

Analisis Pemanfaatan Playwright untuk Automasi Pengujian Aplikasi Berbasis Web (Studi Kasus : Sistem Manajemen Jaringan)

by John Doe

Submission date: 27-Nov-2021 11:26AM (UTC+0700)

Submission ID: 1713127751

File name: an_Playwright_untuk_Automasi_Pengujian_Aplikasi_Berbasis_Web.pdf (415.32K)

Word count: 3494

Character count: 23105

Analisis Pemanfaatan Playwright untuk Automasi Pengujian Aplikasi Berbasis Web

(Studi Kasus : Sistem Manajemen Jaringan)

Abstract—Pengujian menjadi fase penting dalam *System Development Life Cycle* (SDLC) karena disetiap pengembangan perangkat lunak adanya *bug* tidak bisa dihindari. Dalam pengujian sistem apalagi yang memiliki cakupan luas, terdapat ratusan *test case* yang harus dieksekusi. Untuk mempermudah pengujian dan menghindari *human error* selama pengujian, diperlukan bantuan *automation testing tools*. Banyak *tools* yang dapat digunakan untuk pengujian web, seperti *tools* populer Selenium dan yang terbaru yaitu Playwright. Untuk mengetahui penggunaan Playwright sebagai alat automasi terbaru, dilakukan percobaan pengujian Playwright dan menganalisis perbandingan dengan Selenium. Disimpulkan bahwa Playwright lebih cocok diimplementasikan oleh penguji yang baru mengenal *automation test* karena kemudahan dalam penggunaannya. Adapun Selenium lebih mudah diimplementasikan oleh penguji yang terbiasa dengan pengujian automasi dan memiliki *skills* pemrograman handal. *Skills* tersebut diperlukan untuk pembuatan *script test* dan pengintegrasian dengan *tools* lain.

Keywords—*Pengujian Automasi, Playwright, Selenium*

I. PENDAHULUAN

Pengujian perangkat lunak berperan penting dalam *System Development Life Cycle* (SDLC). Fase pengujian bertujuan untuk mencari sebanyak mungkin kesalahan yang terdapat pada sistem [1]. Pengujian didefinisikan sebagai proses verifikasi serta validasi hasil pengembangan sistem berdasarkan kebutuhan pemangku kepentingan. Proses tersebut dapat dilakukan dengan cara mengevaluasi perangkat lunak dari sisi spesifikasi kebutuhan, logika pemrograman, dan *user experience*. Selanjutnya hasil dari pengujian dapat dibandingkan dengan ekspektasi hasil uji yang sudah didokumentasikan dalam sebuah kasus uji (*test case*). Penting untuk memperhatikan kualitas perangkat lunak yang dikembangkan, apalagi untuk sebuah perusahaan *software house* yang mengembangkan perangkat lunak dalam modul besar dan harus bertanggung jawab terhadap klien mengenai kualitas produknya. Hal tersebut selain untuk menciptakan kepercayaan klien kepada perusahaan juga meminimalisir kerugian biaya maupun sumber daya manusia.

Terdapat 2 metode yang dapat digunakan untuk pengujian perangkat lunak berbasis web yaitu pengujian manual dan automasi. Pengujian manual adalah pengujian yang mengandalkan ketelitian penguji karena dilakukan manual tanpa menggunakan bantuan alat automasi. Adapun pengujian automasi dijalankan menggunakan bantuan *automation testing tools* dan bergantung kepada *script test*. Pengeksekusian *script test* akan membandingkan hasil uji aktual dengan hasil uji ekspektasi yang sudah didokumentasikan dalam *script* tersebut. Alat pengujian automasi yang dapat digunakan untuk pengujian aplikasi berbasis web adalah Selenium dan Playwright.

Selama proses pengujian terdapat ratusan *test case* yang harus diuji sebelum sistem tersebut dirilis. Jika dilakukan secara manual maka proses pengujian akan membutuhkan waktu yang lama serta permasalahan *human error* ketika pengujian tidak bisa dihindarkan. Permasalahan tersebut dapat diatasi dengan pengujian automasi. Berdasarkan penelitian dari [2] dijabarkan bahwa pengujian automasi mengeksekusi kasus uji lebih cepat daripada pengujian manual dan dapat diandalkan dari sisi ketepatan pengujian karena mampu menemukan cacat aplikasi lebih banyak. Pengujian manual juga tidak efektif untuk melakukan pengujian regresi yaitu pengujian yang dilakukan ketika terdapat perbaikan atau pembaruan yang bertujuan untuk memastikan bahwa perbaikan tersebut tidak menyebabkan *bug* baru pada aplikasi.

