

# PERBANDINGAN KINERJA FRAMEWORK AUTOMATION UI TESTING MENGGUNAKAN METODE THE DISTANCE TO THE IDEAL ALTERNATIVE

Gilang Aries Prasetyo<sup>1)</sup>, Novi Setiani<sup>2)</sup>

1. Universitas Islam Indonesia, Indonesia
2. Universitas Islam Indonesia, Indonesia

## Article Info

**Kata Kunci:** Framework otomasi pengujian; Pengujian UI; Selenium; Cypress; Playwright

**Keywords:** Test automation framework; UI testing; Selenium; Cypress; Playwright

## Article history:

Received 10 October 2024  
Revised 22 December 2024  
Accepted 1 March 2025  
Available online 1 March 2025

## DOI :

<https://doi.org/10.29100/jupi.v10i1.5760>

\* Corresponding author.

Corresponding Author

E-mail address:

[20523112@students.uii.ac.id](mailto:20523112@students.uii.ac.id)

## ABSTRAK

Penelitian ini bertujuan untuk membandingkan kinerja tiga framework otomatisasi pengujian, yaitu Selenium, Cypress, dan Playwright, dalam pengujian UI pada platform website. Metode yang digunakan adalah The Distance to the Ideal Alternative (DIA) untuk menilai performa framework berdasarkan parameter Technical and Economic View, Testing Process View, dan Quality View. Studi ini diawali dengan analisis literatur untuk menentukan parameter pengujian, diikuti oleh pembobotan parameter melalui survei pada software tester dan programmer profesional. Framework diuji pada website iconhub.io, dan hasil pengujian menunjukkan bahwa Cypress adalah framework terbaik dengan skor tertimbang terendah, menunjukkan keunggulan dalam kecepatan dan efisiensi. Hasil penelitian ini memberikan panduan bagi pengembang dalam memilih framework pengujian UI yang tepat dan efisien, berkontribusi pada peningkatan kualitas dan efektivitas pengujian perangkat lunak.

## ABSTRACT

This study aims to compare the performance of three test automation frameworks, namely Selenium, Cypress, and Playwright, in UI testing on a website platform. The method used is The Distance to the Ideal Alternative (DIA) to evaluate the frameworks performance based on the parameters Technical and Economic View, Testing Process View, and Quality View. The study began with a literature review to determine the testing parameters, followed by weighting the parameters through a survey of professional software testers and programmers. The frameworks were tested on the iconhub.io website, and the test results showed that Cypress is the best framework with the lowest weighted score, demonstrating superiority in speed and efficiency. This study provides guidance for developers in choosing the appropriate and efficient UI testing framework, contributing to the improvement of software testing quality and effectiveness.

## I. PENDAHULUAN

DALAM pengembangan sistem informasi, pengujian perangkat lunak penting untuk menghasilkan output berkualitas tinggi dan mendeteksi kesalahan sistem. Pengujian perangkat lunak merupakan bagian integral dari pembuatan sistem development [1]. Pengujian perangkat lunak, baik manual maupun otomatis, penting dalam proses pengembangan perangkat lunak. Pengujian otomatis meningkatkan keakuratan dan efisiensi, namun pemilihan alat yang tepat semakin rumit seiring dengan maraknya berbagai alat pengujian otomatis [2].

Menurut Nagabushanam dan Vijayasree [3], pengujian perangkat lunak membantu mengevaluasi kesesuaian desain dan arsitektur perangkat lunak dengan fungsionalitasnya. Pengujian manual, di mana penguji bertindak sebagai pengguna akhir, bisa memakan waktu dan tidak selalu efektif dalam mengidentifikasi semua cacat [4]. Sebaliknya, pengujian otomatis mempercepat proses pengujian dan memungkinkan cakupan pengujian yang lebih luas dalam waktu terbatas, yang pada gilirannya meningkatkan kualitas perangkat lunak [3].

Framework pengujian otomatis memudahkan pengembang mendeteksi bugs dan regresi kode program [5]. Terdapat beberapa test automation framework yang baik, menurut Badkar [5] terdapat 8 test automation framework terbaik yang digunakan oleh developer di seluruh dunia yaitu Selenium, Cypress, Playwright, WebDriverIO, TestCafe, NightwatchJS, Appium, dan Cucumber. Penelitian ini membandingkan tiga test automation framework terbaik. Selenium, Cypress, dan Playwright yang populer dan kompatibel dengan web browser utama, selain itu

popularitasnya diakui pada industri perangkat lunak dibuktikan pada *used by* di github pada framework cypress sebanyak 660 ribu pengguna, 175 ribu pengguna pada selenium, dan 46 ribu pengguna pada playwright. Serta banyaknya penelitian yang membahas dari ketiga framework tersebut dari hasil riset mereka [6] [7]. Uji coba akan dilakukan pada website iconhub.io, menggunakan parameter Technical and Economic View, Testing Process View, dan Quality View [2]. Metode The Distance to The Ideal Alternative (DIA) dipilih karena memberikan keunggulan dalam menentukan jarak alternatif ideal positif dan negatif menggunakan Manhattan Distance. Metode ini memungkinkan penetapan Positif Ideal Alternatif (PIA) yang memberikan hasil lebih akurat dibandingkan metode lain seperti AHP atau TOPSIS [8]. Dalam konteks perbandingan multi hirarki dan memiliki jenis atribut yang berbeda beda. Dalam konteks ini, pada penelitian ini penulis akan menggunakan multi parameter dengan sub parameter yang kompleks dan bertingkat yang mana penggunaan metode DIA memungkinkan hasil analisis yang komprehensif dengan mempertimbangkan berbagai aspek performa framework, seperti efisiensi waktu eksekusi, kemudahan setup, dan fleksibilitas scripting language [9][10].

Penelitian mengenai perbandingan performa berbagai alat dan framework pengujian telah banyak dilakukan sebelumnya. Abdulwareth dan Al-Shargabi [2] dalam penelitiannya "Toward a Multi-Criteria Framework for Selecting Software Testing Tools" menggunakan metode Analytic Hierarchy Process (AHP) dan TOPSIS untuk mengembangkan framework yang mempermudah pemilihan alat uji perangkat lunak berdasarkan taksonomi komprehensif dan peringkat yang akurat. Namun, penelitian Abdulwareth dan Al-Shargabi menggunakan taksonomi yang rumit dan metode AHP serta TOPSIS yang kompleks, sehingga sulit diimplementasikan secara praktis. Untuk mengatasi batasan tersebut, penelitian ini menyederhanakan taksonomi dengan fokus pada tiga parameter utama, yaitu Technical and Economic View, Testing Process View, dan Quality View. Metode The Distance to the Ideal Alternative (DIA) yang digunakan menawarkan pendekatan kuantitatif yang lebih sederhana dan terukur. Selain itu, implementasi praktis dilakukan dengan pengujian langsung pada website iconhub.io, bobot parameter ditentukan melalui survei empiris dari software tester dan programmer profesional, sehingga lebih realistis dan mudah diterapkan. Penelitian ini memberikan hasil yang langsung dapat digunakan oleh pengembang dalam memilih framework pengujian UI yang tepat, mengatasi batasan pada taksonomi yang rumit dan keterbatasan implementasi praktis dari penelitian sebelumnya.

