|  |  |
| --- | --- |
|  | **https://lh3.googleusercontent.com/t9H-h3vRocgqRhzCc9V_RgpVj2tgeF4C46EA3DW_4SHoUcyDQehUNWlBG9SmpNJr9LH_OGfDuRxzzvew6eswYyXiHX6mtmEn2N2JwN4NZYqWir41XDIwQeSzv4fGpWq8IZg_Vz28** |

**PT. SOFCO GRAHA  
SIDOARJO  
(8 Juli 2019 s/d 8 Agustus 2019)**

**KERJA PRAKTIK**

**PENGUJIAN FUNGSIONAL MERATUS HUMAN RESOURCES SOFTWARE MENGGUNAKAN SELENIUM**

YUSUF UMAR HANAFI NRP. 0721 16 4000 0006  
MUHAMMAD RUSYDI AL AFFAN NRP. 0721 16 4000 0029

DOSEN PEMBIMBING  
Dr. SURYA SUMPENO, ST., M.Sc.

DEPARTEMEN TEKNIK KOMPUTER  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019

|  |
| --- |
| *(Halaman ini sengaja dikosongkan)* |

# LEMBAR PENGESAHAN

**PENGUJIAN FUNGSIONAL MERATUS HUMAN RESOURCES SOFTWARE MENGGUNAKAN SELENIUM**

Laporan Kerja Praktik ini disusun untuk memenuhi persyaratan akademik Departemen Teknik Komputer - Fakultas Teknologi Elektro - Institut Teknologi Sepuluh Nopember

**Menyetujui,**

**Pembimbing Kerja Praktik**

**Dr. Surya Sumpeno, ST., M.Sc.**

**NIP. 196906131997021003**

**Mengetahui,**

**Kepala Departemen Teknik Komputer FTE-ITS**

**Dr. I Ketut Eddy Purnama, ST., MT.,**

**NIP. 196907301995121001**

**SURABAYA**

**SEPTEMBER 2019**

|  |
| --- |
| *(Halaman ini sengaja dikosongkan)* |

**LEMBAR PENGESAHAN II**

**PENGUJIAN FUNGSIONAL MERATUS HUMAN RESOURCES SOFTWARE MENGGUNAKAN SELENIUM**



**PT. Sofco Graha**

**Berbek Industri III no 17, Berbek, Waru,**

**Sidoarjo - Jawa Timur 61256**

Laporan Kerja Praktik ini disusun untuk memenuhi persyaratan akademik Departemen Teknik Komputer - Fakultas Teknologi Elektro - Institut Teknologi Sepuluh Nopember

**Menyetujui ;**

**Pembimbing Kerja Praktik**

**SPV QA & Training**

**Henrie Prawiro**

**SIDOARJO**

**SEPTEMBER 2019**

|  |
| --- |
| *(Halaman ini sengaja dikosongkan)* |

# KATA PENGANTAR

Puji syukur atas kehadirat Tuhan yang Maha Esa, karena atas rahmat-Nya penulis telah dapat menyelesaikan Kerja Praktik di PT. Sofco Graha, Sidoarjo yang dilaksanakan tanggal 8 Juli 2019 hingga 8 Agustus 2019. Dalam penyelesaian Laporan Kerja Praktik ini, kami mengucapkan terima kasih kepada berbagai pihak yang telah membantu terselesaikannya laporan ini :

* Bapak Dr. I Ketut Eddy Purnama, ST., MT. selaku Kepala Departemen Teknik Komputer FTE-ITS
* Bapak Eko Pramunanto, ST., MT. selaku Koordinator Kerja Praktik Departemen Teknik Komputer ITS
* Bapak Dr. Surya Sumpeno, ST., M.Sc. selaku dosen pembimbing kerja Praktik
* Bapak Henrie Prawiro. Selaku SPV QA & Training yang menjadi Pembimbing Lapangan dari PT. Sofco Graha selama Kerja Praktik ini
* Bapak dan Ibu Karyawan PT. Sofco Graha, Sidoarjo

Penulis menyampaikan permohonan maaf jika selama pelaksanaan kerja Praktik terdapat hal yang kurang berkenan dan jikalau ada salah dalam penulisan laporan ini.

                Surabaya,1 September 2019

                 Penulis

|  |
| --- |
| *(Halaman ini sengaja dikosongkan)* |

# DAFTAR ISI

[LEMBAR PENGESAHAN iii](#_Toc23229812)

[LEMBAR PENGESAHAN II v](#_Toc23229813)

[KATA PENGANTAR vii](#_Toc23229814)

[DAFTAR ISI ix](#_Toc23229815)

[DAFTAR GAMBAR xi](#_Toc23229816)

[DAFTAR TABEL xii](#_Toc23229817)

[BAB I PENDAHULUAN 14](#_Toc23229818)

[1.1 LATAR BELAKANG 14](#_Toc23229819)

[1.2 BATASAN PERMASALAHAN 15](#_Toc23229820)

[1.3 TUJUAN 15](#_Toc23229821)

[1.4 BENTUK KEGIATAN 16](#_Toc23229822)

[1.5 TEMPAT DAN WAKTU PELAKSANAAN 17](#_Toc23229823)

[1.6 METODE PENULISAN 17](#_Toc23229824)

[1.7 SISTEMATIKA PENULISAN 18](#_Toc23229825)

[BAB II TINJAUAN UMUM PERUSAHAAN 20](#_Toc23229826)

[2.1 SEJARAH SINGKAT PERUSAHAAN 20](#_Toc23229827)

[2.2 PROFIL PERUSAHAAN 21](#_Toc23229828)

[2.3 FILOSOFI PERUSAHAAN 21](#_Toc23229829)

[2.4 VISI DAN MISI 21](#_Toc23229830)

[2.5 PRODUK DAN LAYANAN 22](#_Toc23229831)

[BAB III TINJAUAN PUSTAKA 24](#_Toc23229832)

[3.1 WEB APPLICATION 24](#_Toc23229833)

[3.2 SELENIUM 25](#_Toc23229834)

[3.3 SOFTWARE TESTING 26](#_Toc23229835)

[3.3.1 Sasaran Pengujian 26](#_Toc23229836)

[3.3.2 Karakteristik Pengujian 27](#_Toc23229837)

[3.4 PRINSIP PENGUJIAN 27](#_Toc23229838)

[3.5 QUALITY ASSURANCE 28](#_Toc23229839)

[3.4.1 Mindset Pengujian 29](#_Toc23229840)

[3.4.2 Analisa dan Pengujian Fungsional 29](#_Toc23229841)

[3.4.3 Perbaikan Proses 29](#_Toc23229842)

[3.4.4 Pengujian Keamanan 29](#_Toc23229843)

[3.4.5 Pengujian Performa 30](#_Toc23229844)

[3.3.6 User Acceptance Testing 30](#_Toc23229845)

[3.6 JENIS-JENIS PENGUJIAN SOFTWARE 30](#_Toc23229846)

[3.7 PENGUJIAN SISTEM 34](#_Toc23229847)

[3.7 PENGUJIAN APLIKASI MENGGUNAKAN SELENIUM 36](#_Toc23229848)

[3.8 HUMAN RESOURCES DEVELOPMENT 36](#_Toc23229849)

[BAB IV 41](#_Toc23229850)

[DESAIN DAN IMPLEMENTASI SISTEM 41](#_Toc23229851)

[3.1 DESAIN 41](#_Toc23229852)

[BAB V PEMBAHASAN 43](#_Toc23229853)

[4.1 MELAKUKAN MANUAL TESTING 43](#_Toc23229854)

[4.2 PEMBUATAN SKENARIO TES 45](#_Toc23229855)

[4.3 MEMASANG LIBRARY YANG DIBUTUHKAN 50](#_Toc23229856)

[4.4 PEMBUATAN KODE PROGRAM PAGE OBJECT 51](#_Toc23229857)

[4.5 PEMBUATAN KODE PROGRAM AUTOMATED TEST 55](#_Toc23229858)

[Setelah membuat kode program automated testing, maka kode tersebut dijalankan untuk memulai tes. Tes dinyatakan berhasil jika tidak ditemukan proses yang gagal dalam suatu modul pada saat dijalankan. 59](#_Toc23229859)

[4.6 PEMBUATAN KODE PROGRAM RUNNING TEST 60](#_Toc23229860)

[4.7 PEMBUATAN AUTOMATED TESTING REPORT 65](#_Toc23229861)

[BAB VI PENUTUP 67](#_Toc23229862)

[5.1 KESIMPULAN 67](#_Toc23229863)

[5.2 SARAN 67](#_Toc23229864)

[DAFTAR PUSTAKA 69](#_Toc23229865)

[LAMPIRAN 71](#_Toc23229866)

# DAFTAR GAMBAR

[Gambar 2.1.1 Logo PT. Sofco Graha 19](#_Toc20862392)

[Gambar 3.1.1 Proses Testing 31](#_Toc20862394)

[Gambar 4.3.1 Tangkap Layar Configure Build Path 44](file:///C:\Users\asus2\OneDrive\Documents\Buku%20KP%20SOfco%20Graha\Buku%20KP%20bisa%20diedit.docx#_Toc20862396)

[Gambar 4.3.2 Tangkap Layar Menambahkan Library 44](file:///C:\Users\asus2\OneDrive\Documents\Buku%20KP%20SOfco%20Graha\Buku%20KP%20bisa%20diedit.docx#_Toc20862397)

[Gambar 4.4.1 Contoh Program Page Object untuk Aktifiktas Test 46](file:///C:\Users\asus2\OneDrive\Documents\Buku%20KP%20SOfco%20Graha\Buku%20KP%20bisa%20diedit.docx#_Toc20862398)

[Gambar 4.4.2 Contoh Tombol Simpan dimana Memiliki Deteksi Alert dan Memindah Controller Automatic Browser ke Alert Tersebut 46](file:///C:\Users\asus2\OneDrive\Documents\Buku%20KP%20SOfco%20Graha\Buku%20KP%20bisa%20diedit.docx#_Toc20862399)

[Gambar 4.4.3 Contoh Program untuk Menekan Tombol yang Ada dan Berpindah Halaman 47](file:///C:\Users\asus2\OneDrive\Documents\Buku%20KP%20SOfco%20Graha\Buku%20KP%20bisa%20diedit.docx#_Toc20862400)

[Gambar 4.4.5 Contoh Program untuk Menyimpan Laporan dan Mendeteksi Sukses Tidaknya Sebuah Fungsi Berjalan dalam Test 48](file:///C:\Users\asus2\OneDrive\Documents\Buku%20KP%20SOfco%20Graha\Buku%20KP%20bisa%20diedit.docx#_Toc20862401)

[Gambar 4.4.6 Contoh Program untuk Navigasi Menuju Suatu Halaman 48](file:///C:\Users\asus2\OneDrive\Documents\Buku%20KP%20SOfco%20Graha\Buku%20KP%20bisa%20diedit.docx#_Toc20862402)

[Gambar 4.5.1 Contoh Program Yang Memberikan Akses menuju Database 49](file:///C:\Users\asus2\OneDrive\Documents\Buku%20KP%20SOfco%20Graha\Buku%20KP%20bisa%20diedit.docx#_Toc20862403)

[Gambar 4.5.2 Contoh Program yang Menyimpan Sebuah Status Apakah Tes Skenario Tersebut Sudah Dijalankan Apa Tidak 50](file:///C:\Users\asus2\OneDrive\Documents\Buku%20KP%20SOfco%20Graha\Buku%20KP%20bisa%20diedit.docx#_Toc20862404)

[Gambar 4.5.3 Contoh Program untuk men-setup Selenium 50](file:///C:\Users\asus2\OneDrive\Documents\Buku%20KP%20SOfco%20Graha\Buku%20KP%20bisa%20diedit.docx#_Toc20862405)

[Gambar 4.5.4 Contoh Program untuk Mendeteksi Keberhasilan Sebuah Test 51](file:///C:\Users\asus2\OneDrive\Documents\Buku%20KP%20SOfco%20Graha\Buku%20KP%20bisa%20diedit.docx#_Toc20862406)

[Gambar 4.4.5 Contoh Program Login Secara Otomatis 51](file:///C:\Users\asus2\OneDrive\Documents\Buku%20KP%20SOfco%20Graha\Buku%20KP%20bisa%20diedit.docx#_Toc20862407)

[Gambar 4.5.6 Contoh Program Lengkap Automatic Test Satu Bagian Test Case 53](file:///C:\Users\asus2\OneDrive\Documents\Buku%20KP%20SOfco%20Graha\Buku%20KP%20bisa%20diedit.docx#_Toc20862408)

[Gambar 4.5.7 Contoh Program untuk Merekam Aktifitas ke Database untuk Tujuan Dokumentasi Otomatis 53](file:///C:\Users\asus2\OneDrive\Documents\Buku%20KP%20SOfco%20Graha\Buku%20KP%20bisa%20diedit.docx#_Toc20862409)

[Gambar 4.6.1 Contoh Program XML Menjalankan Berbagai Test Dalam Satu Program 54](file:///C:\Users\asus2\OneDrive\Documents\Buku%20KP%20SOfco%20Graha\Buku%20KP%20bisa%20diedit.docx#_Toc20862410)

[Gambar 4.6.2 Contoh Program Mereset Status Running Untuk Kebutuhan Pencatatan Dokumentasi Otomatis 55](file:///C:\Users\asus2\OneDrive\Documents\Buku%20KP%20SOfco%20Graha\Buku%20KP%20bisa%20diedit.docx#_Toc20862411)

[Gambar 4.6.3 Contoh Program Lengkap XML Untuk Pengujian Otomatis Hris Meratus 57](file:///C:\Users\asus2\OneDrive\Documents\Buku%20KP%20SOfco%20Graha\Buku%20KP%20bisa%20diedit.docx#_Toc20862412)

[Gambar 4.7.1 Contoh Program Untuk Melakukan Perekaman Dokumentasi Otomatis dari Database ke Exel 58](file:///C:\Users\asus2\OneDrive\Documents\Buku%20KP%20SOfco%20Graha\Buku%20KP%20bisa%20diedit.docx#_Toc20862413)

# DAFTAR TABEL

[Table 4.2.1 Contoh Test Scenario 43](#_Toc19959092)

|  |
| --- |
| *(Halaman ini sengaja dikosongkan)* |

# BAB I PENDAHULUAN

## 1.1 LATAR BELAKANG

Pendidikan memberikan dampak besar terhadap segala aspek kehidupan. setiap manusia berhak memperoleh pendidikan yang layak demi terciptanya sumber daya manusia yang berkualitas dan dapat berkontribusi bagi bangsa dan negaranya. perkembangan ilmu pengetahuan dan teknologi saat ini mendorong seluruh aspek, khususnya lembaga pendidikan seperti perguruan tinggi, untuk memperbaiki kualitas metode pendidikan yang nantinya akan menunjang kebutuhan dunia industri. Oleh karenanya, dibutuhkan sebuah kegiatan dimana mahasiswa diberikan kesempatan untuk lebih mengenal dunia kerja dalam industri yang dapat mengimplementasikan disiplin ilmu teori yang didapatkannya di kelas.