Pada makalah ini akan disampaikan hasil analisis implementasi *automation test* untuk pengujian perangkat lunak berbasis web. Metode yang digunakan untuk proses analisis yaitu percobaan pengujian dan perbandingan. Oleh karena itu, akan dilakukan percobaan pengujian fungsionalitas menggunakan *automation testing tools* terbaru yaitu Playwright pada Sistem Manajemen Jaringan. Proses selanjutnya melakukan analisis dengan metode perbandingan yang variabel pembandingnya didapat berdasarkan percobaan pengujian dengan Playwright. Adapun *tools* yang akan dibandingkan yaitu Selenium yang merupakan salah satu alat pengujian automasi web populer. Dari hasil analisis, akan disajikan kelemahan, kelebihan, serta pengimplementasian yang cocok dari masing-masing *tools* yang dapat digunakan sebagai referensi dalam pemilihan web *automation testing tools*.

II. KAJIAN PUSTAKA

Terdapat beberapa penelitian yang telah dilakukan sebelumnya terkait implementasi pengujian aplikasi berbasis web menggunakan Selenium. Dalam penelitian Karuniawati (2015), dilakukan analisis implementasi Selenium dalam pengujian perangkat lunak berbasis web[3]. Pada penelitian tersebut, penguji menggunakan metode *Equivalence Partitioning* yang membagi data input menjadi 2 kategori yaitu data valid dan invalid. Kedua kategori data tersebut menjadi data input untuk setiap pengujian. Hasil pengujian menyebutkan bahwa Selenium memiliki performa pengujian yang baik karena mampu menghasilkan *output* pengujian yang sesuai dengan kategori data input dengan persentase lulus uji 100%.

Terdapat juga penelitian yang membahas mengenai perbandingan 3 *Automation Testing Tool* yaitu Selenium, Quick Test Professional, dan TestComplete[4]. Dalam studi kasus tersebut, peneliti melakukan perbandingan berdasarkan

lisensi, *application support*, bahasa, sistem operasi, *programming skills*, penggunaan, *report generation*, serta *platform dependency*. Hasil dari penelitian tersebut menyebutkan bahwa Selenium dan Quick Test Professional merupakan *Automation Testing Tool* terbaik yang dapat digunakan untuk melakukan pengujian aplikasi berbasis web. Akan tetapi, Selenium unggul dari sisi *open source* sedangkan Quick Test Professional memiliki biaya lisensi dan *maintenance* yang mahal.

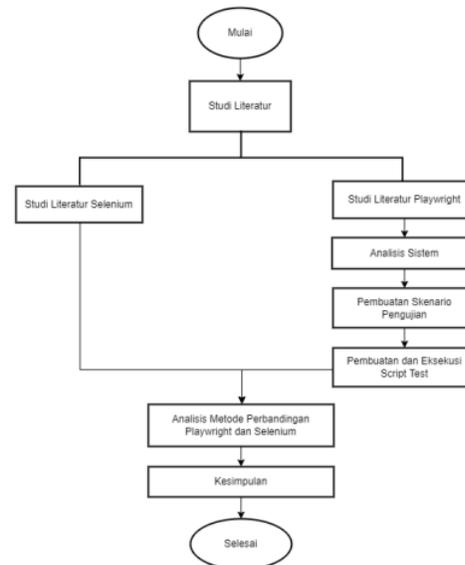
Terdapat penelitian yang melakukan survei terkait dengan Selenium[5]. Survei disebarluaskan menggunakan kuisioner Google Forms ke beberapa komunitas Selenium yaitu Selenium Conf Slack yang calon respondennya ahli dan memiliki ketertarikan dengan Selenium. Penguji juga menyebarkan survei kedalam komunitas Selenium Reddit yang berisikan calon respon memiliki minat proaktif pada Selenium. Peneliti melakukan survei dalam 7 kategori yaitu informasi umum Selenium, tes pengembangan, sistem unit, frameworks, infrastruktur, dukungan komunitas, dan yang terakhir yaitu pertanyaan terbuka mengenai pengalaman menggunakan Selenium. Salah satu informasi yang didapatkan adalah Selenium WebDriver merupakan jenis Selenium yang paling banyak digunakan dengan persentase kurang lebih 99% (71 dari 72 responden). Selenium juga merupakan automation testing tools yang populer dan memiliki komunitas luas yang tersebar di berbagai negara.