Selanjutnya, Merina [11] dalam "A Comparative Analysis of Test Automation Frameworks Performance for Functional Testing in Android-Based Applications using the Distance to the Ideal Alternative Method" menggunakan metode Distance to The Ideal Alternative (DIA) untuk membandingkan kinerja Espresso, Calabash, dan Appium pada aplikasi Android, dengan hasil menunjukkan Calabash sebagai framework terbaik. Christy dan Siburian [12] membandingkan alat pengujian otomatis Selendroid, Calabash, dan UI Automator untuk aplikasi Android, dengan temuan bahwa Selendroid dan Calabash lebih cocok untuk aplikasi hybrid dan mobile web, sementara UI Automator lebih baik untuk aplikasi native. Berbeda dengan penelitian tersebut, penelitian ini fokus pada pengujian framework otomatisasi pada platform web, yang memiliki kompleksitas dan skala penggunaan teknologi yang lebih besar. Pengujian UI pada platform web melibatkan berbagai browser dan konfigurasi yang lebih bervariasi dibandingkan dengan aplikasi mobile, sehingga memerlukan pendekatan pengujian yang lebih robust dan fleksibel. Dengan demikian, penelitian ini tidak hanya mengukur kinerja framework seperti Selenium, Cypress, dan Playwright, tetapi juga memberikan panduan praktis yang relevan bagi pengembang web yang bekerja dalam lingkungan yang lebih kompleks dan dinamis dibandingkan pengujian aplikasi mobile.

Selain itu, Pelivani dan Cico [13] dalam "A comparative study of automation testing tools for web applications" mengevaluasi dan membandingkan Katalon Studio dan Selenium untuk pengujian aplikasi web, menyoroti kelebihan dan kekurangan masing-masing serta implikasi AI dalam pengujian otomatis. Mobaraya dan Ali [14] dalam "Technical Analysis of Selenium and Cypress as Functional Automation Framework for Modern Web Application" menemukan bahwa Cypress lebih unggul dalam waktu eksekusi dan efisiensi dibandingkan Selenium, dengan saran untuk pengujian lebih lanjut pada berbagai browser. Penelitian ini mengkaitkan hasil tersebut dengan draf penelitian yang menggunakan parameter kompleks untuk mengevaluasi kinerja framework otomatisasi pengujian UI pada platform web. Dengan menggunakan metode The Distance to the Ideal Alternative (DIA), penelitian ini menilai framework berdasarkan parameter yang lebih beragam dan terukur, seperti Technical and Economic View, Testing Process View, dan Quality View. Kompleksitas parameter ini memungkinkan evaluasi yang lebih komprehensif dan mendalam, memberikan panduan yang lebih akurat bagi pengembang dalam memilih framework terbaik untuk kebutuhan pengujian mereka. Hal ini mendukung pernyataan bahwa penelitian ini menawarkan analisis yang lebih holistik dibandingkan studi sebelumnya yang mungkin hanya fokus pada beberapa aspek saja, tanpa mempertimbangkan parameter yang lebih luas dan kompleks seperti yang dilakukan dalam penelitian ini.

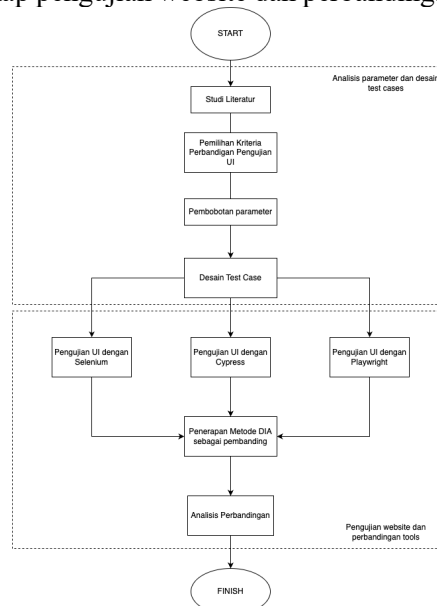
Dalam analisis penelitian ini, beberapa framework otomatisasi pengujian UI telah dibandingkan, masing-masing dengan kelebihan dan kekurangannya. Selenium, meskipun fleksibel dan mendukung berbagai bahasa

pemrograman, memiliki kelemahan dalam waktu eksekusi yang lebih lama dan kompleksitas setup yang tinggi. Cypress unggul dalam kecepatan eksekusi dan kemudahan penggunaan, tetapi terbatas pada bahasa JavaScript dan tidak mendukung banyak browser secara native seperti Selenium. Playwright menawarkan keseimbangan antara kemudahan penggunaan dan dukungan multi-bahasa, tetapi bisa lebih rumit dibandingkan Cypress dalam beberapa aspek setup dan dokumentasi. Oleh karena itu, penelitian ini menjadi penting untuk memberikan panduan yang lebih komprehensif bagi pengembang dalam memilih framework yang sesuai dengan kebutuhan spesifik mereka, mempertimbangkan berbagai parameter kompleks yang dievaluasi menggunakan metode The Distance to the Ideal Alternative (DIA).

Mengacu pada berbagai penelitian tersebut, penelitian ini bertujuan untuk melakukan analisis perbandingan kinerja framework otomatis pengujian pada UI testing dengan menggunakan metode The Distance to the Ideal Alternative (DIA). Fokus utama penelitian ini adalah untuk mengevaluasi dan membandingkan kinerja framework pengujian otomatis yang ada dan menentukan framework terbaik yang dapat digunakan dalam pengujian UI. Penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam pemilihan framework pengujian yang tepat dan efisien untuk berbagai jenis aplikasi serta penelitian ini berkontribusi dalam bidang pengujian otomatis dengan menyediakan panduan komprehensif bagi pengembang perangkat lunak dalam memilih framework pengujian yang sesuai. Dalam penelitian ini mempertimbangkan berbagai parameter kompleks yang mana bisa menjadi panduan bagi pengembang.

## II. METODOLOGI

Bagian ini menjelaskan metodologi penelitian yang digunakan dalam studi ini, mengikuti alur proses pada gambar 1, yang terdiri dari dua tahap besar. Tahap pertama meliputi analisis parameter dan desain test cases, melalui studi literatur untuk mengumpulkan informasi terkait parameter dan desain test case yang relevan dengan pengujian UI. Kemudian, dilakukan pemilihan dan pembobotan parameter untuk perbandingan pengujian UI, serta perancangan test case berdasarkan parameter tersebut. Bab ini akan membahas secara detail tahap-tahap tersebut hingga desain test case, sementara tahap pengujian website dan perbandingan tools akan dibahas di bab berikutnya.



