

Analisis Pemanfaatan Playwright untuk Pengujian Otomatis Aplikasi Berbasis Web

(Studi Kasus: Sistem Manajemen Jaringan)

Anisa Amalia
Program Studi Informatika
Universitas Islam Indonesia
Yogyakarta, Indonesia
18523157@students.uii.ac.id

Andhik Budi Cahyono
Program Studi Informatika
Universitas Islam Indonesia
Yogyakarta, Indonesia
andhikbudi@uui.ac.id

Abstrak—Pengujian menjadi fase penting dalam *System Development Life Cycle* (SDLC) karena di setiap pengembangan perangkat lunak adanya *bug* yang tidak bisa dihindari. Dalam pengujian sistem yang memiliki cakupan luas, terdapat ratusan *test case* yang harus dieksekusi. Untuk mempermudah pekekseskuan 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 otomatis terbaru, dilakukan percobaan pengujian Playwright dan menganalisis menggunakan metode komparatif 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 otomatis dan memiliki *skills* pemrograman handal. *Skills* tersebut diperlukan untuk pembuatan *script test* dan pengintegrasian dengan *tools* lain.

Kata Kunci—*Pengujian Otomatis, Playwright, Selenium, Sistem Manajemen Jaringan*

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 mengevaluasi perangkat lunak dari sisi spesifikasi kebutuhan, logika pemrograman, dan *user experience*. Penting untuk memperhatikan kualitas perangkat lunak, apalagi untuk sebuah perusahaan *software house* yang mengembangkan perangkat lunak dalam modul besar. Hal tersebut selain untuk menciptakan kepercayaan klien kepada perusahaan juga meminimalisir kerugian biaya maupun sumber daya manusia.

Pengujian perangkat lunak dapat dilakukan dengan dua cara yaitu pengujian manual dan otomatis. Pengujian manual adalah pengujian yang mengandalkan ketelitian penguji karena dilakukan manual tanpa menggunakan bantuan alat uji. Adapun pengujian otomatis dijalankan menggunakan bantuan *automation testing tools* dan bergantung kepada *script test*. Pekekseskuan *script test* akan membandingkan hasil uji aktual dengan hasil uji ekspektasi yang sudah didokumentasikan dalam *script* tersebut.

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

Salah satu hal yang perlu diperhatikan dalam pengujian otomatis adalah pemilihan *automation testing tools* yang tepat. Pertimbangan dalam pemilihan alat pengujian dapat disesuaikan dengan kebutuhan pengujian serta kondisi sumber daya manusia dalam tim proyek. Tidak hanya itu, kelebihan serta kelemahan dari masing-masing alat uji juga harus dipertimbangkan.

Pada makalah ini akan disampaikan hasil analisis implementasi pengujian otomatis pada aplikasi berbasis web. Alat uji yang akan dilakukan pembahasan yaitu Playwright dan Selenium. Hal tersebut dikarenakan Playwright merupakan alat uji web berbasis lintas browser yang digunakan oleh PT Javan Cipta Solusi dalam melakukan pengujian Sistem Manajemen Jaringan. Adapun Selenium yaitu alat uji web berbasis lintas browser yang disebutkan akan menjadi tolak ukur alat uji pada masa mendatang [3].

Proses analisis menggunakan percobaan pengujian dan metode komparatif. Oleh karena itu, akan dilakukan pengujian fungsionalitas menggunakan alat uji terbaru yaitu Playwright pada Sistem Manajemen Jaringan. Adapun variabel yang digunakan pada metode komparatif didapatkan melalui hasil pengujian Playwright dan pengalaman penguji selama menggunakan Playwright. Hasil tersebut akan dibandingkan dengan Selenium dan menggunakan penelitian yang telah ada sebelumnya sebagai sumber data pengujian.