Tidak hanya itu, terdapat penelitian yang menguji Selenium dengan mengintegrasikan dengan tools pihak ketiga[6]. Selenium memiliki batasan tidak bisa melakukan *generate* hasil pengujian yang manfaatnya cukup penting untuk pendokumentasian dan analisis *root cause* jika terjadi kegagalan uji. Oleh karena itu, peneliti melakukan percobaan pengujian dan mengintegrasikan dengan TestNg. Hasil dari pengujian tersebut menyebutkan bahwa laporan pengujian berhasil dibuat dan berjalan dengan baik dalam Selenium. Hal tersebut dapat dijadikan solusi terkait batasan fitur pelaporan di Selenium.

Dari penelitian tersebut, Selenium merupakan *automation testing tool* yang populer dan unggul dari sisi *open source* maupun performa. Jenis Selenium yang populer dan paling banyak digunakan adalah Selenium WebDriver karena performa stabil kelengkapan fiturnya daripada jenis Selenium yang lain[5]. Akan tetapi, dari semua penelitian belum ada yang melakukan pengujian menggunakan alat automasi terbaru yaitu Playwright dan membandingkan *tools* tersebut dengan alat automasi yang lain. Oleh karena itu, dalam makalah ini akan melakukan percobaan pengujian web menggunakan Playwright dan melakukan perbandingan dengan Selenium WebDriver.

III. METODOLOGI

Proses analisis pemanfaatan dan pengimplementasian Playwright sebagai alat automasi terbaru dilakukan selama proses pengujian. Hasil analisis pengujian menggunakan Playwright akan dibandingkan dengan Selenium yang menjadi salah satu *web automation tools* populer. Adapun alur metodologi dalam makalah ditunjukkan pada Gambar 1.



Gambar 1. Alur Metodologi

A. Studi Literatur

Pengumpulan informasi didapatkan melalui studi literatur yang diambil dari jurnal ilmiah yang sesuai dengan topik pembahasan Playwright dan Selenium.

B. Analisis Sistem

Percobaan pengujian dengan Playwright diawali dengan memahami proses Sistem Manajemen Tower yang terdapat pada *Technical Document* dan *User Guide*. *Technical document* yaitu dokumen yang mendokumentasikan semua kebutuhan dalam pengembangan sistem seperti informasi arsitektur, database, alur proses bisnis BPMN, dokumentasi API, dan kebutuhan fungsional sistem. Adapun *user guide* merupakan dokumentasi panduan pengguna dalam menjalankan sebuah sistem.

C. Pembuatan Skenario Pengujian

Test case atau skenario pengujian merupakan pendokumentasian skenario kasus uji yang mencakup langkah-langkah pengujian, hasil ekspektasi, hasil aktual, dan status pengujian. Dalam pembuatan skenario tes terdapat 2 tipe pengujian yaitu pengujian positif dan negatif. Tipe pengujian positif adalah skenario untuk memastikan bahwa sistem melakukan proses yang seharusnya dilakukan sedangkan pengujian negatif yaitu memastikan bahwa sistem tidak melakukan proses yang seharusnya tidak dilakukan. Dokumen skenario digunakan sebagai acuan dalam menjalankan pengujian sistem

D. Pembuatan dan Eksekusi *Script Test*

Setelah proses skenario pengujian selesai, langkah selanjutnya yaitu pembuatan *script test*. *Script test* dibuat berdasarkan skenario uji yang telah didokumentasikan. Setiap skenario memiliki 1 *script test* untuk dieksekusi. *Script test* akan memvalidasi proses aktual yang terjadi di dalam sistem. Dalam pengekseskuan *script test* akan

dihasilkan hasil aktual pengujian. Hasil aktual tersebut memiliki 2 status pengujian yaitu *passed* dan *failed*.

E. Analisis Perbandingan Playwright dan Selenium

Dalam proses analisis Playwright dan Selenium menggunakan metode perbandingan. Variabel pembanding yang digunakan dalam proses analisis menggunakan variabel yang didapat pada proses pengujian menggunakan Playwright. Dalam metode perbandingan tersebut, informasi Playwright didapatkan dari percobaan pengujian sistem, sedangkan informasi mengenai perbandingan Selenium didapatkan dari studi literatur penelitian yang telah ada.