Dengan program kerja Praktik ini, mahasiswa mempelajari secara langsung dengan pekerjaan yang ada di lapangan untuk memperluas wawasan dan cara berpikirnya. Para mahasiswa tidak hanya dituntut untuk memiliki ilmu pengetahuan yang luas namun juga harus memiliki keterampilan dan kemampuan untuk menerapkan ilmu yang dimilikinya. Sehingga tidak hanya memiliki ilmu pengetahuan secara hardskill, mahasiswa juga dilengkapi dengan softskill berupa problem solving yang akan dihadapi di lapangan nantinya.

Sebagai salah satu lembaga pendidikan, Institut Teknologi Sepuluh Nopember (ITS) juga menerapkan kerja praktik ini sebagai metode pembelajaran bagi mahasiswa yang diharapkan mampu memberikan pandangan terkait kondisi lingkungan kerja sesuai dengan bidang studi yang ditempuh dan mampu menyelaraskan pengetahuan teoritis dengan aplikasi praktis di lapangan.

PT. Sofco Graha sebagai salah satu perusahaan nasional dengan bidang disiplin yang linier dengan Teknik Komputer FTE-ITS diharapkan mampu menjadi pendukung upaya-upaya perguruan tinggi dalam meningkatkan mutu pendidikan melalui program kerja praktik ini.

## 1.2 BATASAN PERMASALAHAN

Setiap perusahaan memerlukan sebuah standardisasi atau pedoman utama agar setiap proses yang dijalankan teratur dan sesuai prosedur serta memiliki kualitas produk yang siap untuk dipasarkan, tidak terkecuali PT. Sofco Graha. PT. Sofco Graha sebagai sebuah perusahaan yang memiliki klien dari kalangan perusahaan-perusahaan besar tentunya juga menerapkan hal tersebut khususnya dalam proses komunikasi baik secara internal maupun eksternal dengan partner kerjasama maupun dengan publik sebagai sasarannya guna menjaga kerahasian dan privasi perusahaan serta menerapkan kontrol terhadap kualitas produk guna menjaga performa baik dari produk yang akan jual kepada klien perusahaan. Oleh karena itu, dalam kerja praktik di PT. Sofco Graha kali ini kami akan membahas tentang Pengujian Fungsional Meratus Human Resources Software menggunakan Selenium yang diterapkan di PT. Sofco Graha dalam menunjang proses produksi perangkat lunak berbasis *web application* yang menjadi produknya.

## 1.3 TUJUAN

Adapun tujuan dari kerja Praktik ini dapat dilihat dari dua sudut pandang sebagai berikut :

1. *Secara Umum*
   1. Terciptanya suatu hubungan yang sinergis dan terarah antara dunia perguruan tinggi dan dunia kerja sebagai pengguna outputnya.
   2. Membuka wawasan mahasiswa agar dapat mengetahui dan memahami aplikasi ilmunya di dunia industri.
   3. Mahasiswa dapat memahami dan mengetahui sistem kerja di dunia industri sekaligus mampu mengadakan pendekatan masalah secara utuh.
   4. Menumbuhkan dan menciptakan pola berpikir konstruktif yang lebih berwawasan bagi mahasiswa.
2. *Secara Khusus*
   1. Untuk memenuhi beban satuan kredit semester (SKS) yang harus ditempuh sebagai persyaratan akademis di Departemen Teknik Komputer ITS.
   2. Memperdalam pengetahuan, sikap, dan keterampilan profesi melalui penerapan ilmu, latihan kerja, dan pengalaman teknik yang diterapkan di PT. Sofco Graha
   3. Memperdalam pengetahuan mahasiswa dengan mengenal dan mempelajari secara langsung penerapan teknologi mengenai komputerisasi dibidang *Human Resources* dalam suatu perusahaan.
   4. Membuat dan mempelajari cara pembuatan *Automated testing* yang diterapkan pada Meratus Human Resources Software sebagai salah satu produk PT. Sofco Graha
   5. Mengembangkan hubungan baik antara pihak perguruan tinggi dengan PT. Sofco Graha.

## 1.4 BENTUK KEGIATAN

Dalam proses Kerja Praktik di PT. Sofco Graha, kami ingin mengetahui secara langsung proses-proses yang ada di dalamnya dan sebagai bahan perbandingan terhadap apa yang telah kami terima di dalam perkuliahan. Oleh karena itu, sebagai wujud nyatanya kami terjun langsung dalam proses tersebut sehingga kami mengetahui secara nyata bagaimana bentuk dunia kerja sebenarnya, khususnya terhadap hal-hal yang berhubungan dengan disiplin ilmu yang kami pelajari. Bentuk kegiatan kami ini disesuaikan dengan kebijaksanaan di perusahaan. Berikut bentuk teknis kegiatan :

* + 1. Penyusunan laporan sementara yang dilakukan pada minggu terakhir kegiatan kerja Praktik dilaksanakan. Selanjutnya laporan sementara ini dimintakan persetujuan kepada pembimbing kerja Praktik yang bersangkutan. Dalam memberikan persetujuan terhadap laporan sementara ini maka pembimbing lapangan memberikan penilaian terhadap pelaksanaan Kerja Praktik yang dilaksanakan oleh mahasiswa. Laporan sementara ini selanjutnya dijadikan dasar untuk pembuatan laporan resminya. Penilaian dari pembimbing lapangan akan diserahkan pada dosen pembimbing kerja Praktik di Departemen Teknik Komputer - ITS Surabaya.
    2. Penilaian kedua diberikan berdasarkan hasil penyusunan laporan resmi yang dibuat oleh mahasiswa selesai dilaksanakannya kerja Praktik.
    3. Penilaian ketiga diberikan berdasarkan hasil presentasi laporan resmi kerja Praktik.
    4. Perusahaan tempat dilaksanakannya kerja Praktik berhak menerima laporan resmi dari mahasiswa peserta kerja Praktik.

## 1.5 TEMPAT DAN WAKTU PELAKSANAAN

Berikut tempat dan waktu pelaksanaan kerja Praktik yang dilakukan :

Tempat : PT. Sofco Graha

Berbek Industri III no 17, Berbek, Waru, Sidoarjo - Jawa Timur 61256

Waktu : 8 Juli – 8 Agustus 2018

## 1.6 METODE PENULISAN

Metode penulisan yang digunakan dalam pelaksanaan Kerja Praktik ini antara lain :

1. Tahap Persiapan
   * 1. Metode Pengumpulan data melalui pengamatan langsung mengenai budaya kerja dan proses kerja yang ada di PT. Sofco Graha
     2. Metode Literatur yang diperoleh dari buku, jurnal, website, maupun referensi yang tersedia
     3. Metode diskusi yang dilakukan dengan pembimbing Kerja Praktik dan karyawan di PT. Sofco Graha
2. Tahap Pembuatan dan Pengembangan Program

Pembuatan *Automatic Sofwere Tester* dengan Selenium sesuai arahan dari pembimbing Kerja Praktik di PT. Sofco Graha

## 1.7 SISTEMATIKA PENULISAN

Laporan Kerja Praktik ini terdiri atas lima bab dengan rincian sebagai berikut :

1. BAB I : PENDAHULUAN

Pada BAB I dibahas mengenai latar belakang, tujuan, waktu dan tempat pelaksanaan Kerja Praktik, metode penulisan, serta sistematika penulisan.

1. BAB II : TINJAUAN UMUM PERUSAHAAN

Pada BAB II dibahas mengenai profil singkat dari PT. Sofco Graha.

1. BAB III : TINJAUAN PUSTAKA

Pada BAB III dibahas mengenai teori-teori penunjang dalam pembuatan program *Automatic Softwere Tester* dengan Menggunakan Selenium yang diterapkan kepada Meratus Human Resources Software oleh PT. Sofco Graha.

1. BAB IV : DESAIN

Pada BAB IV dibahas mengenai desain dan rencana dari apa yang akan kami buat selama kerja praktik dalam hal pembuatan *Automatic Softwere Tester* dengan Menggunakan Selenium yang diterapkan kepada Meratus Human Resources Software.

1. BAB V PEMBAHASAN

Pada BAB IV dibahas mengenai pembuatan program *Automatic Softwere Tester* dengan Menggunakan Selenium yang diterapkan oleh PT. Sofco Graha untuk produknya yaitu Meratus Human Resources Software

1. BAB VI PENUTUP

Pada BAB V dibahas mengenai kesimpulan dan saran selama melakukan Kerja Praktik yang telah dilakukan oleh penulis.

1. DAFTAR PUSTAKA

Pada bagian ini berisi sumber dari literatur yang berkaitan dengan topik yang dibahas dalam laporan kerja praktik.

1. LAMPIRAN

Pada bagian ini berisi dokumen tambahan yang melengkapi laporan kerja praktik.

# BAB II TINJAUAN UMUM PERUSAHAAN

## 2.1 SEJARAH SINGKAT PERUSAHAAN



Gambar 2.1.1 Logo PT. Sofco Graha

Sumber: https://urbanhire.co/

PT Sofco Graha didirikan pada tahun 1983, dengan visi untuk membantu perusahaan-perusahaan di Indonesia masuk ke Era Komputerisasi dalam hal penerapan Komputerisasi Akuntansi di perusahaan. Pada tahun 2012, sebagai salah satu klien terbesar, PT Konimex ikut ambil bagian dalam saham PT Sofco Graha.

Produk-produk yang telah PT. Sofco Graha hasilkan meliputi ERP (Enterprise Resources Planning) untuk Perusahaan Manufacturing dan Consumer Goods Distribution, HRIS (Human Resource Information System).

Dengan pengalaman selama lebih dari 30 tahun, membuat PT. Sofco Graha sangat mengenal karakteristik proses bisnis dan perilaku perusahaan-perusahaan yang ada di Indonesia. Selama itu pula PT. Sofco Graha ikut mensupport bisnis klien dengan memberikan layanan maintenance untuk aplikasi yang dihasilkan, sehingga klien bisa lebih fokus kepada bisnisnya tanpa harus kuatir apabila timbul kebutuhan untuk mengembangkan atau memodifikasi aplikasi mengikuti perkembangan proses bisnisnya. PT. Sofco Graha juga selalu membantu mendorong perusahaan untuk memanfaatkan Teknologi Informasi lebih jauh, untuk membantu perencanaan, pengambilan keputusan, operasional perusahaan melalui berbagai fitur seperti DSS (Decision Support System), EWS (Early Warning System), Business Intelligent dan lainnya. Penerapan Open Source Applications, Web based Applications, Mobile Applications, Cloud systems pada produk kami juga menunjukkan bahwa PT. Sofco Graha selalu mau memberikan yang terbaik bagi kliennya.

## 2.2 PROFIL PERUSAHAAN

Nama Instansi : PT. Sofco Graha

Alamat : Berbek Industri III no 17, Berbek, Waru,

Sidoarjo - Jawa Timur 61256

(Gedung PT. Marga Nusantara Jaya)

No. Telp/Fax : (031) 842 0861/ (031) 843 5460

Website : <https://www.sofcograha.co.id/>

## 2.3 FILOSOFI PERUSAHAAN

Sepeti yang tertera pada *tagline* yang dimiliki PT. Sofco Graha yaitu *“Solutions for Companies”,* perusahaan selalu berusaha memberikan dan membantu perusahaan-perusahaan klien untuk dapat masuk ke era Komputeriasi dalam proses menjalankan perusahaanya. Memberikan kemudahan dalam pemanfaatan teknologi informasi yang mudah, cepat, dan aman demi menunjang proses bisnis suatu perusahaan yang menjadi klien PT. Sofco Graha.

## 2.4 VISI DAN MISI

**Visi**

Membantu perusahaan-perusahaan di Indonesia masuk ke Era Komputerisasi dalam hal penerapan Komputerisasi Akuntansi di perusahaan.

**Misi**

* + 1. Membangun SDM yang bertanggungjawab, bermutu, dan dapat dipercaya
    2. Melihat kondisi pasar dan lapangan untuk selalu mengetahui kebutuhan konsumen terkini
    3. Menghasilkan produk yang berkualitas dan selalu mengikuti teknologi terkini
    4. Membangun relasi yang baik dengan konsumen
    5. Membuka peluang bermitra dengan pihak lain untuk memperluas penetrasi pasar

## 2.5 PRODUK DAN LAYANAN

Berikut adalah produk dan layanan dari PT. Sofco Graha:

Payroll Softwere

gaji.id adalah sebuah *payroll application* berbasis *Cloud* yang dikembangkan dengan tujuan membantu perusahaan untuk mempermudah pengelolaan pembayaran gaji karyawan setiap bulannya, termasuk variasi tunjangan dan potongan gaji kepada karyawan, juga perhitungan pajak dan premi ke BPJS. Di samping itu juga membantu perusahaan untuk menghasilkan dokumen pelaporan pajak penghasilan karyawan yang akan diberikan kepada kantor pajak, serta form-form BPJS yang akan diberikan kepada BPJS

IT Consulting

Memiliki pengalaman selama lebih dari 30 tahun untuk memberikan berbagai solusi untuk berbagai masalah administratif perusahaan, membantu melakukan berbagai analisa data perusahaan, mencari bottleneck dan sumber masalah dalam operasional perusahaan, dengan tujuan untuk meningkatkan tidak hanya efisiensi kerja perusahaan namun juga daya saing perusahaan dalam dunia bisnis.

Tailor Made Applications

Mengembangkan aplikasi ERP (enterprise resource planning) Manufacturing, ERP Distribution bagi Consumer Goods Distribution, Human Resources Information System secara tailor made bagi klien kami. Melakukan maintenance terhadap aplikasi-aplikasi tersebut, untuk menjamin keberlangsungan jalannya aplikasi tersebut dan agar aplikasi kami dapat terus dikembangkan sesuai dengan perkembangan proses bisnis klien kami.

# BAB III TINJAUAN PUSTAKA

## WEB APPLICATION

*Website*merupakan halaman situs sistem informasi yang dapat diakses secara cepat.*Website*ini didasari dari adanya perkembangan teknologi informasi dan komunikasi.Sedangkan *Web Application*  adalah suatu [aplikasi](https://id.wikipedia.org/wiki/Aplikasi) yang diakses menggunakan [penjelajah web](https://id.wikipedia.org/wiki/Penjelajah_web) melalui suatu jaringan seperti Internet atau Intranet. Bahasa yang digunakan untuk *Web Application* yaitu HTML, CSS, dan JavaScript. Selain itu juga menggunakan bahasa pemrograman web seperti PHP, Ruby, atau Python. Biasanya untuk kebutuhan yang lebih kompleks dapat menggunakan *framework* seperti *Rails,*[*Django*](https://www.djangoproject.com/) dan *CakePHP.* *Web Application* hampir selalu menggunakan database sehingga dapat disebut dinamis.

Keunggulan *Web Application* adalah tidak mengharuskan spesifikasi komputer yang tinggi terutama RAM yang besar. Selain itu, dapat dijalankan pada sistem operasi manapun seperti pada *Linux, Windows* maupun MAC. Walaupun dapat diakses sengan sistem operasi manapun tetapi tetap membutuhkan *Web Browser* dan akses Internet. Memogram web application dapat dilakukan dengan mudah apabila sudah memahami bahasa pemograman web seperti HTML, CSS, JavaScript dan PHP.

Kekurangan *Web Application* adalah membutuhkan koneksi ke web server yang stabil agar saat aplikasi dijalankan akan berjalan dengan baik dan lancar, serta untuk kebutuhan pengujian terhadap aplikasi yang telah dikembangkan akan membutuhkan waktu lama jika tidak diimbangi dengan kecepatan web server. Selain itu, *Web Application* masih memiliki fitur yang terbatas daripada *Mobile Application*. *Mobile Application* dapat diprogram dalam keadaan Offline sehingga, lebih efisien. Dan juga *Mobile Application* dapat melakukan akses dan kontrol terhadap fitur-fitur yang dimiliki oleh *smartphone* untuk memperkaya fungsinya. Pada jaman sekarang, *Mobile Application* lebih banyak digunakan oleh *programmer* Aplikasi.

## SELENIUM

Selenium adalah sebuah *framework* portable yang digunakan untuk tujuan pengujuan terhadap sebuah *web application*. Selenium merupakan *open-source softwere* yang rilis berdasarkan lisensi dari Apache License 2.0. Selenium menyediakan alat untuk dapat membuat test fungsional dari web application tanpa harus mempelajari *test scripting language* (Selenium IDE). Selenium juga menyediakan test dalam domain-specific language (Selenese), agar supaya kita dapat menuliskan test dalam berbagai Bahasa pemrograman popular seperti C#, Groovy, Java, Perl, PHP, Python, Ruby, dan Scala. Test yang dihasilkan dapat dijalankan di berbagai *web browser*.

Dalam mengguakan Selenium dibutuhkan sebuah *Locators*. *Locators* adalah sebuah format code untuk memberitahu GUI Elements yang mana yang ingin dioperasikan oleh Selenium IDE. Jenis-jenis *Locators* yang terdapat pada Selenium dapat dilihat pada tabel dibawah ini.

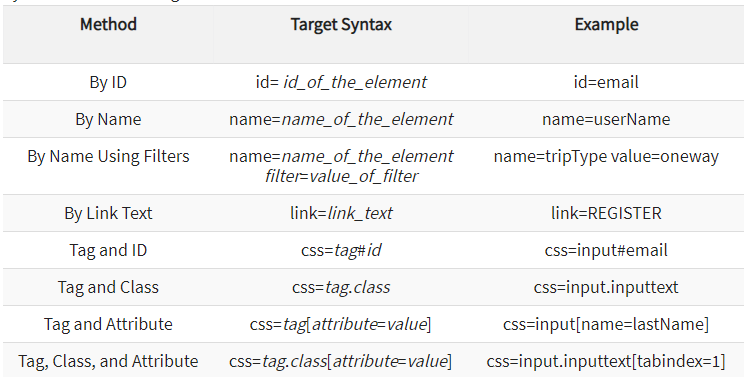


Table 3.1 Tabel Locators Selenium

## SOFTWARE TESTING

Software testing adalah proses eksekusi dan menjalankan suatu program aplikasi dengan tujuan untuk menemukan kesalahan yang dihasilkan selama proses pengemabangan program aplikasi tersebut. Kesalahan tersebut dapat berupa masukan, proses yang terjadi, keluaranan, dan respon aplikasi terhadap kondisi-kondisi tertentu yang mungkin terjadi selama program aplikasi tersebut digunakan oleh pengguna. *Software* testing merupakan sebuah elemen kritis dalam mempertahankan kualiatas dari suatu *software* dan merepresentasikan kajian pokok dari spesifikasi, desain dan pengkodean yang telah dibangun oleh sebuah perusahaan penghasil *software*. Dalam proses pengembangan sebuah aplikasi, *software testing* akan menghabisakan upaya sebesar 30-40% dari keseluruhan pekerjaan proyek. Dalam melakukan pengujian tersebut, perlu adanya sasaran pengujian dan karakteristik pengujian, yaitu:

### Sasaran Pengujian

Dalam melakukan pengujian terdapat sasaran yang diharapkan dapat tercapai yaitu:

1. Pengujian didesain secara sitematis untuk mecari kelas kesalahan yang berbeda sehingga dapat menjangkau.

2. Pengujian dilakukan dalam waktu dan usaha minimum untuk meminimalisir keterlambatan dari jadwal yang telah dibuat dan membuat alokasi usaha dan waktu yang lebih besar untuk usaha memperbaiki error yang ditemukan dan melanjutkan proses pengembangan aplikasi.

3. Pengujian yang sukses adalah pengujian yang dapat menghasilkan kesalahan dalam software dan dapat menunjukkan reabilitas software, serta dapat melaporkan hasil pengujianya dalam suatu dokumentasi yang dapat dimengerti oleh orang lain.

4. Pengujian tidak dapat memperlihatkan tiak adanya kesalahan karena jika hal tersebut terjadi berarti proses pengujian tidak berjalan dengan baik. Jika hal tersebut memang benar-benar terjadi keluaran yang dihasilkan oleh proses pengujian adalah hasil reabilitas aplikasi.

### Karakteristik Pengujian

Karakteristik yang dilakukan selama pengujian adalah:

1. Testing dimulai pada level modul dan bekerja keluar kearah integrasi pada sistem berbasiskan computer.

2. Teknik testing yang berbeda sesuai dengan poin-poin yang berbeda pada waktunya.

3. Testing diadakan oleh software developer dan untuk proyek software besar yang tingkat komplesitanya tinggi dilakukan oleh grup testing independent.

4. Testing dan Debugging adalah aktifitas yang berbeda tetapi debugging harus diakomodasikan pada setiap strategi testing

## PRINSIP PENGUJIAN

Dalam melakukan pengujian terhadap suatu software, perlu diperhatikanya beberapa pronsip:

Pengujian harus ditelusuri sampai ke persyaratan pelanggan

Pengujian harus direncanakan lama sebelum dimulai sesuai spesifikasi software yang akan dibuat

Prinsip Pareto berlaku untuk poengujian perangkat lunak, maksudnya dari 80% kesalahan yang ditemukan selama pengujian dapat ditelusuri sampai 20% dari semua modul program.

Pengujian dimuali dari “yang kecil” dan berkembang ke pengujian “yang besar”.

Pengujian yang medalam tidak mungkin dilakukan karena tidak mungkin mengeksekusi setiap kombinasi jalur skema pengujian, dikarenakan jumlah permutasi untuk program menengah-pun sangat besar

Untuk hasil dan proses yang efektif, pengujian dilakukanoleh pihak ketiga yang *independent.*

Sasaran utama dari pembuatan desain test case adalah untuk mendapatkan serangkaian pengujian yang memiliki kemungkinan tertinggi di dalam pengungkapan kesalahan atau *error* pada perangkat lunak.

## QUALITY ASSURANCE

Quality Assurance atau yang lebih dikenal dengan sebutan QA merupakan pengujian atau testing terhadap suatu produk sistem atau aplikasi untuk memastikan bahwa sistem yang dikembangkan sesuai dengan kebutuhan yang telah ditentukan sebelumnya sehingga menghasilkan sistem yang terjamin kualitasnya. Pengujian terhadap aplikasi biasa dijalankan oleh QA Engineer.

QA Engineer biasanya ditemukan di beberapa Perusahaan IT dan memiliki peranan penting dalam pembangunan sebuah sistem. QA Engineer akan melakukan testing mulai dari alur sistem, desain, sampai hasil akhir aplikasi. Hasil pengujian QA Engineer berupa alur sistem, bug, serta masukan aplikasi yang diuji akan didokumentasikan dalam sebuah laporan Quality Assurance. Seorang QA Engineer juga harus mampu berkomunikasi dengan pihak-pihak yang berhubungan dengan sistem seperti designer, back end, serta project manager. Dalam melakukan proses testing, seorang QA Engineer harus memiliki beberapa kriteria kemampuan tertentu. Kemampuan tersebut diantaranya yaitu:

### 3.4.1 Mindset Pengujian

Seorang QA Engineer yang memiliki tugas utama untuk melakukan pengetesan terhadap suatu sistem serta memastikan sistem memiliki kualitas terjamin harus memiliki pola pikir yang luas mengenai penggunaan sistem. QA Engineer harus mampu memikirkan berbagai kemungkinan yang akan dilakukan oleh pengguna saat akan menggunakan sistem.

### 3.4.2 Analisa dan Pengujian Fungsional

Seorang QA Engineer harus mampu memahami analisis terhadap kebutuhan fungsional sistem yang akan dibangun serta melakukan pengujian terhadap sistem untuk memastikan bahwa sistem yang telah selesai dibuat sudah memenuhi kebutuhan yang telah ditentukan sebelumnya.

### 3.4.3 Perbaikan Proses

Seorang QA Engineer harus mampu memahami setiap proses yang terdapat pada sistem serta memberikan arahan perbaikan untuk beberapa proses yang dirasa kurang efisien.

### 3.4.4 Pengujian Keamanan

Seorang QA Engineer harus mampu melakukan pengujian keamanan secara detail sehingga sistem yang ada memiliki jaminan keamanan yang tinggi. Pengujian keamanan bisa dilakukan dengan beberapa hal, salah satu contohnya yaitu pengujian terhadap proses login. QA Engineer harus memastikan bahwa proses login yang ada sudah sesuai dengan standar ketentuan yang ada. Contoh pengujian pada proses login dapat dilakukan dengan melakukan pengujian apakah password yang dimasukkan memiliki batas minimal karakter, pengaruh antara kombinasi huruf besar dan kecil pada password, dll.

### 3.4.5 Pengujian Performa

Seorang QA Engineer harus mampu melakukan pengujian terhadap performa dari sistem. Beberapa contoh pengujian yang dapat dilakukan untuk menguji performa sistem dapat dilakukan dengan melakukan pengujian sistem terhadap respon tertentu apakah sistem akan menerima respon dengan cepat atau tidak. Dengan adanya pengujian performa, maka diharapkan QA Engineer dapat memastikan bahwa sistem yang dihasilkan memiliki performa serta kualitas yang sesuai.

### 3.3.6 User Acceptance Testing

Seorang QA Engineer harus mampu menghasilkan dokumen yang dijadikan bukti bahwa sistem yang telah dibuat sudah dapat diterima oleh pengguna apabila hasil pengujian tersebut sudah bisa dianggap memenuhi kebutuhan pengguna. Dari hasil dokumentasi tersebut, maka dapat diambil kesimpulan apakah sistem yang telah diuji dapat diterima atau tidak.

Proses testing dapat dilakukan dengan 2 cara yaitu testing manual serta testing otomatis. Testing manual dilakukan oleh QA Engineer langsung dengan berbagai test case saat menjalankan sistem. Sedangkan testing otomatis merupakan suatu testing yang dilakukan oleh QA Engineer dengan memanfaatkan software testing yang ada sehingga proses pengujian menjadi lebih efisien. Salah satu contoh software testing yang digunakan untuk testing yaitu Selenium IDE. Selenium IDE akan merekam kegiatan QA Engineer saat menjalankan suatu sistem lalu menampilkan detail kegiatan yang telah dilakukan QA Engineer terhadap suatu sistem berdasarkan test case tertentu.

## JENIS-JENIS PENGUJIAN SOFTWARE

Pada dasarnya terdapat banyak jenis pengujian yang dilakukan pada software yang dihasilkan oleh suatu perusahaan IT. Banyak kasus uji yang mewakili situasi baik yang normal maupun kejadian tertentu yang harus disiapkan untuk masing – masing scenario. Tetapi ada 8 jenis pengujian yang sering dilakukan oleh seorang Software Tester, diantaranya:

1. Performance Testing

Performance test adalah integration dan usability test yang menentukan apakah sistem atau sub-sistem dapat memenuhi kriteria kinerja berbasis waktu seperti response time atau throughput. Response time menentukan batas waktu maksimum yang diijinkan dari respon software untuk query dan update. Throughput menentukan jumlah minimum query dan transaksi yang harus diproses per menit atau per jam.

1. System Testing

System test adalah integration test dari behavior seluruh sistem atau independent sub-sistem. Sistem testing biasanya dilakukan pertama kali oleh pengembang atau personil pengujian untuk memastikan bahwa keseluruhan sistem berfungsi dan bahwa sistem telah memenuhi persyaratan pengguna (user requirement). System testing biasanya dilakukan di akhir setiap iterasi untuk mengidentifikasi isu – isu penting, seperti masalah performance yang perlu ditangani di iterasi berikutnya. Biasanya test ini harus dilakukan sesering mungkin.

1. Unit Testing

Unit testing adalah proses metode pengujian individual, class, atau komponen sebelum mereka terintegrasi dengan perangkat lunak lainnya. Tujuan dari unit testing adalah untuk mengidentifikasi dan memperbaiki kesalahan sebanyak mungkin sebelum modul – modul digabungkan menjadi unit perangkat lunak yang lebih besar, seperti program, class dan subsistem. Kesalahan menjadi lebih sulit dan mahal untuk ditemukan dan diperbaiki ketika banyak unit telah digabungkan.

Unit testing memerlukan implementasi dari driver dan/atau stub. Stub adalah class atau method dummy yang dapat dipanggil namun biasanya tidak melakukan apapun kecuali mengembalikan tipe yang diperlukan. Modul driver adalah program yang menjalankan method atau fungsi dari class yang akan ditest.

Berikut adalah langkah yang harus dilakukan :

Menentukan nilai dari parameter input

Memanggil unit yang dites, melewatkannya dengan parameter input

Menerima parameter kembalian dari unit yang dites dan mencetaknya, menampilkannya, atau mengetes hasilnya terhadap hasil yang diharapkan.

1. Integration Testing

Integration test adalah mengevaluasi behavior dari kelompok method atau class. Tujuan dari integration test adalah untuk mengidentifikasi kesalahan yang tidak dapat dideteksi oleh unit testing. Kesalahan tersebut mungkin disebabkan oleh beberapa masalah, diantaranya :

Interface incompatibility, misalnya sebuah method melewatkan parameter dengan tipe data yang salah ke method lainnya

Parameter values, misalnya sebuah method mengembalikan nilai yang tidak terduga seperti nomor negatif untuk harga

Run-time exceptions, misalnya method menyebabkan kesalahan seperti ―out of memory atau ― file already in use karena ada konflik kebutuhan sumber daya

Unexpected state interactions, misalnya state dari dua atau lebih objek yang berinteraksi menyebabkan kesalahan yang kompleks seperti ketika method class Order menjalankan satu kesalahan dari semua kemungkinan state objek Customer

Beberapa masalah di atas merupakan kesalahan paling umum yang sering ditemui dalam integration testing, tetapi sebenarnya masih banyak masalah lainnya yang dapat menjadi penyebab kesalahan (error).

1. Usability Testing

Usability test adalah test untuk menentukan apakah method, class, subsistem, atau sistem telah memenuhi persyaratan pengguna. Oleh karena banyaknya tipe persyaratan sistem baik yang fungsional maupun non-fungsional, maka banyak tipe dari usability test yang harus dilakukan di waktu yang berbeda. Umumnya usability test mengevaluasi persyaratan fungsional dan kualitas dari user interface. User berinteraksi dengan sistem untuk menentukan apakah fungsi telah seperti yang diharapkan dan apakah user interface membuat sistem dapat mudah digunakan. Pengujian ini sering dilakukan untuk mendapatkan feedback yang cepat dalam meningkatkan interface dan mengkoreksi kesalahan dalam komponen perangkat lunak.

1. Smoke Testing

Smoke testing adalah system test yang biasanya dilakukan setiap hari atau beberapa kali per minggu. Build and smoke test sangat penting karena menyediakan feedback yang cepat dalam masalah yang signifikan.

1. Stress Testing

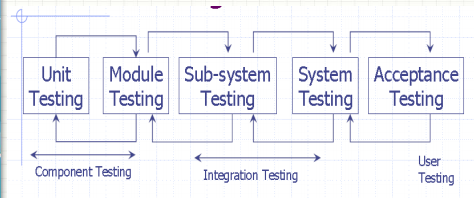
Stress Testing adalah pengujian yang biasanya dilakukan dalam membuat sebuah website, dimana stress testing dilakukan untuk mengetahui sekuat apa server website kita menampung visitor dalam website tersebut, dengan cara melakukan hit dummy ke website menggunakan tools.

1. User Acceptence Test

User acceptance test digunakan untuk menentukan apakah sistem yang dikembangkan telah memenuhi kebutuhan pengguna. Dalam beberapa proyek, acceptance testing dilakukan pada putaran terakhir proses pengujian yaitu sebelum sistem diserahkan kepada user.

Acceptance Testing biasanya dilakukan setelah rangkaian testing seperti Unit Testing, Integration Testing, dan System Testing selesai dan menggunakan metode Black Box Testing, dengan mengunakan dokumen test case untuk di presentasikan di akhir ke user / client.

Sehingga, secara umum gambaran dari proses testing dapat dilihat pada gambar dibawah ini:



1. Gambar 3.1.1 Proses Testing  
   Sumber : <http://staffnew.uny.ac.id/>

## 3.7 PENGUJIAN SISTEM

Pengujian sistem adalah pengujian program perangkat lunak yang lengkap dan terintegrasi. Pengujian perangkat lunak dapat dibedakan menjadi dua yaitu Black Box Testing dan White Box Testing.

* + 1. Black Box Testing

Black Box Testing atau yang sering dikenal dengan sebutan pengujian fungsional merupakan metode pengujian Perangkat Lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau Program. Dalam pengujian ini, tester menyadari apa yang harus dilakukan oleh program tetapi tidak memiliki pengetahuan tentang bagaimana melakukannya.

Kelebihan *Black Box Testing* yaitu:

1. Efisien untuk segmen kode besar
2. Akses kode tidak diperlukan
3. Pemisahan antara perspektif pengguna dan pengembang

Kelemahan *Black Box Testing*yaitu:

1. Cakupan terbatas karena hanya sebagian kecil dari skenario pengujian yang dilakukan
2. Pengujian tidak efisien karena keberuntungan *tester* dari pengetahuan tentang perangkat lunak internal
   * 1. White Box Testing

White Box Testing merupakan metode pengujian perangkat lunak di mana struktur internal diketahui untuk menguji siapa yang akan menguji perangkat lunak. Pengujian ini membutuhkan pengetahuan internal tentang kemampuan sistem dan pemrograman.

Kelebihan *White Box Testing* yaitu:

1. Efisien dalam menemukan kesalahan dan masalah
2. Diperlukan pengetahuan tentang internal perangkat lunak yang sedang diuji bermanfaat untuk pengujian menyeluruh
3. Memungkinkan menemukan kesalahan tersembunyi
4. Membantu mengoptimalkan kode

Kelemahan *White Box Testing* yaitu:

1. Membutuhkan pengetahuan tingkat tinggi dari perangkat lunak internal yang sedang diuji
2. Membutuhkan akses kode

## PENGUJIAN APLIKASI MENGGUNAKAN SELENIUM

Penggunaan Selenium sebagai media testing merupakan hal yang biasa dilakukan untuk melakukan *Automated Testing*. *Automated Testing* adalah suatu test yang dilakukan dengan bergantung pada pra-scripted tes yang berjalan secara otomatis, fungsinya untuk membandingkan hasil yang diharapkan dengan hasil yang sebenarnya. Sehingga dapat mengetahui apakah aplikasi berjalan sesuai dengan apa yang diharapkan, menggunakan automated testing dapat dilakukan secara berulang. Sehingga jika hasilnya tidak sama dengan yang diharapkan maka akan mendapatkan bug. Sebelum melakukanya biasanya melakukan *Manual Testing* untuk mendapatkan hasil sebenarnya dari sebuah test dengan membuat sebuah alur pengujian.

*Manual Testing* sendiri adalah langkah untuk mencari cacat atau bug pada program perangkat lunak, pada metode ini tester/penguji memiliki peran penting sebagai pengguna akhir untuk pengecekan semua fitur aplikasi bekerja dengan benar. Penguji melakukan pengecekan secara manual tanpa menggunakan bantuan dari tools atau scripts, tujuannya adalah untuk memastikan jika aplikasi yang di uji bebas cacat dan aplikasi perangkat lunak dapat bekerja sesuai apa yang diharapkan. Manual testing juga berperan penting saat pengujian visual dimana automation tools tidak dapat melakukan.

Manual tester dapat mengetahui kontras antara button dan background yang lebih terang sehingga membuat user kesulitan mencari button dan memahami tindakan yang perlu diambil. User Interface (UI)feedback adalah hal yang tidak dapat ditemukan menggunakan automated testing.

## HUMAN RESOURCES DEVELOPMENT

HRD atau Human Resources Development, adalah bagian atau departemen dari perusahaan yang tugas utamanya mengelola sumber daya manusia di perusahaan, mulai dari tugas perencanaan yang sering disebut perencanaan SDM, rekrutmen sering disebut Rekrutmen dan Seleksi, pengembangan sering disebut Pelatihan dan Pengembangan, Manajemen Kinerja sering disebut Performance Management, gaji sering disebut Kompensasi dan Benefit dan menumbuhkan hubungan kerja yang sering disebut sebagai Hubungan Industrial atau hubungan Industrial.

Arti lain dari HRD Pengembangan Sumber Daya Manusia adalah proses menangani berbagai masalah pada ruang lingkup karyawan, pegawai, buruh, manajer dan pekerja lainnya untuk dapat menunjang aktivitas organisasi atau perusahaan untuk mencapai tujuan yang telah ditetapkan. Bagian atau unit yang biasanya mengurusi sdm adalah departemen sumber daya manusia.

Manajemen sumber daya manusia juga dapat diartikan sebagai suatu prosedur yang berkelanjutan yang bertujuan untuk memasok suatu organisasi atau perusahaan dengan orang yang tepat untuk ditempatkan pada posisi yang tepat dan posisi saat organisasi memerlukannya.

Pada prinsipnya dapat diartikan sebagai mengelolaan sumber daya manusia HRD terdiri dari berbagai kegiatan, yaitu: Perencanaan SDM, HR Mendapatkan, Mengembangkan SDM, HR Mempertahankan, HR Mempertahankan untuk membantu perusahaan menyadari Goal, Visi, Misi dan valuenya.

Sehingga HRD memiliki tugas dan tanggung jawab di bidang Perencanaan SDM, HR Mendapatkan, Mengembangkan SDM, HR Mempertahankan, HR Mempertahankan. Tugas, Tanggung Jawab dan Peran HRD dalam perusahaan, antara lain:

Persiapan

Ada dua hal yang perlu diperhatikan dalam hal persiapan, antara faktor internal dan faktor eksternal lainnya. Faktor internal meliputi jumlah kebutuhan dalam persiapan untuk karyawan baru, struktur organisasi, departemen terkait, dan sebagainya. Sedangkan faktor eksternal dalam hal persiapan termasuk hukum ketenagakerjaan, pangsa kondisi tenaga kerja, dan sebagainya.

Rekruitmen Tenaga Kerja

Rekrutmen adalah suatu proses untuk mencari calon atau karyawan yang dapat memenuhi kebutuhan sumber daya manusia dari organisasi atau perusahaan. Pada tahap HRD ini perlu menganalisis posisi yang ada untuk membuat deskripsi pekerjaan (job description) dan juga spesifikasi pekerjaan (job spesifikasi).

Seleksi Tenaga Kerja

Definisi Seleksi tenaga kerja adalah suatu proses yang dilakukan untuk menemukan pekerjaan yang tepat dari sekian banyak kandidat. Langkah yang diambil dalam proses seleksi tenaga kerja, yang melihat resume / CV, melakukan seleksi awal berdasarkan CV pelamar, memanggil pelamar untuk tes wawancara, tes calon karyawan dengan tes tertulis, proses wawancara / wawancara kerja, dan proses selanjutnya.

Pengembangan dan Evaluasi Karyawan

Agar tenaga kerja atau karyawan dapat memberikan kontribusi optimal kepada perusahaan atau organisasi, maka dia harus menguasai tugas pekerjaan dan tanggung jawab. Proses pengembangan dan evaluasi karyawan dilakukan sebagai pembekalan agar tenaga kerja dapat lebih terkontrol dan ahli di bidangnya, serta meningkatkan kinerja yang ada.

Penyedia Konpensasi dan Perlindungan Karyawan

Kompensasi adalah imbalan atau upah bagi karyawan secara teratur kontribusi organisasi atau perusahaan. Kompensasi harus tepat dan sesuai dengan kondisi pasar tenaga kerja yang ada di lingkungan eksternal untuk menghindari masalah tenaga kerja atau membahayakan organisasi atau perusahaan.

Sedangkan Fungsi HRD adalah:

* **Internal** : HRD bekerja sebagai pelatih untuk karyawan di perusahaan.
* **Eksternal** : HRD Karyawan yang memeilki konseling di luar kategori perusahaan dapat dilihat dari tingginya tingkat kemampuan dan kemauan dari kemampuan akan mengikuti.

|  |
| --- |
| *(Halaman ini sengaja dikosongkan)* |

# BAB IV

# DESAIN DAN IMPLEMENTASI SISTEM

Kerja praktek ini bertujuan untuk membuat tes skenario serta kode program untuk melakukan automated testing pada aplikasi HRIS Meratus yang digunakan untuk menguji fungsional aplikasi berjalan sesuai skenario atau tidak.

## 3.1 DESAIN

Secara garis besar, terdapat lima proses dalam pembuatan automated testing menggunakan Selenium.

Pembuatan Tes Skenario

Skenario tes berisi langkah-langkah tiap fitur yang ada yang didapatkan dari manual testing.

Memasang Library yang Dibutuhkan

Dalam pembuatan automated testing, kami menggunakan bahasa pemrograman Java pada Eclipse. Sebelum membuat kode program, terlebih dahulu memasang library baik dari library Selenium maupun library yang sudah disiapkan oleh perusahaan.

* + 1. Pembuatan Kode Program Page Object

Berisi Web Element yang diidentifikasi melalui atribut seperti ID, Name maupun Class guna dipanggil untuk dieksekusi pada Selenium sesuai alur tes skenario.

* + 1. Pembuatan Kode Program Automated Test

Setelah tes skenario telah dibuat, selanjutnya implementasi tes skenario dalam program. Setiap langkah pada tes skenario harus diimplementasikan. Setiap automated testing selalu dimulai dari Login system. Pada pembuatan kode program dibutuhkan library Selenium dan TestNG.

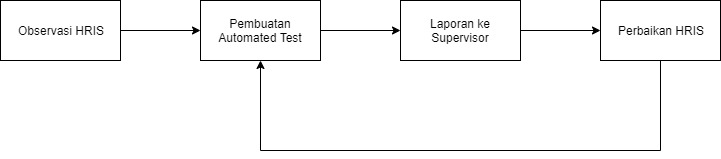
* + 1. Pembuatan Kode Program Running Test

Dalam pembuatan kode program Running Test menggunakan Bahasa pemrograman XML (eXtensible Markup Language) yang terbagi menjadi tiga bagian. Bagian pertama Running Test yang meliputi satu modul. Dalam satu modul terdapat beberapa kode program automated testing yang harus dijalankan secara bersamaan karena merupakan satu-kesatuan dalam satu modul sesuai dengan tes skenario. Selain itu kode program Running Test digunakan untuk menjalankan program yang sudah dijalankan jika program tersebut dibutuhkan pada modul lain.

* + 1. Pembuatan Automated Testing Report

Setelah kode program automated test dijalankan, maka secara otomatis diproses dan ditampilkan dalam bentuk data xls, hal ini bertujuan sebagai Laporan atau Dokumentasi setelah melakukan tes.

BAB V  
PEMBAHASAN

Dalam kerja praktik di PT. Sofco Graha, kami melakukan test terhadap produk yang telah dibuat dan dikembangkan oleh PT. Sofco Graha yaitu sebuah *Human Resources Software* untuk PT. Meratus Line. Dalam pelaksanaannya, kami melalui beberapa tahapan proses, yaitu melakukan manual testing untuk mengetahui alur kerja dan aliran data yang terjadi pada *software* tersebut, lalu membuat sebuah test scenario untuk patokan pembuatan program test otomatis yang akan dijalankan di server perusahaan, dan yang terakhir menelaah hasil dokumentasi otomatis sebagai keluaran proses test yang telah dijalankan untuk mendapatkan hasil test. Alur kerja Quality Assurance di PT. Sofco Graha sebagai berikut:

Gambar 4 Alur Kerja Quality Assurance (QA)

## 4.1 MELAKUKAN MANUAL TESTING

Manual Testing diperlukan untuk penyusunan alur pengujian. Setiap fungsi dalam sebuah website dilakukan Manual Testing untuk mengetahui alur pengujiannya yang kemudian dideskripsikan menjadi tes skenario. Setiap skenario dimulai dari proses login sampai eksekusi data seperti Input, Ubah dan Hapus. Berikut daftar modul yang akan di tes:

Surat Lamaran

Kebutuhan Tenaga Kerja

Man Power Planning

Proses Seleksi Karyawan

Rencana Tenaga Kerja

Mutasi Rekrut Baru

Mutasi Rangkap Jabatan

Mutasi Lepas Rangkap Jabatan

Mutasi Status Pekerjaan

Mutasi Homebase

Mutasi Terminasi

Mutasi Pembinaan

Absensi Karyawan Mutasi/Terminasi

Kebutuhan Training

Rencana Training

Realisasi Training

Trainee Performance Evaluation

Personal Competency Assessment

Evaluasi Pasca Training

Performance Appraisal

Rencana Perjalanan Dinas

Uang Muka

Realisasi Perjalanan Dinas

Penyelesaian Uang Muka

Surat Perintah Lembur

Surat Ijin dan Cuti (Selain Cuti A dan B)

Pindah Lokasi Kerja

Surat Cuti A

Surat Cuti B

Koreksi Sisa Cuti

Koreksi Tanggal Masa Berlaku

Cuti Masal

Daftar Peserta Askes

Pengajuan Klaim ke ASKES

Penggantian klaim dari ASKES

Daftar Peserta Jamsostek

Pengajuan Klaim Karyawan BukanPeserta Askes

Pinjaman

Anggota Koperasi

Potongan Koperasi

ESS

Administrasi Medical

Proses Hitung Premi JAMSOSTEK Bulanan

Proses Hitung Premi Askes Bulanan

Proses Jatah Cuti Tahunan

Absensi dari Check Clock

Proses Ambil dan Lengkapi Data di Lokasi Kerja

Proses Absensi Harian di Cabang

Data Hasil Proses Ambil Data Absensi

Data Hasil Proses Absensi Harian

Batal Proses Absensi

Tunjangan & Potongan Untuk Beberapa Periode

Proses Transfer Data dari Mesin Absensi

Absensi dari Finger Scan

Proses Hitung Tunjangan & Potongan

Data Hasil Proses Tunjangan & Potongan

Proses Batal Konfirmasi

Proses Batal Tunjangan & Potongan

## 4.2 PEMBUATAN SKENARIO TES

Skenario tes berisi langkah-langkah tiap fitur yang ada yang didapatkan dari manual testing. Isi dari Skenario tes sebagai berikut:

1. Nama Skenario

Berisi nama instruksi yang ada dilakukan pada suatu fitur beserta file kode program untuk memudahkan pencarian.

1. Precondition

Berisi data yang sudah tersedia didalam database dan siap digunakan.

1. Steps

Berisi langkah-langkah dalam mencapai sebuah instruksi.

1. Hasil yang diharapkan

Berisi perintah balasan atau respon ketika mengeksekusi setiap step.









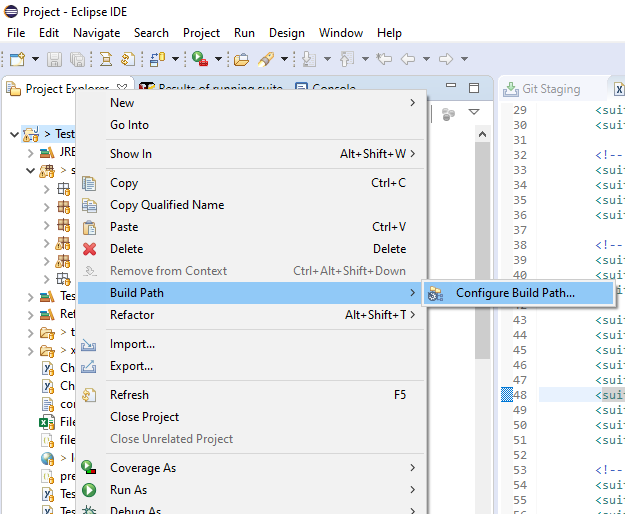
Table 4.2.1 Contoh Test Scenario

## 4.3 MEMASANG LIBRARY YANG DIBUTUHKAN

Dalam pembuatan automated testing, kami menggunakan bahasa pemrograman Java pada Eclipse. Sebelum membuat kode program, terlebih dahulu memasang library baik dari library Selenium maupun library yang sudah disiapkan oleh perusahaan. Berikut langkah-langkah dalam menambahkan library pada Eclipse IDE:

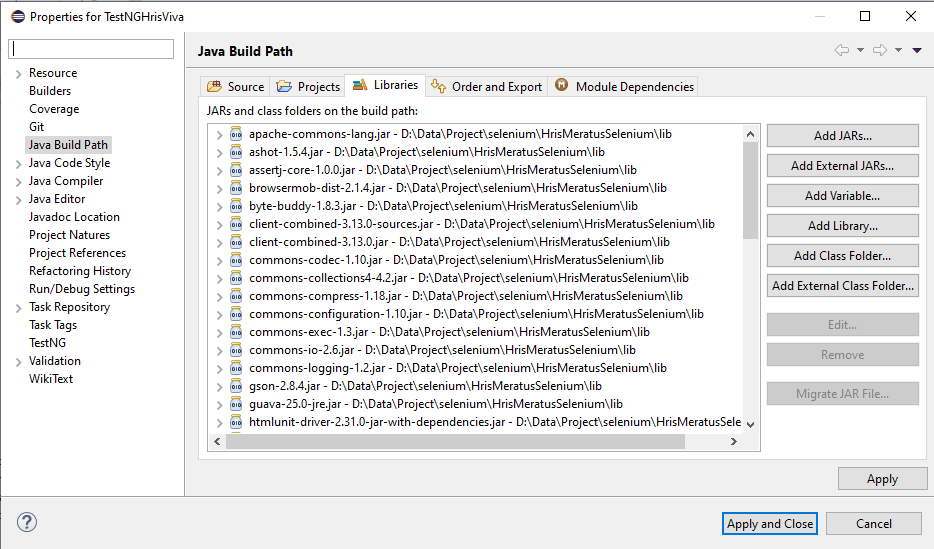
Buka project yang telah dibuat yang nanti berisi kode program untuk testing.

Kemudian klik kanan pada project dan pilih Build Path kemudian Configure Build Path­.



Gambar 4.3.1 Tangkap Layar Configure Build Path

Kemudian pada Libraries, tambahkan library yang telah disediakan.



Gambar 4.3.2 Tangkap Layar Menambahkan Library

## 4.4 PEMBUATAN KODE PROGRAM PAGE OBJECT

Dalam pembuatan kode program, dibagi menjadi empat bagian yaitu:

* Page Object: Berisi Web Element yang diidentifikasi melalui atribut seperti ID, Name maupun Class.
* Automated Testing: Berisi perintah untuk mengeksekusi Page Object secara terstruktur sesuai scenario tes yang telah dibuat.
* Running Test: Dalam satu modul terdapat beberapa eksekusi seperti Input, Edit dan Hapus. Setelah kode program dibuat, maka semua kode program tersebut di gabung dalam satu modul melalui program yang dibuat menggunakan Bahasa pemrograman XML.
* Automated Testing Report: Setelah kode program automated test dijalankan, maka secara otomatis diproses dan ditampilkan dalam bentuk data xls, hal ini bertujuan sebagai Laporan atau Dokumentasi setelah melakukan tes.

Berikut langkah-langkah pembuatan kode program Page Object:

1. Identifikasi terlebih dahulu atribut Web Element yang akan dieksekusi sesuai skenario tes melalui Inspect Element pada Google Chrome.
2. Dalam pembuatan tes skenario, banyak ditemukan perintah seperti mengisi form atau menekan tombol. Dalam pembuatan fungsi Page Object untuk pengisian form, alur program dimulai dari menunggu Web Element yang terkait muncul pada saat website sedang berproses, identifikasi atribut yang dideteksi berdasarkan nama dari Web Element tersebut. Setelah Web Element tersebut muncul, maka proses selanjutnya yaitu memberikan perintah atau masukan pada Web Element tersebut. Perintah tersebut seperti:

* sendKeys(): Mengakses Web Element dengan perintah memasukkan nilai.
* click(): Mengakses Web Element dengan perintah menekan.
* clear(): Mengakses Web Element dengan perintah membersihkan nilai pada Web Element tersebut.

Sehingga untuk contoh kode programnya seperti berikut:

public void isiTanggalRecruit(String tanggal, String bulan, String tahun) {  
wait.isElementVisible(By.name("stringMapped(TglTransRecruitdd)"));  
WebElement inputTanggal = driver.findElement(By.name("stringMapped(TglTransRecruitdd)"));  
inputTanggal.sendKeys(tanggal);  
  
wait.isElementVisible(By.name("stringMapped(TglTransRecruitmm)"));  
WebElement inputBulan = driver.findElement(By.name("stringMapped(TglTransRecruitmm)"));  
inputBulan.sendKeys(bulan);  
  
wait.isElementVisible(By.name("stringMapped(TglTransRecruityy)"));  
WebElement inputTahun = driver.findElement(By.name("stringMapped(TglTransRecruityy)"));  
inputTahun.sendKeys(tahun);}

Gambar 4.4.1 Contoh Program Page Object untuk Aktifiktas Test

Kemudian dalam pembuatan fungsi Page Object untuk akses tombol, dalam web HRIS Meratus, ketika ingin mengeksekusi data seperti menyimpan ataupun menghapus data, terlebih dahulu akan muncul Alert Pop Up yang berisi konfirmasi sebelum melakukan eksekusi tersebut, maka ditambahkan program untuk berpindah dari Web Page menuju Alert seperti berikut:

public void klikSimpan() {  
WebElement button = driver.findElement(By.name("save"));  
button.click();  
Alert alert = driver.switchTo().alert();  
alert.accept();  
pause(2000);}

Gambar 4.4.2 Contoh Tombol Simpan dimana Memiliki Deteksi Alert dan Memindah Controller Automatic Browser ke Alert Tersebut

1. Selain untuk mengakses tombol dan mengisi form, Terdapat juga Page Object untuk berpindah halaman. Atribut tersebut sering digunakan pada tombol Simpan dan Hapus. Berikut kode program untuk berpindah halaman:

@FindBy(name = "addTblRencana")  
private WebElement buttonTambah;  
  
public ManPowerPlanningTambahRencana tambahRencanaManPowerPlanning() {  
pause(350);  
buttonTambah.click();  
return new ManPowerPlanningTambahRencana();  
}

Gambar 4.4.3 Contoh Program untuk Menekan Tombol yang Ada dan Berpindah Halaman

1. Setiap melakukan aktivitas, dibutuhkan parameter pada Web Element untuk mengetahui status Automated Test tersebut sukses atau gagal. Pada fungsi Page Object ini, program akan mendeteksi tanda ketika aktivitas tersebut sukses dieksekusi. Fungsi ini sangat penting sebagai Laporan setelah dilakukan Automated Testing. Berikut kode program untuk mendeteksi string pada web element:

public boolean isSimpanSukses() {  
boolean result = false;  
    
try {  
wait.forElementVisible(messagePanel);  
if(messagePanel.isDisplayed()) {  
WebElement message = messagePanel.findElement(By.tagName("font"));  
String messageTag = message.getText();  
if(messageTag.trim().equals("Info")) {  
System.out.println("--> info 1 ");  
WebElement messageParent = driver.findElement(By.xpath("//blink//ancestor::tbody"));  
WebElement messageContent = messageParent.findElement(By.tagName("li"));  
String messageText = messageContent.getText();  
if(messageText.contains("INF0001")) {  
result = true;  
}  
}  
}  
}  
catch (NoSuchElementException|TimeoutException e) {  
// Panel berhasil disimpan tidak ada sebab simpan tidak berhasil disimpan  
}  
  
return result;  
}

}  
}  
}  
catch (NoSuchElementException|TimeoutException e) {  
// Panel berhasil disimpan tidak ada sebab simpan tidak berhasil disimpan  
}  
  
return result;  
}

Gambar 4.4.5 Contoh Program untuk Menyimpan Laporan dan Mendeteksi Sukses Tidaknya Sebuah Fungsi Berjalan dalam Test

1. Pembuatan Page Object untuk berpindah halaman, dalam fungsi ini, atribut yang didefinisikan menggunakan Link Text, sehingga jika terdapat web element yang mengandung string sesuai dengan Link Text, maka web element tersebut akan dieksekusi. Untuk menggunakan atribut berdasarkan Link Text, string yang dituju harus bersifat unik atau tidak terdapat string yang sama dalam satu webpage.

public ManPowerPlanningBrowserPage navigateToTransaksiManPowerPlanning() {  
pause(350);  
menuRecruitment.click();  
  
pause(350);  
WebElement transaksi = driver.findElement(By.linkText("Transaksi"));  
transaksi.click();  
  
pause(350);  
transaksi = driver.findElement(By.linkText("MPP - Man Power Planning"));  
transaksi.click();  
return new ManPowerPlanningBrowserPage();  
}

Gambar 4.4.6 Contoh Program untuk Navigasi Menuju Suatu Halaman

## PEMBUATAN KODE PROGRAM AUTOMATED TEST

Setelah tes skenario telah dibuat, selanjutnya implementasi tes skenario dalam program. Setiap langkah pada tes skenario harus diimplementasikan. Setiap automated testing selalu dimulai dari Login system. Pada pembuatan kode program dibutuhkan library Selenium dan TestNG. Berikut kode program automated test berdasarkan langkah yang dieksekusi:

* + - 1. Membuat variable yang berisi akses menuju PostgreSQL, langkah ini digunakan untuk mengetahui aktivitas pada database seperti proses Create, Read, Update dan Delete yang nantinya akan dilampirkan pada Automated Testing Report.

Connection koneksi;  
  
String driver = "org.postgresql.Driver";  
String user = "xxxxx";  
String password = "xxxx";  
String url = "jdbc:postgresql://192.168.23.242:5432/hr\_meratus\_xxxxx";

Gambar 4.5.1 Contoh Program Yang Memberikan Akses menuju Database

* + - 1. Membuat kode program yang menyimpan sebuah status apakah tes skenario tersebut sudah dijalankan apa tidak, karena ada beberapa tes skenario yang dijalankan berulang kali, sementara pada Automated Testing Report tidak boleh terdapat aktivitas tes skenario lebih dari satu kali. Pada saat salah satu modul dijalankan pertama kali statusnya dalam keadaan false, kemudian ketika telah selesai dijalankan, maka akan merekam aktivitas pada database dan keadaannya berubah menjadi true, kondisi tersebut disimpan menggunakan shared preferences. Sehingga pada saat modul tersebut dijalankan untuk kedua kalinya, maka kondisinya true dan tidak merekam aktivitas pada database.

final String statusMPPInput = "boolean";  
Preferences prefs = Preferences.userNodeForPackage(this.getClass());  
….  
@Test(priority = 4)  
public void inputManPowerPlanning() throws ClassNotFoundException, SQLException {  
testCase = "HVST-103";  
boolean status = prefs.getBoolean(statusMPPInput, false);

if (!errorDetected) {  
try {  
System.out.println("STATUS SAAT INI " + status);  
  
if(status == false) {  
System.out.println("--Belum tercetak di Excel");

// insert proses ke t\_curr\_proses  
insertCurrentProsesName("Man Power Planning (BAI010)", "Add MPP", 0);  
prefs.putBoolean(statusMPPInput, true);  
} else {  
System.out.println("--Sudah tercetak di Excel");  
}  
}  
}

testCase = "HVST-103";  
boolean status = prefs.getBoolean(statusMPPInput, false);

if (!errorDetected) {  
try {  
System.out.println("STATUS SAAT INI " + status);  
  
if(status == false) {  
System.out.println("--Belum tercetak di Excel");

// insert proses ke t\_curr\_proses  
insertCurrentProsesName("Man Power Planning (BAI010)", "Add MPP", 0);  
prefs.putBoolean(statusMPPInput, true);  
} else {  
System.out.println("--Sudah tercetak di Excel");  
}  
}  
}

Gambar 4.5.2 Contoh Program yang Menyimpan Sebuah Status Apakah Tes Skenario Tersebut Sudah Dijalankan Apa Tidak

1. Sebelum menjalankan automated test, Persiapkan terlebih dahulu fungsi yang berisi pengaturan untuk Selenium seperti Webdriver yang digunakan dan alamat yang dituju.

@BeforeTest  
public void setupSelenium() {  
homePage = new HrisHomePage();  
KonfigurasiHris.openConfigurationProperties("config.properties");  
}

Gambar 4.5.3 Contoh Program untuk men-setup Selenium

1. Fungsi Assert digunakan untuk mendeteksi apakah proses tersebut berhasil dijalankan dan mengeluarkan output sesuai dengan hasil pada tes skenario, jika Assert bernilai false, maka program akan terhenti pada langkah tersebut. Sebaliknya jika selama bernilai true, maka program akan tetap terus berjalan sampai selesai.

Assert.assertEquals(hrisMeratusHome.getJudulTransaksi(), "Man Power Planning",  
"Content yang tampil salah");

Gambar 4.5.4 Contoh Program untuk Mendeteksi Keberhasilan Sebuah Test

1. Kemudian melakukan proses login dengan memasukkan isi form dengan username dan password yang sudah disediakan. Jika setelah login diarahkan menuju halaman Homepage, maka Assert bernilai true.

@Test(priority = 1)  
public void loginSukses() {  
//prefs.remove(statusMPPInput);  
homePage.open();  
LoginPage loginPage = homePage.navigateToLoginPage();  
loginPage.loggingIn("SofcoGraha", "g4kt4u"); // login ini login yang valid  
  
try {  
Assert.assertTrue(loginPage.isLoginSukses());  
  
errorDetected = false;  
} catch (AssertionError e) {  
  
errorDetected = true;  
throw e;  
}  
}

Gambar 4.4.5 Contoh Program Login Secara Otomatis

1. Page Object yang telah dibuat, dipanggil dengan parameter sesuai langkah-langkah pada tes skenario

@Test(priority = 4)  
public void inputManPowerPlanning() throws ClassNotFoundException, SQLException {  
testCase = "HVST-103";  
boolean status = prefs.getBoolean(statusMPPInput, false);  
  
if (!errorDetected) {  
try {  
System.out.println("STATUS SAAT INI " + status);  
  
if(status == false) {  
System.out.println("--Belum tercetak di Excel");  
// insert proses ke t\_curr\_proses  
insertCurrentProsesName("Man Power Planning (BAI010)", "Add MPP", 0);  
prefs.putBoolean(statusMPPInput, true);  
} else {  
System.out.println("--Sudah tercetak di Excel");  
}  
  
// jalankan input  
manPowerPlanningInputPage.isiPT("05");  
manPowerPlanningInputPage.isiDivisi("17");  
manPowerPlanningInputPage.isiCabang("02");  
manPowerPlanningInputPage.isiLokasiKerja("TB");  
manPowerPlanningInputPage.isiNomor("02/MPP/2019");  
manPowerPlanningInputPage.isiTanggalRecruit("08", "07", "2019");  
manPowerPlanningInputPage.periodeTahun("2019");  
manPowerPlanningInputPage.nomorJob("200708.05   .012");  
manPowerPlanningInputPage.jumlahKaryawan("3");  
manPowerPlanningInputPage.rencana("0", "1", "0", "1", "2");  
manPowerPlanningInputPage.keterangan("");  
manPowerPlanningInputPage.klikSimpan();  
  
Assert.assertTrue(manPowerPlanningInputPage.isSimpanSukses(), "Simpan Man Power Planning gagal dilakukan");  
  
errorDetected = false;  
} catch (AssertionError e) {  
  
errorDetected = true;  
throw e;  
}  
} else {  
  
throw new AssertionError("Karena error dari test sebelumnya");  
}  
}

}  
// jalankan input  
manPowerPlanningInputPage.isiPT("05");  
manPowerPlanningInputPage.isiDivisi("17");  
manPowerPlanningInputPage.isiCabang("02");  
manPowerPlanningInputPage.isiLokasiKerja("TB");  
manPowerPlanningInputPage.isiNomor("02/MPP/2019");  
manPowerPlanningInputPage.isiTanggalRecruit("08", "07", "2019");  
manPowerPlanningInputPage.periodeTahun("2019");  
manPowerPlanningInputPage.nomorJob("200708.05   .012");  
manPowerPlanningInputPage.jumlahKaryawan("3");  
manPowerPlanningInputPage.rencana("0", "1", "0", "1", "2");  
manPowerPlanningInputPage.keterangan("");  
manPowerPlanningInputPage.klikSimpan();  
  
Assert.assertTrue(manPowerPlanningInputPage.isSimpanSukses(), "Simpan Man Power Planning gagal dilakukan");  
  
errorDetected = false;  
} catch (AssertionError e) {  
  
errorDetected = true;  
throw e;  
}  
} else {  
  
throw new AssertionError("Karena error dari test sebelumnya");  
}  
}

System.out.println("STATUS SAAT INI " + status);  
  
if(status == false) {  
System.out.println("--Belum tercetak di Excel");  
// insert proses ke t\_curr\_proses  
insertCurrentProsesName("Man Power Planning (BAI010)", "Add MPP", 0);  
prefs.putBoolean(statusMPPInput, true);  
} else {  
System.out.println("--Sudah tercetak di Excel");  
}  
  
// jalankan input  
manPowerPlanningInputPage.isiPT("05");  
manPowerPlanningInputPage.isiDivisi("17");  
manPowerPlanningInputPage.isiCabang("02");  
manPowerPlanningInputPage.isiLokasiKerja("TB");  
manPowerPlanningInputPage.isiNomor("02/MPP/2019");  
manPowerPlanningInputPage.isiTanggalRecruit("08", "07", "2019");  
manPowerPlanningInputPage.periodeTahun("2019");  
manPowerPlanningInputPage.nomorJob("200708.05   .012");  
manPowerPlanningInputPage.jumlahKaryawan("3");  
manPowerPlanningInputPage.rencana("0", "1", "0", "1", "2");  
manPowerPlanningInputPage.keterangan("");  
manPowerPlanningInputPage.klikSimpan();  
  
Assert.assertTrue(manPowerPlanningInputPage.isSimpanSukses(), "Simpan Man Power Planning gagal dilakukan");  
  
errorDetected = false;  
} catch (AssertionError e) {  
  
errorDetected = true;  
throw e;  
}  
} else {  
  
throw new AssertionError("Karena error dari test sebelumnya");  
}  
}

}  
// jalankan input  
manPowerPlanningInputPage.isiPT("05");  
manPowerPlanningInputPage.isiDivisi("17");  
manPowerPlanningInputPage.isiCabang("02");  
manPowerPlanningInputPage.isiLokasiKerja("TB");  
manPowerPlanningInputPage.isiNomor("02/MPP/2019");  
manPowerPlanningInputPage.isiTanggalRecruit("08", "07", "2019");  
manPowerPlanningInputPage.periodeTahun("2019");  
manPowerPlanningInputPage.nomorJob("200708.05   .012");  
manPowerPlanningInputPage.jumlahKaryawan("3");  
manPowerPlanningInputPage.rencana("0", "1", "0", "1", "2");  
manPowerPlanningInputPage.keterangan("");  
manPowerPlanningInputPage.klikSimpan();  
  
Assert.assertTrue(manPowerPlanningInputPage.isSimpanSukses(), "Simpan Man Power Planning gagal dilakukan");  
  
errorDetected = false;  
} catch (AssertionError e) {  
  
errorDetected = true;  
throw e;  
}  
} else {  
  
throw new AssertionError("Karena error dari test sebelumnya");  
}  
}

manPowerPlanningInputPage.isiDivisi("17");  
manPowerPlanningInputPage.isiCabang("02");  
manPowerPlanningInputPage.isiLokasiKerja("TB");  
manPowerPlanningInputPage.isiNomor("02/MPP/2019");  
manPowerPlanningInputPage.isiTanggalRecruit("08", "07", "2019");  
manPowerPlanningInputPage.periodeTahun("2019");  
manPowerPlanningInputPage.nomorJob("200708.05   .012");  
manPowerPlanningInputPage.jumlahKaryawan("3");  
manPowerPlanningInputPage.rencana("0", "1", "0", "1", "2");  
manPowerPlanningInputPage.keterangan("");  
manPowerPlanningInputPage.klikSimpan();  
  
Assert.assertTrue(manPowerPlanningInputPage.isSimpanSukses(), "Simpan Man Power Planning gagal dilakukan");  
  
errorDetected = false;  
} catch (AssertionError e) {  
  
errorDetected = true;  
throw e;  
}  
} else {  
  
throw new AssertionError("Karena error dari test sebelumnya");  
}  
}

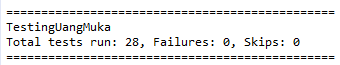
Gambar 4.5.6 Contoh Program Lengkap Automatic Test Satu Bagian Test Case

1. Kemudian pembuatan fungsi untuk merekam aktivitas antara program automated testing dengan database.

Gambar 4.5.7 Contoh Program untuk Merekam Aktifitas ke Database untuk Tujuan Dokumentasi Otomatis

insertCurrentProsesName("Man Power Planning (BAI010)", "Add MPP", 0);

## Setelah membuat kode program automated testing, maka kode tersebut dijalankan untuk memulai tes. Tes dinyatakan berhasil jika tidak ditemukan proses yang gagal dalam suatu modul pada saat dijalankan.



*Gambar 4.5.8 Output Program ketika modul berhasil dieksekusi*

## 4.6 PEMBUATAN KODE PROGRAM RUNNING TEST

Dalam pembuatan kode program Running Test menggunakan Bahasa pemrograman XML (eXtensible Markup Language) yang terbagi menjadi tiga bagian. Bagian pertama Running Test yang meliputi satu modul. Dalam satu modul terdapat beberapa kode program automated testing yang harus dijalankan secara bersamaan karena merupakan satu-kesatuan dalam satu modul sesuai dengan tes skenario. Selain itu kode program Running Test digunakan untuk menjalankan program yang sudah dijalankan jika program tersebut dibutuhkan pada modul lain.program yang sudah dijalankan jika program tersebut dibutuhkan pada modul lain.

<suite name="TestingManPowerPlanning">  
<test name="ManPowerPlanningInputMPP" preserve-order="true">  
<classes>  
<class name="test.recruitment.TRecruitment\_ManPowerPlanning\_InputMPP" />  
</classes>  
</test>  
<test name="ManPowerPlanningInputRencana" preserve-order="true">  
<classes>  
<class name="test.recruitment.TRecruitment\_ManPowerPlanning\_InputRencana" />  
</classes>  
</test>  
<test name="ManPowerPlanningHapusRencana" preserve-order="true">  
<classes>  
<class name="test.recruitment.TRecruitment\_ManPowerPlanning\_HapusRencana" />  
</classes>  
</test>  
<test name="ManPowerPlanningHapusMPP" preserve-order="true">  
<classes>  
<class name="test.recruitment.TRecruitment\_ManPowerPlanning\_HapusMPP" />  
</classes>  
</test>  
</suite>

Gambar 4.6.1 Contoh Program XML Menjalankan Berbagai Test Dalam Satu Program

Bagian kedua yaitu program untuk reset ulang status running pada kode program automated testing dalam keadaan false atau belum pernah dijalankan sama sekali. Hal ini bertujuan untuk menandai kembali semua modul dalam keadaan belum dijalankan jika semua modul ingin dijalankan ulang, sehingga menghindari terjadi data rekam aktivitas ganda.

<test name="ChangeTruePreferenceValue" preserve-order="true">  
<classes>  
<class name="test.ChangeTruePreferenceValue" />  
</classes>  
</test>  
  
<test name="ChangeFalsePreferenceValue" preserve-order="true">  
<classes>  
<class name="test.ChangeFalsePreferenceValue" />  
</classes>  
</test>

Gambar 4.6.2 Contoh Program Mereset Status Running Untuk Kebutuhan Pencatatan Dokumentasi Otomatis

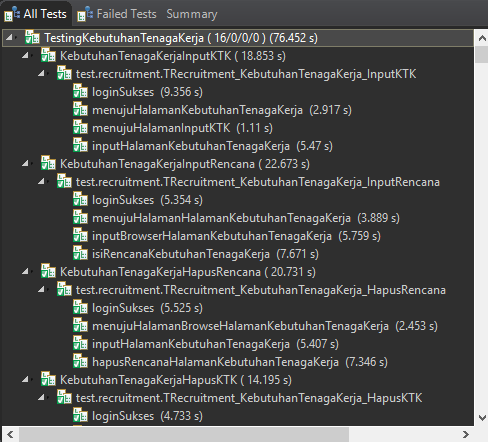
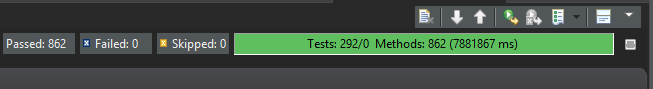
Bagian ketiga yaitu program untuk testing secara keseluruhan modul yang sudah dibuat. Tujuannya untuk pembuatan automated testing report.

<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">  
<suite name="Suite">  
<suite-files>  
<suite-file path="ChangeFalseAll.xml"></suite-file>  
<!-- RECRUITMENT -->  
    <suite-file path="TestingKebutuhanTenagaKerja.xml"></suite-file>  
    <suite-file path="TestingManPowerPlanning.xml"></suite-file>  
    <suite-file path="TestingProsesSeleksiKaryawan.xml"></suite-file>  
    <suite-file path="TestingRencanaTenagaKerja.xml"></suite-file>  
    <suite-file path="TestingSuratLamaran.xml"></suite-file>  
  
<!-- MUTASI DAN PEMBINAAN KARYAWAN -->      
    <suite-file path="TestingMutasiRekrutBaru.xml"></suite-file>  
    <suite-file path="TestingMutasiRangkapJabatan.xml"></suite-file>  
    <suite-file path="TestingMutasiLepasRangkapJabatan.xml"></suite-file>  
    <suite-file path="TestingMutasiStatusPekerjaan.xml"></suite-file>  
    <suite-file path="TestingMutasiHomebase.xml"></suite-file>  
    <suite-file path="TestingMutasiTerminasi.xml"></suite-file>  
    <suite-file path="TestingMutasiPembinaan.xml"></suite-file>  
      
    <!-- TRAINING DAN PENILAIAN KARYAWAN (TRANSAKSI TRAINING) -->  
    <suite-file path="TestingKebutuhanTraining.xml"></suite-file>  
    <suite-file path="TestingRencanaTraining.xml"></suite-file>  
    <suite-file path="TestingRealisasiTraining.xml"></suite-file>  
      
    <!-- TRAINING DAN PENILAIAN KARYAWAN (TRANSAKSI PENILAIAN KARYAWAN) -->  
    <!-- <suite-file path="TestingPenilaianOrientasiKaryawan.xml"></suite-file> -->  
    <suite-file path="TestingTraineePerformanceEvaluation.xml"></suite-file>  
    <suite-file path="TestingPerformanceAppraisal.xml"></suite-file>  
      
    <!-- PERJALANAN DINAS -->  
    <suite-file path="TestingRencanaPerjalananDinas.xml"></suite-file>  
    <suite-file path="TestingUangMuka.xml"></suite-file>  
    <suite-file path="TestingRealisasiPerjalananDinas.xml"></suite-file>  
    <suite-file path="TestingPenyelesaianUangMuka.xml"></suite-file>  
      
    <!-- ABSENSI -->  
    <suite-file path="TestingSuratPerintahLembur.xml"></suite-file>  
    <suite-file path="TestingSuratIzinDanCuti.xml"></suite-file>  
    <suite-file path="TestingPindahLokasiKerja.xml"></suite-file>  
      
    <suite-file path="TestingAbsensidariCheckClock.xml"></suite-file>  
    <suite-file path="TestingTransferDatadariMesinAbsensi.xml"></suite-file>  
    <suite-file path="TestingAbsensidariFingerScan.xml"></suite-file>  
    <suite-file path="TestingProsesAmbildanLengkapiDatadiLokasiKerja.xml"></suite-file>  
    <suite-file path="TestingDataHasilProsesAmbilDataAbsensi.xml"></suite-file>  
    <suite-file path="TestingProsesAbsensiHariandiCabang.xml"></suite-file>  
    <suite-file path="TestingDataHasilProsesAbsensiHarian.xml"></suite-file>  
    <suite-file path="TestingBatalProsesAbsensi.xml"></suite-file>  
    <suite-file path="TestingProsesAbsensiHariandiCabang.xml"></suite-file>  
      
    <!-- CUTI A DAN B -->  
    <suite-file path="TestingSuratCutiA.xml"></suite-file>  
    <suite-file path="TestingSuratCutiB.xml"></suite-file>  
    <suite-file path="TestingKoreksiSisaCuti.xml"></suite-file>  
    <!-- <suite-file path="TestingKoreksiTanggalMasaBerlaku.xml"></suite-file> -->   
    <suite-file path="TestingCutiMasal.xml"></suite-file>  
    <suite-file path="TestingProsesJatahCutiTahunan.xml"></suite-file>  
      
    <!-- ASKES DAN JAMSOSTEK -->  
    <suite-file path="TestingDaftarPesertaAskes.xml"></suite-file>  
    <suite-file path="TestingPengajuanKlaimAskes.xml"></suite-file>  
    <suite-file path="TestingPenggantianKlaimAskes.xml"></suite-file>  
    <suite-file path="TestingProsesHitungPremiAskesBulanan.xml"></suite-file>  
    <suite-file path="TestingProsesHitungPremiJamsostekBulanan.xml"></suite-file>  
      
    <!-- BIAYA KESEHATAN -->  
    <suite-file path="TestingPengajuanKlaimKaryawanBukanPesertaAskes.xml"></suite-file>  
    <suite-file path="TestingAdministrasiMedical.xml"></suite-file>  
      
    <!-- PINJAMAN KARYAWAN -->  
    <suite-file path="TestingPinjamanKaryawan.xml"></suite-file>  
      
    <!-- PENDAPATAN DAN POTONGAN -->  
    <suite-file path="TestingTunjanganDanPotonganUntukBeberapaPeriode.xml"></suite-file>  
    <suite-file path="TestingProsesHitungTunjangandanPotongan.xml"></suite-file>  
    <suite-file path="TestingDataHasilProsesTunjangandanPotongan.xml"></suite-file>  
    <suite-file path="TestingProsesBatalKonfirmasi.xml"></suite-file>  
    <suite-file path="TestingProsesBatalTunjangandanPotongan.xml"></suite-file>  
      
    <!-- ESS -->  
    <suite-file path="TestingESSMasterUser.xml"></suite-file>  
      
    <!-- KOPERASI -->  
    <suite-file path="TestingAnggotaKoperasi.xml"></suite-file>  
    <suite-file path="TestingPotonganKoperasi.xml"></suite-file>  
      
</suite-files>  
</suite>

    <suite-file path="TestingKebutuhanTraining.xml"></suite-file>  
    <suite-file path="TestingRencanaTraining.xml"></suite-file>  
    <suite-file path="TestingRealisasiTraining.xml"></suite-file>  
      
    <!-- TRAINING DAN PENILAIAN KARYAWAN (TRANSAKSI PENILAIAN KARYAWAN) -->  
    <!-- <suite-file path="TestingPenilaianOrientasiKaryawan.xml"></suite-file> -->  
    <suite-file path="TestingTraineePerformanceEvaluation.xml"></suite-file>  
    <suite-file path="TestingPerformanceAppraisal.xml"></suite-file>  
      
    <!-- PERJALANAN DINAS -->  
    <suite-file path="TestingRencanaPerjalananDinas.xml"></suite-file>  
    <suite-file path="TestingUangMuka.xml"></suite-file>  
    <suite-file path="TestingRealisasiPerjalananDinas.xml"></suite-file>  
    <suite-file path="TestingPenyelesaianUangMuka.xml"></suite-file>  
      
    <!-- ABSENSI -->  
    <suite-file path="TestingSuratPerintahLembur.xml"></suite-file>  
    <suite-file path="TestingSuratIzinDanCuti.xml"></suite-file>  
    <suite-file path="TestingPindahLokasiKerja.xml"></suite-file>  
      
    <suite-file path="TestingAbsensidariCheckClock.xml"></suite-file>  
    <suite-file path="TestingTransferDatadariMesinAbsensi.xml"></suite-file>  
    <suite-file path="TestingAbsensidariFingerScan.xml"></suite-file>  
    <suite-file path="TestingProsesAmbildanLengkapiDatadiLokasiKerja.xml"></suite-file>  
    <suite-file path="TestingDataHasilProsesAmbilDataAbsensi.xml"></suite-file>  
    <suite-file path="TestingProsesAbsensiHariandiCabang.xml"></suite-file>  
    <suite-file path="TestingDataHasilProsesAbsensiHarian.xml"></suite-file>  
    <suite-file path="TestingBatalProsesAbsensi.xml"></suite-file>  
    <suite-file path="TestingProsesAbsensiHariandiCabang.xml"></suite-file>  
      
    <!-- CUTI A DAN B -->  
    <suite-file path="TestingSuratCutiA.xml"></suite-file>  
    <suite-file path="TestingSuratCutiB.xml"></suite-file>  
    <suite-file path="TestingKoreksiSisaCuti.xml"></suite-file>  
    <!-- <suite-file path="TestingKoreksiTanggalMasaBerlaku.xml"></suite-file> -->   
    <suite-file path="TestingCutiMasal.xml"></suite-file>  
    <suite-file path="TestingProsesJatahCutiTahunan.xml"></suite-file>  
      
    <!-- ASKES DAN JAMSOSTEK -->  
    <suite-file path="TestingDaftarPesertaAskes.xml"></suite-file>  
    <suite-file path="TestingPengajuanKlaimAskes.xml"></suite-file>  
    <suite-file path="TestingPenggantianKlaimAskes.xml"></suite-file>  
    <suite-file path="TestingProsesHitungPremiAskesBulanan.xml"></suite-file>  
    <suite-file path="TestingProsesHitungPremiJamsostekBulanan.xml"></suite-file>  
      
    <!-- BIAYA KESEHATAN -->  
    <suite-file path="TestingPengajuanKlaimKaryawanBukanPesertaAskes.xml"></suite-file>  
    <suite-file path="TestingAdministrasiMedical.xml"></suite-file>  
      
    <!-- PINJAMAN KARYAWAN -->  
    <suite-file path="TestingPinjamanKaryawan.xml"></suite-file>  
      
    <!-- PENDAPATAN DAN POTONGAN -->  
    <suite-file path="TestingTunjanganDanPotonganUntukBeberapaPeriode.xml"></suite-file>  
    <suite-file path="TestingProsesHitungTunjangandanPotongan.xml"></suite-file>  
    <suite-file path="TestingDataHasilProsesTunjangandanPotongan.xml"></suite-file>  
    <suite-file path="TestingProsesBatalKonfirmasi.xml"></suite-file>  
    <suite-file path="TestingProsesBatalTunjangandanPotongan.xml"></suite-file>  
      
    <!-- ESS -->  
    <suite-file path="TestingESSMasterUser.xml"></suite-file>  
      
    <!-- KOPERASI -->  
    <suite-file path="TestingAnggotaKoperasi.xml"></suite-file>  
    <suite-file path="TestingPotonganKoperasi.xml"></suite-file>  
      
</suite-files>  
</suite>

    <!-- ASKES DAN JAMSOSTEK -->  
    <suite-file path="TestingDaftarPesertaAskes.xml"></suite-file>  
    <suite-file path="TestingPengajuanKlaimAskes.xml"></suite-file>  
    <suite-file path="TestingPenggantianKlaimAskes.xml"></suite-file>  
    <suite-file path="TestingProsesHitungPremiAskesBulanan.xml"></suite-file>  
    <suite-file path="TestingProsesHitungPremiJamsostekBulanan.xml"></suite-file>  
      
    <!-- BIAYA KESEHATAN -->  
    <suite-file path="TestingPengajuanKlaimKaryawanBukanPesertaAskes.xml"></suite-file>  
    <suite-file path="TestingAdministrasiMedical.xml"></suite-file>  
      
    <!-- PINJAMAN KARYAWAN -->  
    <suite-file path="TestingPinjamanKaryawan.xml"></suite-file>  
      
    <!-- PENDAPATAN DAN POTONGAN -->  
    <suite-file path="TestingTunjanganDanPotonganUntukBeberapaPeriode.xml"></suite-file>  
    <suite-file path="TestingProsesHitungTunjangandanPotongan.xml"></suite-file>  
    <suite-file path="TestingDataHasilProsesTunjangandanPotongan.xml"></suite-file>  
    <suite-file path="TestingProsesBatalKonfirmasi.xml"></suite-file>  
    <suite-file path="TestingProsesBatalTunjangandanPotongan.xml"></suite-file>  
      
    <!-- ESS -->  
    <suite-file path="TestingESSMasterUser.xml"></suite-file>  
      
    <!-- KOPERASI -->  
    <suite-file path="TestingAnggotaKoperasi.xml"></suite-file>  
    <suite-file path="TestingPotonganKoperasi.xml"></suite-file>  
      
</suite-files>  
</suite>

Gambar 4.6.3 Contoh Program Lengkap XML Untuk Pengujian Otomatis Hris Meratus

****Setelah dibuat, maka kode program tersebut dijalankan, Pada saat menjalankan 58 modul, didapatkan waktu running program sekitar 131 menit (7881867 ms) dan didapatkan semua program berhasil berjalan sesuai dengan tes skenario yang telah dibuat.

Gambar 4.6.5 Automated Test yang telah berhasil melalui pengujian

Gambar 4.6.4 Contoh Program Mereset Status Running Untuk Kebutuhan Pencatatan Dokumentasi Otomatis

## 4.7 PEMBUATAN AUTOMATED TESTING REPORT

Berikut kode program untuk menampilkan hasil Automated Testing modul secara keseluruhan dalam bentuk Excel.

private void insertCurrentProsesName(String namaProses, String aksiLogic, int noUrut) throws ClassNotFoundException, SQLException {  
  
Class.forName(driver);  
koneksi = DriverManager.getConnection(url, user, password);  
  
Connection connection = koneksi;  
      
    String sql = "Select \* " +  
                 "From t\_curr\_proses " +  
          "Where nama = '" + namaProses + "'";  
      
PreparedStatement rsStatement = connection.prepareStatement(sql);  
  
ResultSet resultSet = rsStatement.executeQuery();  
  
int counter = 0;  
if (resultSet.next()) {  
counter = resultSet.getInt("no\_urut");  
}  
  
rsStatement.close();  
  
    Statement statement = connection.createStatement();  
  
sql = "Update t\_curr\_proses Set " +   
      "nama = '" + namaProses + "', " +   
      "aksi\_logic = '" + aksiLogic + "', " +  
  "no\_urut = " + counter + ";";  
statement.executeUpdate(sql);  
     
statement.close();  
koneksi.close();  
  
}

Gambar 4.7.1 Contoh Program Untuk Melakukan Perekaman Dokumentasi Otomatis dari Database ke Exel

# BAB VI PENUTUP

Pelaksanaan Kerja Praktek di PT. Sofco Graha Representative Office telah kami lakukan. Kami mendapatkan ilmu dan pengalaman bekerja di kantor yang menerapkan teknologi, kreativitas, dan ketekunan dalam bekerja. Kami juga belajar mengenai budaya kerja di PT. Sofco Graha sebagai pengusaha perangkat lunak akuntansi. Dan juga dengan terlaksananya kerja praktek ini, kami mendapatkan pengetahuan baru tentang Quality Assurance. Tidak hanya bidang keilmuan yang sejalan dengan departemen kami saja, namun softskill seperti cara berpresentasi, cara berkomunikasi dengan divisi Non-IT dan alur bekerja yang baik kami dapatkan saat kerja praktek kami berlangsung.

## 5.1 KESIMPULAN

Berdasarkan Kerja Praktek yang kami lakukan, diperoleh beberapa kesimpulan sebagai berikut:

* + 1. Terdapat 58 modul yang dibuat kode program untuk tes, saat dijalankan semuanya membutuhkan waktu running program sekitar 131 menit (7881867 ms).
    2. Semua modul pada HRIS Meratus saat dilakukan automated testing, didapatkan semua modul berjalan dengan normal.
    3. Dengan Automated Testing, seorang Quality Assurance tidak perlu melakukan manual testing secara berulang-ulang.
    4. Automated Testing report memudahkan Software Engineer dalam memperbaiki perangkat lunak yang sudah dibuat.

## 5.2 SARAN

Berdasarkan Kerja Praktek yang kami lakukan, diperoleh beberapa kesimpulan sebagai berikut:

Selain untuk tes fungsional, Selenium juga dapat digunakan untuk tes keamanan dan kekuatan pada perangkat lunak.

Form yang berhubungan dengan tanggal harus selalu diperbarui, karena ada beberapa form mempunyai syarat batas tanggal yang dapat dimasukkan.

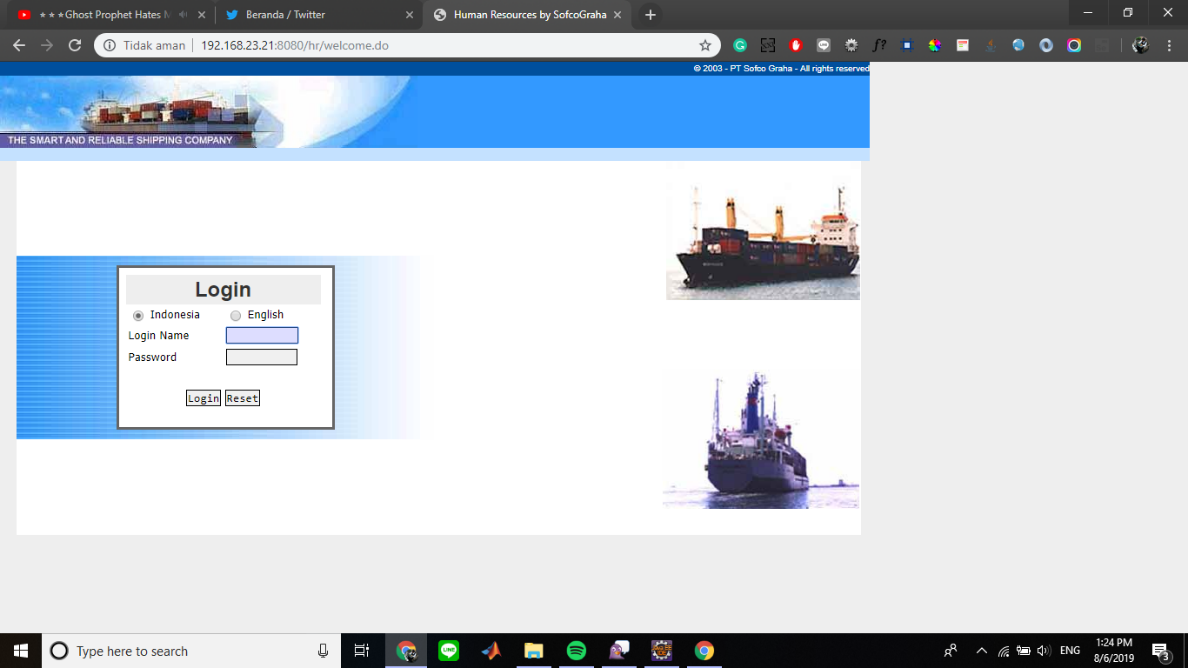
Perlunya mengatur posisi modul berdasarkan modul yang sering dijalankan untuk mempersingkat waktu saat menjalankan semua modul.

# DAFTAR PUSTAKA

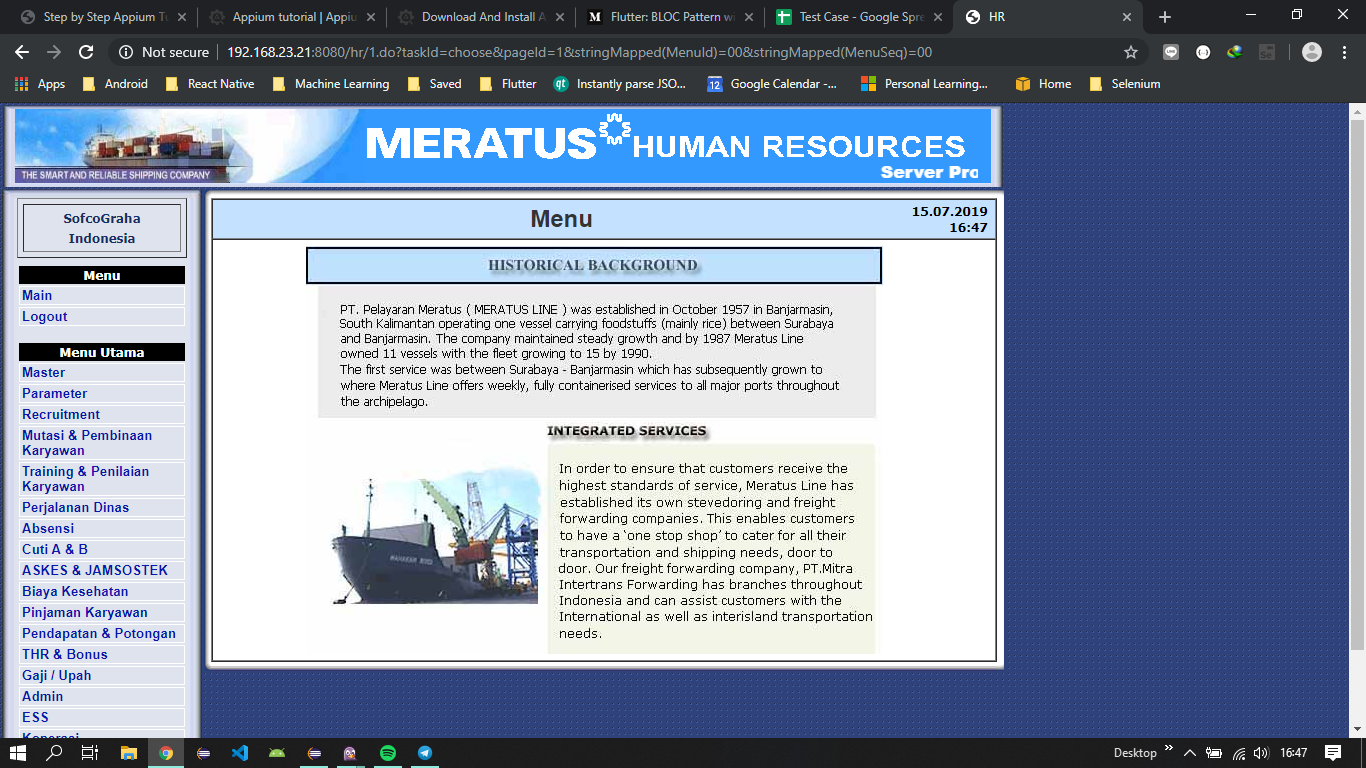
1. Admin. 2016. *Kontak kami*. <https://www.sofcograha.co.id/>. [30 Juli 2019]
2. Admin. 2016. *Produk & Layanan*. <https://www.sofcograha.co.id/>. [30 Juli 2019]
3. Admin. 2016. *Sekilas tentang Sofco*. <https://www.sofcograha.co.id/>. [30 Juli 2019]
4. Ammann, Paul & Offutt, Jeff 2008. *Introduction Of Software Testing,* Cambridge University Press, New York
5. Anonim. 2018. *8 Jenis Testing Wajib Seorang Software Tester / Software QA*. [https://toghr.com](https://toghr.com/). [1 Oktober 2019]
6. Anonim. 2019. *Prinsip Pareto*. [https://id.wikipedia.org](https://id.wikipedia.org/) . [1 Oktober 2019]
7. Anonim. 2019. *Selenium (software)*. [https://en.wikipedia.org](https://en.wikipedia.org/)/. [30 Agustus 2019]
8. Anonim. 2019. *What is Selenium?.* <https://www.seleniumhq.org/>. [26 Agustus 2019]
9. Bagus, Dimas. 2018. *Apa itu Software Tester ? Jobdesknya ngapain?.* [https://medium.com](https://medium.com/). [1 Oktober 2019]
10. Bertolino, Antonia 2007. 'Software Testing Research: Achievements, Challenges, Dreams'. *Future of Software Engineering (FOSE '07)*. Minneapolis, USA. 23-25 Mei 2007
11. Dosen Pendidikan 3. 2019. *Pengertian, Fungsi Dan Tugas Penting HRD Dalam Perusahaan.* [https://www.dosenpendidikan.co.id](https://www.dosenpendidikan.co.id/). [1 Oktober 2019]
12. HC Indonesia Editor. 2017. *MENGENAL JURUSAN DAN PROSPEK KERJA JURUSAN HUMAN RESOURCES/HR.* [https://www.hotcourses.co.id](https://www.hotcourses.co.id/). [1 Oktober 2019]
13. Kurniawati, Peni. 2018. *Pengujian Sistem.* [https://medium.com](https://medium.com/). [1 Oktober 2019]
14. Kurniawati, Peni. 2018. *Quality Assurance dalam dunia IT*. <https://medium.com/>. [30 Agustus 2019]
15. ProgressTech. 2017. *Perbedaan Website dan Web Application.* <https://medium.com/>. [8 Agustus 2019]
16. Richard Kuhn, D 2004,’IEEE Transactions On Software Engineering’, *Software Fault Interactions and Implications for Software Testing*, vol. 30, no. 6, hal.1-4
17. Rouse, Margaret. 2019. *Web application (Web app)*.<https://searchsoftwarequality.techtarget.com/>. [31 Agustus 2019]
18. Ryandhita, Pietro. *Perbedaan Manual Testing dan Automated Testing.* [https://medium.com](https://medium.com/). [1 Oktober 2019]
19. Tian, Jeff 2005, *Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement,* Wiley-IEEE Press, New Jersey
20. Wardani, Ratna 2017, *Software Testing*, dilihat 1 Oktober 2019 <https://docplayer.info/37387686-Software-testing-ratna-wardani.html>

# LAMPIRAN

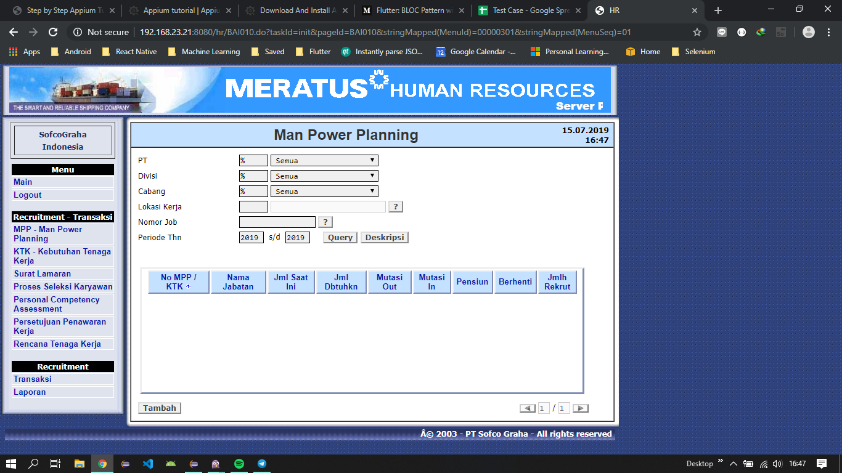
* + - 1. Tampilan Aplikasi HRIS Meratus



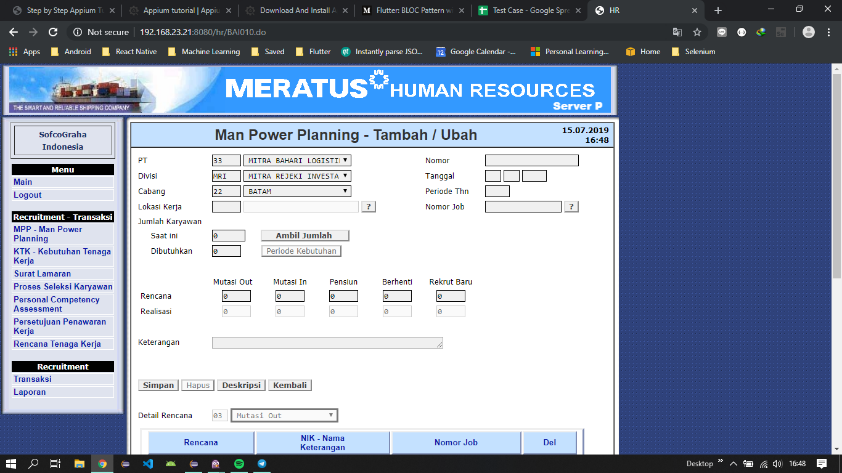
Tampilan *login*



Tampilan Beranda Awal

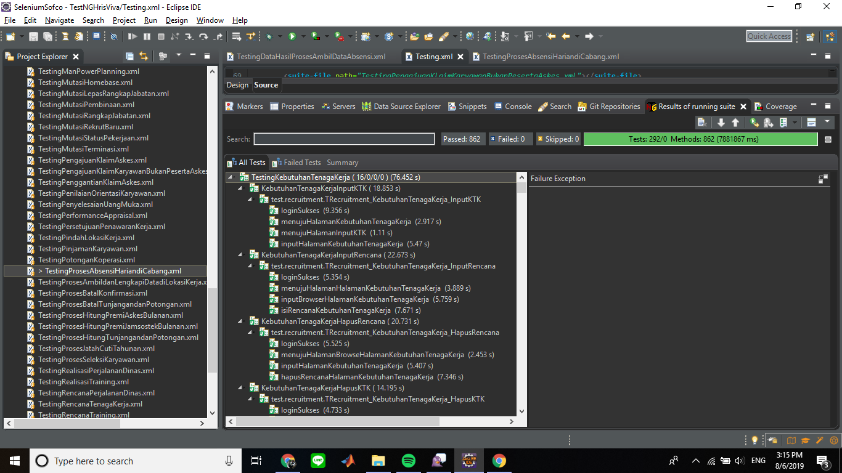


Tampilan Menu *Man Power Planning*

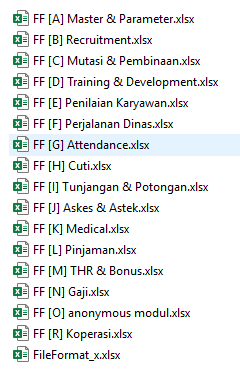


Tampilan Menu *Man Power Planning – Tambah/ubah*

* + - 1. Tangkap Layar Proses Automatic Test Keseluruhan Aplikasi HRIS Meratus



3. Tangkap Layar File Hasil Dokumentasi Otomatis



* + - 1. Dokumentasi Kerja Praktik

|  |  |
| --- | --- |
|  | |
|  |  |

Foto Bersama Pembimbing Kerja Praktik, Bapak Henrie Prawiro saat perpisahan

|  |
| --- |
|  |

Foto Bersama Karyawan PT. Sofco Graha