Tujuan dari pembahasan studi kasus ini yaitu untuk mengimplementasikan pengujian otomatis menggunakan Playwright. Adapun dalam pembahasan ini akan disajikan

kelemahan, kelebihan, serta pengimplementasian yang cocok dari masing-masing alat uji Playwright dan Selenium. Hal tersebut dapat digunakan sebagai referensi dalam pemilihan web *automation testing tools*.

II. KAJIAN PUSTAKA

Terdapat beberapa penelitian yang telah dilakukan sebelumnya terkait pengujian aplikasi berbasis web dengan menggunakan *automation testing tools*. Sebuah penelitian melakukan analisis implementasi Selenium dalam pengujian perangkat lunak berbasis web [4]. Pada penelitian tersebut, penguji menggunakan metode *Equivalence Partitioning* yang membagi data input menjadi dua 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%.

Penelitian yang membahas mengenai perbandingan tiga *automation testing tools* yaitu Selenium, Quick Test Professional, dan TestComplete [5]. Perbandingan ketiga *tools* dilakukan berdasarkan kriteria 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 tools* 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 survei pengalaman penggunaan Selenium dari beberapa ahli dan orang-orang yang memiliki minat serta kepentingan terhadapnya. Survei dilakukan melalui Google Forms ke beberapa komunitas Selenium seperti Selenium Conf Slack dan Selenium Reddit [6]. 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.

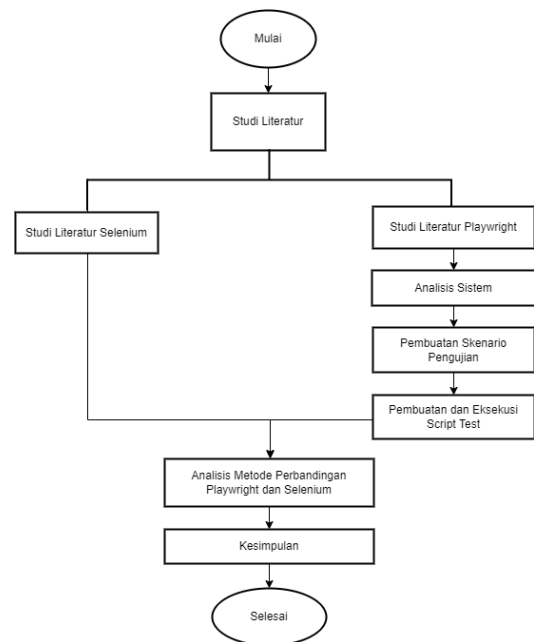
Tidak hanya itu, terdapat penelitian yang menguji Selenium dengan mengintegrasikan dengan *tools* pihak ketiga [7]. Selenium memiliki batasan tidak bisa melakukan *generate* hasil pengujian yang manfaatnya cukup penting untuk pendokumentasian dan analisis *root cause* jika terjadi kegagalan uji. Oleh 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 semua penelitian [4] hingga [7] menunjukkan bahwa Selenium merupakan alat pengujian otomatis yang populer untuk menguji aplikasi berbasis web. Jenis Selenium yang populer dan paling banyak digunakan adalah Selenium WebDriver karena performa stabil daripada jenis Selenium yang lain [6]. Data dan informasi yang disajikan dalam penelitian-penelitian tersebut akan menjadi basis pengetahuan yang digunakan untuk membandingkan hasil pengujian dengan salah satu alat pengujian terbaru yaitu Playwright.

Hal yang dibandingkan dalam studi kasus ini yaitu informasi mendasar mengenai kedua alat uji tersebut seperti bahasa pemrograman, dukungan web browser, proses instalasi, ketersediaan fitur laporan pengujian, mode eksekusi, hingga komunitas pendukung. Sumber informasi Playwright didapatkan melalui pengalaman penguji selama menggunakan Playwright sedangkan untuk Selenium didapatkan melalui analisis hasil dari penelitian-penelitian sebelumnya.

III. METODOLOGI

Proses analisis pemanfaatan dan pengimplementasian Playwright 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. Studi literatur tersebut menjadi sumber informasi dalam proses analisis dan perbandingan kasus uji.