IV. HASIL DAN PEMBAHASAN

A. Percobaan Pengujian Menggunakan Playwright

1) Pemahaman dan Analisis Sistem

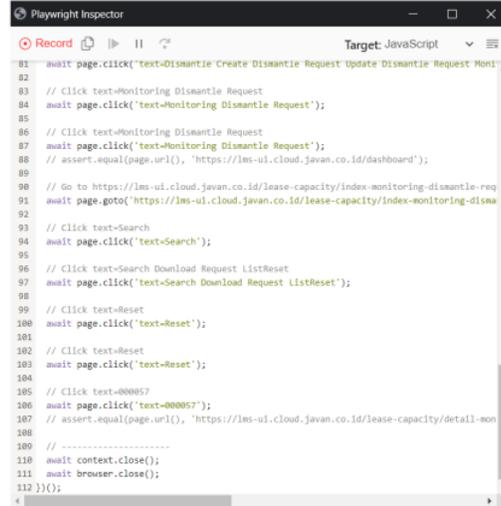
Pengujian web yang dilakukan menggunakan Playwright yaitu Sistem Manajemen Jaringan pada sebuah Perusahaan Telekomunikasi di modul *Upgrade Downgrade* proses *Create Link*. Sistem Manajemen Jaringan memiliki 8 role pengguna dan 12 proses yang saling berhubungan. Setiap proses didalam modul tersebut saling berhubungan, untuk itu dipilih proses *Create Link* sebagai studi kasus dikarenakan proses penginputan data pertama dilakukan di menu tersebut. Dalam proses tersebut, pengguna harus melanjutkan proses jaringan yang telah diselesaikan dalam modul yang lain. Pengguna diberikan pilihan untuk melakukan *upgrade* maupun *downgrade* kapasitas jaringan dan terdapat pilihan untuk *submit* maupun *save draft* data

2) Pembuatan Skenario Pengujian

Setelah memahami proses sistem maka langkah selanjutnya yaitu membuat skenario pengujian. Skenario tersebut berisi pengujian *Create Link* tipe *Upgrade* dengan 2 pilihan proses yaitu *save draft* dan *submit*. Proses *submit* dapat dieksekusi jika semua *field mandatory* telah diinputkan, sedangkan proses *save draft* tidak diwajibkan untuk menginputkan *field mandatory* kecuali *field request type*. Adapun 4 *field mandatory* yaitu *request type*, *capacity adjustment*, *project*, dan *purpose*. Adapun dalam skenario *Create Link* terdapat 10 skenario dengan 6 tipe pengujian positif dan 4 negatif.

3) Pembuatan dan Eksekusi *Script Test*

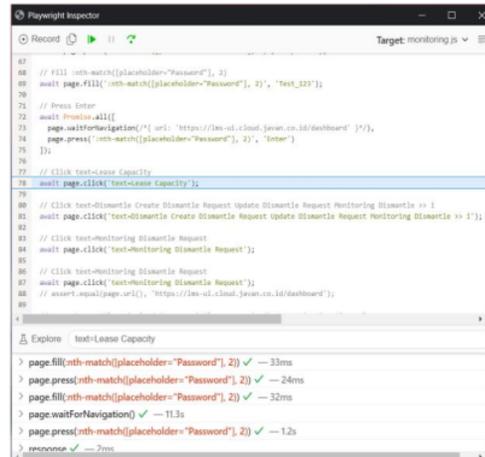
Pembuatan *script test* menggunakan Playwright dapat dilakukan tanpa pemrograman manual. *Script test* Playwright dihasilkan dari rekaman aktivitas pengguna pada proses yang sedang diuji menggunakan web browser dengan bahasa pemrograman JavaScript seperti pada Gambar 2.



```
81 await page.click('text-Dismantle Create Dismantle Request Update Dismantle Request Monitoring Dismantle Request');
82 // Click text-Monitoring Dismantle Request
83 await page.click('text-Monitoring Dismantle Request');
84 // Click text-Monitoring Dismantle Request
85 await page.click('text-Monitoring Dismantle Request');
86 // assert.equal(page.url(), 'https://lms-ui.cloud.javan.co.id/dashboard');
87 // Go to https://lms-ui.cloud.javan.co.id/lease-capacity/index-monitoring-dismantle-request
88 await page.goto('https://lms-ui.cloud.javan.co.id/lease-capacity/index-monitoring-dismantle-request');
89 // Click text-Search
90 await page.click('text-Search');
91 // Click text-Search Download Request ListReset
92 await page.click('text-Search Download Request ListReset');
93 // Click text-Reset
94 await page.click('text-Reset');
95 // Click text-Reset
96 await page.click('text-Reset');
97 // Click text-000057
98 await page.click('text-000057');
99 // assert.equal(page.url(), 'https://lms-ui.cloud.javan.co.id/lease-capacity/detail-monitoring-dismantle-request');
100 // -----
101 await context.close();
102 await browser.close();
103 }());
```

