](#_Toc430390266)

[2.2. Visi dan Misi Perusahaan 31](#_Toc430390267)

[2.3. Struktur Organisasi 31](#_Toc430390268)

[2.4. Divisi R&D 32](#_Toc430390269)

[3 BAB III TINJAUAN PUSTAKA 34](#_Toc430390270)

[3.1. Tinjauan Pustaka Mitrais User Interface Test Automation Framework 34](#_Toc430390271)

[3.1.1. Scrum 34](#_Toc430390272)

[3.1.2. Scrum 34](#_Toc430390273)

[3.1.3. TFS ( *Team Foundation Server* ) 36](#_Toc430390274)

[3.1.4. Git 36](#_Toc430390275)

[3.1.5. Gherkin 37](#_Toc430390276)

[3.1.6. Cucumber 37](#_Toc430390277)

[3.1.7. Share Point 37](#_Toc430390278)

[3.1.8. Selenium 38](#_Toc430390279)

[3.1.9. Ace Js 38](#_Toc430390280)

[3.1.10. Eclipse 38](#_Toc430390281)

[3.1.11. Json 39](#_Toc430390282)

[3.2. Tinjauan Pustaka Performance test terhadap aplikasi rekam medis 39](#_Toc430390283)

[3.2.1. Vivifyscrum 39](#_Toc430390284)

[3.2.2. Jmeter 40](#_Toc430390285)

[3.2.3. Microsoft Sql Server 40](#_Toc430390286)

[3.2.4. Sql Profiler 41](#_Toc430390287)

[3.2.5. Java Mission Control 41](#_Toc430390288)

[3.3. Mitrais User Interface Test Automation 42](#_Toc430390289)

[4 BAB IV ANALISIS DAN PERANCANGAN SISTEM 43](#_Toc430390290)

[4.1. Mitrais User Interface Test Automation Framework 43](#_Toc430390291)

[4.1.1. Analisis Sistem 43](#_Toc430390292)

[4.1.2. Perancangan Sistem 44](#_Toc430390293)

[4.1.3. Pengerjaan Sistem 45](#_Toc430390294)

[4.1.4. Dokumentasi Sistem 46](#_Toc430390295)

[4.2. Performance Test Terhadap Aplikasi Rekam Medis 47](#_Toc430390296)

[4.2.1. Analisis Sistem 47](#_Toc430390297)

[4.2.2. *Testing* 47](#_Toc430390298)

[4.2.3. Pengerjaan *Testing* 50](#_Toc430390299)

[5 BAB V IMPLEMENTASI SISTEM 52](#_Toc430390300)

[5.1. Mitrais User Interface Testing Automation Framework 52](#_Toc430390301)

[5.1.1. Implementasi Antarmuka Pengguna 52](#_Toc430390302)

[5.1.2. Implementasi Fungsi 58](#_Toc430390303)

[5.2. Performance test terhadap aplikasi rekam medis 66](#_Toc430390304)

[5.2.1. Antarmuka Pengguna 66](#_Toc430390305)

[5.2.2. *Testing* 71](#_Toc430390306)

[6 BAB VI PENGUJIAN DAN EVALUASI 83](#_Toc430390307)

[6.1. Lingkungan Pengujian 83](#_Toc430390308)

[6.2. Skenario Pengujian 83](#_Toc430390309)

[6.3. Evaluasi Pengujian 83](#_Toc430390310)

[7 BAB VII KESIMPULAN DAN SARAN 85](#_Toc430390311)

[7.1. Kesimpulan 85](#_Toc430390312)

[7.2. Saran 85](#_Toc430390313)

[DAFTAR PUSTAKA 86](#_Toc430390314)

[LAMPIRAN 87](#_Toc430390315)

[BIODATA PENULIS 88](#_Toc430390316)

# DAFTAR GAMBAR

[Gambar 1. perancangan sistem 45](#_Toc430390412)

[Gambar 2.team foundatio server 46](#_Toc430390413)

[Gambar 3. fifivyscrum 51](#_Toc430390414)

[Gambar 4. halaman utama *text editor* 52](#_Toc430390415)

[Gambar 5. tooltip *save file* 53](#_Toc430390416)

[Gambar 6. *save file* 53](#_Toc430390417)

[Gambar 7. tooltip *load existing file* 54](#_Toc430390418)

[Gambar 8. *load existing file* 54](#_Toc430390419)

[Gambar 9. hasil *load existing file* 55](#_Toc430390420)

[Gambar 10. tooltip *load element keyword* 55](#_Toc430390421)

[Gambar 11. *load element keyword* 56](#_Toc430390422)

[Gambar 12. hasil *load element keyword* 56](#_Toc430390423)

[Gambar 13. *showing list keyword* 57](#_Toc430390424)

[Gambar 14. *drag and drop* 57](#_Toc430390425)

[Gambar 15. *autocomplete* 58](#_Toc430390426)

[Gambar 16. halaman *login* 67](#_Toc430390427)

[Gambar 17. halaman *home* 67](#_Toc430390428)

[Gambar 18. halaman *list patient* 68](#_Toc430390429)

[Gambar 19. halaman tambah pasien 68](#_Toc430390430)

[Gambar 20. halaman memilih tempat klinik 69](#_Toc430390431)

[Gambar 21. halaman ruang tunggu 69](#_Toc430390432)

[Gambar 22. halaman audit 70](#_Toc430390433)

[Gambar 23. halaman *report* 70](#_Toc430390434)

[Gambar 24. *file* jmx *home* *waiting room* 71](#_Toc430390435)

[Gambar 25. *file* jmx *end to end* *waiting room* 72](#_Toc430390436)

[Gambar 26. *file* jmx *search patient* 72](#_Toc430390437)

[Gambar 27. *file* jmx *prerequisite* *create patient* 73](#_Toc430390438)

[Gambar 28. *file* jmx *prerequisite* *make appointment* 73](#_Toc430390439)

[Gambar 29. *file* jmx *performance test* 74](#_Toc430390440)

[Gambar 30. *file* jmx *performace test Lt 25* 74](#_Toc430390441)

[Gambar 31. hasil *performance test* 76](#_Toc430390442)

[Gambar 32. hasil *performance test* 76](#_Toc430390443)

[Gambar 33. sql profiler 77](#_Toc430390444)

[Gambar 34. sql profiler 2 77](#_Toc430390445)

[Gambar 39. hasil *test* baseline 80](#_Toc430390446)

[Gambar 40. hasil *test* baseline + 30.000 pasien 80](#_Toc430390447)

[Gambar 41. hasil *test* baseline + 30.000 pasien + 30.000 *appointmen* 81](#_Toc430390448)

*[Halaman ini sengaja dikosongkan]*

# DAFTAR TABEL

*[Halaman ini sengaja dikosongkan]*

# DAFTAR KODE SUMBER

[Kode Sumber 1. fungsi IP 58](#_Toc430390449)

[Kode Sumber 2. fungsi saveFile 59](#_Toc430390450)

[Kode Sumber 3. fungsi downloadFile 59](#_Toc430390451)

[Kode Sumber 4. fungsi *save server* 60](#_Toc430390452)

[Kode Sumber 5. fungsi *setExistingFeature* 61](#_Toc430390453)

[Kode Sumber 6. *load element keyowrd* 62](#_Toc430390454)

[Kode Sumber 7. *load element keyword 2* 62](#_Toc430390455)

[Kode Sumber 8. *drag and drop* 63](#_Toc430390456)

[Kode Sumber 9. *autocomplete* 64](#_Toc430390457)

[Kode Sumber 10. *autocorrect* 65](#_Toc430390458)

[Kode Sumber 11. *autocorrect 2* 66](#_Toc430390459)

[Kode Sumber 12. *generate provider* 75](#_Toc430390460)

[Kode Sumber 13. *prosedur insert patient* 78](#_Toc430390461)

[Kode Sumber 14. *execute insert patient* 78](#_Toc430390462)

[Kode Sumber 15. *prosedur insert appointment* 79](#_Toc430390463)

[Kode Sumber 16. *execute insert appointment* 79](#_Toc430390464)

*[Halaman ini sengaja dikosongkan]*

# BAB I PENDAHULUAN

## Latar Belakang

Latar belakang ini dibagi menjadi dua bagian yakni mitrais *user interface test automation framework* dan *performance test terhadap aplikasi rekam medis*

### Mitrais *User Interface Test Automation Framework*

*User interface testing* adalah proses *testing* terhadap suatu antar muka sistem untuk memastikan semua kebutuhan terpenuhi dan memastikan bahwa sistem melakukan keluaran yang diharapkan. Contoh *user interfac test* seperti mengecek apakah suatu halaman ( dalam kasus ini aplikasi berbasis web ) sudah memuat atribut – atribut yang dibutuhkan. Bisanya untuk *user interface test* menggunakan banyak *test case* untuk mencoba interface yang di test.

Salah satu riset yang dilakukan pihak mitrais adalah pembuatan *framework* untuk *testing interface* suatu sistem, nama riset tersebut adalah MUIT ( Mitrais *User Interface Test* ). Tujuannya adalah membuat suatu *framework* yang akan digunakan untuk *testing*  *interface* sebelum sistem tersebut diberikan kepada *client*. Dalam riset ini, *framework* yang dibangun digunakan untuk menguji aplikasi berbasis web ( *web based* ).

Diharapkan dengan *user interface test automation framework* memudahkan dalam pengujian *user interface*, dan diharapkan produk yang dikirim ke *client*  sesuai dengan kebutuhan *client*. Pada kesempatan kerja praktik kali ini penulis diberitugas untuk melakukan menambahkan fungsionalitas Mitrais *user interface test framework* yang terdiri dari dua bagian yaitu bagian *user interface test automation framework* dan *text editor.*

### *Performance test terhadap aplikasi rekam medis*

*Performance test* adalah proses dimana pengujian terhadap kecepatan, efektivitas dari suatu sistem atau perangkat lunak yang telah dibuat. Salah satu contoh dari test ini adalah mengukur waktu respon dari suatu sistem atau perangkat lunak ketika perangkat lunak itu dijalankan. Nantinya atribut-atribut tersebut dianalisa apakah sesuai dengan yang diinginkan. Dari atribut tersebut bisa diketahui proses mana yang mengakibatkan perangkat lunak tersebut tidak sesuai dengan apa yang diinginkan. Ada beberapa kemungkinan yang dapat terjadi jika ada ketidak sesuaian tersebut, contohnya adalah adanya *query*  ke database yang membuat kinerja perangkat lunak melambat atau melakukan *request* berkali-kali ke *server*.

Kali ini, *testing* dilakukan terhadap *project* Mitrais dimana *project* tersebut dimiliki oleh client PT. Mitrais. *Client* menginginkan aplikasi berbasis web untuk rekam medis . Sistem yang akan dibangun adalah sebuah sistem informasi yang digunakan rumah sakit di Australia yang akan menghubungkan dokter, pegawai rumah sakit dengan pasien.

*Performance test* kali ini menguji apakah sistem sudah sesuai dengan apa yang diharapkan, seperti kesusuaian data yang dimasukkan atau diambil, kecepatan respon sistem, dan keslahan yang mungkin terjadi. *Test* ini dilakukan menggunakan jmeter, dimana jmeter mempunyai fitur untuk melakukan *test* menggunakanlebih dari satu *user*. Selain itu data yang ada pada database diperbesar, menggunakan metode *inject sql*  pada sql server.

## Tujuan

* Mengembangkan *framework* dan *texteditor* untuk melakukan *user interface tesing*
* Membuat dokumentasi dari yang telah dikerjakan untuk mempermudah pemakaian atau pengembangan selanjutnya
* Melakukan *testing* dari aplikasi rekam medis
* Menemukan kesalahan – kesalahan yang mungkin terjadi ketika pada sistem

## Manfaat

Manfaat yang diperoleh antara lain :

* Mempermudah kegiatan *testing user interface*
* Mempermudah dalam pembuatan *user interface testing* dengan menggunakan *text editor* yang dibuat.
* Mempermudah pengembangan *framework* selanjutnya
* Sistem yang telah dibuat siap untuk diberikan kepada *client* karna sudah dilakukan *performance* *testing*.
* Menemukan kekurangan dari aplikasi rekam medis yang di *test*
* Memberikan laporan apa saja yang tidak sesuai dengan apa yang diharapkan

## Rumusan Permasalahan

* Bagaimana mengembangkan *text editor* berbasis web yang digunakan sebagai media menuliskan perintah untuk melakukan *testing*?
* Bagaimana melakukan *performance testing* ?
* Bagaimana cara membuat laporan untuk *performance testing* ?
* Bagaimana menuliskan dokumentasi dari aplikasi yang sudah dikembangkan ?

## Lokasi dan Waktu Kerja Praktik

Lokasi kerja praktik berada di Jl. Sidobali, Umbulharjo, Kota Yogyakarta, Daerah Istimewa Yogyakarta. Untuk tanggal kerja praktik dimulai pada tanggal 22 juni 2015 hingga 21 agustus 2015 ( 9 minggu ). Diwajibkan masuk setiap hari senin hingga jum’at dari jam 07.00 – 16.00 ( durasi waktu 8 jam kerja ).

## Metodologi Kerja Praktik

1. Perumusan Masalah

Pada tahap perumusan masalah penulis menemukan titik pokok hal yang harus penulis kerjakan, untuk kerja praktik di Mitrais, yaitu “Bagaimana meningkatkan aplikasi *User Interface Test Automation Framework* dan melakukan *performance test* terhadap aplikasi rekam medis” sehingga pada tahap selanjutnya penulis harus menjawab pertanyaan tersebut.

1. Studi Literatur

Pada tahap studi literatur penulis melakukan pembelajaran terhadap *user interface test automation framework* milik mitrais, menggunakan bahasa pemrograman java, HTML, javascript, dan CSS. Melakukan pembelajaran terhadap *tools* jmeter, sql server, sql profiler, java mission controller. Hal ini dilakukan untuk menunjang pengetahuan untuk meningkatkan aplikasi *user interface test automation framework* dan *performance test* untuk aplikasi rekam medis.

1. Analisis Sistem

Pada tahap ini dilakukan analisis sistem untuk *user interface test automation framework* dan analisis sistem aplikasi rekam medis, analisis dilakukan menggunakan dokumentasi yang telah dituliskan sebelumya. Analisis sistem untuk *user interface test automation framework* dilakukan dengan membaca dokumen dari *developer* sebelumnya dan membaca code dari *framework* tersebut, sedangkan analisis sistem untuk *performance test* dilakukan dengan menganalisa kerja dari aplikasi rekam medis tersebut

1. Implementasi Sistem

Implementasi merupakan tahap membangun rancangan tahap yang telah ada. Pada tahapan ini merealisasikan apa yang terdapat pada tahapan sebelumnya, sehingga sesuai dengan apa yang telah direncanakan.

1. Pengujian dan Evaluasi

Pada tahapan ini dilakukan uji coba pada aplikasi yang telah diimplementasi. Pengujian dan evaluasi dilakukan dengan menggunakan *testing* web aplikasi sederhana. Tahapan ini bermaksud untuk mengevaluasi kesesuaian sistem dan aplikasi yang dibuat apakah dapat dilakukan dengan lancar atau tidak. Selain itu juga untuk mencari masalah yang mungkin timbul dan tidak lupa mengadakan perbaikan jika terdapat kesalahan

1. Kesimpulan dan Saran

Mitrais *user interface test automation framework* ini telah memenuhi syarat yang diinginkan, dari *framework* dan *text editor* telah berjalan dengan baik dan lancar. Untuk *performance test* telah dilakukan sesuai dengan prosedur yang ada, hasil yang dilakukan diberikan kepada tim pengembang aplikasi yang di ujikan.

## Sistematika Laporan

Laporan kerja praktik ini terbagi menjadi 7 bab dengan rincian sebagai berikut :

* **BAB 1 : PENDAHULUAN**

Berisi pendahuluan yang terdiri dari latar belakang, tujuan, manfaat, rumusan permasalahan, lokasi dan waktu kerja praktik, metodologi kerja praktik, dan sistematika penulisan.

* **BAB 2 : PROFIL PERUSAHAAN**

Berisi tentang gambaran umum dari tempat kerja praktik yaitu PT Mitrais, mulai dari sejarah perusahaan, visi dan misi perusahaan, struktur organisasi, dan divisi tempat kerja praktik dilakukan.

* **BAB 3 : TINJAUAN PUSTAKA**

Bab ini berisi dasar teori dari metode/teknologi yang digunakan dalam meyelesaikan proyek kerja praktik.

* **BAB 4 : ANALISIS SISTEM**

Pada bab ini dijelaskan mengenai desain antarmuka aplikasi serta yang digunakan dalam aplikasi.

* **BAB 5 : IMPLEMENTASI SISTEM**

Pada bab ini berisi uraian tahap-tahap yang dilakukan untuk proses implementasi aplikasi

* **BAB 6 : UJI COBA DAN EVALUASI**

Pada bab ini, dijelaska mengenai hasil uji coba dan evaluasi dari perangkat lunak yang telah dikembangkan selama pelaksanaan kerja praktik.

