

## Reading Material

### Memahami Automation Testing - Selenium Intro Part 1



# Automation Testing

**Automation Testing** merupakan teknik pengujian perangkat lunak yang dilakukan oleh alat perangkat lunak pengujian otomatis khusus untuk menjalankan test cases. Sebaliknya, Manual Testing dilakukan oleh manusia atau user yang menjalankan semua langkah-langkah pengujian.

Automation testing software juga dapat memasukkan data pengujian ke dalam sistem yang sedang diuji, membandingkan hasil yang diharapkan dan hasil yang ditemukan, juga menghasilkan laporan pengujian yang lengkap dan terperinci. Automation testing software menuntut investasi yang cukup besar dari sumber daya. Tujuan Otomatisasi adalah untuk mengurangi jumlah kasus pengujian yang akan dijalankan secara manual namun tidak menghilangkan Pengujian Manual sama sekali.

Jadi peran Automation Testing adalah untuk mengoptimalkan pengujian, bukan untuk menggantikan manual testing sepenuhnya.

## Mengapa Test Automation?

Automation Test adalah cara terbaik untuk meningkatkan efektivitas, cakupan tes, dan kecepatan eksekusi dalam pengujian perangkat lunak. Automation testing software menjadi penting karena beberapa alasan berikut:

- Pengujian manual mengharuskan menjalankan semua workflow, semua scope, semua skenario negatif yang tentunya akan menghabiskan waktu dan uang
- Sulit untuk menguji situs multibahasa secara manual
- Automation testing software tidak memerlukan intervensi manusia ketika menjalankannya. Kita dapat menjalankan automation testing kapan saja diperlukan.
- Automation Testing meningkatkan kecepatan eksekusi pengujian

- Otomasi membantu meningkatkan cakupan pengujian
- Pengujian Manual dapat menjadi membosankan dan karenanya rawan terjadi kesalahan, karena penguji sudah sangat terbiasa menggunakan platform nya.

## **Manfaat Automation Testing**

Berikut ini adalah beberapa manfaat dari penerapan automation testing:

- 70% lebih cepat dari pengujian manual
- Cakupan pengujian yang lebih luas dari fitur aplikasi
- Hasil yang dapat diandalkan
- Dapat memastikan konsistensi
- Menghemat Waktu dan Biaya
- Meningkatkan akurasi
- Intervensi Manusia tidak diperlukan saat eksekusi
- Meningkatkan Efisiensi
- Kecepatan yang lebih baik dalam menjalankan tes
- Test script yang dapat digunakan kembali
- Pengujian secara berulang, sering dan menyeluruh
- Lebih banyak siklus eksekusi dapat dicapai melalui otomatisasi
- Bisa lebih awal untuk rilis

## **Test Cases mana saja yang bisa di otomatisasi?**

Kasus pengujian yang akan otomatisasi dapat dipilih menggunakan kriteria berikut untuk meningkatkan ROI (Return on Investment) otomatisasi

- Risiko Tinggi  
Critical test cases yang berhubungan dengan proses bisnis
- Test cases yang dijalankan berulang kali  
Seperti End to end scenario
- Test Cases yang sangat membosankan atau sulit dilakukan secara manual

- Kasus Uji yang memakan waktu  
Seperti regression testing jika platform nya besar

Namun tidak semua test cases cocok untuk diotomatisasi, ada kategori test cases yang tidak cocok untuk otomatisasi:

- Fitur yang masih dalam proses pengembangan
- Test Case yang baru dirancang dan belum dieksekusi secara manual sama sekali
- Test Case yang persyaratannya sering berubah
- Test Case yang dieksekusi secara ad-hoc atau hanya ketika dibutuhkan saja.

## **Proses Automation Testing**

Langkah langkah berikut perlu dilakukan dalam proses pembuatan atau pemberlakuan automation testing

1. Pemilihan Alat atau Tools
2. Tentukan lingkup Otomatisasi
3. Perencanaan, Desain dan Pengembangan
4. Eksekusi Test Cases
5. Perbaikan
6. Pemeliharaan

## **Pemilihan Tools Automation Testing**

Pemilihan tools sebagian besar tergantung pada teknologi yang digunakan oleh sistem dan juga experience dari tim yang akan menggunakan. Saat ini ada banyak sekali automation testing tools yang digunakan oleh QA maupun SDET. Tools yang digunakan pun sangat bervariasi, hampir setiap perusahaan menggunakan tools yang berbeda. Memilih tool yang tepat bisa menjadi tugas yang rumit.

Kriteria berikut akan membantu tim memilih tool terbaik:

- Environment Support

- Kemudahan penggunaan
- Pengujian Database
- Identifikasi objek
- Pengujian Gambar
- Pengujian Pemulihan Error
- Pemetaan Objek
- Mendukung berbagai jenis tes termasuk fungsional, manajemen, mobile, dll...
- Dukungan untuk beberapa frameworks
- Mudah untuk debug skrip perangkat lunak otomatisasi
- Kemampuan mengenali objek dalam lingkungan apapun
- Laporan dan hasil tes yang ekstensif
- Meminimalkan biaya pelatihan alat yang dipilih

## **Menentukan Ruang Lingkup otomatisasi**

Ruang lingkup otomatisasi adalah area atau scope yang akan di otomatisasi. Poin-poin berikut dapat membantu menentukan scope of testing:

- Fitur-fitur yang penting bagi bisnis
- Skenario yang memiliki sejumlah besar data
- Fungsionalitas umum di seluruh aplikasi
- Kelayakan teknis
- Sejauh mana komponen bisnis digunakan kembali
- Kompleksitas dari test cases
- Kemampuan untuk menggunakan kasus pengujian yang sama untuk pengujian cross-browser

## **Perencanaan, Desain, dan Pengembangan**

Selama fase ini tim QA membuat strategi dan rencana yang berisi rincian berikut:

- Telah memilih tools yang akan digunakan untuk otomatisasi
- Desain kerangka dan fiturnya
- Item-item otomatisasi In-Scope dan Out-of-scope

- Persiapan testbed otomatisasi
- Jadwal dan Timeline dari penulisan skrip dan eksekusi
- Menyelesaikan pengembangan pengujian otomatis

## Test Execution

Skrip Otomatisasi dieksekusi selama fase ini. Skrip memerlukan data uji yang telah diatur lebih dulu untuk dijalankan. Setelah dieksekusi maka akan memberikan laporan tes yang rinci.

Eksekusi dapat dilakukan menggunakan automation tool secara langsung atau melalui Test Management Tool yang akan memanggil automation tool, jika sudah diintegrasikan.

Contoh:

Quality Center adalah Test Management Tool yang pada eksekusinya akan meminta pelaksanaan skrip otomatisasi. Skrip dapat dieksekusi dalam satu mesin atau sekelompok mesin. Eksekusi bisa dilakukan pada malam hari, untuk menghemat waktu.

## Automation Framework

Framework adalah seperangkat pedoman otomatisasi yang membantu dalam hal berikut ini:

- Mempertahankan konsistensi Pengujian
- Memperbaiki struktur pengujian
- Minimum penggunaan kode
- Kurangi Pemeliharaan kode
- Meningkatkan penggunaan kembali
- Penguji Non Teknis dapat dilibatkan dalam kode
- Masa pelatihan penggunaan tool dapat dikurangi
- Dapat menggunakan data dimanapun data tersebut disimpan

Ada empat jenis framework yang digunakan dalam automation testing:

1. Data Driven Automation Framework
2. Keyword Driven Automation Framework
3. Modular Automation Framework
4. Hybrid Automation Framework

## **Praktik Terbaik Automation Tool**

Untuk mendapatkan ROI otomatisasi secara maksimum, tim dapat mengamati hal-hal berikut:

- Cakupan Otomatisasi perlu ditentukan secara rinci sebelum proyek dimulai. Ini menetapkan ekspektasi dari Automation.
- Pilih tool otomatisasi yang tepat: Sebuah tool tidak boleh dipilih berdasarkan popularitasnya, tetapi sesuai dengan persyaratan otomatisasi.
- Pilih framework yang sesuai
- Standar Scripting - Standar harus diikuti saat menulis skrip untuk Otomasi.
- Buat skrip seragam, komentar, dan indentasi dari kode
- Error handling yang memadai. Bagaimana kesalahan ditangani pada kegagalan sistem atau perilaku tak terduga dari aplikasi.
- Pesan yang ditentukan pengguna harus distandarisasi agar penguji dapat mengerti.

Ukur metrik dari keberhasilan otomasi tidak dapat ditentukan dengan membandingkan upaya manual dengan upaya otomasi tetapi dengan menggunakan matrik berikut:

- Persentase cacat yang ditemukan
- Waktu yang diperlukan untuk pengujian otomatisasi untuk setiap siklus rilis
- Waktu Minimal yang digunakan untuk rilis
- Indeks Kepuasan Pelanggan
- Peningkatan produktivitas

**Automation Testing** bergantung pada pre-scripted test yang berjalan secara otomatis. Automation Testing berfungsi untuk membandingkan hasil yang diharapkan dengan hasil yang sebenarnya. Sehingga dapat mengetahui apakah aplikasi atau website berjalan sesuai dengan apa yang diharapkan yang dapat dilihat dengan menggunakan Automation testing tools.

Menggunakan Automation testing tools bisa dilakukan secara berulang kali sehingga dapat sangat membantu QA dalam proses pengujian. Jika didapatkan hasil pengujian tidak sama dengan yang diharapkan, berarti telah ditemukan bug dan bisa segera dilaporkan kepada tim developer untuk dilakukan perbaikan atau fixing. Proses ini akan sangat terbantu jika QA menggunakan Automation testing tools.

Tetapi tidak semua Automation testing tools bekerja secara sama persis. Beberapa tools pengujian perangkat lunak menghasilkan kinerja yang luar biasa pada apa yang mereka lakukan, sementara yang lain tidak memiliki kemampuan yang maksimal. Ada tiga tipe utama Automation testing tools yaitu : open-source, komersial, dan custom. Dengan open-source, kode sumber diterbitkan dan tersedia bagi siapa saja untuk digunakan. Tools komersial memiliki lebih banyak dukungan namun berbayar. Kemudian, ada custom tools yang dibuat untuk proyek tertentu.

## **Selenium**

Selenium berfungsi untuk menguji otomatisasi dan dianggap sebagai standar industri untuk pengujian otomatisasi antarmuka pengguna aplikasi Web. Untuk pengembang dan penguji yang memiliki pengalaman dan keterampilan dalam pemrograman dan scripting, Selenium menawarkan fleksibilitas yang tidak terlihat dalam banyak alat dan kerangka kerja otomatisasi pengujian.

Pengguna dapat menulis skrip uji dalam berbagai bahasa berbeda (seperti Java, Groovy, Python, C #, PHP, Ruby, dan Perl) dan kompatibel dengan berbagai sistem operasi (Windows, Mac, Linux) dan browser (Chrome, Firefox, IE, dan Headless browsers).



**Kelebihan:**

Tools Pengujian Selenium adalah standar industri dalam hal perangkat lunak dan tools load testing. Selenium bersifat Open-Source dan dikembangkan melalui komponen IDE dan WebDriver yang merupakan kerangka kerja inti untuk beberapa tools automation testing lainnya, termasuk Katalon, Watir, dan Robot Framework. Selenium berjalan pada sistem operasi Windows, Linux, dan OS X dan kompatibel dengan sebagian besar browser, termasuk Internet Explorer, Chrome, Firefox, dan Headless. Selenium juga menawarkan fitur untuk merekam dan memutar ulang. Mungkin yang terbaik dari semuanya yaitu programmer dapat menulis tes dalam bahasa umum seperti Java, C#, Perl, Python, JavaScript, Ruby, Groovy, dan PHP.

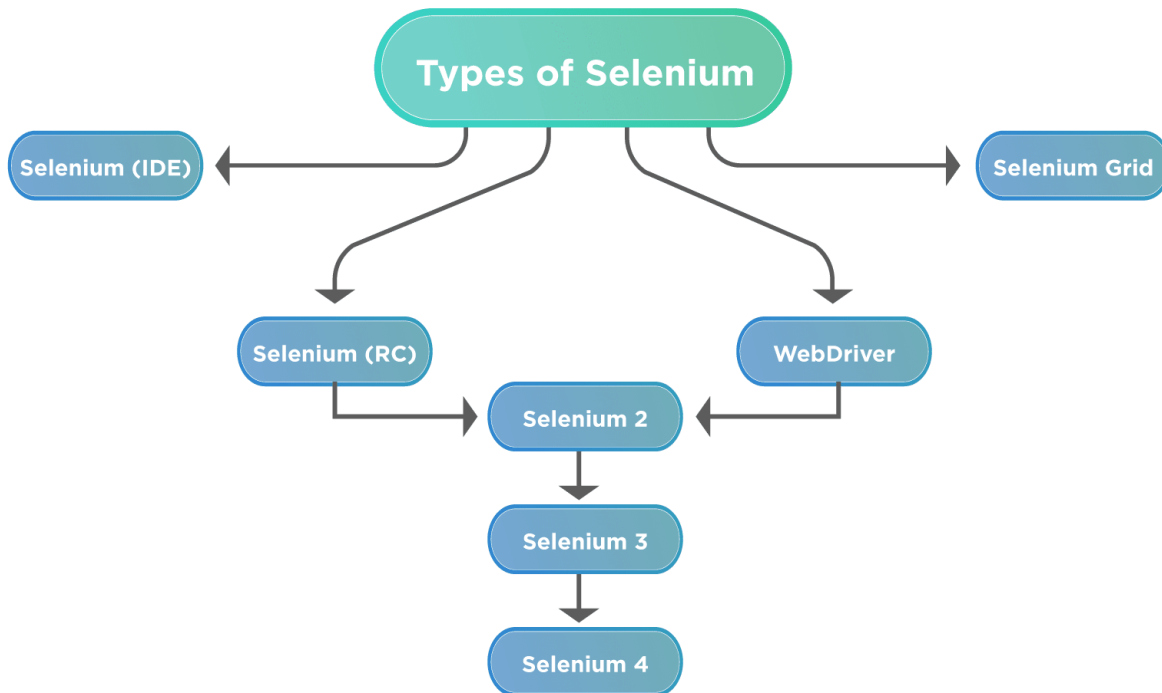
**Fitur:**

- Menyediakan ketentuan untuk mengekspor skrip rekaman dalam bahasa lain seperti Java, Ruby, RSpec, Python, C#, dll
- Dapat digunakan dengan kerangka kerja seperti JUnit dan TestNG
- Dapat menjalankan beberapa tes pada suatu waktu
- Lengkap otomatis untuk perintah Selenium yang umum
- Tes walkthrough
- Mengidentifikasi elemen menggunakan id, nama, jalur-X, dll
- Simpan tes sebagai Ruby Script, HTML, dan format lain
- Menyediakan pilihan untuk menegaskan judul untuk setiap halaman
- Mendukung berkas selenium user-extensions
- Memungkinkan untuk memasukkan komentar di tengah skrip untuk pemahaman dan pengawakutan yang lebih baik

**Kekurangan :** Diperlukan pemahaman dan keterampilan untuk menginstal, menggunakan, dan mengintegrasikan tools Selenium.

Selenium Software sendiri bukan sebuah alat tunggal melainkan sebuah serangkaian perangkat lunak yang masing-masing bagian dari serangkaian perangkat lunak tersebut akan melayani kebutuhan pengujian yang berbeda dari suatu organisasi.

Berikut Jenis-jenis selenium::



## **Selenium IDE (Integrated Development Environment)**

Selenium IDE merupakan automation testing tool yang paling sederhana dan merupakan yang paling mudah untuk dipelajari. Keuntungan menggunakan automation testing tool ini adalah bahwa tes yang direkam melalui plugin dapat diekspor dalam berbagai bahasa pemrograman seperti: Java, Ruby, Python dll.

Kelebihan Selenium IDE:

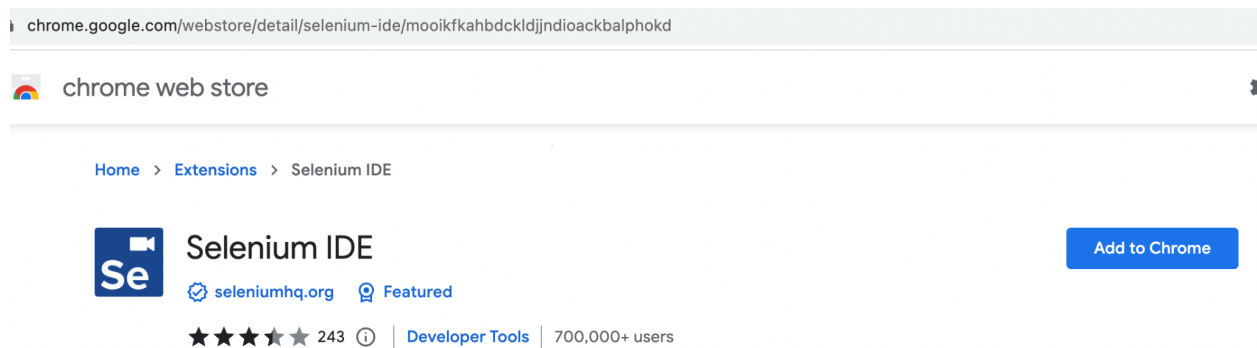
- Perintah Selenium seperti clickAndWait, assert, verify, etc..
- Pencari seperti id, nama, xpath, pemilih css, dll.
- Menjalankan kode JavaScript khusus menggunakan runScript
- Mengekspor kasus uji dalam berbagai format.
- Untuk membuat tes dengan sedikit atau tanpa pengetahuan sebelumnya dalam pemrograman.

- Untuk membuat kotak uji sederhana dan ruang uji yang dapat kamu ekspor nanti ke RC atau WebDriver.
- Untuk menguji aplikasi web hanya terhadap Firefox dan Chrome

Selenium IDE bisa didapatkan pada browser extensions.

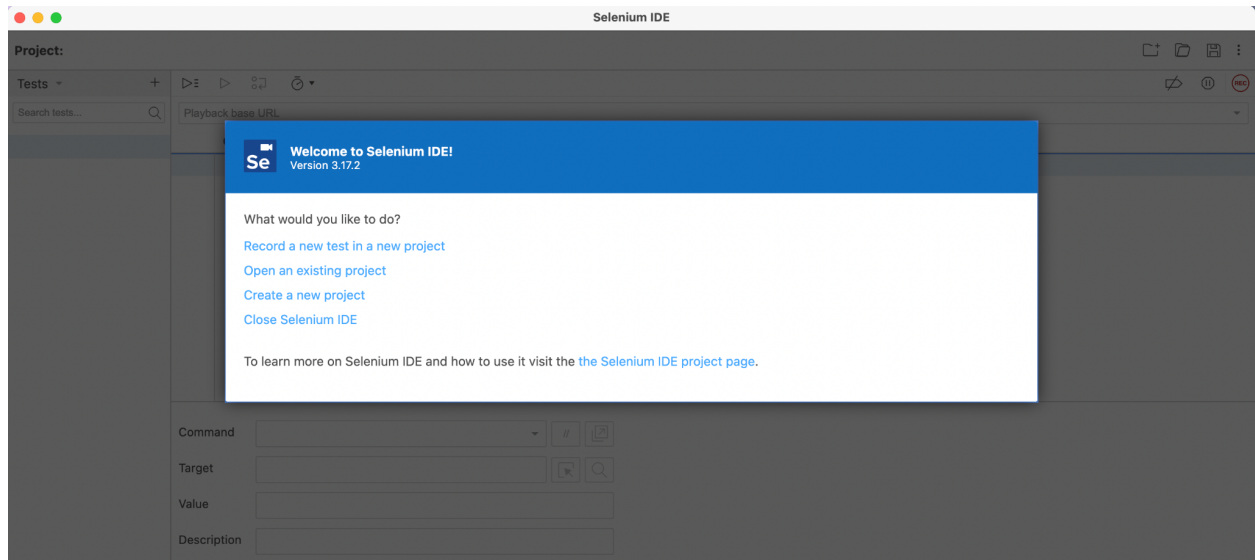
Berikut contoh Selenium IDE pada Chrome extensions:

Klik Add to Chrome untuk menambahkan extension.

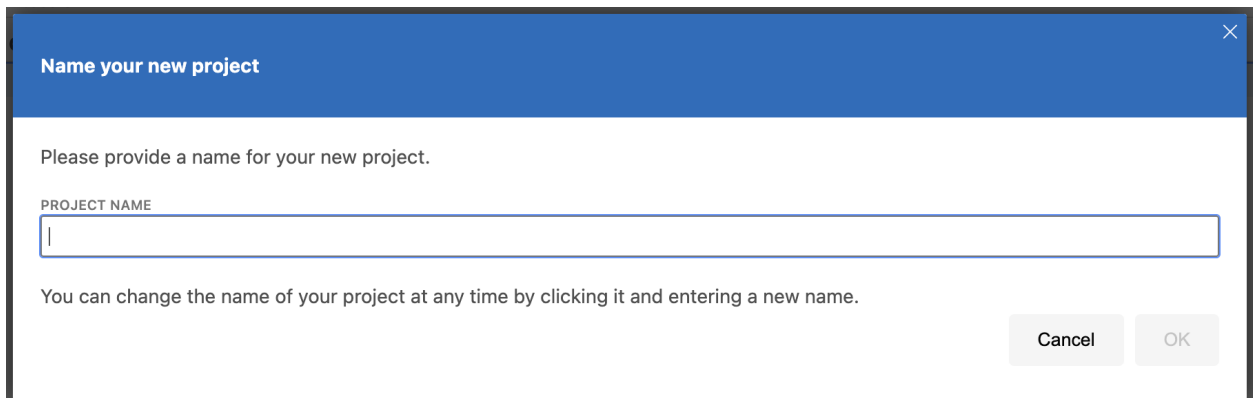


Ketika memulai project, ada beberapa pilihan seperti membuat project baru atau membuka project yang sudah ada.

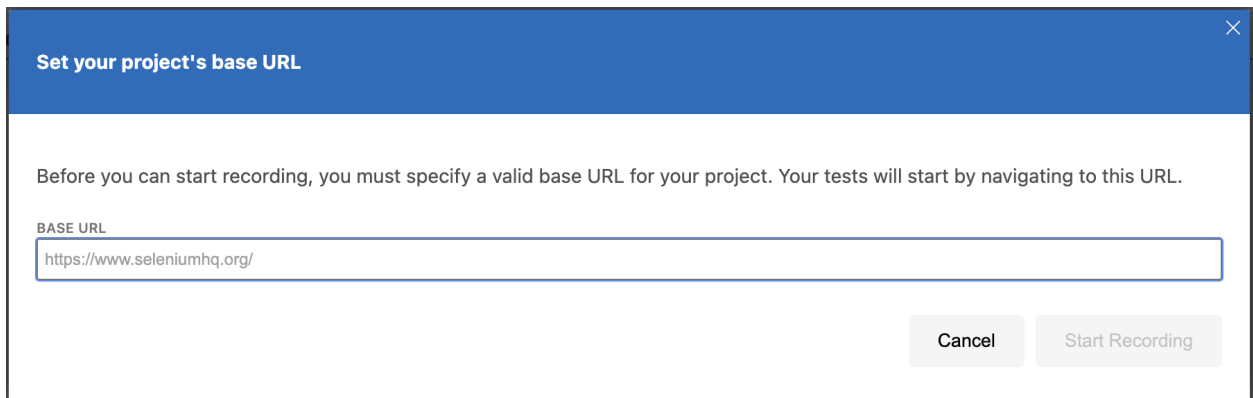
Atau dapat langsung memulai project baru dengan melakukan record pada web dengan melakukan klik Record a new test in a new project.



Inputkan nama project pada kolom Project name, lalu klik OK

The screenshot shows a dialog box titled 'Name your new project'. It has a blue header bar with a close button (X) in the top right corner. The main content area is white and contains the text 'Please provide a name for your new project.' Below this is a text input field labeled 'PROJECT NAME'. Underneath the input field, it says 'You can change the name of your project at any time by clicking it and entering a new name.' At the bottom right, there are two buttons: 'Cancel' and 'OK'.

Inputkan alamat URL project yang akan dilakukan automation test, lalu klik start recording

The screenshot shows a dialog box titled 'Set your project's base URL'. It has a blue header bar with a close button (X) in the top right corner. The main content area is white and contains the text 'Before you can start recording, you must specify a valid base URL for your project. Your tests will start by navigating to this URL.' Below this is a text input field labeled 'BASE URL' containing the text 'https://www.seleniumhq.org/'. At the bottom right, there are two buttons: 'Cancel' and 'Start Recording'.

Setiap langkah langkah yang dilakukan oleh mouse dan keyboard akan direkam (record) oleh Selenium IDE, lalu kita akan mendapatkan hasil sebagai berikut

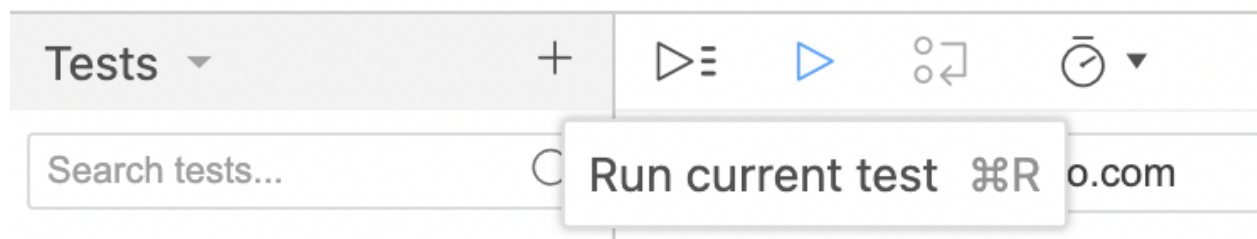
https://www.saucedemo.com

	Command	Target	Value
1	open	/	
2	set window size	1269x875	
3	type	css=[data-test="username"]	username
4	type	css=[data-test="password"]	password
5	click	css=[data-test="login-button"]	
6	click	css=[data-test="error"]	
7	close		

1. Open  
Ketika membuka browser
2. Set window size  
Selenium secara otomatis mengatur ukuran layar
3. Type username  
Menginputkan username pada kolom username
4. Type password  
Menginputkan password pada kolom password
5. Click login button  
Melakukan klik tombol login
6. Click error  
Melakukan klik pada error message
7. Close  
Adalah ketika close browser

Setelah ini, script bisa dijalankan kembali dengan menekan tombol play

## Project: rakamin\*



## **Selenium RC (Remote Control)**

Selenium RC merupakan web automation testing tool pertama yang memungkinkan para penggunanya untuk menggunakan bahasa pemrograman yang mereka sukai.

- Merancang tes menggunakan bahasa yang lebih ekspresif
- Menjalankan tes terhadap berbagai browser (kecuali HtmlUnit) pada sistem operasi yang berbeda.
- Menerapkan tes pada web apps di berbagai lingkungan menggunakan Selenium Grid.
- Menguji pada web apps terhadap browser baru yang mendukung JavaScript.
- Menguji aplikasi web dengan skenario berbasis AJAX yang kompleks.

### **Kekurangan Selenium RC**

- Arsitektur yang Rumit
- Eksekusi skrip pengujian memakan waktu karena Selenium RC menggunakan perintah JavaScript sebagai instruksi ke browser dan menghasilkan kinerja yang lambat
- API kurang berorientasi objek
- Tidak ada dukungan untuk browser Headless HTMLUnit (Browser tak terlihat)

Keterbatasan Selenium RC menyebabkan pengembangan kerangka otomatisasi baru, yaitu Selenium WebDriver. Setelah pengenalan WebDriver pada tahun 2006, komplikasi yang timbul di RC dapat diatasi dan diselesaikan.

## **Selenium WebDriver**

Selenium WebDriver hadir sebagai automation testing tool yang lebih baik daripada Selenium IDE dan Selenium RC dalam berbagai aspek. Selenium Webdriver sendiri menerapkan pendekatan yang lebih modern dan stabil dalam melakukan otomatisasi pada browser.

### **Kelebihan Selenium WebDriver**

- Menggunakan bahasa pemrograman tertentu dalam merancang uji kasus.
- Menguji aplikasi yang kaya akan fungsionalitas berbasis AJAX.
- Dapat menjalankan tes pada browser HtmlUnit.
- Mampu membuat hasil tes yang disesuaikan.

## **Selenium Grid**

Selenium Grid adalah automation testing tool yang dapat menjalankan pengujian paralel (menjalankan beberapa tes sekaligus) di berbagai mesin dan browser yang berbeda pada saat. Selenium grid dapat bersamaan digunakan dengan Selenium RC

- Menjalankan skrip Selenium RC di beberapa browser dan sistem operasi secara bersamaan.
- Menjalankan test suite yang sangat besar, yang perlu diselesaikan dalam waktu secepat mungkin.

## **Mengapa Selenium?**

Selenium merupakan salah satu automation testing tool yang harus kita pelajari sebagai Quality Assurance, karena

### **Open source**

Selenium bersifat open source sehingga user bisa mengunduh dan menggunakan berbagai fitur yang tersedia pada selenium secara gratis. Selain itu, kita juga bisa berbagi, memperluas, dan memodifikasi kode yang tersedia.

### **Support banyak bahasa pemrograman dan framework**

Selenium mendukung dan terintegrasi dari berbagai bahasa pemrograman, framework, dan berbagai platform pengembangan.

## **Fleksibel**

Kemampuannya untuk regrouping dan refactoring test case dapat membantu developer atau QA untuk mempercepat perubahan kode, mengecek dan mengurangi duplikasi kode dan meminimalkan komplikasi. Sehingga dapat membuat manajemen pengujian menjadi jauh lebih sederhana dan efisien.

## **Support berbagai browser dan sistem operasi**

Pengujian dapat dilakukan di berbagai sistem operasi seperti: Windows , Mac atau Linux dan menggunakan browser apa saja seperti: Google Chrome, Mozilla Firefox, Internet Explorer, Microsoft Edge, Safari atau Opera.

## **Element Locator**

Element Locator (pencari elemen) adalah istilah yang digunakan dalam konteks pengujian otomatis aplikasi web, terutama ketika menggunakan alat seperti Selenium. Ini merujuk pada teknik atau metode yang digunakan untuk mengidentifikasi dan memilih elemen-elemen tertentu di halaman web yang akan diinteraksikan oleh skrip pengujian.

Dalam pengujian otomatis, diperlukan interaksi dengan elemen-elemen di dalam halaman web, seperti tombol, tautan, formulir, teks, dan lain-lain. Namun, elemen-elemen ini perlu diidentifikasi secara unik oleh alat pengujian agar dapat diakses dan dimanipulasi. Inilah peran dari "Element Locator".

Beberapa metode umum yang digunakan untuk mengidentifikasi elemen-elemen ini melalui teknik pencarian elemen atau "element locators" termasuk:

- ID: Menggunakan atribut id unik yang diberikan pada elemen.
- Name: Menggunakan atribut name pada elemen.
- Class Name: Menggunakan nama kelas CSS element.
- Tag Name: Menggunakan tag HTML element.



- XPath: Menggunakan jalur XPath dari elemen.
- CSS Selector: Menggunakan selektor CSS untuk elemen.
- Link Text: Menggunakan teks tautan (link text) dari tautan.
- Partial Link Text: Menggunakan sebagian dari teks tautan.
- DOM: Menggunakan metode Document Object Model (DOM).

Ketika membuat skrip pengujian otomatis dengan alat seperti Selenium, Anda perlu menentukan lokasi elemen di halaman web menggunakan salah satu dari teknik ini. Alat tersebut kemudian akan menggunakan metode tersebut untuk menemukan elemen yang sesuai dan melakukan interaksi yang diinginkan, seperti mengklik, mengisi formulir, atau memverifikasi teks.

Pemilihan metode pencarian yang tepat sangat penting untuk memastikan kestabilan dan konsistensi skrip pengujian. Karena struktur halaman web dapat berubah, harus dipastikan bahwa elemen locator stabil dan tidak mudah terpengaruh oleh perubahan dalam kode atau tampilan halaman.