Gambar 2. Pembuatan *Script Test*

Adapun pengujian kali ini akan dieksekusi menggunakan mode *debug headless* dan *headful* untuk mengetahui secara rinci proses pengujian. Mode *headless* akan mengeksekusi tes browser tanpa antarmuka (UI) sistem sedangkan mode *headful* akan menampilkan keseluruhan antarmuka sistem yang sedang diuji. Pengeksekusian *script test* diperlihatkan pada Gambar 3.



```
67 // Fill nth-match[placeholder="Password", 2]
68 await page.fill('nth-match[placeholder="Password", 2]', 'Test123');
69 // Press Enter
70 await Promise.all([
71   page.waitForNavigation({url: 'https://lms-ui.cloud.javan.co.id/dashboard'}),
72   page.press('nth-match[placeholder="Password", 2]', 'Enter')
73 ]);
74 // Click text-Lease Capacity
75 await page.click('text-Lease Capacity');
76 // Click text-Dismantle Create Dismantle Request Update Dismantle Request Monitoring Dismantle Request
77 await page.click('text-Dismantle Create Dismantle Request Update Dismantle Request Monitoring Dismantle Request');
78 // Click text-Monitoring Dismantle Request
79 await page.click('text-Monitoring Dismantle Request');
80 // Click text-Monitoring Dismantle Request
81 await page.click('text-Monitoring Dismantle Request');
82 // assert.equal(page.url(), 'https://lms-ui.cloud.javan.co.id/dashboard');
83 // -----
84 await context.close();
85 await browser.close();
86 }());
```

Gambar 3. Pengeksekusian *Script Test*

Tabel I memperlihatkan 10 skenario dan hasil eksekusi pengujian menggunakan Playwright. Dari 10 skenario pengujian, Playwright berhasil mengeksekusi kasus uji sesuai dengan ekspektasi uji.

TABEL I. SKENARIO TES

ID	Sub Modul	Deskripsi	Skenario	Ekspektasi	Hasil Aktual	Status
1	Search by Parameters	Positif - Melakukan searching data menggunakan parameters yang ada	1. Buka web sistem manajemen tower 2. Klik Login, masukkan username dan password 3. Klik menu Manajemen Tower 4. Klik menu Create Link Upgrade & Downgrade 5. Input valid keyword di search bar 6. Klik Search	Dapat menampilkan data sesuai dengan parameters yang dicari	Hasil aktual sesuai dengan ekspektasi	<i>Passed</i>
2	Search by Parameters	Negatif - Melakukan searching data invalid menggunakan parameters yang ada	1. Buka web sistem manajemen tower 2. Klik Login, masukkan username dan password 3. Klik menu Manajemen Tower 4. Klik menu Create Link Upgrade & Downgrade 5. Input invalid keyword atau simbol di search bar untuk setiap parameters 6. Klik Search	Data tidak ditemukan, sistem menampilkan pesan "Data tidak ditemukan"	Hasil aktual sesuai dengan ekspektasi	<i>Passed</i>
3	Upgrade - Submit Data	Positif - Submit data tipe Upgrade dengan menginputkan semua field mandatory	1. Buka web sistem manajemen tower 2. Klik Login, masukkan username dan password 3. Klik menu Manajemen Tower 4. Klik menu Create Link Upgrade & Downgrade 5. Klik ID Data yang akan diproses 6. Pilih type Upgrade. Input semua field mandatory 7. Klik Submit	Data sukses terkirim, sistem menampilkan pesan "Data telah sukses terkirim!".	Hasil aktual sesuai dengan ekspektasi	<i>Passed</i>
4	Upgrade - Submit Data	Negatif - Submit data tipe Upgrade dengan tidak menginputkan field mandatory "Capacity Adjustment"	1. Buka web sistem manajemen tower 2. Klik Login, masukkan username dan password 3. Klik menu Manajemen Tower 4. Klik menu Create Link Upgrade & Downgrade 5. Klik ID Data yang akan diproses 6. Pilih type Upgrade. Input semua field kecuali Capacity Adjustment 7. Klik Submit	Button Submit tidak aktif, pengguna tidak bisa melakukan proses Submit	Hasil aktual sesuai dengan ekspektasi	<i>Passed</i>
5	Upgrade - Submit Data	Negatif - Submit data tipe Upgrade dengan tidak menginputkan field mandatory "Project Name"	1. Buka web sistem manajemen tower 2. Klik Login, masukkan username dan password 3. Klik menu Manajemen Tower 4. Klik menu Create Link Upgrade & Downgrade 5. Klik ID Data yang akan diproses 6. Pilih type Upgrade. Input semua field kecuali Project Name 7. Klik Submit	Button Submit tidak aktif, pengguna tidak bisa melakukan proses Submit	Hasil aktual sesuai dengan ekspektasi	<i>Passed</i>
6	Upgrade - Submit Data	Negatif - Submit data tipe Upgrade dengan tidak menginputkan field mandatory "Purpose"	1. Buka web sistem manajemen tower 2. Klik Login, masukkan username dan password 3. Klik menu Manajemen Tower 4. Klik menu Create Link Upgrade & Downgrade 5. Klik ID Data yang akan diproses 6. Pilih type Upgrade. Input semua field kecuali Purpose 7. Klik Submit	Button Submit tidak aktif, pengguna tidak bisa melakukan proses Submit	Hasil aktual sesuai dengan ekspektasi	<i>Passed</i>
7	Upgrade - Save Draft Data	Positif - Save draft data tipe Upgrade dengan menginputkan semua field	1. Buka web sistem manajemen tower 2. Klik Login, masukkan username dan password 3. Klik menu Manajemen Tower 4. Klik menu Create Link Upgrade & Downgrade 5. Klik ID Data yang akan diproses 6. Pilih type Upgrade. Input semua field 7. Klik Save Draft	Data sukses terkirim. Sistem menampilkan pesan "Data telah sukses tersimpan!"	Hasil aktual sesuai dengan ekspektasi	<i>Passed</i>
8	Upgrade - Save Draft Data	Positif - Save Draft data tipe Upgrade dengan tidak menginputkan field mandatory "Capacity Adjustment"	1. Buka web sistem manajemen tower 2. Klik Login, masukkan username dan password 3. Klik menu Manajemen Tower 4. Klik menu Create Link Upgrade & Downgrade 5. Klik ID Data yang akan diproses 6. Pilih type Upgrade. Input semua field kecuali Capacity Adjustment 7. Klik Save Draft	Data sukses terkirim. Sistem menampilkan pesan "Data telah sukses tersimpan!"	Hasil aktual sesuai dengan ekspektasi	<i>Passed</i>
9	Upgrade - Save Draft Data	Positif - Save Draft data tipe Upgrade dengan tidak menginputkan field	1. Buka web sistem manajemen tower 2. Klik Login, masukkan username dan password 3. Klik menu Manajemen Tower 4. Klik menu Create Link Upgrade & Downgrade 5. Klik ID Data yang akan diproses	Data sukses terkirim. Sistem menampilkan pesan "Data telah sukses tersimpan!"	Hasil aktual sesuai dengan ekspektasi	<i>Passed</i>