Gambar. 1. Alur proses

### A. Studi Literatur

Dalam melakukan penelitian ini, penulis membutuhkan bahan, referensi, informasi, dan data untuk mendukung pembahasan penelitian yang sedang dilakukan. Dengan mengumpulkan data dan informasi yang relevan dengan topik penelitian, diharapkan dapat memperkuat kevalidan dan kebenaran hasil penelitian. Melalui studi literatur, yaitu dengan membaca dan mempelajari penelitian serupa yang telah dilakukan sebelumnya, diharapkan dapat menjadi sumber referensi, informasi, dan dasar teoritis bagi penulis. Literatur yang diacu meliputi jurnal-jurnal baik nasional maupun internasional.

### B. Pemilihan Parameter Perbandingan Pengujian

Penulis memprioritaskan pemilihan parameter dalam penelitian ini. Parameter yang diambil dari studi literatur akan memperkuat keabsahan perbandingan kinerja penelitian. Oleh karena itu, penulis merujuk penelitian

sebelumnya dan fokus pada pemilihan parameter pengujian, dengan menyajikan tiga parameter utama yang masing-masing memiliki beberapa subparameter.

Penulis menentukan parameter dan sub parameter berdasarkan studi literatur yang ditulis oleh Abdulwareth dan Al-Shargabi [2], Pelivani [13], Melia dan Putra [15], serta Merina [11]. Menyebutkan bahwa parameter atau parameter dalam memilih test automation framework ada tiga. Parameter pertama adalah Technical Economic View yaitu kemudahan untuk penulis mengklasifikasikan automation testing framework berdasarkan aspek teknis dan ekonomi. Terdapat juga Sub Parameter pada Technical economic view yaitu:

1) *Scripting Language*

Semakin banyak pilihan Bahasa pemrograman yang bisa digunakan dalam satu framework, semakin baik fleksibilitas framework tersebut.

2) *Cost*

Biaya yang dikeluarkan untuk melakukan pengujian atau menggunakan framework tersebut, apakah gratis ataukah berbayar.

3) *Product Support*

Mempunyai komunitas yang besar dan aktif dengan dokumentasi dan sumber daya pendukung yang baik.

Parameter kedua adalah Testing Process View yaitu parameter saat penulis mengklasifikasikan automation testing framework berdasarkan proses yang dialami dan seperti apa proses yang dirasakan penguji dan penulis saat mencoba alat tersebut yang mencakup hasil eksekusi, seberapa sulit untuk melakukan setup serta proses pembuatan code nya. Terdapat juga Sub Parameter pada Testing Process View yaitu:

1) *Easy of Setup*

Kemudahan saat proses instalasi atau konfigurasi awal, dan juga setup pada framework tersebut, tidak memerlukan banyak tools untuk terintegrasi.

2) *Script Creation Time*

Seberapa banyak memakan waktu saat membuat script atau code pengujian pada framework tersebut.

3) *Dokumentasi hasil report*

Apakah terdapat dokumentasi dari hasil report pengujian pada framework tersebut ataukah sudah built in atau perlu tambahan library tools lagi.

4) *Waktu hasil eksekusi*

Seberapa banyak memakan waktu saat menjalankan pengujian dari script sesuai test case yang sudah dibuat.

5) *Inspection Element*

Apakah ada kemudahan untuk mencari atau menemukan element (locator) pada objek yang akan diuji pada framework tersebut.

Parameter ketiga adalah Quality View yaitu salah satu aspek penting dalam pengujian pada automation testing framework. Berdasarkan model kualitas perangkat lunak, terdapat berbagai karakteristik yang umumnya diidentifikasi, antara lain fungsionalitas, keandalan, keamanan, maintainabilitas, kegunaan, portabilitas, dan efisiensi. Terdapat juga sub parameter pada Quality View yaitu:

1) *Usability*

Mencakup Understandability, Learnability, Operability, Attractiveness, Usability compliance pada framework tersebut atau secara mudahnya adalah System Usability Scale (SUS).

2) *Portability*

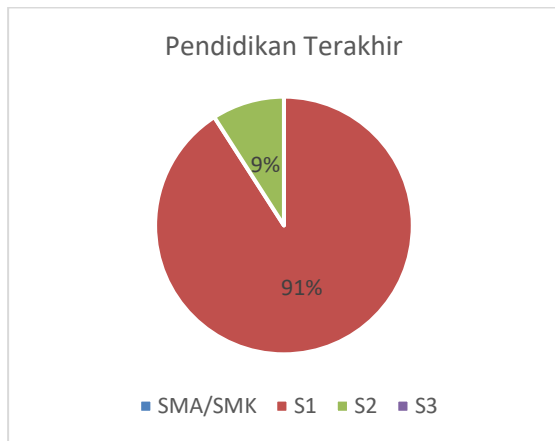
Apakah framework bisa cross platform atau tidak.

### C. Pembobotan Parameter dan Nilai

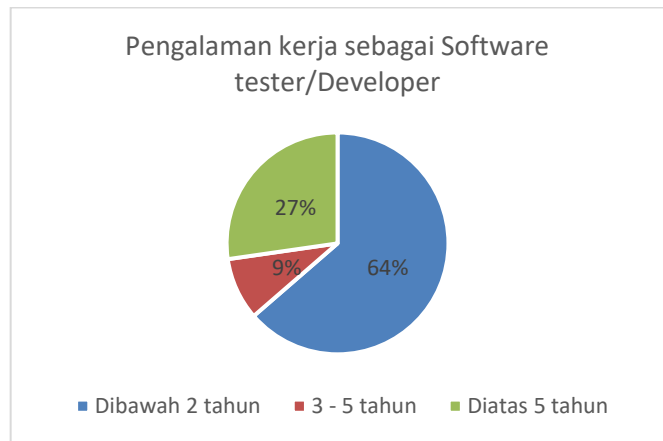
Penulis melakukan pembobotan parameter untuk membandingkan efektivitas berbagai framework menggunakan metode The Distance to The Ideal Alternative (DIA). Bobot ditentukan berdasarkan pengaruhnya dalam mengukur kinerja framework otomasi pengujian, menggunakan prinsip-prinsip Multiple Attribute Decision Making (MADM). Bobot yang lebih tinggi menunjukkan pengaruh yang lebih besar terhadap hasil akhir, dengan total bobot sama dengan 1 sesuai ketentuan metode DIA. Proses pembobotan dilakukan secara empiris melalui distribusi kuesioner kepada software tester dan programmer profesional untuk memahami prioritas dan efektivitas berbagai parameter dalam framework pengujian otomatis.

1) *Demografi Responden*

Data demografi dari responden yang terlibat dalam survei ini memberikan konteks yang berharga mengenai keberagaman latar belakang dan pengalaman mereka. Distribusi umur responden bervariasi, dengan sebagian besar berada di kisaran 23 hingga 29 tahun, mencerminkan keaktifan generasi muda dalam industri pengembangan perangkat lunak.

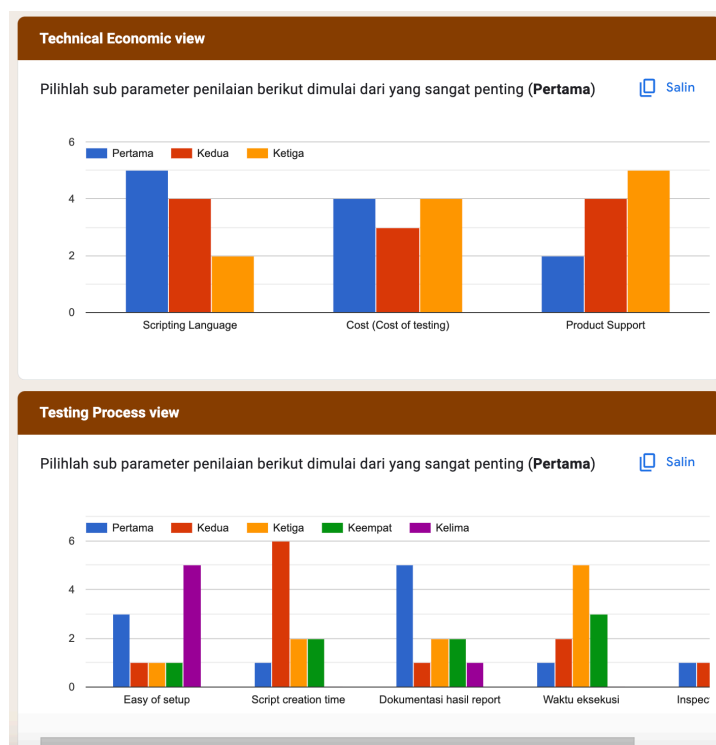


Gambar. 3. Data pendidikan terakhir responden



Gambar. 4. Data pengalaman kerja responden

Sebagian besar responden (91%) memiliki gelar sarjana (S1), dan adapula responden yang memiliki gelar magister (S2) sebanyak (1%). Responden menunjukkan variasi pengalaman yang luas, mulai dari mereka yang memiliki pengalaman di bawah dua tahun hingga yang memiliki pengalaman lebih dari lima tahun. Variasi ini memungkinkan penelitian untuk memperoleh perspektif yang kaya dan beragam tentang penggunaan dan efektivitas framework pengujian, mencerminkan berbagai tingkat keahlian dan pendekatan dalam pengujian perangkat lunak. Dengan pengalaman kerja yang berbeda beda akan mempengaruhi hasil variasi perspektif pengembang dalam menentukan tingkat kepentingan suatu parameter, sebagai contoh developer dengan pengalaman yang lebih tinggi akan melihat aspek yang berbeda atau seimbang daripada developer yang lain, pengalaman yang lebih rendah akan melihat aspek yang berbeda pula cara pandanganya seperti *cost* yang dikeluarkan dan *setup* yang lebih mudah.



Gambar. 5. Contoh pertanyaan pada survey yang dilakukan serta hasilnya.

Pada gambar 5, tertampil contoh dari pertanyaan yang penulis ajukan pada kuesioner. Tipe jawaban yang penulis ajukan berupa perangkingan sesuai jumlah sub-parameter, lalu data tersebut diolah agar menjadi sebuah bobot nilai parameter dihitung dari jumlah ranking pertama terbanyak dari setiap parameter dan sub parameter, seperti pada persamaan 1 berikut ini.

$$Pembobotan = \frac{\text{Responden yang memilih sub parameter}}{\text{Jumlah responden}} \quad (1)$$

## 2) Bobot Nilai Parameter dan Sub Parameter

TABEL 1  
 BOBOT NILAI PARAMETER DAN SUB PARAMETER

Parameter	Bobot	Sub Parameter	Bobot Sub Parameter
Technical Economic View	0.27	Scripting Language	0.46
		Cost	0.36
		Product Support	0.18
Testing Process View	0.64	Easy of Setup	0.27
		Scripting Creation Time	0.09
		Dokumentasi Hasil Report	0.46
		Waktu Hasil Eksekusi	0.09
		Inspection Element	0.09
Quality View	0.09	Usability	0.91
		Portability	0.09

Dari hasil kuesioner yang telah disebar, selanjutnya dihitung nilai bobot untuk sub parameter dan parameter. Untuk menentukan bobot yakni dengan menghitung jumlah responden yang memilih sub parameter sebagai aspek paling penting, lalu dibagi dengan jumlah responden keseluruhan. Pada parameter Technical Economic View memiliki bobot 0.27 dengan sub parameter Scripting Language sebesar 0.46, Cost dengan bobot 0.36 dan Product Support dengan bobot 0.18. Lalu pada parameter Testing Process View memiliki bobot 0.64 dengan sub parameter Easy of Setup sebesar 0.27, Scripting Creation Time sebesar 0.09, Dokumentasi hasil report sebesar 0.46, dan Waktu hasil eksekusi sebesar 0.09 serta Inspection Element memiliki bobot sebesar 0.09. Lalu pada parameter terakhir yakni Quality View memiliki bobot sebesar 0.09 dengan sub parameter Usability yang memiliki bobot sebesar 0.91 dan Portability sebesar 0.09, seperti pada tabel 1.

## 3) Metode The Distance to The Ideal Alternative (DIA)

Metode DIA merupakan salah satu teknik yang termasuk dalam MADM dan baru dikembangkan oleh beberapa peneliti. DIA terdiri dari penentuan nilai ideal positif dan negatif untuk setiap atribut, yang sebanding dengan metode TOPSIS. Perbedaannya terletak pada penggunaan manhattan *distance* untuk menentukan jarak, menentukan Positif Ideal Alternatif (PIA) dengan nilai minimal + Dj dan maksimal - Dj, serta formula urutan nilai pada Ri untuk menentukan pengklasifikasi alternatif [16]. Langkah-langkah penyelesaian masalah dengan menggunakan metode DIA adalah sebagai berikut [8].

### 1) Menentukan matriks keputusan

Menggunakan bobot yang telah ditetapkan

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} \quad (2)$$

### 2) Normalisasi matriks keputusan

Matriks normalisasi diperoleh dari setiap elemen pada matriks keputusan dibagi dengan akar dari jumlah setiap elemen kolom matriks keputusan yang dikuadratkan. Persamaan ini dapat dilihat sebagai berikut.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad (3)$$

3) Pembobotan pada matriks yang telah dinormalisasi

Setelah proses normalisasi matriks keputusan, kemudian menentukan matriks pembobotan. Dimana setiap elemen matriks pembobotan diperoleh dari hasil setiap elemen bobot dikali dengan setiap elemen pada matriks normalisasi. Persamaan ini dapat dilihat sebagai berikut.



(4)

4) Menentukan solusi ideal positif dan ideal negatif

Solusi ideal positif dinotasikan dengan  $A^{+}$  dan solusi ideal negatif dinotasikan dengan  $A^{-}$ . Persamaan ini dapat dilihat sebagai berikut.



(5)



5) Hitung jarak manhattan untuk atribut positif dan negatif

Dimana,  $a_i^{+}$  adalah nilai dari setiap solusi ideal positif. Dimana,  $a_i^{-}$  adalah nilai dari setiap solusi ideal negatif.



(6)



6) Menentukan Positif Ideal Alternatif (PIA)

PIA minimal dinotasikan dengan  $\min(D_i^{+})$ , nilai ini diperoleh dari pencarian nilai terkecil pada setiap baris  $D_i^{+}$ . Untuk PIA maksimal dinotasikan dengan  $\max(D_i^{-})$ , dan nilai ini diperoleh dari pencarian nilai terbesar pada setiap baris  $D_i^{-}$ . Persamaan ini dapat dilihat sebagai berikut.



(7)

7) Melakukan Identifikasi Peringkat

Peringkat dapat ditentukan dengan membandingkan nilai identifikasi peringkat ( $R_i$ ) yang diperoleh dari persamaan sebagai berikut



(8)

D. Desain Test Case

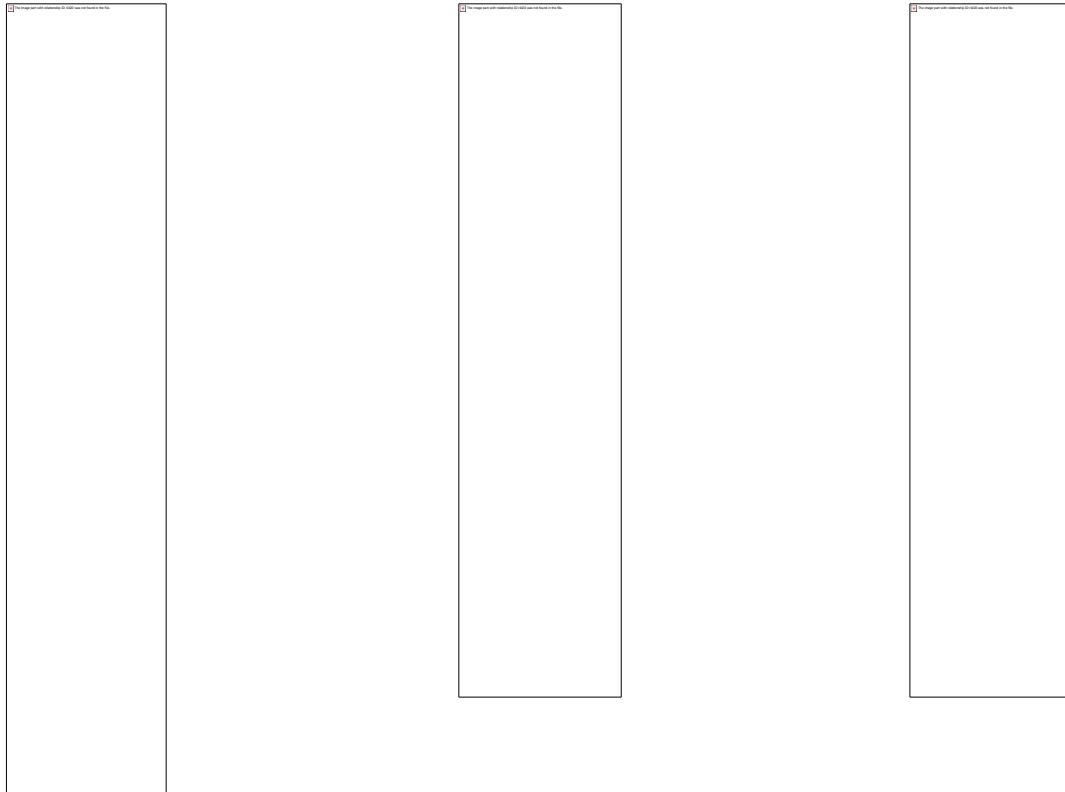
Pada tahap ini, penulis mendesain test case yang akan diuji pada masing-masing fitur, test case yang penulis desain berupa test case positif maupun test case negatif. Test case digunakan untuk menunjukkan bahwa suatu sistem perangkat lunak bekerja sesuai dengan yang diharapkan (pengujian positif) dan tidak melakukan hal-hal yang tidak diharapkan (pengujian negatif) [17], [18]. Test case yang penulis desain antara lain Menampilkan halaman utama, membuat akun baru, masuk ke akun, gagal masuk ke akun, memilih *icon style*, memilih *icon size*, memilih *icon stroke size*, memilih *icon*, mengunduh *icon*, dan memilih *icon* berdasarkan kategori.

### III. HASIL DAN PEMBAHASAN

#### A. Automated UI Testing

Penulis melakukan pengujian UI menggunakan tiga *framework* utama: Selenium, Cypress, dan Playwright. Untuk Selenium, penulis melakukan setup dengan bahasa Java karena meskipun Selenium dapat dijalankan dengan berbagai bahasa seperti Java, C#, Ruby, Python, dan JavaScript, pengalaman sebelumnya dengan Java dan keberhasilan dalam mengatasi *error* membuat pilihan ini optimal. Cypress digunakan dengan setup *framework* dan

bahasa JavaScript, karena Cypress hanya mendukung bahasa ini, yang juga selaras dengan kemampuan Cypress untuk pengujian UI yang cepat dan efisien. Untuk Playwright, penulis menggunakan bahasa TypeScript setelah mengalami hambatan berupa *error* pada pengujian dengan bahasa lain, dan berhasil melakukan pengujian dengan TypeScript. Penulis akan melakukan pengujian UI untuk setiap *automation testing framework* dengan ketentuan dan aturan masing-masing *framework*. Berikut adalah langkah-langkah dari masing masing *framework* pada Gambar 5.



Gambar. 6. Alur Pengujian pada tiap Framework

### B. Penerapan Metode DIA Sebagai Pembandingan

Setelah menyelesaikan pengujian UI fungsional otomatis pada ketiga test automation framework yang dipilih, langkah berikutnya adalah mengumpulkan data, menganalisis hasil pengujian, dan menerapkan Metode The Distance to The Ideal Alternative (DIA). Metode DIA digunakan untuk mengevaluasi dan membandingkan kinerja masing-masing framework berdasarkan parameter yang telah ditentukan, sehingga dapat menentukan framework yang paling optimal dari segi kinerja untuk pengujian UI.

#### 1) Pengumpulan Data

Berikut pada tabel 2 merupakan data hasil pengujian otomatis yang telah dilakukan oleh penulis pada ketiga test automation framework.

TABEL 2  
 DATA HASIL PENGUJIAN DARI SETIAP FRAMEWORK

Nama Framework	Waktu Hasil Eksekusi	Script Creation Time	Easy of Setup	Cost	Scripting Language
Selenium	28 detik	7 jam	Sulit	Gratis	5 bahasa
Cypress	23 detik	2 jam	Mudah	Gratis	1 bahasa
Playwright	25 detik	4 jam	Mudah	Gratis	4 bahasa



2) *Menentukan matriks keputusan*

Berdasarkan hasil data yang sudah dikumpulkan lalu data tersebut di normalisasi sehingga menghasilkan seperti pada tabel 3, tabel 4 dan tabel 5.

TABEL 3  
 MATRIKS KEPUTUSAN PARAMETER TECHNICAL ECONOMIC VIEW

Alternatif	Bobot Sub Parameter		
	Scripting Language	Cost	Product Support
Selenium	1	1	1
Cypress	0	1	1
Playwright	0.75	1	1

TABEL 4  
 MATRIKS KEPUTUSAN PARAMETER TESTING PROCESS VIEW

Alternatif	Bobot Sub Parameter				
	Easy of Setup	Script Creation Time	Product Support	Dokumentasi hasil report	Inspection Element
Selenium	0	0	0	0	0
Cypress	1	1	1	1	1
Playwright	1	0.5	1	0.5	1

TABEL 5  
 MATRIKS KEPUTUSAN PARAMETER QUALITY VIEW

Alternatif	Bobot Sub Parameter	
	Usability	Portability
Selenium	0.25	1
Cypress	1	1
Playwright	0.75	1

3) *Normalisasi matriks keputusan*

Berdasarkan hasil pada tabel 3, tabel 4 dan tabel 5 maka dihasilkan normalisasi matriks keputusan seperti pada tabel 6, tabel 7 dan tabel 8 berikut ini.

TABEL 6  
 NORMALISASI MATRIKS KEPUTUSAN PARAMETER TECHNICAL ECONOMIC VIEW

Alternatif	Bobot Sub Parameter		
	Scripting Language	Cost	Product Support
Selenium	0.80	0.58	0.58
Cypress	0	0.58	0.58
Playwright	0.60	0.58	0.58

TABEL 7  
 NORMALISASI MATRIKS KEPUTUSAN PARAMETER TESTING PROCESS VIEW

Alternatif	Bobot Sub Parameter				
	Easy of Setup	Script Creation Time	Product Support	Dokumentasi hasil report	Inspection Element
Selenium	0	0	0	0	0
Cypress	0.71	0.89	0.71	0.89	0.71
Playwright	0.71	0.45	0.71	0.45	0.71

TABEL 8  
 NORMALISASI MATRIKS KEPUTUSAN PARAMETER QUALITY VIEW

Alternatif	Bobot Sub Parameter	
	Usability	Portability
Selenium	0.20	0.58
Cypress	0.78	0.58
Playwright	0.59	0.58

4) Pembobotan pada matriks yang telah di normalisasi

Berdasarkan normalisasi matriks keputusan maka nilai tersebut dimasukkan kedalam pembobotan pada matriks yang telah di normalisasi sebagai berikut ini.

TABEL 9  
 PEMBOBOTAN NORMALISASI MATRIKS KEPUTUSAN PARAMETER

Alternatif	Technical Economic View			Testing Process View				Quality View		
	Scripting Language	Cost	Product Support	Easy of Setup	Script Creation Time	Dokumen-tasi hasil report	Waktu hasil eksekusi	Inspection Element	Usability	Portabil-ity
Selenium	0.368	0.209	0.104	0	0	0	0	0	0.182	0.052
Cypress	0	0.209	0.104	0.192	0.81	0.327	0.081	0.064	0.710	0.052
Playwright	0.276	0.209	0.104	0.192	0.041	0.037	0.041	0.064	0.537	0.052

5) Menentukan solusi ideal positif dan ideal negatif

Pada tahap ini, hasil dari pembobotan pada matriks yang telah di normalisasi, selanjutnya menentukan solusi ideal positif dan negatif berdasarkan pada tabel sebelumnya yakni tabel 9. Berikut merupakan contoh pada sub parameter technical economic view pada tabel 10 dan 11.

TABEL 10  
 HASIL IDEAL POSITIF TECHNICAL ECONOMIC VIEW

Alternatif Positif	Sub Parameter	$A^+ = \max V_{ij} = [V_1^+, V_2^+ \dots V_n^+]$
$V_1^+$	Scripting Language	0.368
$V_2^+$	Cost	0.209
$V_3^+$	Product Support	0.104

TABEL 11  
 HASIL IDEAL NEGATIF TECHNICAL ECONOMIC VIEW

Alternatif Positif	Sub Parameter	$A^+ = \max V_{ij} = [V_1^+, V_2^+ \dots V_n^+]$
$V_1^+$	Scripting Language	0
$V_2^+$	Cost	0.209
$V_3^+$	Product Support	0.104

6) Hitung Jarak Manhattan untuk attribute positif dan negatif

Berikut merupakan contoh pada sub parameter technical economic view pada tabel 12 dan 13.

TABEL 12  
 HASIL JARAK MANHATTAN MAKSIMUM TECHNICAL ECONOMIC VIEW

Alternatif	Manhattan Maximum	Nilai
Selenium	$D_1^+$	0
Cypress	$D_2^+$	0.368
Playwright	$D_3^+$	0.092

TABEL 13  
 HASIL JARAK MANHATTAN MINIMUM TECHNICAL ECONOMIC VIEW

Alternatif	Manhattan minimum	Nilai
Selenium	$D_1^-$	0.368
Cypress	$D_2^-$	0
Playwright	$D_3^-$	0.276

7) *Menentukan Positif Ideal Alternatif (PIA)*

Untuk parameter Technical Economic View, nilai PIA adalah :

$$\mathbf{PIA = 0 ; 0.368}$$

Lalu untuk parameter Testing Process View, nilai PIA adalah :

$$\mathbf{PIA = 0 ; 0.745}$$

Sedangkan untuk parameter Quality View, nilai PIA adalah :

$$\mathbf{PIA = 0 ; 0.528}$$

8) *Melakukan Identifikasi Peringkat*

Nilai  $R_i$  yang minimum mengindikasikan bahwa alternatif tersebut lebih terpilih sebagai peringkat yang lebih utama. Dari persamaan yang sudah dijabarkan, penulis akan melanjutkan perhitungan mulai dari persamaan 5 sampai selesai berdasarkan pembobotan yang sudah ditetapkan dari setiap sub parameter dan parameter yakni, pada parameter Technical Economic View memiliki bobot 0.27 dengan sub parameter Scripting Language sebesar 0.46, Cost dengan bobot 0.36 dan Product Support dengan bobot 0.18. Lalu pada parameter Testing Process View memiliki bobot 0.64 dengan sub parameter Easy of Setup sebesar 0.27, Scripting Creation Time sebesar 0.09, Dokumentasi hasil report sebesar 0.46, dan Waktu hasil eksekusi sebesar 0.09 serta Inspection Element memiliki bobot sebesar 0.09. Lalu pada parameter terakhir yakni Quality View memiliki bobot sebesar 0.09 dengan sub parameter Usability yang memiliki bobot sebesar 0.91 dan Portability sebesar 0.09.

Sehingga dihasilkan pada persamaan terakhir yaitu identifikasi peringkat berdasarkan parameter yakni pada Tabel 14, Tabel 15, dan Tabel 16.

TABEL 14  
 HASIL PERINGKAT PARAMETER TECHNICAL ECONOMIC VIEW

Framework	$R_i$	Peringkat
Selenium	0	1
Cypress	0.520	3
Playwright	0.130	2

TABEL 15  
 HASIL PERINGKAT PARAMETER TESTING PROCESS VIEW

Framework	$R_i$	Peringkat
Selenium	1.053	3
Cypress	0	1
Playwright	0.113	2

TABEL 16  
 HASIL PERINGKAT PARAMETER QUALITY VIEW

Framework	$R_i$	Peringkat
Selenium	0.747	3
Cypress	0	1
Playwright	0.245	2

### C. Analisis Perbandingan

Pada tahap ini, penulis akan melakukan proses analisis perbandingan. Setelah mendapat hasil nilai  $R_i$  dari ketiga parameter yang telah ditetapkan, selanjutnya penulis akan melakukan analisis kembali untuk membandingkan nilai dari ketiga parameter. Menurut Ayan dan Ezell [19] menggunakan perhitungan Weighted Scoring Model ini mempermudah dan mengkuantifikasi proses pengambilan keputusan yang kompleks dengan mempertimbangkan berbagai faktor penting, terutama untuk mendapatkan hasil akhir yang lebih valid agar perbandingan kinerja test automation framework bisa terukur secara keseluruhan. Berikut merupakan tahap tahap pada Weighted Scoring Model.

#### 1) Peringkat untuk setiap parameter

Berdasarkan hasil peringkat yang penulis peroleh, berikut merupakan tabel peringkat beserta bobot ( $w$ ) pada Tabel 17 untuk setiap *framework*.

TABEL 17  
 PERINGKAT DARI SETIAP PARAMETER DAN BOBOT

Parameter	Selenium	CYPRESS	Playwright	Bobot
Technical Economic View	1	3	2	0.27
Testing Process View	3	1	2	0.64
Quality View	2	1	2	0.09

#### 2) Menghitung Skor Tertimbang (Weighted Scoring)

Berikut merupakan contoh perhitungan Skor tertimbang pada *framework* Selenium.

$$\text{Skor} = (1 \times 0.27) + (3 \times 0.64) + (3 \times 0.09) = 0.27 + 1.92 + 0.27 = 2.46$$

Untuk melihat hasil lengkap dari perhitungan pada setiap *framework*, data tersebut disajikan dalam Tabel 18.

TABEL 18  
 HASIL PERHITUNGAN SKOR TERTIMBANG

Framework	Total Skor	Peringkat
Selenium	2.46	3
Cypress	1.54	1
Playwright	2.00	2

### 3) Alternatif Terbaik

Berdasarkan perhitungan di atas, alternatif terbaik adalah Cypress dengan total skor terkecil (1.54). Cypress menempati peringkat pertama, diikuti oleh Playwright di peringkat kedua, dan Selenium di peringkat ketiga.

Dalam penelitian ini, ketiga framework diuji berdasarkan parameter Technical and Economic View, Testing Process View, dan Quality View. Dari sisi Technical and Economic View, Selenium menunjukkan fleksibilitas tinggi dalam scripting language dan dukungan browser, tetapi memiliki setup yang kompleks dan waktu eksekusi yang lama. Cypress unggul dalam kemudahan setup dan kecepatan eksekusi, namun terbatas pada JavaScript dan dukungan browser yang kurang lengkap. Playwright menawarkan keseimbangan dengan dukungan multi-bahasa dan browser, namun setup yang rumit menjadi tantangan. Dari sisi Testing Process View, Selenium memerlukan waktu lebih lama dalam pembuatan skrip dan dokumentasi hasil eksekusinya tidak seintuitif Cypress. Cypress menawarkan proses pembuatan skrip yang cepat dan dokumentasi hasil yang otomatis dan lengkap, tetapi terbatas pada beberapa browser. Playwright cukup efisien dalam pembuatan skrip dengan dukungan lintas platform yang baik, namun dokumentasi hasil bisa lebih ditingkatkan. Dari sisi Quality View, Selenium memiliki usability dan portability yang baik, tetapi reliabilitas dalam pengujian elemen dinamis masih menjadi masalah. Cypress sangat user-friendly dan portable, namun terbatas pada JavaScript dan beberapa browser. Playwright memiliki usability yang baik dengan dukungan multi-platform, namun setup bisa memerlukan waktu lebih lama dan troubleshooting yang lebih intensif. Hasil pengujian menunjukkan bahwa Cypress unggul dalam hal kecepatan dan efisiensi, namun memiliki keterbatasan dalam dukungan browser. Selenium, meskipun lebih fleksibel, kalah dalam kecepatan dan kemudahan setup. Playwright menawarkan keseimbangan yang baik tetapi memerlukan setup yang lebih kompleks. Penelitian ini memberikan panduan komprehensif bagi pengembang untuk memilih framework pengujian UI yang paling sesuai dengan kebutuhan spesifik mereka, berdasarkan analisis mendalam dari berbagai parameter yang dievaluasi.

Hasil pengujian ini memberikan panduan yang komprehensif dan terstruktur bagi pengembang perangkat lunak dalam memilih framework otomatisasi pengujian UI yang paling sesuai dengan kebutuhan spesifik mereka. Dengan menganalisis framework berdasarkan parameter yang telah ditentukan, pembaca dapat memahami kelebihan dan kekurangan masing-masing framework (Selenium, Cypress, dan Playwright) dalam konteks praktis. Pembaca dapat menggunakan informasi ini untuk membandingkan framework berdasarkan poin bobot yang telah ditetapkan. Jika pembaca sepakat dengan hasil pembobotan yang digunakan dalam penelitian ini, mereka dapat langsung menerapkan rekomendasi framework yang dianggap terbaik. Namun, jika mereka tidak sepakat dengan bobot yang diberikan, mereka memiliki fleksibilitas untuk menyesuaikan bobot dan taksonomi sesuai dengan kebutuhan dan prioritas mereka sendiri. Dengan demikian, penelitian ini tidak hanya memberikan hasil spesifik tetapi juga menyediakan kerangka kerja yang dapat disesuaikan untuk membantu pembaca membuat keputusan yang lebih tepat dan informatif dalam pemilihan framework pengujian UI.

Hasil penelitian ini didukung oleh studi sebelumnya seperti penelitian oleh Mobaraya dan Ali [14] yang menemukan keunggulan Cypress dalam waktu eksekusi dan efisiensi dibandingkan selenium. Penelitian ini menunjukkan bahwa penggunaan Cypress dalam pengujian otomatisasi memiliki efisiensi yang lebih tinggi dalam eksekusi waktu dibandingkan Selenium WebDriver. Dalam sebuah studi oleh Hasan [20] yang mana studi ini menganalisis tiga alat pengujian otomatis untuk aplikasi satu halaman berbasis React, yaitu Jest, Enzyme, dan Cypress. Hasil menunjukkan bahwa setiap alat memiliki fitur dan keterbatasan masing-masing, dengan Cypress menunjukkan efisiensi yang signifikan.

## IV. KESIMPULAN

Penelitian ini bertujuan untuk membandingkan kinerja tiga framework otomatisasi pengujian, yaitu Selenium, Cypress, dan Playwright, dalam konteks pengujian UI (User Interface) atau functional test pada platform website. Penelitian ini dilakukan untuk menjawab rumusan masalah yang telah ditentukan, yaitu merancang test case untuk fitur-fitur pada website dan membandingkan hasil pengujian UI dari masing-masing framework berdasarkan parameter Technical and Economic View, Testing Process View, dan Quality View dengan metode Distance to the Ideal Alternative (DIA). Dari hasil penelitian dan analisis yang telah dilakukan, dapat disimpulkan bahwa, framework Terbaik, Berdasarkan perhitungan skor tertimbang, Cypress terbukti menjadi framework terbaik untuk pengujian UI pada platform website dengan skor tertimbang terendah. Pengujian UI, pengujian yang dilakukan pada 10 test case regresi menunjukkan variasi kinerja yang signifikan antara ketiga framework, dengan Cypress menunjukkan kinerja terbaik dalam hal kecepatan dan efisiensi. Parameter Perbandingan semua parameter yang digunakan dalam perbandingan (Technical and Economic View, Testing Process View, dan Quality View) memberikan kontribusi yang signifikan dalam menentukan framework terbaik. Kesimpulan ini menunjukkan bahwa penelitian yang dilakukan telah berhasil menjawab rumusan masalah dan mencapai tujuan yang telah ditetapkan, yaitu menentukan framework otomatisasi pengujian terbaik berdasarkan kinerja UI

## DAFTAR PUSTAKA

- [1] B. D. Saputra and A. Stefanie, "Automation Testing Api, Android, dan Website Menggunakan Serenity Bdd Pada Software Sistem Manajemen Rumah Sakit," *Jurnal Ilmiah Wahana Pendidikan, Mei*, vol. 2023, no. 10, pp. 114–126, 2023, doi: 10.5281/zenodo.7983405.
- [2] A. J. Abdulwareth and A. A. Al-Shargabi, "Toward a Multi-Criteria Framework for Selecting Software Testing Tools," *IEEE Access*, vol. 9, pp. 158872–158891, 2021, doi: 10.1109/ACCESS.2021.3128071.
- [3] D. Nagabushanam, S. S. D. Vijayasree, N. Sai Roopa, and A. Arun, "A Review on the Process of Automated Software Testing," Oct. 2022. doi: 10.48550/arXiv.2209.03069.
- [4] H. V. Gamido and M. V. Gamido, "Comparative review of the features of automated software testing tools," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 5, pp. 4473–4478, Oct. 2019, doi: 10.11591/ijece.v9i5.pp4473-4478.
- [5] Badkar A, "Best Test Automation Frameworks (Updated 2023)," <https://www.browserstack.com/guide/best-test-automation-frameworks>.
- [6] M. Psujek, A. Radzik, and G. Koziel, "Comparative analysis of solutions used in Automated Testing of Internet Applications," *Journal of Computer Sciences Institute*, vol. 18, pp. 7–14, Mar. 2021, doi: 10.35784/jesi.2373.
- [7] D. Zhyhulin, K. Kasian, and M. Kasian, "Combined method of prioritization and automation of software regression testing," in *2022 IEEE 16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, IEEE, Feb. 2022, pp. 751–755. doi: 10.1109/TCSET55632.2022.9767034.
- [8] P. N. Tran and N. Boukhatem, "The Distance to the Ideal Alternative (DiA) Algorithm for Interface Selection in Heterogeneous Wireless Networks," in *Proceedings of the 6th ACM International Symposium on Mobility Management and Wireless Access*, in MobiWac '08. New York, NY, USA: Association for Computing Machinery, 2008, pp. 61–68. doi: 10.1145/1454659.1454671.
- [9] U. Sa'adah *et al.*, "Automatic Testing Framework Based On Serenity And Jenkins Automated Build," *Jurnal Ilmiah Teknologi Informasi*, vol. 19, no. 2, pp. 102–110, 2021, doi: 10.12962/j24068535.v19i2.a1017.
- [10] Raihansyah, Yuyun Dwi Lestari, and Yessi Fitri Anissa Lubis, "Sistem Pendukung Keputusan Pemilihan Tempat Kegiatan Olahraga di Medan Dengan Metode The Distance To The Ideal Alternative (DIA)," *Jurnal Komputer Teknologi Informasi Sistem Komputer*, vol. 1, no. 2, pp. 1–9, Sep. 2022.
- [11] C. Merina, N. Anggraini, S. H. Afrizal, and N. Hakiem, "A Comparative Analysis of Test Automation Frameworks Performance for Functional Testing in Android-Based Applications using the Distance to the Ideal Alternative Method," 2018.
- [12] A. C. Barus and L. Siburian, "STUDI PERBANDINGAN ALAT PENGUJIAN OTOMATIS UNTUK APLIKASI ANDROID," vol. 6, no. 6, pp. 645–654, 2019, doi: 10.25126/jtiik.20196953.
- [13] E. Pelivani, A. Besimi, and B. Cico, "A Comparative Study of UI Testing Framework," in *2021 10th Mediterranean Conference on Embedded Computing, MECO 2021*, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/MECO55406.2022.9797165.
- [14] F. Mobaraya and S. Ali, "Technical Analysis of Selenium and Cypress as Functional Automation Framework for Modern Web Application Testing," Academy and Industry Research Collaboration Center (AIRCC), Dec. 2019, pp. 27–46. doi: 10.5121/csit.2019.91803.
- [15] S. Melia and P. F. Putra, "Analisis Perbandingan Tools Pengujian Otomatis pada GUI Aplikasi Berbasis WEB," *SENTIMAS : Seminar Nasional Penelitian dan Pengabdian Masyarakat*, vol. PROSIDING 2023, pp. 267–273, Aug. 2023.
- [16] M. Rijoly, N. T. Sapulette, B. P. Tomasouw, and D. Patty, "Penerapan Metode The Distance To The Ideal Alternative (DIA) Untuk Menyelesaikan Pegawai Di PT. Fast Food Indonesia (KFC Indonesia) Kakiyaly Tanah Tinggi, Ambon," *Tensor: Pure and Applied Mathematics Journal*, vol. 4, no. 1, pp. 13–20, Jun. 2023, doi: 10.30598/tensorvol4iss1pp13-20.
- [17] H. Buender and H. Kuchen, "Towards behavior-driven graphical user interface testing," *ACM SIGAPP Applied Computing Review*, vol. 19, pp. 5–17, Jun. 2019, doi: 10.1145/3357385.3357386.
- [18] O. Kilincceker, A. Silistre, F. Belli, and M. Challenger, "Model-Based Ideal Testing of GUI Programs—Approach and Case Studies," *IEEE Access*, vol. 9, pp. 68966–68984, 2021, doi: 10.1109/ACCESS.2021.3077518.
- [19] B. Ayan, S. Abacıoğlu, and M. P. Basilio, "A Comprehensive Review of the Novel Weighting Methods for Multi-Criteria Decision-Making," *MDPI Information*, vol. 14, no. 5, 2023, doi: 10.3390/info14050285.
- [20] M. Hasan *et al.*, "Testing React Single Page Web Application using Automated Testing Tools," in *Proceedings of the 17th International Conference on Evaluation of Novel Approaches to Software Engineering*, SCITEPRESS - Science and Technology Publications, 2022, pp. 469–476. doi: 10.5220/0011077900003176.