B. Analisis Sistem

Percobaan pengujian dengan Playwright diawali dengan memahami proses Sistem Manajemen Jaringan 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 pengeksekusian *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 komparatif. Variabel perbandingan 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 21 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 menggunakan

Microsoft Excel. 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*. Dalam skenario *Create Link* terdapat 10 skenario dengan 6 tipe pengujian positif dan 4 negatif.

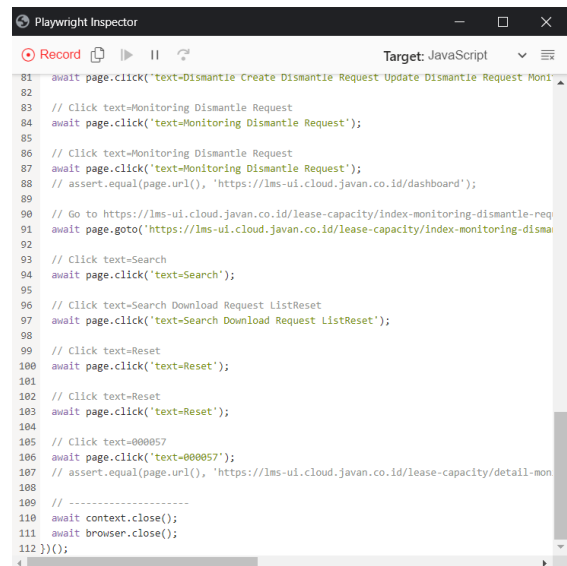
3) Pembuatan dan Eksekusi *Script Test*

Pembuatan *script test* dapat dilakukan tanpa pemrograman manual dengan menggunakan Playwright Codegen. *Script* yang dibuat menggunakan Playwright Codegen dihasilkan dari rekaman aktivitas pengguna pada proses yang sedang diuji menggunakan web browser. Adapun untuk perintah perekaman, penguji menggunakan *command line* yang dijalankan melalui terminal. Perintah perekaman dapat dilihat pada Gambar 2.

```
PS D:\testing> npx playwright codegen https://lms-ui.cloud.javan.co.id/
```

Gambar 2. Perintah Perekaman Sistem

Setelah perintah perekaman dieksekusi, akan muncul 2 jendela browser yaitu jendela web sistem dan playwright *inspector*. Jendela web sistem digunakan penguji untuk melakukan pengujian seperti biasa dan semua aktivitas penguji dalam browser tersebut akan di-*generate* ke dalam playwright *inspector* menjadi kode pemrograman. Adapun tampilan playwright *inspector* dengan bahasa pemrograman JavaScript dapat dilihat pada Gambar 3.