		mandatory "Project Name"	6. Pilih type Upgrade. Input semua field kecuali Project Name 7. Klik Save Draft			
10	Upgrade - Save Draft Data	Positif - Save Draft data tipe Upgrade dengan tidak menginputkan field mandatory "Purpose"	1. Buka web sistem manajemen tower 2. Klik Login, masukkan username dan password 3. Klik menu Manajemen Tower 4. Klik menu Create Link Upgrade & Downgrade 5. Klik ID Data yang akan diproses 6. Pilih type Upgrade. Input semua field kecuali Purpose 7. Klik Save Draft	Data sukses terkirim. Sistem menampilkan pesan "Data telah sukses tersimpan!"	Hasil aktual sesuai dengan ekspektasi	Passed

instalasi Playwright sudah mendukung penginstalan semuanya sehingga tidak diperlukan instalasi satu persatu.

B. Perbandingan Selenium dan Playwright

Playwright merupakan *automation testing tool* lintas browser terbaru berbasis *open source* yang dikembangkan oleh Microsoft yang menyediakan API untuk mengendalikan web browser. *Library* Node.js menyediakan API tingkat tinggi untuk mengontrol web browser melalui protokol DevTools[7]. Playwright dapat melakukan pengujian web dengan menggunakan *browser* Chromium, Webkit, Firefox dan mendukung 5 bahasa pemrograman yaitu JavaScript, Python, Java, .NET, dan TypeScript. Cara kerja Playwright yaitu merekam aktivitas pengujian pada web yang sedang diuji melalui browser dan melakukan *generate* kode pemrograman dari setiap aktivitas uji. Pengeksekusian script kode dapat dijalankan dengan mode *headless* dan *headful*.

Adapun Selenium adalah seperangkat *automation testing tool* yang mendukung hampir semua web browser dan menyediakan kompatibilitas dengan bahasa pemrograman populer yaitu C#, Java, JavaScript, Ruby, Python, dan PHP. Selenium mendukung automasi semua browser melalui penggunaan WebDriver. Setiap browser memiliki WebDriver tertentu yang disebut dengan *driver*. *Driver* adalah komponen yang bertanggung jawab untuk mendelegasikan ke browser dan menjadi perantara antara Selenium dengan web browser. Terdapat 3 jenis Selenium yaitu Selenium IDE yang melakukan perekaman aktifitas pengujian pada sistem, Selenium WebDriver yang menggunakan protokol WebDriver sebagai perantara script pengujian dengan web browser dan Selenium Grid yang dapat digunakan bersamaan dengan Selenium WebDriver untuk melakukan pengujian *multiple test case*.