* **BAB 7 : KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dan saran dari proses pelaksanaan kerja praktik.

*[Halaman ini sengaja dikosongkan]*

# BAB II PROFIL PERUSAHAAN

## Sejarah Perusahaan

Mitrais berasal dari bahasa Sansekerta, Mitra, yang berarti teman atau rekan. Kami memilih Bahasa Sansekerta - yang merupakan bahasa Hindu kuno - sebagai nama perusahaan karena karena peran dan pengaruhnya dalam sejarah berbagai bahasa dan kebudayaan di Asia Tenggara, wilayah yang merupakan rumah bagi Mitrais.

Nama tersebut mencerminkan filosofi yang dianut Mitrais; yakni klien sebagai partner kerja dalam jangka panjang.

Bali merupakan development center pertama Mitrais. Selain sebagai tempat tujuan wisata, Bali juga menarik bagi perusahaan yang ingin membentuk captive team untuk memenuhi kebutuhan software development. Mereka, layaknya para turis yang mengunjungi Bali, tertarik dengan keindahan, keamanan dan biaya yang murah.

Mitrais bekerjasama dengan beberapa perusahaan ternama untuk memberikan pelayanan terbaik bagi kliennya.

Melalui kerjasamanya dengan Mincom pada masa lalu, Mitrais merupakan partner terlama ABB. Mitrais adalah partner bisnis yang menjual software pertambangan dari ABB serta merupakan partner dalam pengembangan software mereka.

Mitrais tergabung dalam Microsoft Partner Network dengan kompetensi Gold Application Development yang mencerminkan kompetensi dan keahlian tertinggi dalam teknologi Microsoft dan hubungan kerja terdekat dengan Microsoft.

Mitrais merupakan partner MuleSoft yang tersertifikasi, siap untuk menyediakan layanan integrasi dengan Mule ESB, ESB yang paling banyak diinstal di dunia.

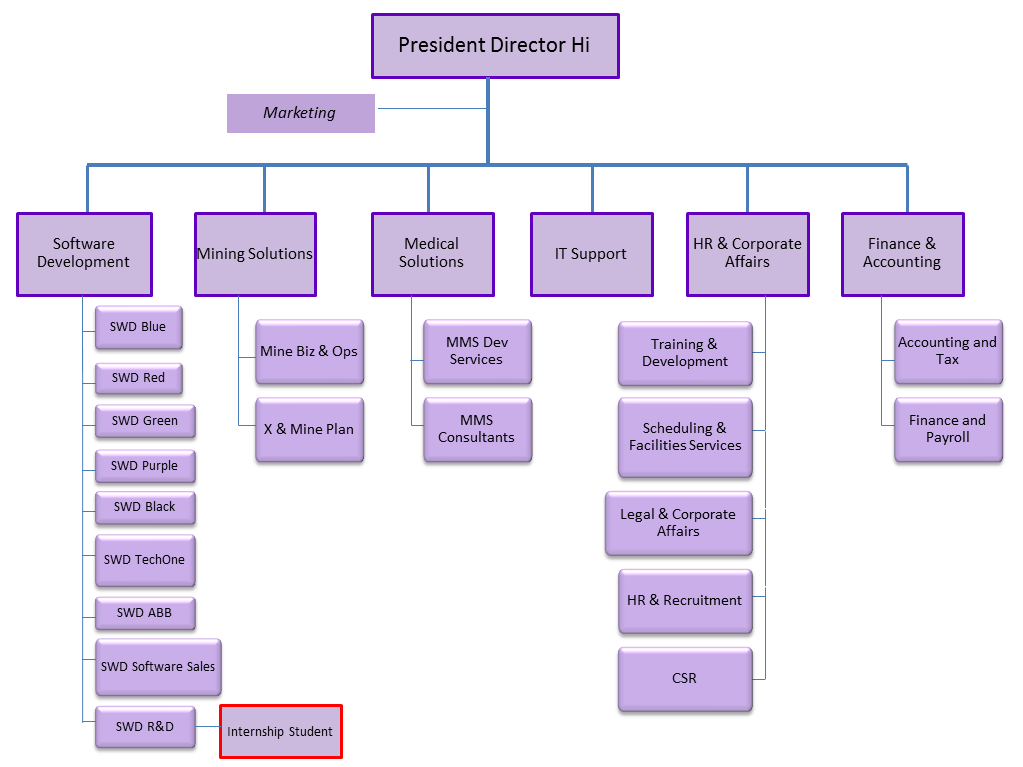
Mitrais bekerjasama dengan JasperSoft menyediakan Jaspersoft Business Intelligence Suite yang digunakan untuk mengolah data mentah menjadi informasi yang berguna bagi bisnis perusahaan.

## Visi dan Misi Perusahaan

“To build term, high trust relationships with our client as we provide products and services which add value to their business.”

* A company we can be proud to work for
* Adding value to society
* Flexible and able to adapt to a changing world
* Understanding and adding value to our clients
* Competent in what we do
* A leader in the fields we engage
* A profitable and efficient company

## Struktur Organisasi



## Divisi R&D

Untuk kegiatan *internship* berada dibawah departemen SWD R&D ( *software development Research and development* ). R&D sendiri bertugas untuk melakukan riset dan pengembangan. Segala bentuk project internal yang dilakukan oleh pihak mitrais akan diserahkan kepada bagian R&D, jadi R&D ini hanya bertugas untuk melakukan riset dan pengembangan aplikasi iternal yang akan digunakan untuk keperluan pihak mitrais dalam membangun proyek yang diberikan dari *client*.

*[Halaman ini sengaja dikosongkan]*

# BAB III TINJAUAN PUSTAKA

## Tinjauan Pustaka Mitrais User Interface Test Automation Framework

### Scrum

Agile *Development Methods* adalah sekelompok metodologi pengembangan perangkat lunak yang didasarkan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun. Agile *development methods* merupakan salah satu dari Metodologi pengembangan perangkat lunak yang digunakan dalam pengembangan perangkat lunak. Agile memiliki pengertian bersifat cepat, ringan, bebas bergerak, dan waspada Sehingga saat membuat perangkat lunak dengan menggunakan agile *development methods* diperlukan inovasi dan responsibiliti yang baik antara tim pengembang dan klien agar kualitas dari perangkat lunak yang dihasilkan bagus dan kelincahan dari tim seimbang.

### Scrum

Scrum adalah suatu *framework* yang bersifat iteratif dan *incremental* dalam manajemen proyek yang berbasis agile *software development* (salah satu tipe *software engineering*). Fokus utama Scrum adalah manajemen pengembangan produk.

Sekitar tahun 1986 Hirotaka Takeuchi dan Ikujiro Nonaka mengembakan suatu solusi dalam pengembangan produk komersial yang dapat meningkatkan kecepatan dan flesibilitas dimana ini didasarkan pada studi kasus industri manufacturing, automative, komputer, photocopier dan printer. Metode ini mereka sebut dengan pendekatan holistic atau rugby.

Sedangkan DeGrace dan Stahl pertama kali memperkenalkan istilah Scrum pada metodologi rugby pada tahun 1991. Istilah metodologi Scrum diperkenalkan pada workshop Business Object Design and Implementation yang merupakan bagian OOPSLA pada tahun 1995 dimana Sutherland dan Schwaber membawakan makalah mengenai metodologi Scrum.

Beberapa orang istilah Scrum kadang ditulis dengan SCRUM (semua huruf besar) walaupun SCRUM bukanlah suatu singkatan tertentu.

Scrum Roles

Pada implementasi metodologi Scrum pembagian team dibagi menjadi tiga role yaitu :

• Product Owner

• ScrumMaster

• Team

Product owner mewakili suara customer dan bertanggungjawab terhadap team yang akan mengimplementasi dari requirement ke implementasi. Product owner biasanya menulis daftar fitur produk berdasarkan diskusi dalam model user story dan memperioritaskan daftar fitur yang dimasukkan kedalam product backlog. Satu team Scrum akan mempunyai satu product owner dan juga anggota team development. Direkomendasikan bahwa role product owner tidak digabungkan dengan role ScrumMaster.

ScrumMaster bertugas untuk membawa team dari hambatan-hambatan dalam pengembangan produk. ScrumMaster bertanggungjawab atas kemajuan pengembangan produk.

Team yang bertanggungjawab dalam realisasi produk jadinya. Biasanya satu team terdiri sampai 5-9 orang dengan ketrampilan yang dimiliki bervariasi yaitu analisa, desain, develop, test, technical communication hingga dokumentasi. Setiap anggota team dituntut untuk bekerja sendiri dan mengatur manajemen sendiri dalam koridor dalam satu team.

### TFS ( *Team Foundation Server* )

Team Foundation Server (TFS) merupakan produk berbasis produk server yang ditujukkan untuk programmer, tester, architect, project manager, business analysis dan orang yang ingin berkontribusi dalam pengembangan perangkat lunak.

Secara umum TFS terdiri dari dua bagian yang dapat diimplementasi dalam satu atau lebih mesin komputer. Dua bagian ini meliputi

• Application Tier. Ini merupakan kumpulan web service yang mana mesin client dapat berkomunikasi pada protokol web service.

• Data Tier. Ini terdiri dari database SQL Server yang menyimpan logic database dari TFS. Data ini termasuk data reporting. Seluruh data disimpan kedalam database termasuk kode program.

Team foundation server ini digunakan dalam project sebagai komunikasi menganai *task* apa saja yang harus dilakukan.

### Git

Apa itu git? git adalah sebuah system versioning control (sistem kontrol versi) terdistribusi, ada istilah lain dari git ini adalah source code management software (aplikasi manajemen kode sumber), atau menurut wikipedia git adalah perangkat lunak pengontrol versi atau proyek manajemen kode perangkat lunak yang diciptakan oleh Linus Torvalds, yang pada awalnya ditujukan untuk pengembangan kernel Linux.

### Gherkin

Gherkin adalah bahasa yang dimengerti oleh cucumber. Gherkin digunakan oleh cucumber untuk mendeskripsikan tingkah laku suatu perangkat lunak tanpa memberi detail bagaimana tingkah laku tersebut diimplementasikan

Gherkin mempunyai dua tujuan, dokumentasi dan *test* automasi. Untuk bahasanya sendiri, tidak seperti syntax yang bisa dituliskan oleh programmer. Bahasa gherkin sendiri dibuat dengan bahasa sehari-hari yang mudah dipahami

### Cucumber

Cucumber adalah *tools* yang digunakan *programmers* untuk testing suatu *software*. Cucumber menjalankan *automated test* yang ditulis menggunakan bahasa gherkin. Cucumber dibuat menggunakan bahasa pemrograman Ruby.

### Share Point

SharePoint adalah sebuah platform sistem manajemen dokumen berbasis web yang diciptakan oleh Microsoft Corporation. SharePoint dapat digunakan untuk menjalankan situs-situs web yang terdiri atas ruangan kerja (shared workspace) dan dokumen yang digunakan secara bersama-sama (shared documents).

Microsoft Sharepoint 2010 memungkinkan individu dalam suatu perusahaan dengan mudah berintegrasi satu sama lain. Sharepoint memungkinkan informasi disimpan dan dikelola secara terpusat serta menyederhanakan proses komunikasi dalam sebuah perusahaan. Sharepont memiliki beberapa kegunaan yaitu :

1. Mempermudah pembuatan website terintegrasi.

2. Pengelolaan informasi secara efisien.

3. Memfasilitasi kolaborasi atau kerja sama tim.

4. Mempermudah komunikasi.

5. Otomatisasi proses bisnis

6. Pengolahan laporan.

7. Terintegrasi

### Selenium

Selenium adalah *brwoser automation tool*, biasanya digunakan untuk menuliskan *end-to-end* aplikasi berbasis web. Selenium ini berupa API yang akan diintegrasikan dengan bahasa pemgrogaman java sehingga kegiatan *UI test automation*  bisa dilakukan.

### Ace Js

Ace adalah *code editor* yang berdiri sendiri menggunakan bahasa javascript. Tujuannya adalah membuat suatu *editor* berbasis web yang cocok dan bisa dikembangkan dari segi fitur, penggunaan, dan performa dari text editor yang sudah ada seperti TextMate, Vim atau Eclipse. Ace bisa dengan mudah tanamkan ke halaman atau aplikasi berbasis javascript.

### Eclipse

Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent). Berikut ini adalah sifat dari Eclipse:

* Multi-platform : Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
* Mulit-language : Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
* Multi-role : Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan open source, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plug-in

### Json

JSON adalah kependekan dari Javascript Object Notation. JSON dipakai untuk dua hal; yang pertama untuk format transfer data antar device/OS/bahasa pemerograman yang berbeda dan sebagai media penyimpanan data seperti MongoDB. JSON dalam artikel ini lebih menekankan pada transfer data. Sebenarnya selain JSON, format transfer data antar sistem bisa dilakukan XML, plaintext ataupun HTML, namun 3 format tersebut punya banyak kekurangan, terutama dari sisi ukuran. Saingant terdekat dari JSON adalah XML, dalam layanan Web API atau webservice, sebagian layanan menggunakan JSON, sebagian yang lain menggunakan format XML. Dua format ini bisa saling menggantikan.

## Tinjauan Pustaka Performance test terhadap aplikasi rekam medis

### Vivifyscrum

Vivifyscrum adalah sebuah website yang fungsinya sama seperti TFS, vivifyscrum digunakan untuk komunikasi mengenai *task* yang akan dikerjakan. Vivifyscrum digunakan untuk mengimplementasikan metode scrum, untuk memberikan kemudahan dan kejelasaan mengenai kegiatan apa saja yang harus dikerjakan. Sehingga dalam pengerjaan suatu project akan lebih cepat dan jelas.

### Jmeter

Apache JMeter adalah perangkat lunak open-source, murni 100% aplikasi Java yang dirancang untuk melakukan uji fungsional dan mengukur kinerja suatu server perangkat lunak (seperti aplikasi web).Dan perlu juga diketahui bahwa JMeter bukanlah browser. Dari sisi web-service mungkin Jmeter terlihat seperti browser (atau lebih tepatnya beberapa browser) , namun sesungguhnya JMeter tidak melakukan semua tindakan layaknya browser pada umumnya. Secara

khusus, JMeter tidak mengeksekusi Java script yang ditemukan di HTML. Juga tidak merender halaman HTML sebagaimana layaknya browser.

### Microsoft Sql Server

Microsoft SQL Server merupakan produk RDBMS (Relational Database Management System) yang dibuat oleh Microsoft. Orang sering menyebutnya dengan SQL Server saja. Microsoft SQL Server juga mendukung SQL sebagai bahasa untuk memproses query ke dalam database. Microsoft SQL Server Mirosoft SQL Server banyak digunakan pada dunia bisnis, pendidikan atau juga pemerintahan sebagai solusi database atau penyimpanan data. Pada tahun 2000 Microsoft mengeluarkan SQL Server 2000 yang merupakan versi yang banyak digunakan. Berikut ini adalah beberapa fitur yang dari sekian banyak fitur yang ada pada SQL Server 2000 [Rado05]:

XML Support. Dengan fitur ini, Anda bisa menyimpan dokumen XML dalam suatu tabel, meng-query data ke dalam format XML melalui Transact-SQL dan lain sebagainya.

Multi-Instance Support. Fitur ini memungkinkan Anda untuk menjalankan beberapa database engine SQL Server pada mesin yang sama.

Data Warehousing and Business Intelligence (BI) Improvements. SQL Server dilengkapi dengan fungsi-fungsi untuk keperluan Business Intelligence melalui Analysis Services. Selain itu, SQL Server 2000 juga ditambahi dengan tools untuk keperluan data mining.

Performance and Scalability Improvements. SQL Server menerapkan distributed partitioned views yang memungkinkan untuk membagi workload ke beberapa server sekaligus. Peningkatan lainnya juga dicapai di sisi DBCC, indexed view, dan index reorganization.

Query Analyzer Improvements. Fitur yang dihadirkan antara lain: integrated debugger, object browser, dan fasilitas object search.

DTS Enhancement. Fasilitas ini sekarang sudah mampu untuk memperhatikan primary key dan foreign key constraints. Ini berguna pada saat migrasi tabel dari RDBMS lain.

Transact-SQL Enhancements. Salah satu peningkatan disini adalah T-SQL sudah mendukung UDF (User-Definable Function). Ini memungkinkan Anda untuk menyimpan rutin-rutin ke dalam database enginen.

### Sql Profiler

Sql profiler adalah sebuah interface yang digunakan untuk membuat, memanajemen, menganalisis dan membalas hasil dari sql yang berjalan. *Event* akan di simpan pada *trace file* dimana nantinya bisa dianalisis atau digunakan untuk mendiagnosa suatu masalah. Setiap *event* yang terjadi pada sql server akan dicatat.

### Java Mission Control

Java mission control adalah sebuah set plugin untuk Eclipse IDE yang didesign untuk membantu pengembangan, profile, dan diagnosa aplikasi yang berjalan pada java. Java mission control meliputi *tools* ke monitor, manajemen, dan profile aplikasi berbasis java.