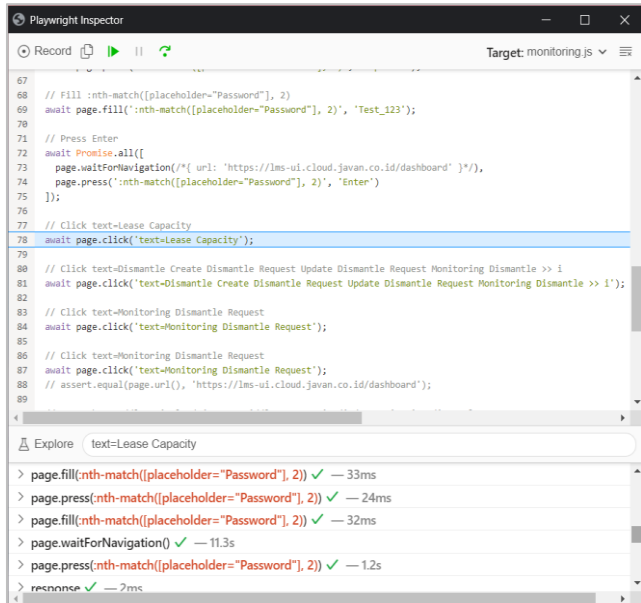
```
Playwright Inspector
Record
Target: JavaScript
81 await page.click('text=Dismantle Create Dismantle Request Update Dismantle Request Moni
82 // Click text-Monitoring Dismantle Request
83 // Click text-Monitoring Dismantle Request
84 await page.click('text=Monitoring Dismantle Request');
85
86 // Click text-Monitoring Dismantle Request
87 await page.click('text=Monitoring Dismantle Request');
88 // assert.equal(page.url(), 'https://lms-ui.cloud.javan.co.id/dashboard');
89
90 // Go to https://lms-ui.cloud.javan.co.id/lease-capacity/index-monitoring-dismantle-req
91 await page.goto('https://lms-ui.cloud.javan.co.id/lease-capacity/index-monitoring-disma
92
93 // Click text-Search
94 await page.click('text=Search');
95
96 // Click text-Search Download Request ListReset
97 await page.click('text=Search Download Request ListReset');
98
99 // Click text-Reset
100 await page.click('text=Reset');
101
102 // Click text-Reset
103 await page.click('text=Reset');
104
105 // Click text=@00057
106 await page.click('text=@00057');
107 // assert.equal(page.url(), 'https://lms-ui.cloud.javan.co.id/lease-capacity/detail-mon
108
109 // -----
110 await context.close();
111 await browser.close();
112 }());
```

Gambar 3. Tampilan Playwright *Inspector*

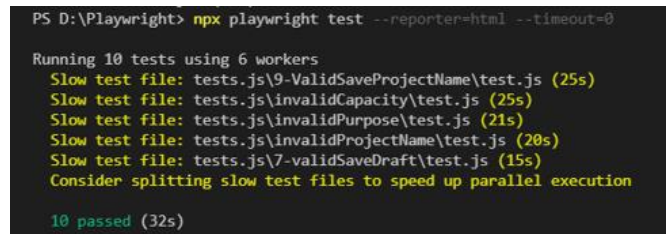
Pengeksekusian *script* Playwright dapat menggunakan mode *headless* dan *headful*. Mode *headless* akan mengeksekusi tes browser tanpa antarmuka (UI) sistem. Adapun untuk mode *headful* akan menampilkan keseluruhan antarmuka sistem yang sedang diuji.

Pengujian *multiple test* dijalankan menggunakan Playwright *Test Runner* mode *headless* yang akan menjalankan semua kasus uji secara bersamaan.tanpa

tampilan UI. Selain itu, penguji juga menjalankan *single test* dengan *debugging* untuk mengetahui alur *script* secara detail. *Debugging* biasanya dieksekusi jika terdapat penambahan kode pemrograman dengan tujuan untuk menguji apakah perubahan kode tersebut sudah sesuai. Proses *debug* dapat dilihat pada Gambar 4. Adapun hasil eksekusi *multiple test* dengan Playwright *Test Runner* dapat dilihat pada Gambar 5.



Gambar 4. Pengeksekusan *Single Script Test* Menggunakan *Debug*



Gambar 5. Hasil Pengeksekusan *Multiple Test* Playwright *Test Runner*

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

TABEL I. SKENARIO TES

ID	Sub Modul	Deskripsi	Skenario	Ekspektasi	Hasil Aktual	Status
1	<i>Search by Parameters</i>	Positif - Melakukan pencarian data menggunakan parameter yang ada	<ol style="list-style-type: none"> Buka web sistem manajemen jaringan Klik Login, masukkan <i>username</i> dan <i>password</i> Klik menu Manajemen Jaringan Klik menu <i>Create Link Upgrade & Downgrade</i> Input valid <i>keyword</i> di search bar Klik <i>Search</i> 	Dapat menampilkan data sesuai dengan parameters yang dicari	Hasil aktual sesuai dengan ekspektasi	<i>Passed</i>
2	<i>Search by Parameters</i>	Negatif - Melakukan pencarian data invalid menggunakan parameter yang ada	<ol style="list-style-type: none"> Buka web sistem manajemen jaringan Klik Login, masukkan <i>username</i> dan <i>password</i> Klik menu Manajemen Jaringan Klik menu <i>Create Link Upgrade & Downgrade</i> Input invalid <i>keyword</i> atau simbol di <i>search bar</i> untuk setiap parameters Klik <i>Search</i> 	Data tidak ditemukan, sistem menampilkan pesan "Data tidak ditemukan"	Hasil aktual sesuai dengan ekspektasi	<i>Passed</i>
3	<i>Upgrade - Submit Data</i>	Positif - <i>Submit</i> data tipe <i>Upgrade</i> dengan menginputkan semua <i>field mandatory</i>	<ol style="list-style-type: none"> Buka web sistem manajemen jaringan Klik Login, masukkan <i>username</i> dan <i>password</i> Klik menu <i>Manajemen Jaringan</i> Klik menu <i>Create Link Upgrade & Downgrade</i> Klik ID Data yang akan diproses Pilih tipe <i>Upgrade</i>, masukkan semua <i>field mandatory</i> Klik <i>Submit</i> 	Data sukses terkirim, sistem menampilkan pesan "Data telah sukses terkirim!".	Hasil aktual sesuai dengan ekspektasi	<i>Passed</i>
4	<i>Upgrade - Submit Data</i>	Negatif - <i>Submit</i> data tipe <i>Upgrade</i> dengan tidak menginputkan <i>field mandatory</i> " <i>Capacity Adjustment</i> "	<ol style="list-style-type: none"> Buka web sistem manajemen jaringan Klik Login, masukkan <i>username</i> dan <i>password</i> Klik menu Manajemen Jaringan Klik menu <i>Create Link Upgrade & Downgrade</i> Klik ID Data yang akan diproses Pilih tipe <i>Upgrade</i>, masukkan semua <i>field</i> 	<i>Button Submit</i> tidak aktif, pengguna tidak bisa melakukan proses <i>submit</i>	Hasil aktual sesuai dengan ekspektasi	<i>Passed</i>

			kecuali <i>Capacity Adjustment</i> 7. Klik <i>Submit</i>			
5	<i>Upgrade - Submit Data</i>	Negatif - <i>Submit</i> data tipe <i>Upgrade</i> dengan tidak menginputkan <i>field mandatory "Project Name"</i>	1. Buka web sistem manajemen jaringan 2. Klik Login, masukkan <i>username dan password</i> 3. Klik menu Manajemen Jaringan 4. Klik menu <i>Create Link Upgrade & Downgrade</i> 5. Klik ID Data yang akan diproses 6. Pilih tipe <i>Upgrade</i> , masukkan semua <i>field</i> kecuali <i>Project Name</i> 7. Klik <i>Submit</i>	<i>Button Submit</i> tidak aktif, pengguna tidak bisa melakukan proses <i>submit</i>	Hasil aktual sesuai dengan ekspektasi	<i>Passed</i>
6	<i>Upgrade - Submit Data</i>	Negatif - <i>Submit</i> data tipe <i>Upgrade</i> dengan tidak menginputkan <i>field mandatory "Purpose"</i>	1. Buka web sistem manajemen jaringan 2. Klik Login, masukkan <i>username dan password</i> 3. Klik menu <i>Manajemen Tower</i> 4. Klik menu <i>Create Link Upgrade & Downgrade</i> 5. Klik ID Data yang akan diproses 6. Pilih tipe <i>Upgrade</i> , masukkan semua <i>field</i> kecuali <i>Purpose</i> 7. Klik <i>Submit</i>	<i>Button Submit</i> tidak aktif, pengguna tidak bisa melakukan proses <i>submit</i>	Hasil aktual sesuai dengan ekspektasi	<i>Passed</i>
7	<i>Upgrade - Save Draft Data</i>	Positif - <i>Save draft</i> data tipe <i>Upgrade</i> dengan menginputkan semua <i>field</i>	1. Buka web sistem manajemen jaringan 2. Klik Login, masukkan <i>username dan password</i> 3. Klik menu Manajemen Jaringan 4. Klik menu <i>Create Link Upgrade & Downgrade</i> 5. Klik ID Data yang akan diproses 6. Pilih tipe <i>Upgrade</i> , masukkan semua <i>field</i> 7. Klik <i>Save Draft</i>	Data sukses terkirim. Sistem menampilkan pesan "Data telah sukses tersimpan!"	Hasil aktual sesuai dengan ekspektasi	<i>Passed</i>
8	<i>Upgrade - Save Draft Data</i>	Positif - <i>Save Draft</i> data tipe <i>Upgrade</i> dengan tidak menginputkan <i>field mandatory "Capacity Adjustment"</i>	1. Buka web sistem manajemen jaringan 2. Klik Login, masukkan <i>username dan password</i> 3. Klik menu Manajemen Jaringan 4. Klik menu <i>Create Link Upgrade & Downgrade</i> 5. Klik ID Data yang akan diproses 6. Pilih tipe <i>Upgrade</i> , masukkan semua <i>field</i> kecuali <i>Capacity Adjustment</i> 7. Klik <i>Save Draft</i>	Data sukses terkirim. Sistem menampilkan pesan "Data telah sukses tersimpan!"	Hasil aktual sesuai dengan ekspektasi	<i>Passed</i>
9	<i>Upgrade - Save Draft Data</i>	Positif - <i>Save Draft</i> data tipe <i>Upgrade</i> dengan tidak menginputkan <i>field mandatory "Project Name"</i>	1. Buka web sistem manajemen jaringan 2. Klik Login, masukkan <i>username dan password</i> 3. Klik menu Manajemen Jaringan 4. Klik menu <i>Create Link Upgrade & Downgrade</i> 5. Klik ID Data yang akan diproses 6. Pilih tipe <i>Upgrade</i> , masukkan semua <i>field</i> kecuali <i>Project Name</i> 7. Klik <i>Save Draft</i>	Data sukses terkirim. Sistem menampilkan pesan "Data telah sukses tersimpan!"	Hasil aktual sesuai dengan ekspektasi	<i>Passed</i>
10	<i>Upgrade - Save Draft Data</i>	Positif - <i>Save Draft</i> data tipe <i>Upgrade</i> dengan tidak menginputkan <i>field mandatory "Purpose"</i>	1. Buka web sistem manajemen jaringan 2. Klik Login, masukkan <i>username dan password</i> 3. Klik menu <i>Manajemen Jaringan</i> 4. Klik menu <i>Create Link Upgrade & Downgrade</i> 5. Klik ID Data yang akan diproses 6. Pilih tipe <i>Upgrade</i> , masukkan semua <i>field</i> kecuali <i>Purpose</i> 7. Klik <i>Save Draft</i>	Data sukses terkirim. Sistem menampilkan pesan "Data telah sukses tersimpan!"	Hasil aktual sesuai dengan ekspektasi	<i>Passed</i>

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 [8]. 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 Codegen yaitu merekam aktivitas pengujian pada web yang sedang diuji melalui browser dan melakukan *generate* kode pemrograman. Pengeksekusi *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 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*. Adapun dalam kasus ini, yang akan dilakukan perbandingan yaitu Playwright dan Selenium

WebDrivers. Hal tersebut dikarenakan Selenium WebDriver merupakan jenis paling populer dan paling banyak digunakan karena performa stabil daripada jenis Selenium yang lain [6].

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

1) Proses Instalasi

Untuk menggunakan Selenium WebDrivers dalam pengujian otomatis, diperlukan instalasi Selenium *libraries* dan *WebDrivers* [9]. 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.

TABEL II. DAFTAR DRIVERS

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

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* instalasi Playwright sudah mendukung penginstalan semuanya sehingga tidak diperlukan instalasi satu persatu. Perintah instalasi Playwright dapat dilihat pada Gambar 6, sedangkan hasil instalasi dapat dilihat pada Gambar 7.

```
# add dependency
npm i -D @playwright/test

# install supported browsers
npx playwright install
```

Gambar 6. Perintah Instalasi Playwright

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19043.1415]
(c) Microsoft Corporation. All rights reserved.

D:\Playwright>npm i -D @playwright/test

added 191 packages, and audited 192 packages in 36s

16 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

D:\Playwright>npx playwright install

D:\Playwright>
```

Gambar 7. Tampilan Hasil Instalasi Playwright

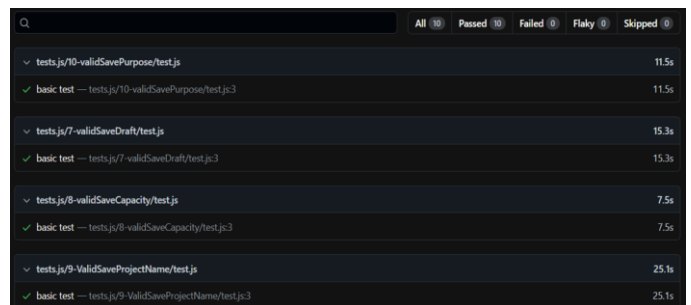
2) Pembuatan *Script Test*

Dalam penelitian [6], 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 [5] 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).

Adapun Playwright melakukan pembuatan *script test* 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*. Dalam membantu pendokumentasian, saat ini Playwright mempunyai *HTML Report* untuk mendokumentasikan hasil pengujian. Selain itu, Playwright juga dapat diintegrasikan dengan *tools* pihak ketiga yaitu *Allure Report*. Hasil pelaporan dengan *HTML Report* dapat dilihat pada Gambar 8.



Gambar 8. Tampilan Laporan Pengujian *HTML Report*

Begitu juga dengan 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 [7] 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 pengekseskuan tes tanpa antarmuka (UI) sedangkan mode *headful* menampilkan keseluruhan antarmuka sistem yang diuji. Mode *headless* membutuhkan lebih sedikit memori dan berjalan lebih cepat sedangkan mode *headful* dapat dimanfaatkan untuk UI debug [8].

Adapun Selenium hanya mencakup mode *headless* dalam eksekusi pengujianya. Untuk pengekseskuan *multiple test*, Selenium WebDriver dapat menggunakan Selenium Grid yang harus dikonfigurasi kembali sebelum melakukan pengekseskuan *multiple test*. Sedangkan Playwright menggunakan Playwright *Test Runner* yang dikonfigurasi bersamaan dengan instalasi Playwright.

5) Komunitas Pendukung

Selenium merupakan *automation testing tools* yang populer dan penggunaannya banyak dipercaya oleh pengujian maupun *developers* [6]. Hal tersebut membuat Selenium memiliki cukup banyak komunitas sehingga referensi pembelajarannya cukup banyak. Komunitas dukungan Selenium tersebar di seluruh dunia seperti Selenium Conf Slack, Selenium Reddit serta Stackoverflow yang menyajikan referensi bantuan dalam permasalahan penggunaan Selenium [6]. Adapun Playwright belum memiliki komunitas sebanyak Selenium.

Selenium dan Playwright memiliki kelebihan dan kekurangannya masing-masing. Adapun untuk semua hasil perbandingan dapat dilihat pada Tabel III.

TABEL III. PERBANDINGAN PLAYWRIGHT DAN SELENIUM

Variabel Perbandingan	Playwright	Selenium WebDrivers
Proses Instalasi	Proses instalasi cukup mudah dan satu <i>command line</i> dapat melakukan instalasi untuk semua browser	Proses instalasi bergantung kepada bahasa pemrograman serta harus menyesuaikan <i>driver</i> sesuai dengan browser yang akan digunakan.
Pembuatan <i>Script Test</i>	Dapat dibuat otomatis menggunakan Playwright Codegen dengan melakukan perekaman yang akan di- <i>generate</i> dalam kode pemrograman	Terdapat kesulitan dalam pembuatan <i>script test</i> secara manual dan membutuhkan <i>skills</i> pemrograman
Dokumentasi Laporan Pengujian	Dapat diintegrasikan dengan tools pihak ketiga <i>Allure Report</i>	Tidak memiliki fitur pendokumentasian laporan pengujian tetapi dapat diintegrasikan dengan <i>tools</i> pihak ketiga TestNg
Mode Eksekusi Tes	Dapat dieksekusi dengan mode <i>headless</i> dan <i>headful</i> . Tidak dibutuhkan pengintegrasian dengan <i>tools</i> lain untuk eksekusi <i>multiple test</i>	Pengekseskuan dengan mode <i>headless</i> dan membutuhkan pengintegrasian dengan Selenium Grid untuk eksekusi <i>multiple test</i>
Komunitas Pendukung	Komunitas pendukung belum seluas Selenium	Memiliki komunitas luas yang tersebar di dunia sehingga dapat memudahkan dalam proses pembelajaran Selenium
Bahasa Pemrograman	JavaScript, Python, Java, .NET, dan TypeScript	C#, Java, JavaScript, Ruby, Python, dan PHP
Web Browser Pendukung	Chrome, Firefox dan Webkit	Mozilla Firefox, Internet Explorer, Google Chrome, Safari dan Opera
Lisensi	Gratis, <i>open source</i>	Gratis, <i>open source</i>

V. KESIMPULAN

Percobaan pengujian Sistem Manajemen Jaringan dalam proses *Create Link* berhasil dilakukan oleh Playwright dengan persentase lulus uji 100%. Kemudahan Playwright Codegen dalam pembuatan *script test* yang terotomatis secara langsung ketika melakukan perekaman kasus uji sangat membantu pengujian. Adapun baik Playwright maupun Selenium merupakan *automation testing tools* yang memiliki performa bagus untuk pengujian web. Masing-masing alat uji memiliki kelebihan dan kelemahan. Oleh karena itu, dalam pemilihan alat uji dapat disesuaikan dengan kebutuhan proyek.

Selenium WebDrivers akan lebih mudah diimplementasikan dan digunakan oleh pengujian yang sudah terbiasa dalam pengujian otomatis dan memiliki *skills* pemrograman handal. *Skills* tersebut digunakan untuk proses pengujian seperti pembuatan *script test* maupun pengintegrasian dengan *tools* pihak ketiga. Adapun Playwright cocok diimplementasikan dan digunakan oleh pengujian 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.

Adapun saran untuk penelitian yang selanjutnya yaitu mengukur kecepatan Playwright dalam mengeksekusi *multiple test* untuk proses pengujian regresi. Selain itu terdapat juga saran untuk melakukan integrasi *tools* pihak ketiga dalam pendokumentasian laporan pengujian menggunakan *Allure Report*.

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] Saravanan, K., & Prasad, E. P. C. (2016). Open source software test automation tools: A competitive necessity. *Scholedge International Journal of Management & Development*, 3(6), 103-110.
- [4] 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).
- [5] Bhagat, B., Bhattacharjee, S., & Ratre, S. (2020). Software testing techniques & automation tools. *Mukt Shabd Journal*, 5957-5962.
- [6] García, B., Gallego, M., Gortázar, F., & Munoz-Organero, M. (2020). A survey of the selenium ecosystem. *Electronics*, 9(7), 1067.
- [7] Gojare, S., Joshi, R., & Gaigaware, D. (2015). Analysis and design of selenium webdriver automation testing framework. *Procedia Computer Science*, 50, 341-346.
- [8] Playwright. (2021). Documentation Playwright. Diakses dari <https://playwright.dev/>
- [9] Selenium. (2021). Documentation Getting Started Installation Selenium. Diakses dari https://www.selenium.dev/documentation/getting_started