Perbandingan Selenium dan Playwright dalam 5 parameter perbandingan, sebagai berikut:

1) Proses Instalasi

Untuk menggunakan Selenium WebDrivers dalam pengujian automasi, diperlukan instalasi Selenium *libraries* dan *WebDrivers*[8]. Proses instalasi Selenium *libraries* bergantung kepada bahasa pemrograman yang akan digunakan. Proses instalasi selanjutnya yaitu *WebDrivers*, pengujian harus melakukan instalasi *driver* yang sesuai dengan browser yang akan digunakan. Daftar *driver* masing-masing browser terdapat pada Tabel II. Adapun proses instalasi Playwright memerlukan NodeJs dan Playwright. Download NodeJs melalui website resminya, setelah itu jalankan *command* di terminal untuk melakukan instalasi Playwright. Berbeda dengan Selenium WebDrivers yang harus menginstalasi setiap bahasa pemrograman dan *drivers* yang sesuai dengan browser yang akan digunakan, *command*

TABEL II. DAFTAR DRIVERS

Web Browser	Driver
Chromium	Chromedriver
Firefox	Geckodriver
Microsoft Edge	Edgedriver
Internet Explorer	Internet Explorer driver
Safari	Safaridriver

2) Pembuatan Script Test

Dalam penelitian [5], peneliti melakukan survei mengenai Selenium dalam kategori pengembangan tes. Dalam kategori tersebut disebutkan bahwa sebanyak 36,11% responden mengalami permasalahan dalam pengujian dan pengembangan *script test*. Jawaban responden meliputi sulit untuk dibuat, sulit untuk ditulis, sulit untuk ditiru sebagai pengguna, dan sulit untuk digunakan kembali. Pengujian pemula yang belum memiliki pengalaman *automation test* maupun *skills* pemrograman yang baik merasa kesulitan ketika mencoba membangun pengujian tes Selenium. Permasalahan tersebut juga disebutkan pada penelitian [4] mengenai perbandingan *automation testing tools* yang menyebutkan bahwa *skills* pemrograman di Selenium sangat dibutuhkan selain untuk pembuatan pengujian juga pengintegrasian dengan *tools* lainnya. Dijelaskan juga bahwa waktu pembuatan *script* di Selenium masuk ke kategori *slow* (lambat). Sedangkan Playwright melakukan pembuatan *test script* dengan perekaman aktivitas pengujian yang langsung di *generate* dalam bentuk kode pemrograman dengan bahasa pemrograman yang dapat dipilih sesuai dengan keinginan pengujian. Kode yang telah di *generate* juga diberikan penjelasan maupun petunjuk singkat terkait proses apa yang dituliskan di dalam *script* tersebut. Hal tersebut dapat membantu pengujian untuk melakukan pembuatan *script test* dengan mudah.

3) Dokumentasi Laporan Pengujian

Laporan pengujian penting untuk mendokumentasikan semua hasil uji yang dilakukan dengan pengujian paralel atau *multiple test case*. Playwright tidak memiliki fitur yang digunakan untuk menghasilkan laporan kasus pengujian. Dalam membantu pendokumentasian, pengujian dapat menggunakan fitur tangkapan layar dan perekaman video, akan tetapi fitur tersebut tidak cukup efektif untuk laporan

pengujian *multiple test* di Playwright. Adapun Selenium secara default tidak memiliki fitur hasil pelaporan tetapi Selenium dapat diintegrasikan dengan *tools* pihak ketiga untuk memperluas fitur-fiturnya. Salah satu *tools* pihak ketiga yang mampu mengatasi permasalahan hasil pelaporan yaitu TestNg. Dalam penelitian [6] disebutkan bahwa integrasi Selenium dengan TestNg berjalan cukup baik dan berhasil untuk memperluas fitur pelaporan hasil uji.