## Mitrais User Interface Test Automation

*Framework* yang dipakai adalah “ Mitrais *User Interaface Test Automation* *Framework*” *framewok* ini dikembangkan sendiri oleh pihak mitrais. Framework ini digunakan untuk melakukan *user interface* *testing* terhadap suatu aplikasi web. Aplikasi ini berbasis java yang menggunakan selenium untuk membuat *testing* bisa berjalan paralel dan dapat berjalan di banyak *environtment*.

Bahasa yang digunakan untuk membangun *framework* ini adalah java. Dengan beberapa *dependency* seperti selenium untuk menjalankan *testing* secara otomatis, dan cucumber yang digunakan untuk melakukan pengecekan mengenai kesusuaian dari *interface* web tersebut.

# BAB IV ANALISIS DAN PERANCANGAN SISTEM

## Mitrais User Interface Test Automation Framework

### Analisis Sistem

#### Deskripsi Umum Sistem

Mitrais *user interface test automation framework* terbagi menjadi dua aplikasi, yaitu *framework* dan *text editor,* yang lebih ditekankan oleh penulis adalah aplikasi *text editor* yang dibuat. Aplikasi *text editor* yang dikembangkan berbasis web ( *web based )*. *Text editor* hanya memiliki satu halaman yang akan digunakan sebagai tempat untuk melakukan kegiatan pembuatan *syntax* *testing*. Aplikasi ini berbasis *server client,* dimana *back end* pada *server* menggunakan node js, sedangkan untuk *front end* menggunakan ace js. *Text editor* yang ada belum memiliki fitur-fitur standard seperti *save file, load existing file, autocorrect*. Oleh kerena itu dilakukan penambahan fungsionalitas yang ada pada *text editor* yang ada.

#### Fitur tambahan pada sistem

Fitur tambahan yang diinginkan pada *text editor* adalah :

* *Save file*

pada fitur ini, diharapkan hasil pekerjaan yang dilakukan pada *text editor* bisa disimpan kedalam lokal file *user*.

* *Load* existing file

Membuka kembali file yang sudah pernah dibuat untuk diedit pada *text editor*. File yang bisa dibuka adalah file ber ekstensi .feature

* *Load element* word

Membaca file yang berisi *list element keyword* yang nantinya akan ditampilkan di *sidebar* dan *autocomplete*. File tersebut berupa file json.

* *Showing list keyword*

Menampilkan *list element keyword* yang telah di *load* sebelumnya, disini *list* yang telah ditampilkan dapat dilakukan *drag and drop* pada kata yang dipilih.

* *Drag and drop side bar menu ( list keyword, element, gherkin )*

Pada sidebar yang berisi *keyword, element, gherkin* mempunyai kemampuan untuk “*drag and drop*” kedalam *text editor*. Tujuannya adalah mempercepat kinerja dari penulisan yang akan *dilakukan testing.*

* *Autocomplete*

Pada saat melakukan *load* file *keyword*, diinginkan adanya penambahan autocomplete sesuai dengan isi file yang di *load* kedalam aplikasi tersebut.

* *Autocorrect*

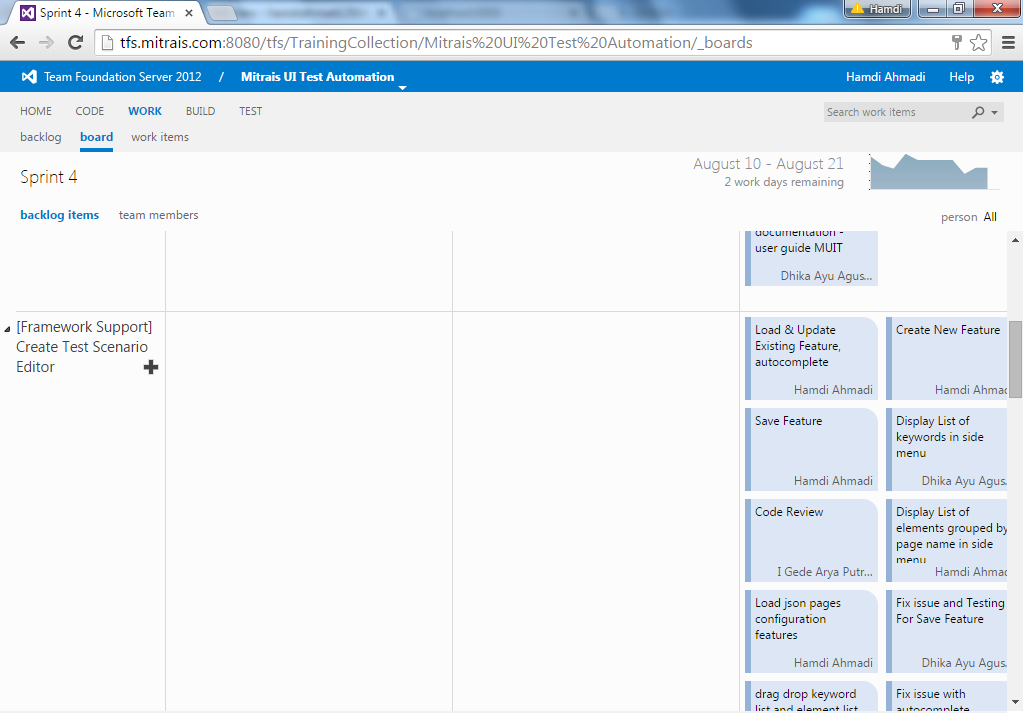
Fitur *autocomplete* yang ada belum memenuhi kriteria yang diinginkan dari pihak mitrais, pada fitur *autocomplete*, diinginkan adanya kemampuan untuk menghapus beberapa kata yang dituliskan ketika *autocomplete* terjadi.

### Perancangan Sistem

Untuk perancangan sistem, semua fitur akan dilakukan pada sisi *client* kecuali untuk fitur *save file*. Pada fitur tersebut, isi dari *text editor* akan dikirimkan ke *server,* *server* akan membuatkan *file* didalam suatu folder, dan akan dikirimkan kembali oleh *server* berbentuk file berextensi .feature. Sedangkan fitur lainnya akan ditangani oleh javascript yang berada pada sisi *client*.

Gambar 1. perancangan sistem

### Pengerjaan Sistem

 Pengerjaan sistem dilakukan menggunakan scrum yang mengimplementasikan metode agile. Disini dalam 1 tim bebas mengerjakan apapun pekerjaan yang sudah dituliskan didalam TFS ( *Team Foundation Server* ). Tiap *task* akan ada bobot lama pengerjaan, sehingga pada akhir *sprint* ( batas sub modul pada pengerjaan aplikasi ) bisa didapatkan kontribusi pengerjaan tiap orang. Disini penulis sebagian besar mengembangkan aplikasi *text editor*.

Gambar 2.team foundatio server

### Dokumentasi Sistem

Sistem yang sudah diselesaikan akan didokumentasikan, dimana tujan dari dokumentasi ini adalah membuat tim pengembang yang selanjutnya memegang aplikasi tersebut cepat untuk menyesuaikan dengan pengembang sebelumnya. Dokumentasi dletakan pada web internal pihak mitrais. Dokumentasi ini berisi penjelasan apa saja langah-langkah yang harus dilakukan untuk menggunakan aplikasi tersebut.

## Performance Test Terhadap Aplikasi Rekam Medis

### Analisis Sistem

#### Deskripsi Umum sistem

Aplikasi rekam medis adalah aplikasi berbasis web ( *web based* ) yang dibertujuan untuk melakukan pendataan kegiatan medis. Aplikasi ini menggunakan bahasa pemrograman java. Hak akses yang diberikan pada aplikasi ini adalah pasien, dokter, dan administrasi. Aplikasi ini memiliki beberapa fitur penting, yaitu membuat *appointmnet* dimana pasien bosa membuat janji dengan dokter, memilih tempat klinik karena satu dokter bisa bekerja lebih dari satu klinik.

Aplikasi ini tergolong besar, karena akan dipakai oleh banyak user dan banyak mengakses data dari database, namun aplikasi ini belum rilis oleh pihak mitrais. Hal ini dikarenakan belum selesainya *testing*. Adanya testing diharapkan dapat menemukan kesalahan jika memang ada agar ketika sudah di rilis tidak terjadi hal yang tidak diinginkan. Selain itu diharapkan adanya optimasi dengan menemukan beberapa hal yang membuat aplikasi tidak optimal.

### *Testing*

#### *Tools Testing*

*Testing* akan dilakukan menggunakan tools, tools yang digunakan adalah :

1. Jmeter

Digunakan untuk melakukan *testing* dengan cara melakukan *record* yang akan disimpan oleh jmeter. Jmeter akan menyimpan *request* dan *resoponse* dari tiap aksi yang dijalankan. Dari *record* yang sudah disimpan, dilakukan *update* terhadap nilai-nilai yang ada pada *variable* *request* , hal tersebut dilakukan untuk mengatur *user*, parameter atau data *input* yang dimasukkan. Selanjutnya bisa menyisipkan *code* kedalam jmeter yang sudah di *record*. Hal ini dilakukan jika menggunakan lebih dari satu *user*, diinginkan adanya perbedaan data yang digunakan ketika menjalankan jmeter.

1. Sql Server

Sql server digunakan untuk melakukan *inject* data kedalam *database*. Memasukkan data pasien dan *appointment* kedalam *database* dengan jumlah besar. Hal ini dimaksudkan agar *testing* bisa berjalan dengan optimal.

1. Sql Profiler

Karena pada saat testing tidak diberikan *code* yang digunakan untuk membuat aplikasi rekam medis, maka digunakan sql profiler. *Tools* ini digunakan untuk menganalisa interaksi kedalam *database*. Parameter apa saja yang digunakan untuk memasukkan data ketika terjadi pembuatan pasien baru dan membuat *appointment* baru.

1. Java Misson Control

Aplikasi berbasis java ini digunakan untuk melihat *performance* web yang dikembangkan. Dilihat bagaimana kinerja sistem tersebut ketika jmeter dijalankan.

#### Alur Kerja *Testing*

Alur kerja *testing* terjadi dari beberapa tahap, tahapan tersebut adalah:

*Testing* akan dilakukan menggunakan tools, tools yang digunakan adalah :

1. Melakukan *record* jmeter *Record* jmeter yang buat adalah :
   * 1. *Prerequisite create patient*
     2. *Prerequisite create user*
     3. *Search Patient*

Untuk *file record* lainnya hanya dilakukan *edit* karena sudah dibuat oleh pihak mitrais

1. Analisis data pada sql profiler

Analisis data pada sql profiler dilakukan dengan menjalankan record jmeter yang sudah di simpan. Dari *record* tersebut dicari bagian mana yang harus dibuatkan prosedur. Karena tidak semua akses ke database perlu dibuatkan prosedur. Hanya *query* untuk manembahkanpasiendan menambahkan *appointment*.

1. Membuat prosedur dan melakukan *inject* data kedalam sql server

Setelah mendapatkan query yang digunakan untuk menambahkan pasien dan menambahkan *appointmnet*, maka dibuatkan prosedur. Prosedur ini digunakan untuk mempermudah proses *inject* kedalam sql server. Setelah membuat prosedur, maka dilakukan inject sesuai dengan kebutuhan.

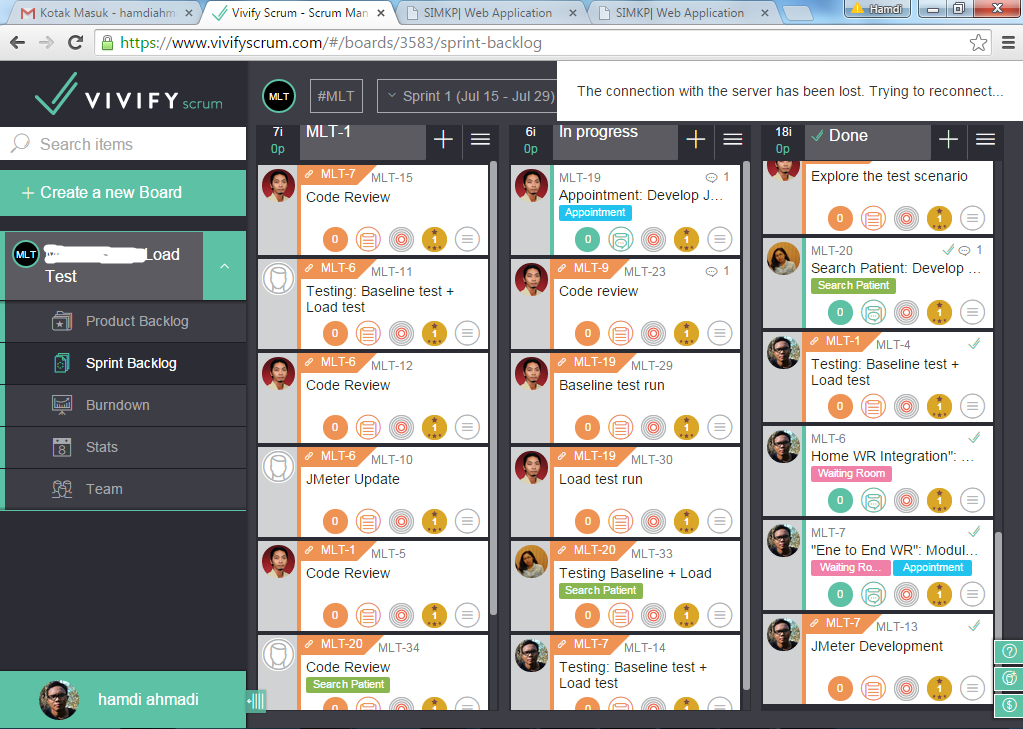
1. Java Misson Control

Setelah melakukan *inject* data, jmeter akan dijalankan sesuai dengan *test* yang diinginkan. Java mission control akan menyimpan data selama *test* dijalankan, *output* dari penggunaan java mission control adalah laporan mengenai kinerja sistem selama jmeter dijalankan.

### Pengerjaan *Testing*

Pengerjaan sistem dilakukan menggunakan scrum yang mengimplementasikan metode agile. Disini dalam 1 tim bebas mengerjakan apapun pekerjaan yang sudah dituliskan didalam Fifivyscrum . Tiap *task* akan ada bobot lama pengerjaan, sehingga pada akhir *sprint* ( batas sub modul pada pengerjaan aplikasi ) bisa didapatkan kontribusi pengerjaan tiap orang.

*Testing* dilakukandenganmelakukan *inject* data pasien dan *appointment sebanyak* 3.0000 pasien dan 30.000 *appointment.* Banyaknya user yang digunakan adalah satu user ( dokter ).



Gambar 3. fifivyscrum

### Laporan *Testing*

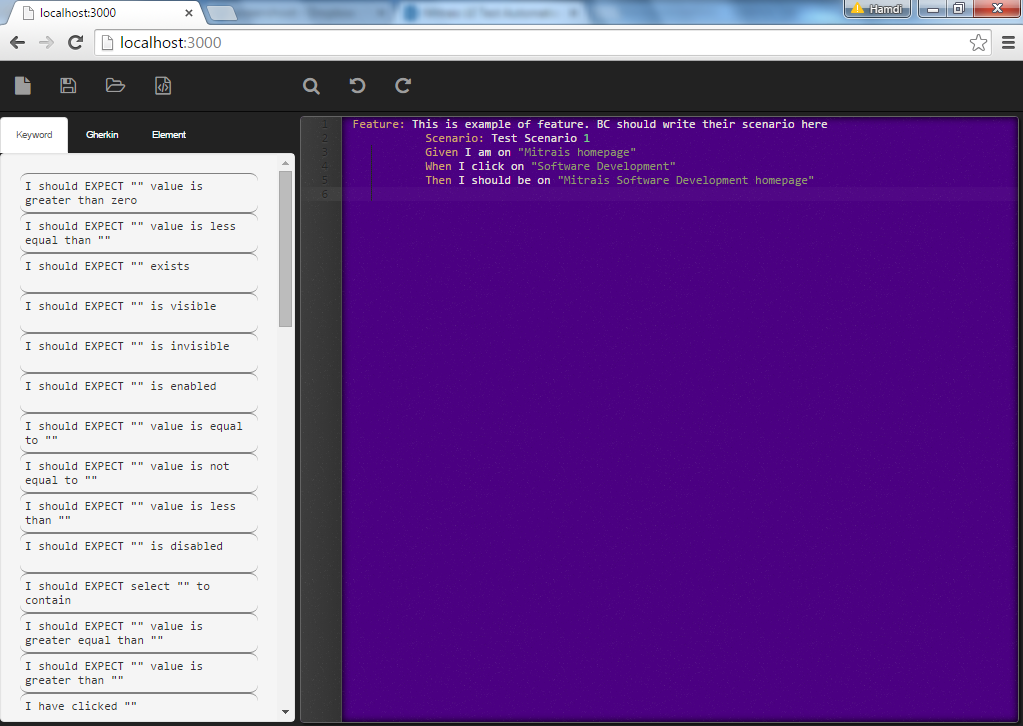
Laporan testing yang dikerjakan berupa gambar, tabel mengenai hasil *testing* dan beberapa *slow query*

# BAB V IMPLEMENTASI SISTEM

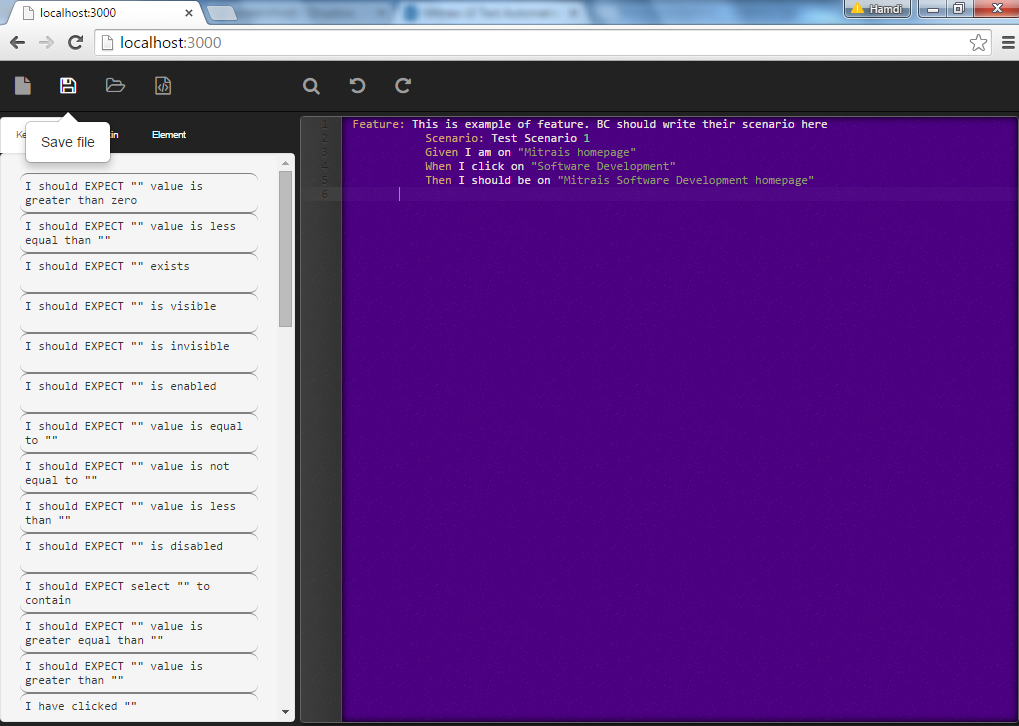
## Mitrais User Interface Testing Automation Framework

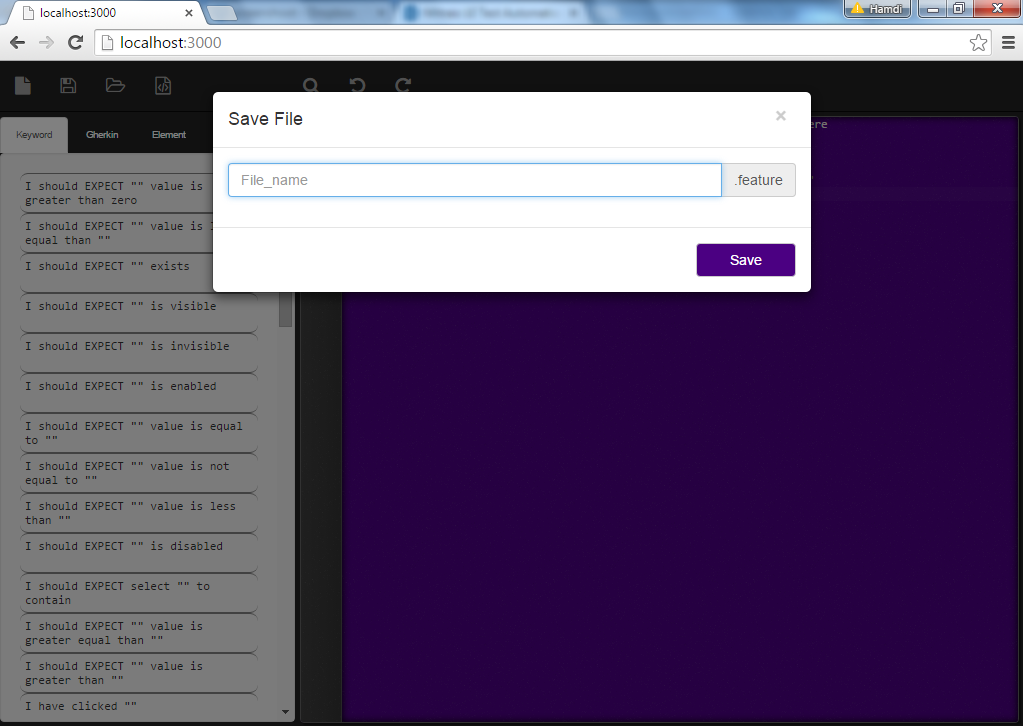
### Implementasi Antarmuka Pengguna

Implementasi antarmuka pengguna untuk aplikasi *text editor* ini hanya berisi satu buah halaman. Pada halaman ini berisi seluruh fitur yang ada, seperti *save file, load existing file* dan lainnya.

Gambar 4. halaman utama *text editor*

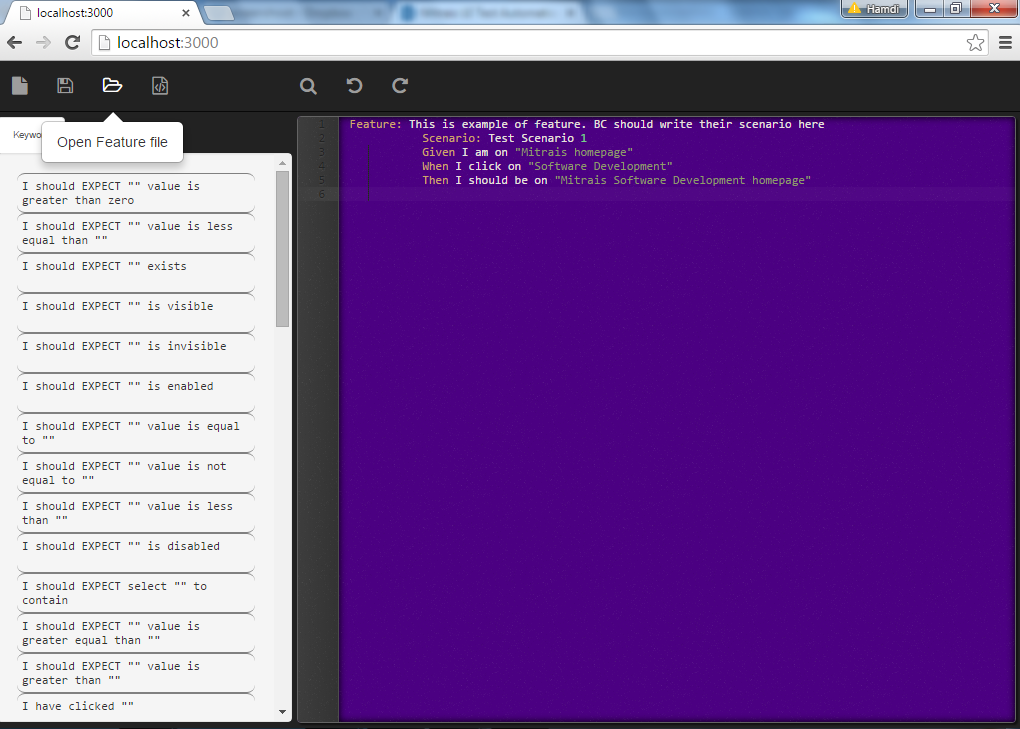
1. *Save File*

Gambar 5. tooltip *save file*

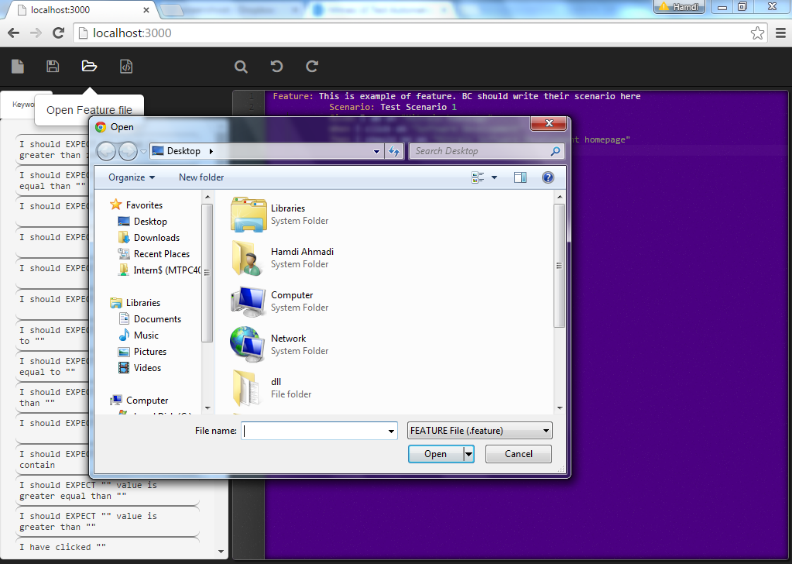
**

Gambar 6. *save file*

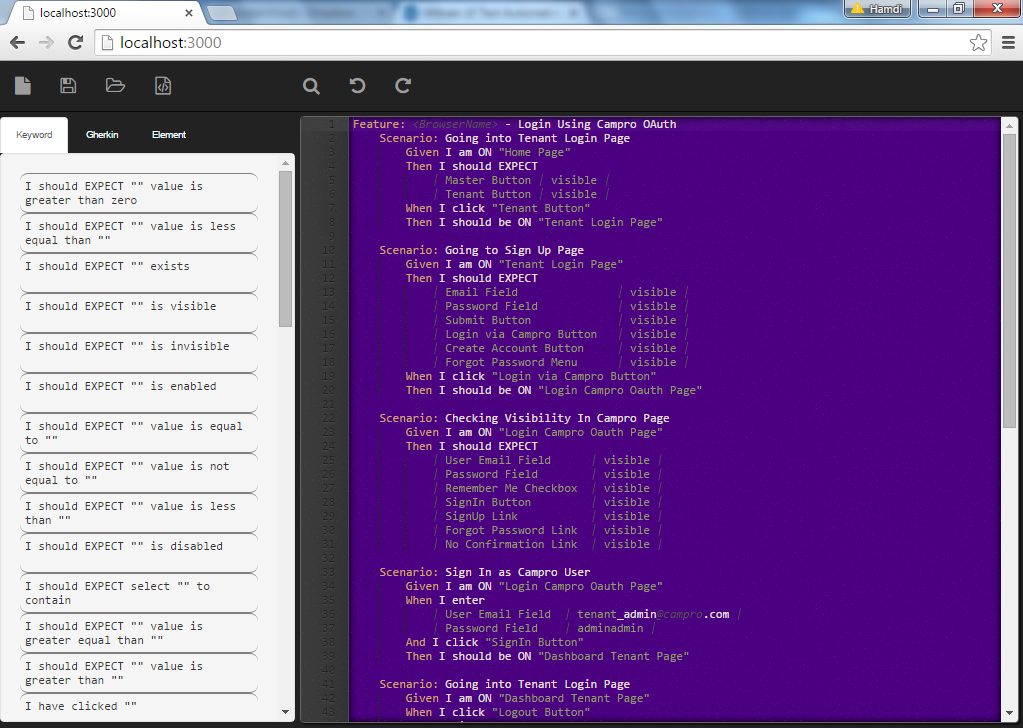
1. *Load existing file*



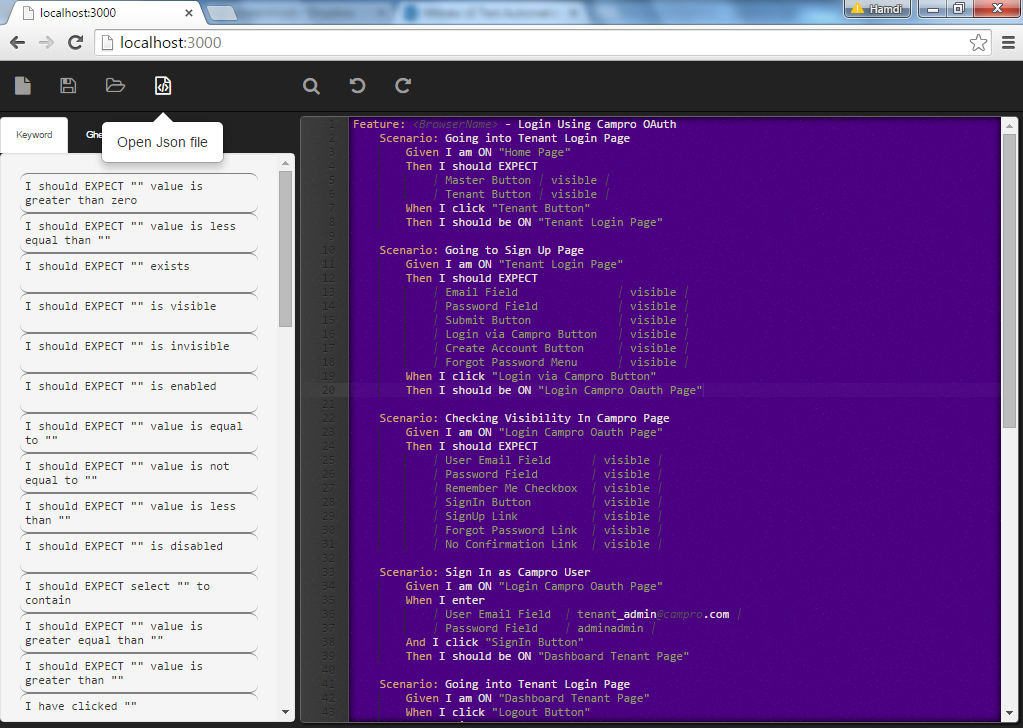
Gambar 7. tooltip *load existing file*

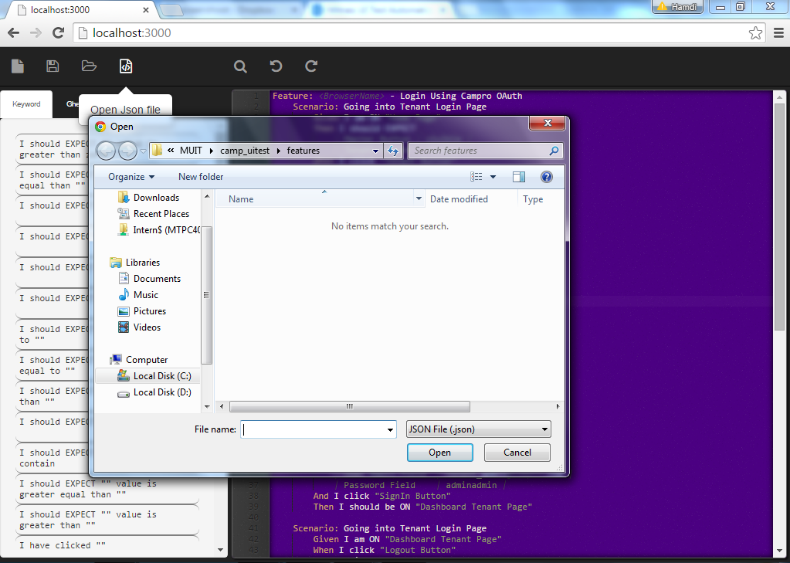
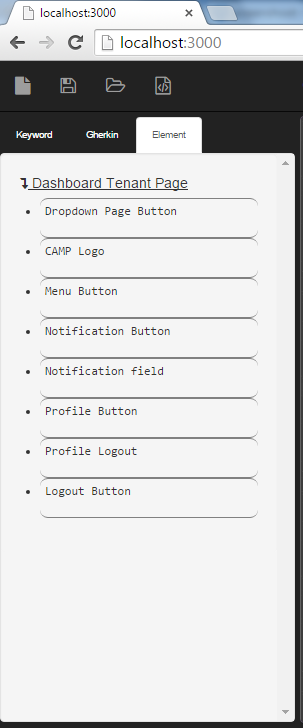


Gambar 8. *load existing file*

Gambar 9. hasil *load existing file*

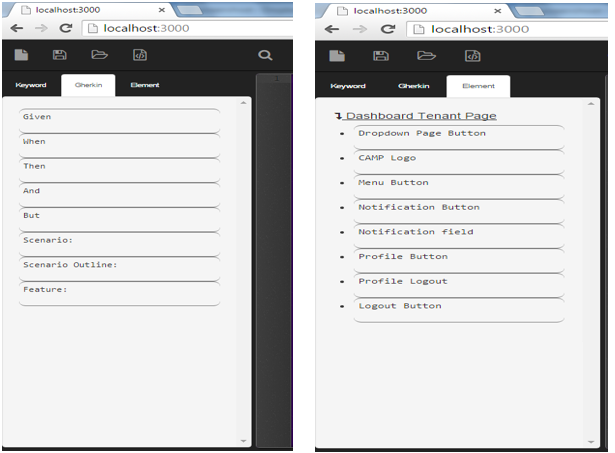
1. *Load element keyword*

Gambar 10. tooltip *load element keyword*

Gambar 11. *load element keyword*

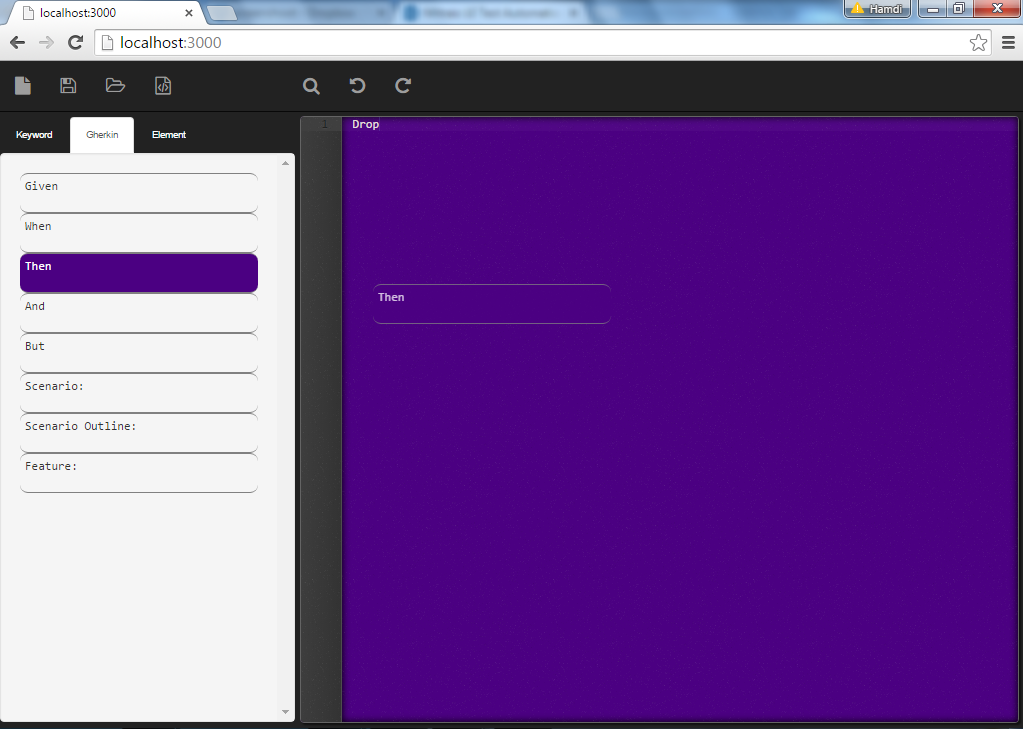
Gambar 12. hasil *load element keyword*

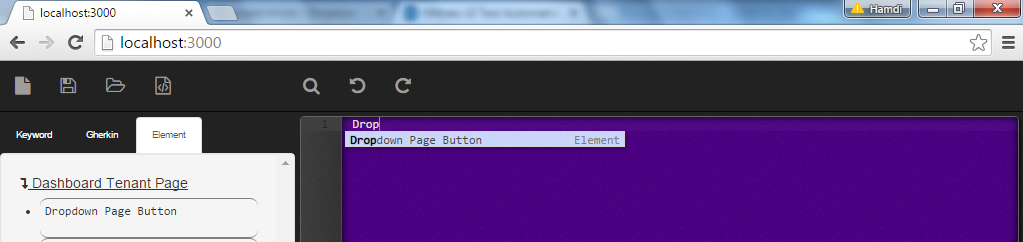
1. *Showing list keyword*



Gambar 13. *showing list keyword*

1. *Drag and Drop*

Gambar 14. *drag and drop*

1. *Autocomplete*

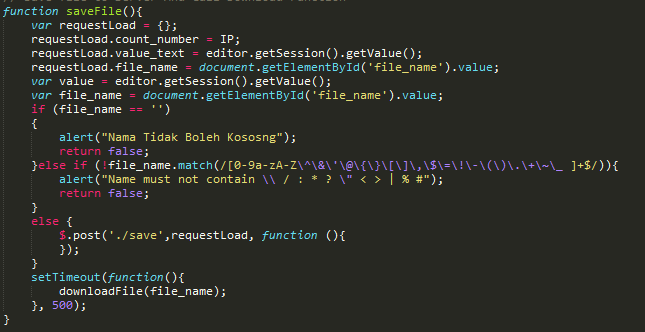
Gambar 15. *autocomplete*

### Implementasi Fungsi



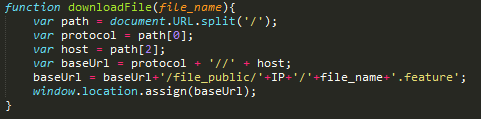
Kode Sumber 1. fungsi IP

Fungsi ini digunakan untuk mendapatkan suatu nilai unik, tujuannya adalah sebagai penanda suatu user. Hampir mirip dengan *session*, hal ini digunakan sebagai pembeda ketika *user* malakukan *save file*. Pada saat *save file*, *client* akan mengirimkan *request* ke server berisi data-data seperti nama *file*, isi teks, dan nomor IP. Pada saat *save file* dilakukan, maka *server* akan membuatkan suatu folder dengan nama sesuai dengan IP. Disana akan dibuatkan *file* dengan nama file sesuai dengan data *file* yang diberikan *client*.

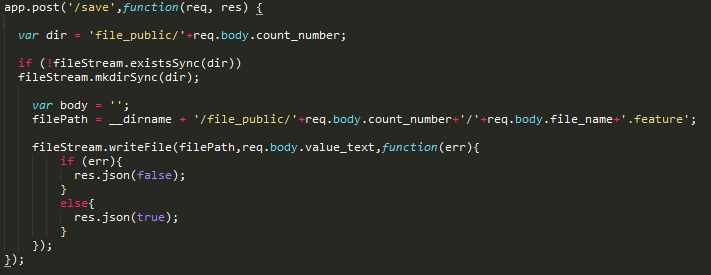
1. *Save File* 
   1. *Client side*

Kode Sumber 2. fungsi saveFile

Ketika *icon save file* di klik, maka *client* akan menjalankan script dengan fungsi saveFile. Fungsi ini digunakan untuk memanggil fungsi “./save“ pada server dengan data parsing “requestload” berupa json. Pada variabel requestload terdiri dari beberapa informasi seperti isi teks, nama *file* yang akan disimpan. Pada akhir fungsi, saveFile akan memanggil fungsi downloadFile yang berada pada *client.* Fungsi tersebut harus menunggu selama 500ms, tujuannya adalah menunggu server selesai menjalankan fungsinya.

Kode Sumber 3. fungsi downloadFile

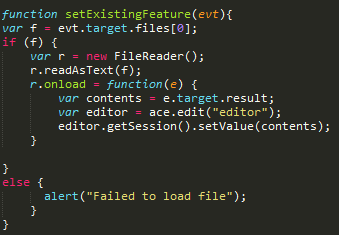
Ketika *request* ke *server* dijalankan dan memanggil fungsi downloadFile, maka pada sisi *server* akan membuatkan suatu *folder* dengan nama sesuai dengan IP , nama *file* sesuai dengan request, dan isi *file* sesuai dengan data teks. Setelah selesai dibuat, maka *client* akan membuka suatu *link* dengan format baseUrl ( localhost:3000 )/file\_public/nomor IP/namafile.feature. secara otomatis file tersebut akan di unduh oleh *client.*

* 1. *Server side*

Kode Sumber 4. fungsi *save server*

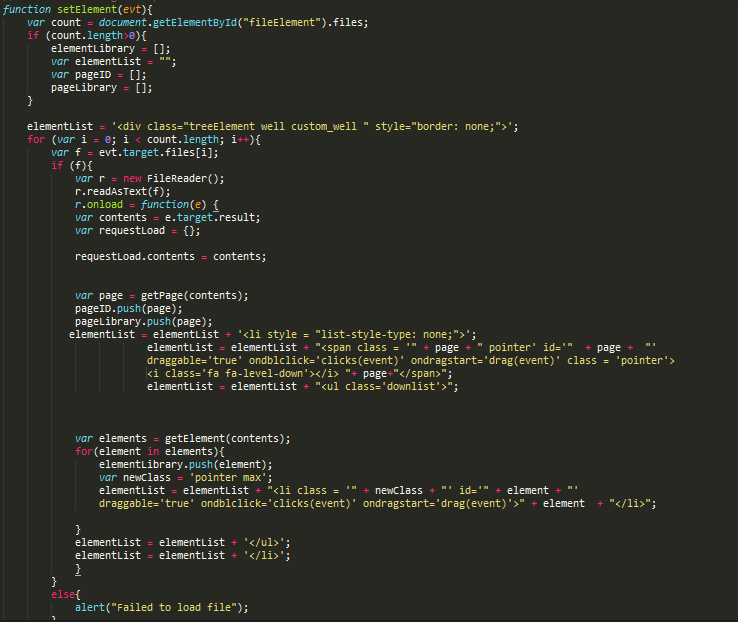
Ketika *client* malekukan *request* ke *server,* maka *server* akan menjalankan fungsi save, dimana fungsi ini bertujuan untuk membuat file pada *folder server*. Pertama kali *server* akan melakukan cek apakah nomor IP yang diberikan sudah dibutkan folder atau belum, jika belum makan *server* akan membuatkan *folder* dengan nama sesuai dengan nomor IP. Jika sudah dibuat maka *server* akan membuat *file* dengan nama *file* yang sudah didefinisikan dan isi *file* sesuai dengan nilai yang sudah diberikan.

1. *Load existing file*

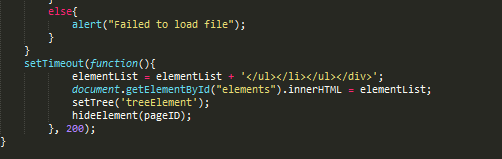


Kode Sumber 5. fungsi *setExistingFeature*

Fungsi ini akan dijalankan ketika ada *event* onclick pada *icon load existing file*. Fungsi ini akan menjalankan fungsi FileReader , FileReader digunakan untuk membaca file yang dimasukkan, dan menaruhnya pada *text editor.*

1. *Load element keyword*

Kode Sumber 6. *load element keyowrd*

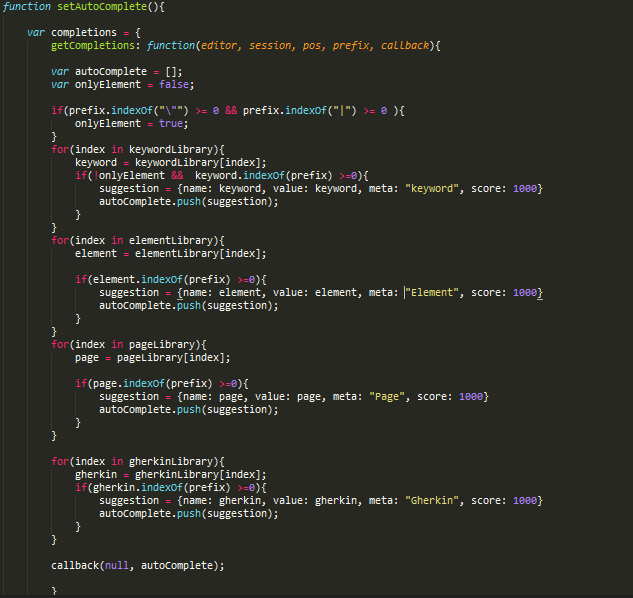
Kode Sumber 7. *load element keyword 2*

Sama seperti *load existing file*, menggunakan fungsi FileReader untuk membaca isi *file* tersebut. Perbedaanya adalah, cara untuk menampilkan isi dari *file* tersebut. Isi dari *file* ini akan dimasukkan kedalam list, dimana list tersebut diberikan parameter dragable agar bisa dilakukan *drag and drop* pada *text editor*.

1. *Drag and Drop*

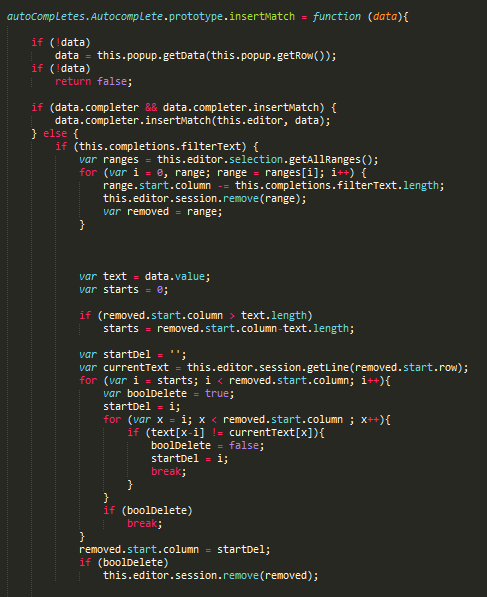
Kode Sumber 8. *drag and drop*

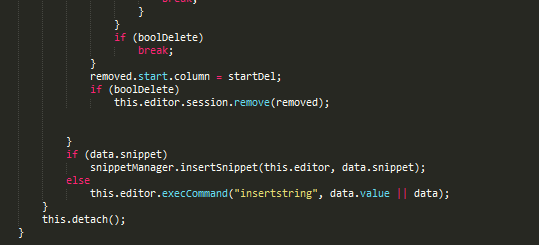
1. *Autocomplete*

Kode Sumber 9. *autocomplete*

*Autocomplete* diimplementasi dengan cara *extend* dari ace js. Dilakukan *extend* agar *code* pada ace js tetap asli ( tidak ada perubahan ).

1. *Autocorrect*

Kode Sumber 10. *autocorrect*

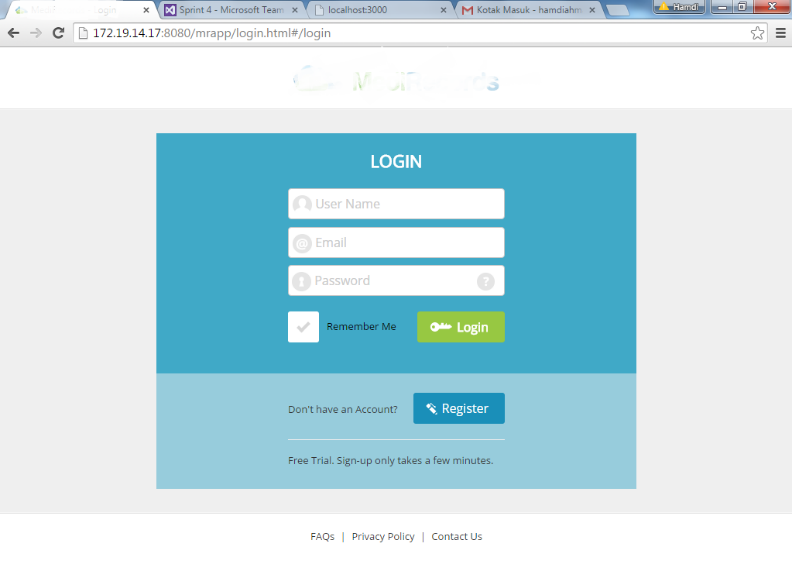
Kode Sumber 11. *autocorrect 2*

*Autocorrect* diimplementasikan dengan cara melakukan *extend* dari code ace js. *Autocorrect* akan terjadi ketika proses *autocomplete* dilakukan, hal ini diakukan dengan cara melakukan cek pada colom tersebut. Dilihat apakah kalimat pada *autocomplete* mempunyai substring yang sama dengan kalimat sebelumnya atau tidak. Jika ada, maka kalimat tersebut akan dihapus. Secara otomatis akan tergantikan oleh kalimat baru.

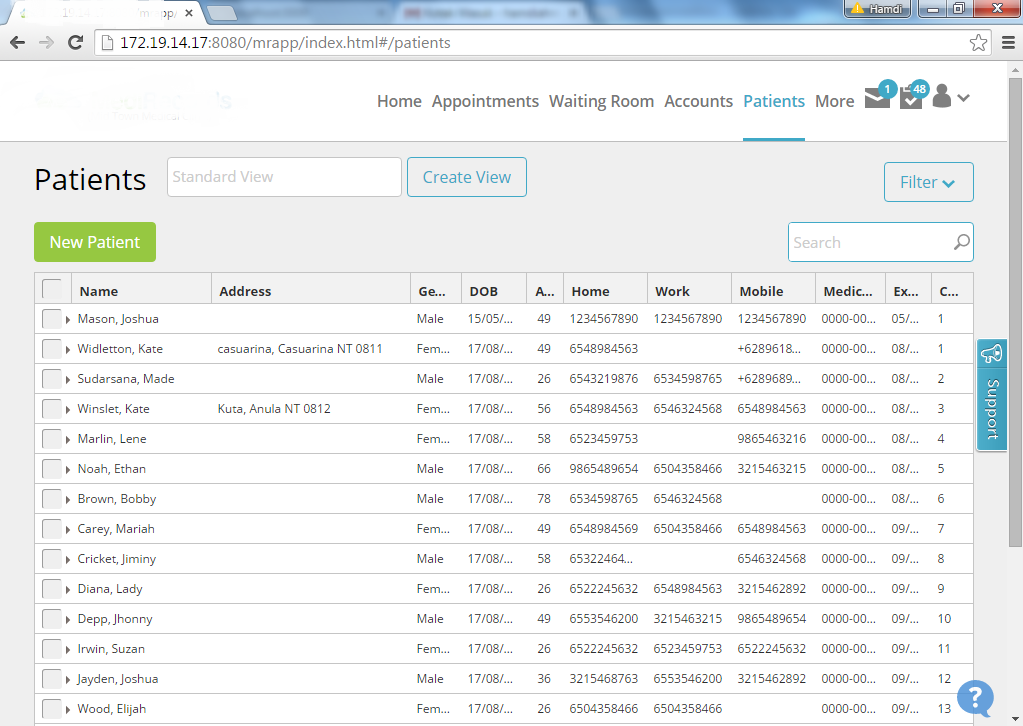
## Performance test terhadap aplikasi rekam medis

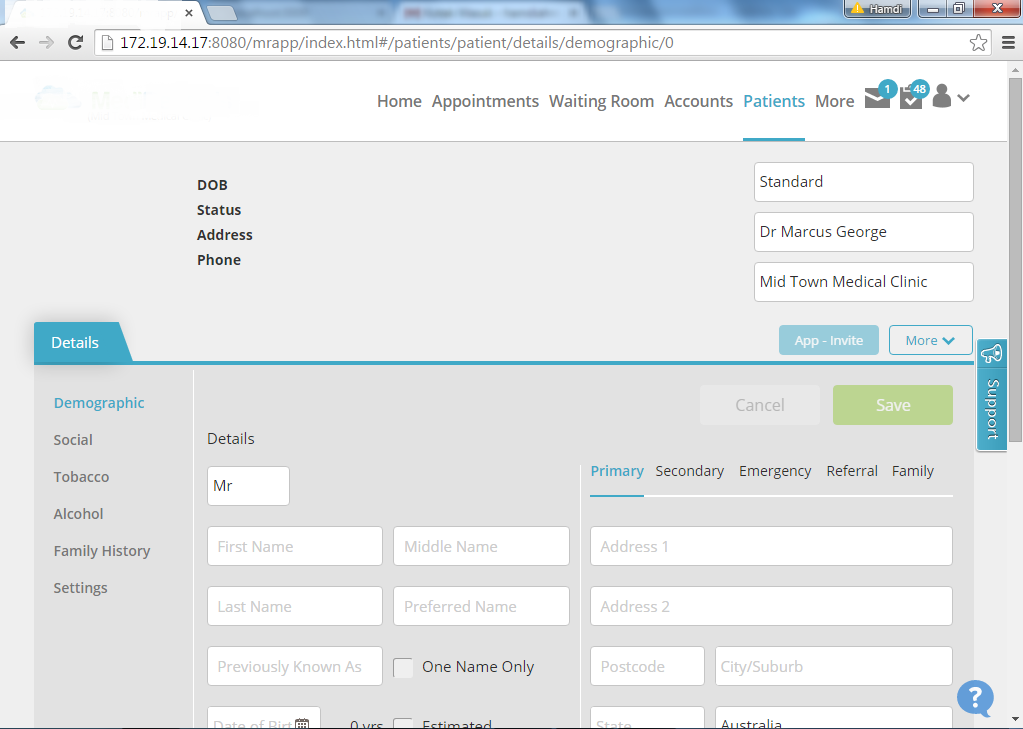
### Antarmuka Pengguna

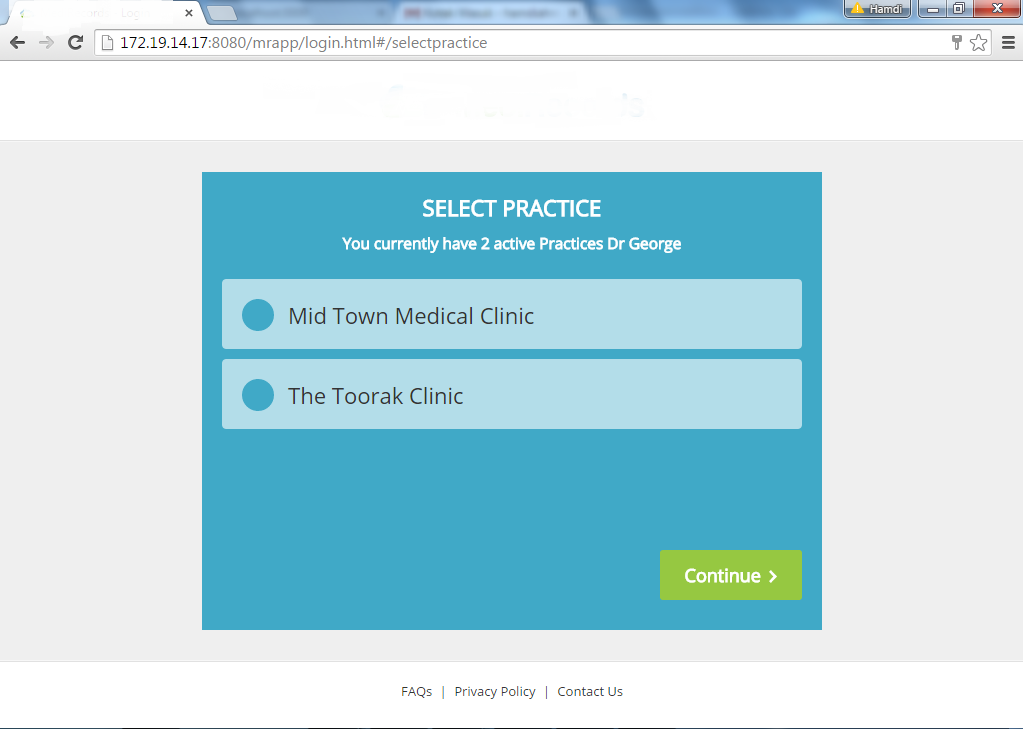
Aplikasi yang dilakukan untuk melakukan *performance testing adalah* aplikasi rekam medis yang sedang dikembangkan oleh pihak mitrais. Berikut adalah beberapa gambar antarmuka pengguna.

Gambar 16. halaman *login*

Gambar 17. halaman *home*

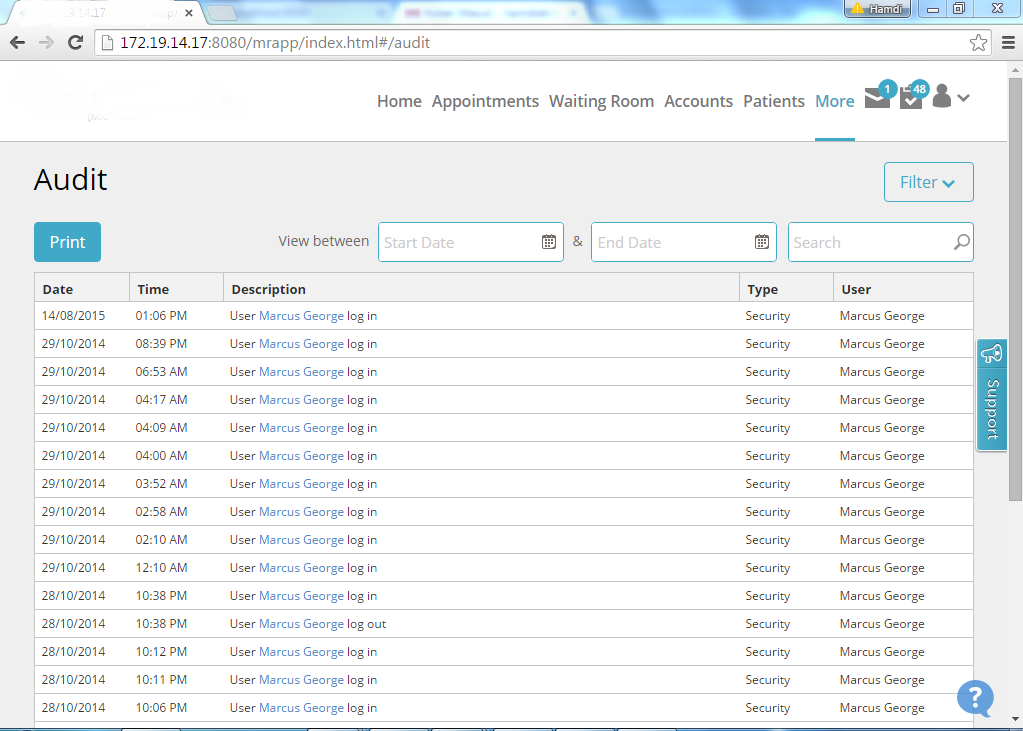
Gambar 18. halaman *list patient*

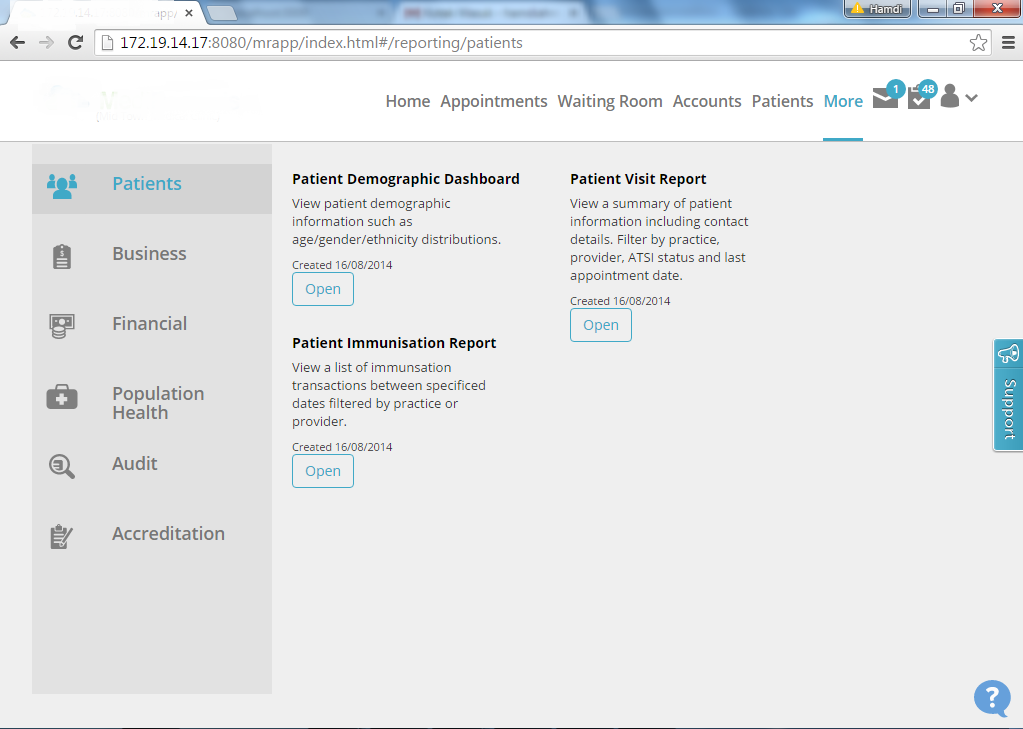
Gambar 19. halaman tambah pasien

Gambar 20. halaman memilih tempat klinik

Gambar 21. halaman ruang tunggu



Gambar 22. halaman audit



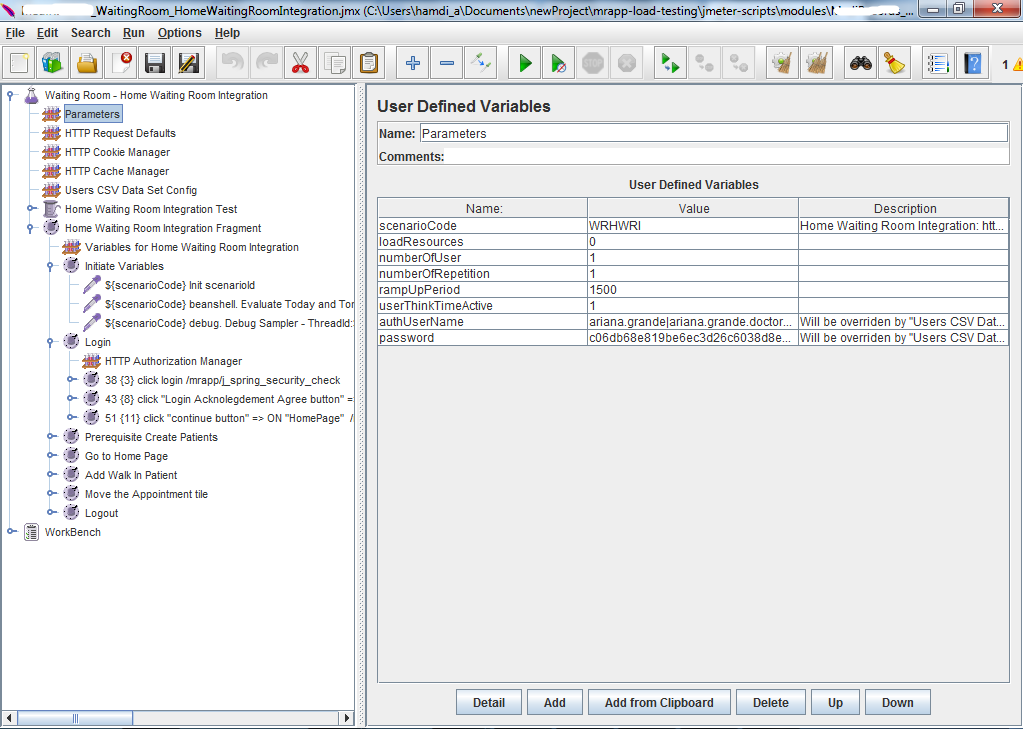
Gambar 23. halaman *report*

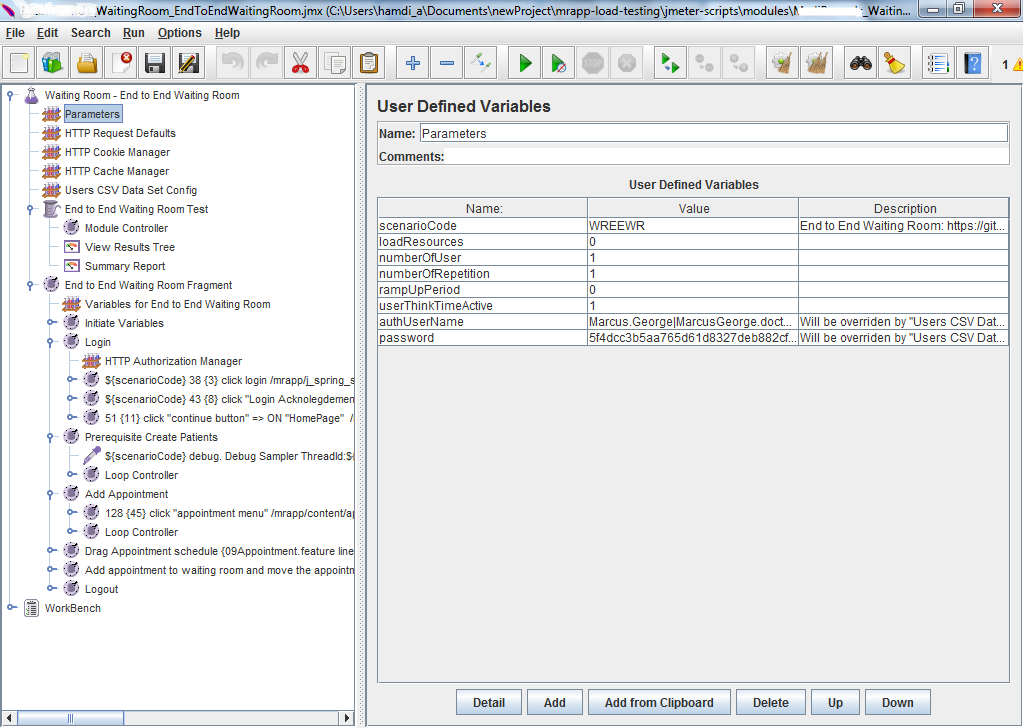
### *Testing*

1. Jmeter

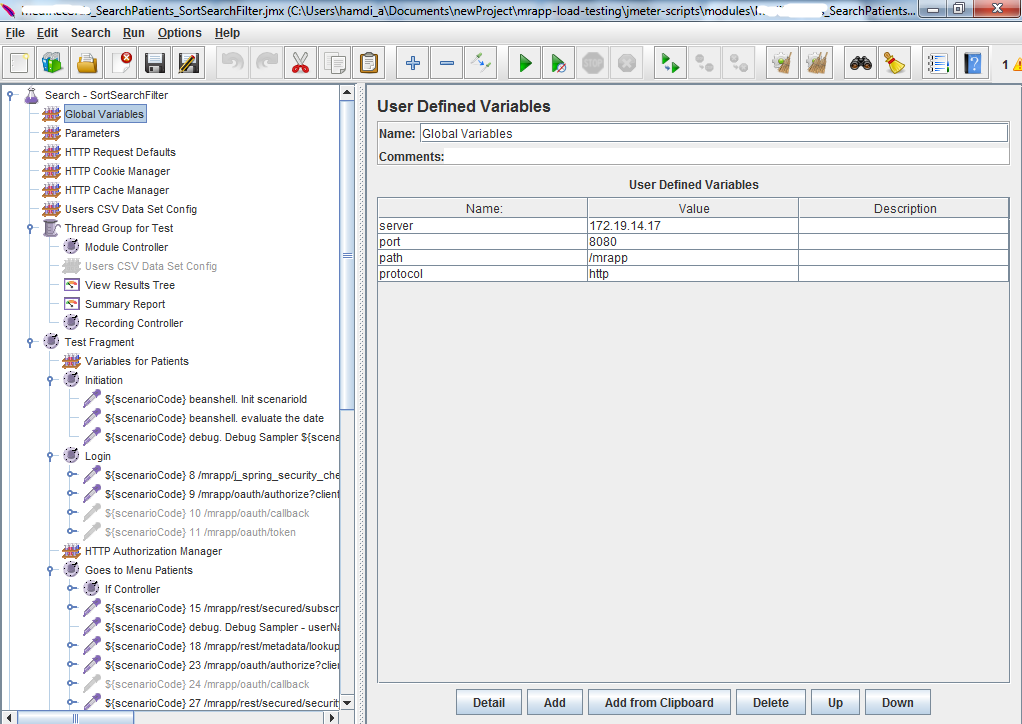
Langkah pertama yang dilakukan untuk membuat *testing* secara otomatis menggunakan jmeter adalah melakukan *recording* terhadap aplikasi yang akan diuji. *Recording* ini nantinya akan disimpan dalam file berextensi “.jmx”. Jmeter akan merekam semua request yang dikirim keserver. Berikut hasil rekaman yang dilakukan

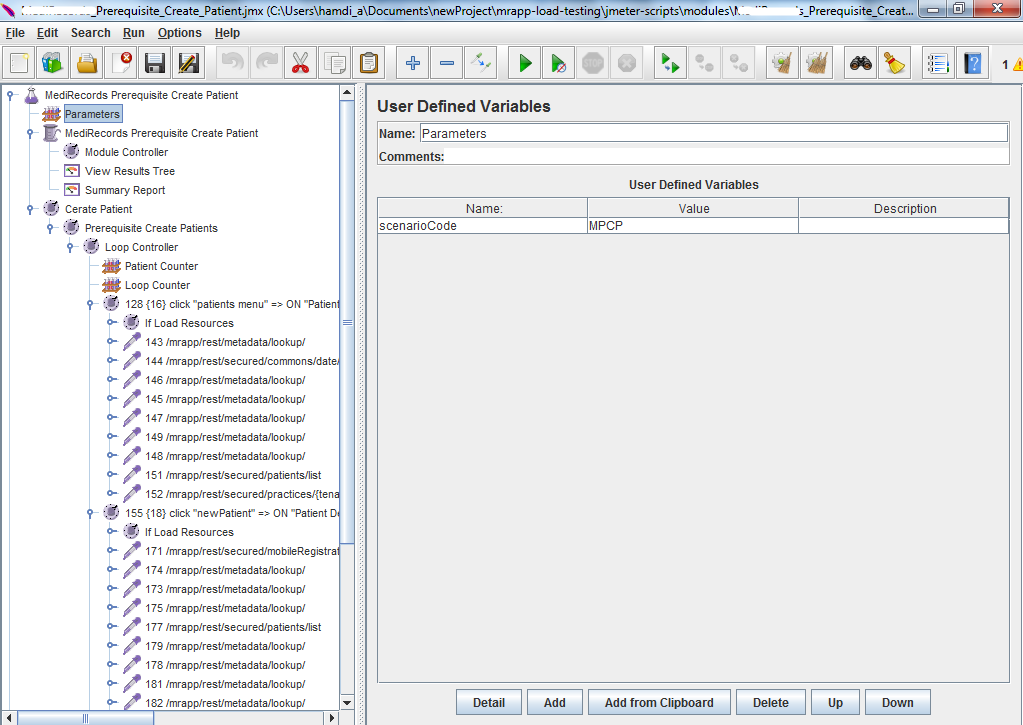
Gambar 24. *file* jmx *home* *waiting room*



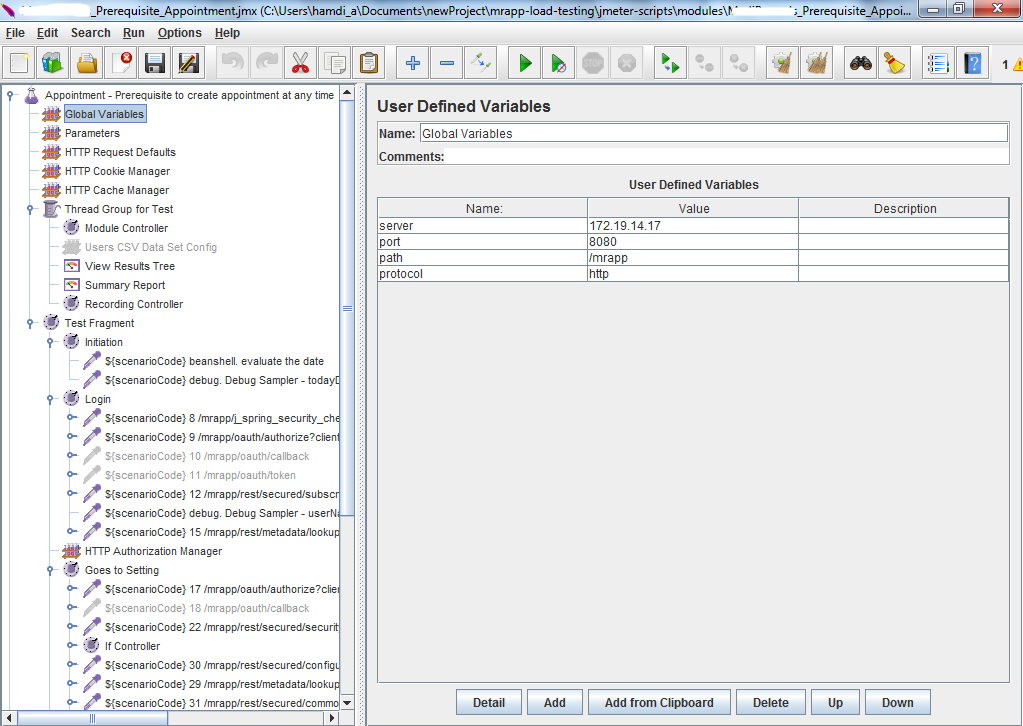
Gambar 25. *file* jmx *end to end* *waiting room*

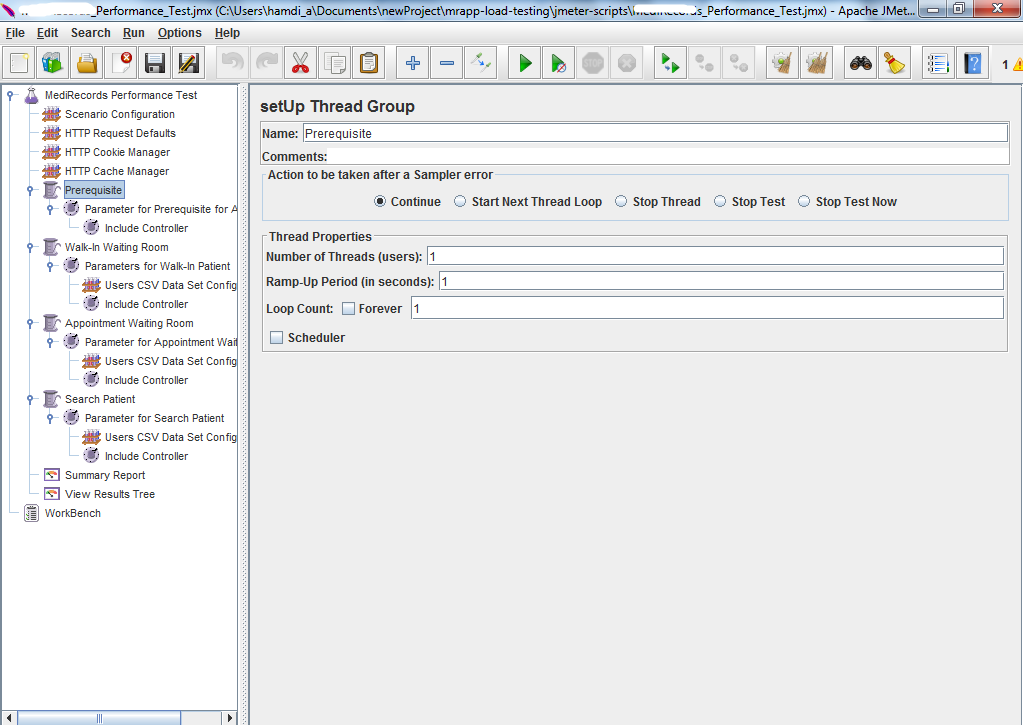
Gambar 26. *file* jmx *search patient*

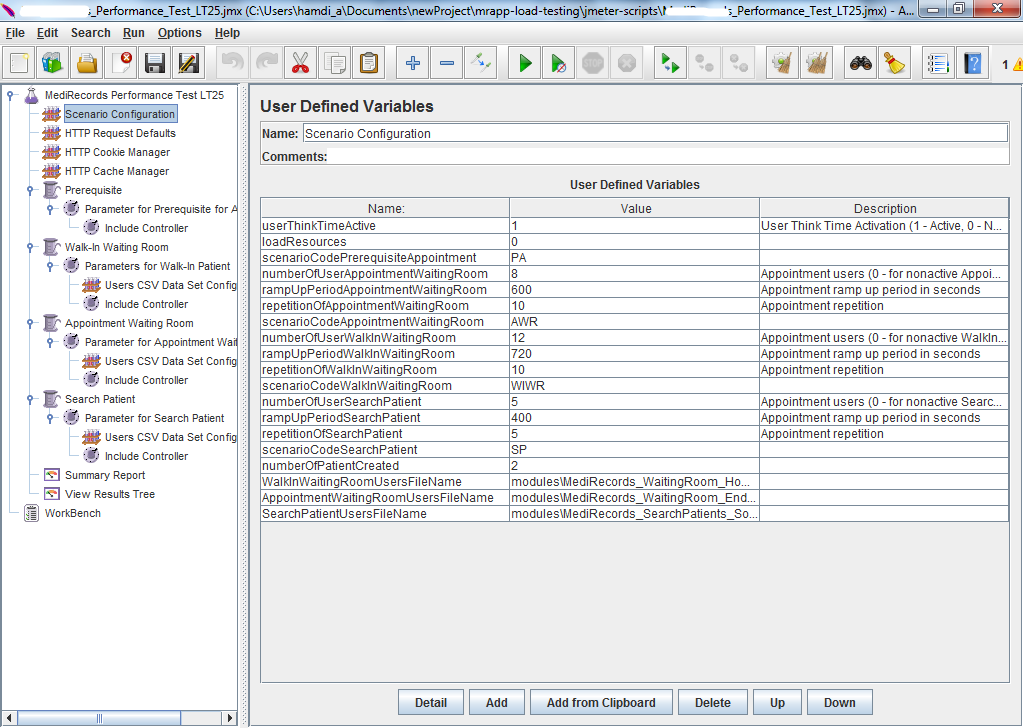


Gambar 27. *file* jmx *prerequisite* *create patient*

Gambar 28. *file* jmx *prerequisite* *make appointment*

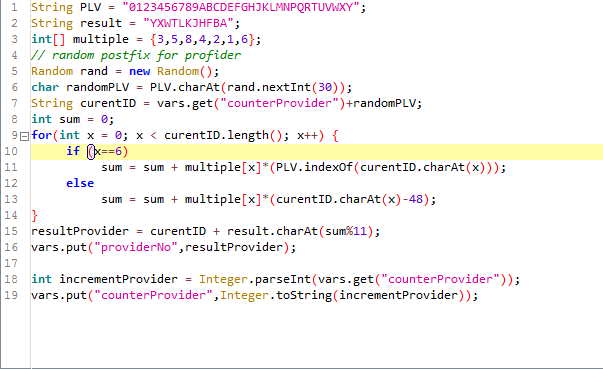


Gambar 29. *file* jmx *performance test*

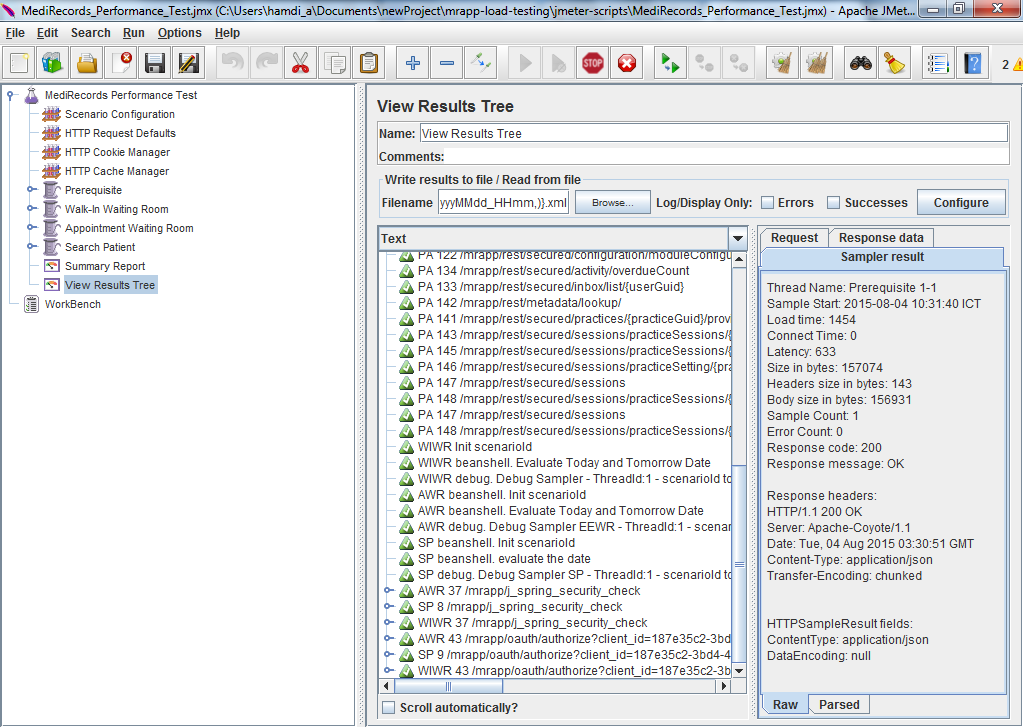
Gambar 30. *file* jmx *performace test Lt 25*

Jmx mempunyai fitur untuk menggunakan *file* jmx lain, memuat file lain untuk dijalankan. Jadi tidak dalam 1 file jmx menjalankan semua *testing*. *File* jmx *performance test* dan *performance test Lt 25*, menggunakan *file home waiting room, end to end waiting room, search patient, create patient, make appointment.* Selain merekam, ada beberapa hal yang harus ditambahkan kedalam jmeter. Pada aplikasi rekam medis, ketika melakukanpenambahan dokter, dokter akan mendapatkan provider. Dimana nomor provider dihasilkan dengan algoritma tertentu.

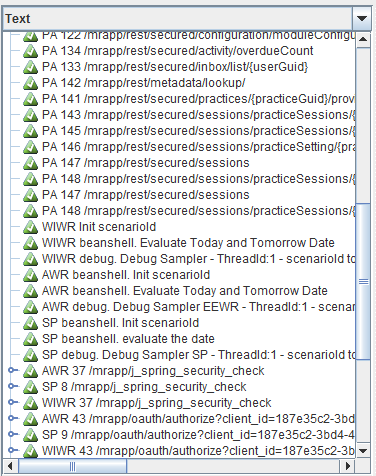
Kode Sumber 12. *generate provider*



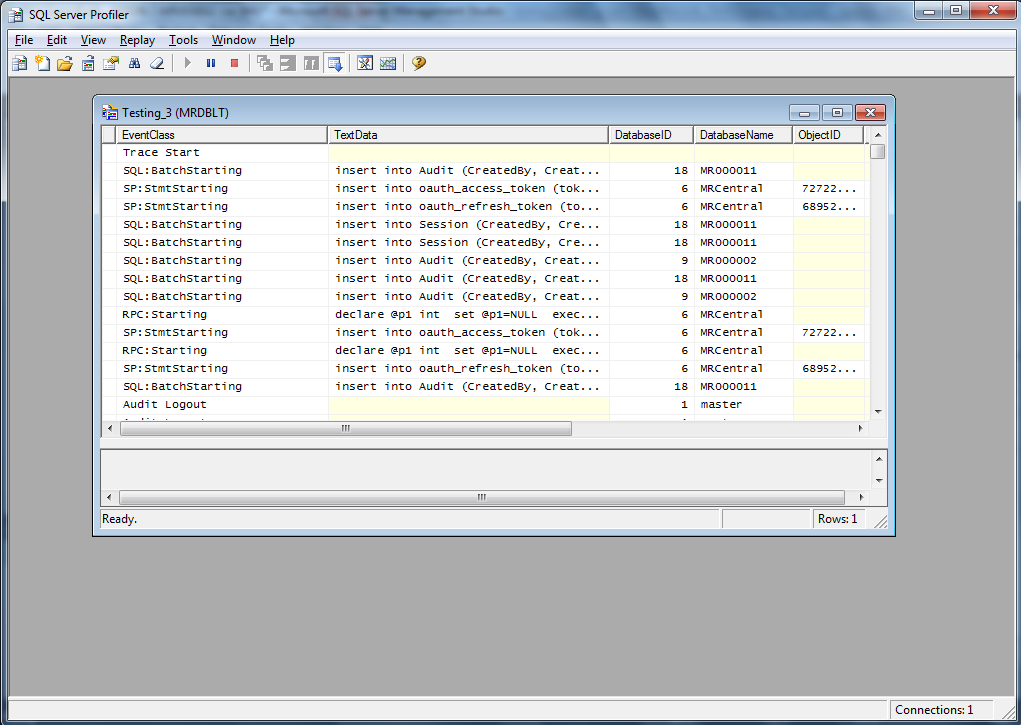
Berikut adalah hasil *testing* ketika jmeter dijalankan, akan muncul tanda ( *check* ) berwarna hijau, hal ini menandakan request berjalan lancar.

Gambar 31. hasil *performance test*

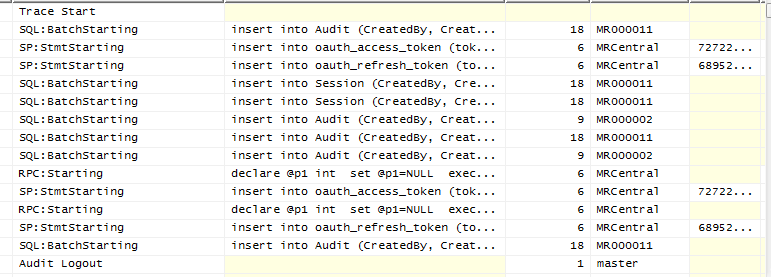
Gambar 32. hasil *performance test*



1. Sql profiler

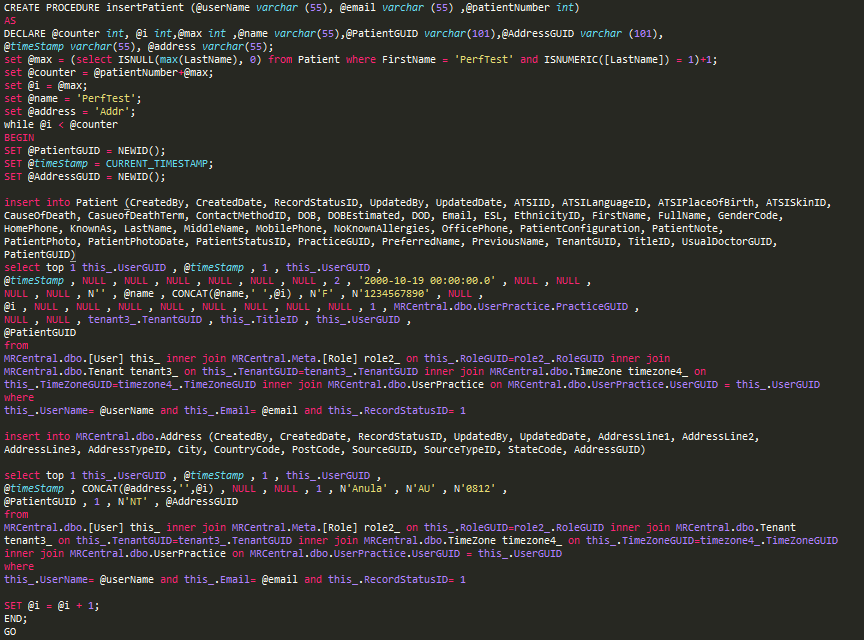
Query yang dianalisa adalah query *insert patient* dan *insert appointment*.

Gambar 33. sql profiler

Gambar 34. sql profiler 2

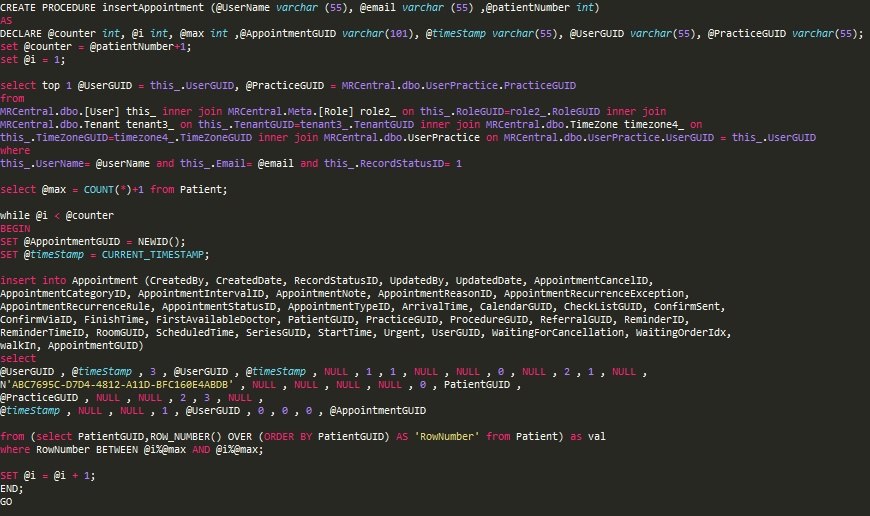
1. Sql server

Prosedur yang dibuat dalam *testing* ini adalah *insert patient* dan *insert appointment*.

Kode Sumber 13. *prosedur insert patient*

F:\KULIAH\KP\intern_folder\project2_hamdi\screenshoot\sql server\new_insert_patient3.png

Kode Sumber 14. *execute insert patient*



Kode Sumber 15. *prosedur insert appointment*

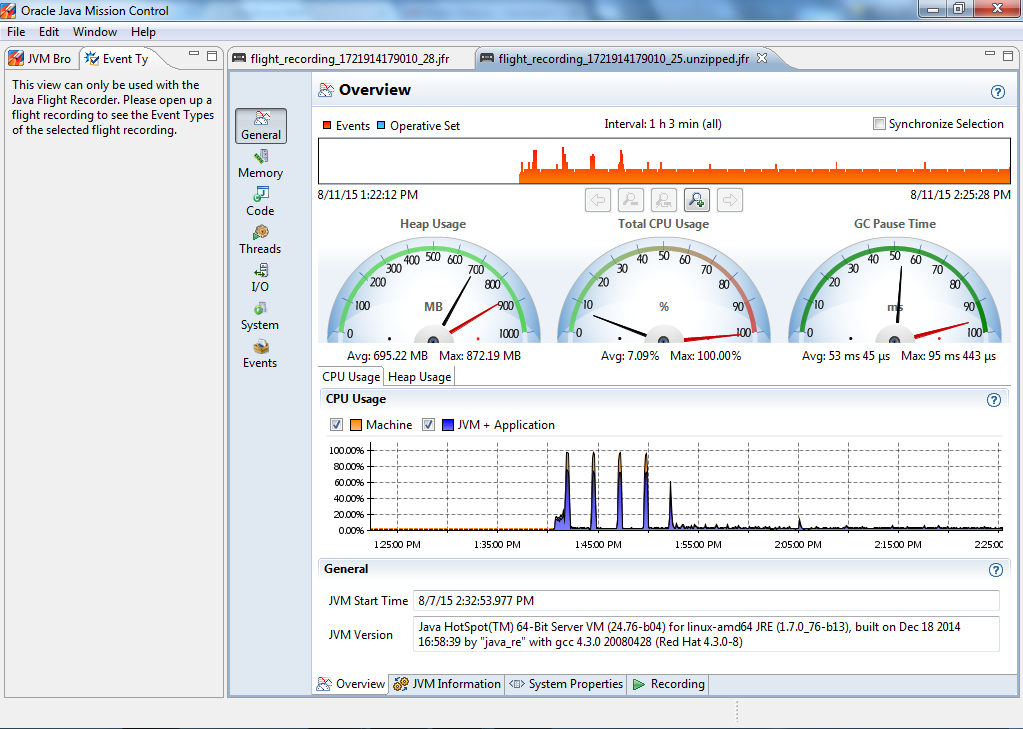
Kode Sumber 16. *execute insert appointment*



1. Java mission control

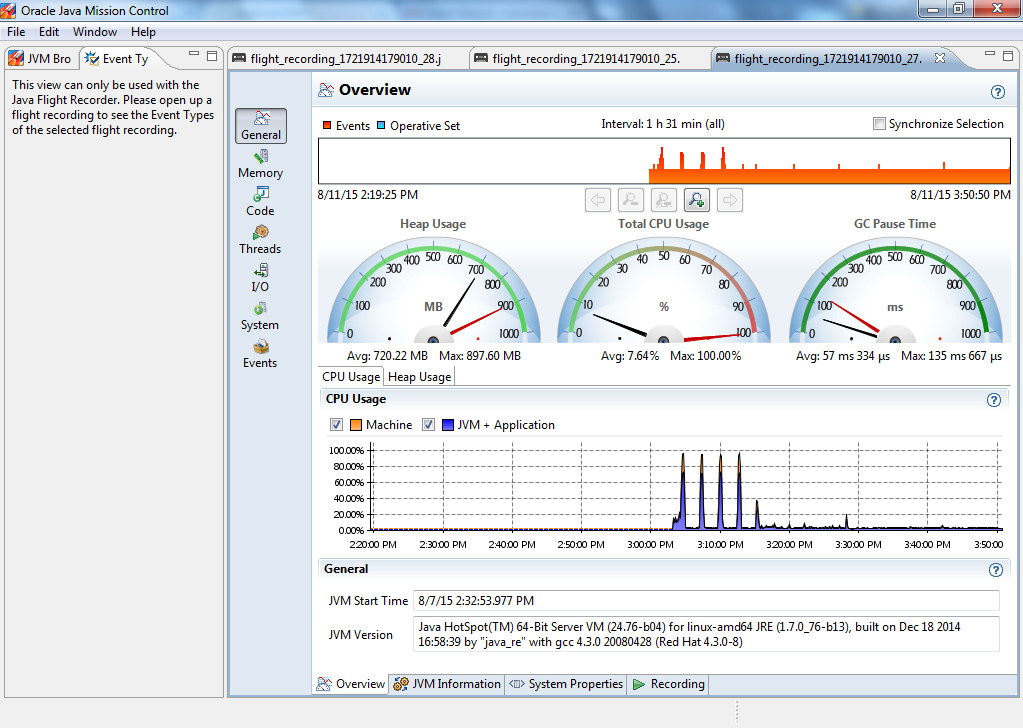
Testing yang dilakukan terdiri dari tiga macam, yaitu :

1. Baseline ( satu dokter )

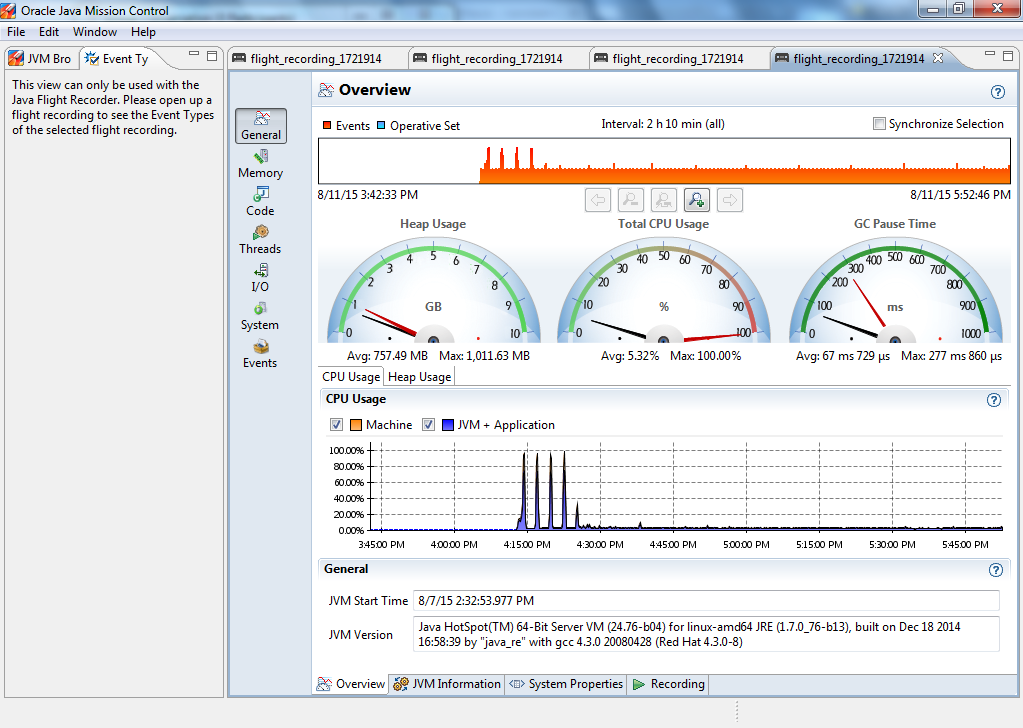
Gambar 39. hasil *test* baseline

1. Baseline + 30.000 pasien

Gambar 40. hasil *test* baseline + 30.000 pasien



1. Baseline + 30.000 pasien + 30.000 *appointment*

Gambar 41.** hasil *test* baseline + 30.000 pasien + 30.000 *appointmen*

*[Halaman ini sengaja dikosongkan]*

# BAB VI PENGUJIAN DAN EVALUASI

[Bab ini membahas pengujian dan evaluasi pada aplikasi yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsionalitas sistem dan pengujian kegunaan sistem atau pengujian usabilitas. Pengujian fungsionalitas mengacu pada …. Pengujian kegunaan sistem dilakukan dengan mengetahui tanggapan dari pengguna terhadap kegunaan sistem. Hasil evaluasi menjabarkan tentang rangkuman hasil pengujian pada bagian akhir bab ini.]

## Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan kerja praktik ini dilakukan pada lingkungan dan alat kakas sebagai berikut:

*Processor* : Intel® Core™ Duo CPU

(2.53 GHz, Cache 3 MB)

*Memory* : 4.00 GB

Jenis *Device* : Personal Computer

Sistem Operasi : Microsoft Windows 7 Profesional

*Browser* : Mozilla Firefox 26.0, Google Chrome

## Skenario Pengujian

[Pada bagian ini akan dijelaskan tentang skenario pengujian yang dilakukan.]

## Evaluasi Pengujian

Pada subbab ini akan diberikan hasil evaluasi dari pengujian-pengujian yang telah dilakukan. Evaluasi yang diberikan meliputi evaluasi pengujian kebutuhan fungsional, evaluasi hasil pengelompokan, dan evaluasi kegunaan sistem.

*[Halaman ini sengaja dikosongkan]*

# BAB VII KESIMPULAN DAN SARAN

[Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan kerja praktik serta saran-saran tentang pengembangan yang dapat dilakukan terhadap kerja praktik ini di masa yang akan datang.]

## Kesimpulan

[Berisi: (1) kesimpulan dari hasil pengujian dan evaluasi, (2) temuan-temuan baru selama pengujian. Kesimpulan **HARUS MENJAWAB rumusan masalah** yang diberikan pada bab Pendahuluan.]

## Saran

[Berisi pengembangan dan perbaikan penelitian kerja praktik ke depannya. Dapat berupa perluasan batasan masalah, variasi data uji, atau penggunaan teknologi dan algoritma lain.]

# DAFTAR PUSTAKA

Catatan: Setiap referensi yang dirujuk di dalam buku laporan, baik dari buku, jurnal, artikel dari web dll harus dituliskan di bagian daftar pustaka ini.

# LAMPIRAN

# BIODATA PENULIS

[Tulislah data riwayat hidup penulis 1 di sini]

**Foto Berwarna**

***Close-up***

**ukuran 3x4**

…………….................................................

……………………………….....................

…………………………………………….

……………………………………………

…………………………………………….

…………………………………………………………………….

[Tulislah data riwayat hidup penulis 2 di sini]

**Foto Berwarna**

***Close-up***

**ukuran 3x4**

…………….................................................

……………………………….....................

…………………………………………….

……………………………………………

…………………………………………….

…………………………………………………………………….

*// margin tiap sub bab beda*