4) Mode Eksekusi Tes

Playwright mendukung eksekusi pengujian dengan mode *headless* dan *headful*. Mode *headless* adalah pengeksekusian tes browser tanpa antarmuka (UI) sistem sedangkan mode *headful* akan menampilkan keseluruhan antarmuka sistem yang sedang diuji. Mode *headless* membutuhkan lebih sedikit memori dan berjalan lebih cepat sedangkan mode *headful* dapat dimanfaatkan untuk UI debug [7]. Adapun Selenium hanya mencakup mode *headless* dalam eksekusi pengujiannya. Untuk pengeksekusian *multiple test*, Selenium WebDriver dapat menggunakan Selenium Grid yang harus dikonfigurasi kembali sebelum melakukan pengeksekusian *multiple test*. Sedangkan Playwright menggunakan *Playwright test* yang dikonfigurasi bersamaan dengan instalasi Playwright.

5) Komunitas Pendukung

Lebih dari 10 tahun Selenium membantu para *developers* untuk melakukan pengujian otomatis berbagai web. Hingga saat ini, Selenium masih menjadi salah satu *automation testing tools* yang populer dan penggunaannya banyak dipercaya oleh penguji maupun *developers* [5]. Hal tersebut membuat Selenium memiliki cukup banyak komunitas sehingga referensi pembelajarannya cukup banyak. Sedangkan Playwright belum memiliki komunitas sebanyak Selenium.

V. KESIMPULAN

Percobaan pengujian Sistem Manajemen Jaringan dalam proses *Create Link* berhasil dilakukan oleh Playwright dengan persentase lulus uji 100%. Kemudahan Playwright dalam pembuatan *script test* yang terotomasi secara langsung ketika melakukan perekaman kasus uji sangat membantu penguji. Akan tetapi, Playwright belum mendukung dokumentasi laporan pengujian yang penting untuk mendokumentasikan hasil uji ketika melakukan pengujian *multiple test case*. Adapun Selenium WebDriver dapat menghasilkan laporan pengujian dengan mengintegrasikan *tools* pihak ketiga TestNg.

Baik Playwright maupun Selenium merupakan *automation testing tools* yang memiliki performa bagus untuk pengujian web. Masing-masing memiliki kelebihan dan kekurangannya, Selenium WebDriver akan lebih mudah diimplementasikan dan digunakan oleh penguji yang sudah terbiasa dalam pengujian otomatis dan memiliki *skills* pemrograman handal. *Skills* tersebut digunakan untuk proses pengujian maupun pengintegrasian dengan *tools* pihak ketiga. Sedangkan Playwright cocok diimplementasikan dan digunakan oleh penguji yang baru saja mengenal *automation testing* karena kemudahannya dalam pengujian. Alat ini dapat digunakan dari pemula hingga profesional, apalagi intensitas tim *development* Playwright yang selalu melakukan pembaharuan *tools* akan membuat Playwright besar di masa mendatang.

REFERENCES

- [1] Jin, J., & Xue, F. (2011, October). Rethinking software testing based on software architecture. In *2011 Seventh International Conference on Semantics, Knowledge and Grids* (pp. 148-151). IEEE.
- [2] Sharma, M., & Angmo, R. (2014). Web based automation testing and tools. *International Journal of Computer Science and Information Technologies*, 5(1), 908-912.
- [3] Karuniawati, S., Widowati, S., & Hakim, I. L. (2015). Implementasi Metode Cause Effect Graphing (CEG) dalam Pengujian Requirement Perangkat Lunak (Studi Kasus: Aplikasi G-College). *eProceedings of Engineering*, 2(2).
- [4] Bhagat, B., Bhattacharjee, S., & Ratre, S. (2020). Software testing techniques & automation tools. *Mukt Shabd Journal*, 5957-5962.
- [5] García, B., Gallego, M., Gortázar, F., & Muñoz-Organero, M. (2020). A survey of the selenium ecosystem. *Electronics*, 9(7), 1067.
- [6] Gojare, S., Joshi, R., & Gaigaware, D. (2015). Analysis and design of selenium webdriver automation testing framework. *Procedia Computer Science*, 50, 341-346.
- [7] Playwright. (2021). Documentation Playwright. Diakses dari <https://playwright.dev/>
- [8] Selenium. (2021). Documentation Getting Started Installation Selenium. Diakses dari https://www.selenium.dev/documentation/getting_start ed

Analisis Pemanfaatan Playwright untuk Automasi Pengujian Aplikasi Berbasis Web (Studi Kasus : Sistem Manajemen Jaringan)

ORIGINALITY REPORT

1 %

SIMILARITY INDEX

1 %

INTERNET SOURCES

0 %

PUBLICATIONS

0 %

STUDENT PAPERS

PRIMARY SOURCES

1

journal.uii.ac.id

Internet Source

<1 %

2

journals.ums.ac.id

Internet Source

<1 %

3

pt.scribd.com

Internet Source

<1 %

4

es.scribd.com

Internet Source

<1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography On