



ANALISIS PERBANDINGAN EFISIENSI DAN PERFORMA PENGUJIAN SELENIUM, CYPRESS, DAN PLAYWRIGHT SEBAGAI ALAT AUTOMATION TESTING UNTUK APLIKASI WEB

Satrio Pratama Wijaya¹, M. Akmal Al Abdilah², Muhammad Farhan³, Afif Firmansyah⁴

^{1,2,3,4} Program Studi Teknik Informatika, Fakultas Teknik, Universitas Pelita Bangsa,
Jl. Inspeksi Kalimalang No.9, Cibatu, Cikarang Sel., Kabupaten Bekasi, Jawa Barat

satrio363@mhs.pelitabangsa.ac.id^{1*}, akmalabdilah6@mhs.pelitabangsa.ac.id²,
mfarhan020603@mhs.pelitabangsa.ac.id³, afif600@mhs.pelitabangsa.ac.id⁴.

Abstract

As information technology advances, software testing becomes a crucial element in developing reliable and high-performing web applications. Automation testing has emerged as a dominant approach to enhance testing efficiency by minimizing manual involvement and reducing human error. This study aims to fill a gap in the existing literature by quantitatively comparing the efficiency and performance of three widely used automation testing tools: Selenium, Cypress, and Playwright. The primary focus of this research is on execution speed, system resource utilization (CPU and memory), and testing stability. The study employed a comparative quantitative approach, with tests conducted repeatedly five times for each tool under identical hardware and system configurations to ensure objectivity. The login process on a public test site, <https://www.saucedemo.com/>, was used as the uniform testing scenario. The results show that Playwright consistently demonstrates superior efficiency and lower resource consumption (CPU 6.25%, memory 89.73 MB, login time 1071.80 ms), making it ideal for large-scale or parallel automation. Cypress occupies a middle ground, offering competitive performance (CPU 8.70%, memory 97.45 MB, login time 1220.40 ms) but with slightly higher resource usage than Playwright. Selenium, despite its extensive support, exhibited the heaviest performance (CPU 12.60%, memory 167.32 MB, login time 2048.60 ms) due to its WebDriver-based architecture. These findings provide valuable insights for software developers and QA practitioners in selecting the most suitable automation testing tool for their projects.

Keywords: Automation Testing, Performance, Selenium, Cypress, Playwright.

Article History

Received: Juni 2025

Reviewed: Juni 2025

Published: Juni 2025

Plagiarism Checker No 235

Prefix DOI :

[10.8734/Kohesi.v1i2.365](https://doi.org/10.8734/Kohesi.v1i2.365)

Copyright : Author

Publish by : Kohesi



This work is licensed under
a [Creative Commons
Attribution-NonCommercial
4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/)



Abstrak

Seiring berkembangnya teknologi informasi, pengujian perangkat lunak menjadi elemen penting dalam pengembangan aplikasi web yang andal dan berkinerja tinggi. Automation testing telah menjadi pendekatan dominan untuk meningkatkan efisiensi pengujian dengan meminimalkan keterlibatan manual dan mengurangi kesalahan manusia. Penelitian ini bertujuan untuk mengisi celah dalam literatur yang ada dengan membandingkan efisiensi dan performa tiga alat automation testing yang banyak digunakan: Selenium, Cypress, dan Playwright. Fokus utama penelitian ini adalah pada kecepatan eksekusi, penggunaan sumber daya sistem (CPU dan memori), dan stabilitas pengujian. Penelitian ini menggunakan pendekatan kuantitatif komparatif, dengan pengujian berulang sebanyak lima kali untuk setiap alat di bawah konfigurasi perangkat keras dan sistem yang identik guna memastikan objektivitas. Skenario pengujian yang seragam adalah proses login pada situs uji coba publik <https://www.saucedemo.com/>. Hasil menunjukkan bahwa Playwright secara konsisten menunjukkan efisiensi superior dan konsumsi sumber daya yang lebih rendah (CPU 6.25%, memori 89.73 MB, waktu login 1071.80 ms), menjadikannya ideal untuk otomatisasi skala besar atau paralel. Cypress menempati posisi tengah, menawarkan performa kompetitif (CPU 8.70%, memori 97.45 MB, waktu login 1220.40 ms) namun dengan penggunaan sumber daya sedikit lebih tinggi dibandingkan Playwright. Selenium, meskipun memiliki dukungan luas, menunjukkan performa terberat (CPU 12.60%, memori 167.32 MB, waktu login 2048.60 ms) karena arsitektur berbasis WebDriver-nya. Temuan-temuan ini memberikan wawasan berharga bagi pengembang perangkat lunak dan praktisi QA dalam memilih alat automation testing yang paling sesuai untuk proyek mereka.

Kata Kunci: Pengujian Otomatisasi, Performa, Selenium, Cypress, Playwright

1. PENDAHULUAN

Pengujian perangkat lunak (*software testing*) menjadi penting dalam proses pengembangan aplikasi web yang andal dan berkinerja tinggi seiring perkembangan teknologi informasi. Untuk meningkatkan efisiensi proses pengujian dengan mengurangi keterlibatan manual dan mengurangi kemungkinan kesalahan manusia, pengujian otomatis telah menjadi metode yang paling populer. Dalam hal ini, banyak alat pengujian otomatis telah dikembangkan, tetapi Selenium, Cypress, dan Playwright adalah yang paling umum di industri saat ini. (Fairuz and Sofani 2024)(Haidar et al. 2025).

Selenium adalah alat pengujian otomatis berbasis *browser* yang telah lama dikenal dalam industri pengujian perangkat lunak. Alat ini bekerja dengan banyak bahasa pemrograman dan kompatibel dengan banyak *browser*, membuatnya pilihan yang menarik bagi banyak pengembang. Namun, Cypress dan Playwright muncul sebagai solusi alternatif di tengah tuntutan kecepatan dan kemudahan integrasi pada arsitektur modern. Keduanya menawarkan keunggulan dalam hal arsitektur, kemudahan *debugging*, dan kecepatan eksekusi; Cypress, dengan arsitektur berbasis JavaScript yang dijalankan langsung dalam browser, memungkinkan



pengujian yang lebih reaktif dan *real-time* dan Playwright, yang dikembangkan oleh *Microsoft*, menonjol dengan fitur canggih seperangkat. (Ismanto, Herlandy, and Renita Rahmadani 2024) (Putri, Muchlas, and Ishafit 2022).

Meskipun masing-masing alat memiliki keunggulan teknis yang berbeda, belum banyak penelitian yang secara eksplisit dan kuantitatif membandingkan efisiensi dan efektivitas pengujian dari ketiganya dalam situasi yang sama. Namun, pemilihan alat pengujian otomatis yang tepat sangat penting untuk kecepatan pengembangan, stabilitas aplikasi, dan efisiensi penggunaan sumber daya sistem. (Fairuz and Sofani 2024) (Utomo, Wijaya, and Setiawan 2023) (Amalia and Cahyono 2019) (Prasetyo and Setiani 2025) (Jevon et al. 2025).

Akibatnya, penelitian ini akan menganalisis dan membandingkan efisiensi dan performa Selenium, Cypress, dan Playwright dalam melakukan pengujian otomatis pada aplikasi web untuk mengisi celah tersebut. Kecepatan eksekusi, penggunaan sumber daya sistem (CPU dan memori), dan stabilitas pengujian dari ketiga alat tersebut adalah topik utama penelitian ini. Penelitian ini diharapkan dapat membantu pengembang perangkat lunak dan praktisi pengendalian kualitas *software* dalam memilih alat yang paling cocok untuk kebutuhan proyek mereka. (Ismanto, Herlandy, and Renita Rahmadani 2024) (Hartono 2021).

Fokus penelitian adalah pertanyaan utama: "Bagaimana perbandingan efisiensi dan performa pengujian antara Selenium, Cypress, dan Playwright dalam melakukan *automation testing* pada aplikasi web?" Pengujian akan berfokus pada skenario *login* aplikasi menggunakan <https://www.saucedemo.com/>. situs uji coba publik. Waktu total proses *login*, waktu *to first byte* (TTFB), waktu *load* total, penggunaan CPU, dan penggunaan memori adalah parameter yang digunakan dalam pengukuran. Untuk menjaga hasil yang objektif, pengujian dilakukan dalam konfigurasi sistem dan kondisi perangkat keras yang seragam. Dengan menggunakan metodologi ini, penelitian ini diharapkan dapat memberikan kontribusi ilmiah dan praktis dalam pemilihan alat pengujian otomatis yang efektif yang sesuai dengan kebutuhan kontemporer dalam pengembangan aplikasi web. (Fairuz and Sofani 2024).

Penelitian ini akan menjawab pertanyaan berikut: "Bagaimana perbandingan efisiensi dan performa pengujian antara Selenium, Cypress, dan Playwright dalam melakukan pengujian otomatis pada aplikasi web?" Oleh karena itu, tujuan utama dari penelitian ini adalah untuk melakukan analisis menyeluruh dari ketiga alat ini dan membuat saran berdasarkan hasilnya. (Lingard 2023).

2. METODE PENELITIAN

Metode kuantitatif komparatif digunakan dalam makalah penelitian ini untuk membandingkan efisiensi dan performa dari tiga alat pengujian otomatis: Selenium, Cypress, dan Playwright. Untuk memastikan bahwa kondisi uji konsisten, penelitian ini dilakukan untuk mengukur dan menganalisis lima parameter utama pengujian: total waktu proses login, waktu *to first byte* (TTFB), total load time, penggunaan CPU, dan penggunaan memori. (Jamaludin et al. 2024).

Proses pengujian dimulai dengan membuka halaman login, mengisi username dan password pengguna, dan mengklik tombol login. Data aktivitas akan dicatat setelah login berhasil diverifikasi. Untuk masing-masing alat, pengujian dilakukan secara berulang sebanyak lima kali untuk memastikan hasilnya objektif. Selanjutnya, data hasil dihitung secara rata-rata untuk menghasilkan nilai representatif. Semua pengujian dilakukan pada perangkat yang sama untuk memastikan bahwa kondisi perangkat keras dan kondisi sistem sama. (Lingard 2023).

Perangkat keras yang digunakan untuk pengujian adalah prosesor Intel Core i7 generasi ke-11, RAM 16 GB, dan Windows 11 Pro versi 64-bit. Selama proses pengujian, tidak ada aplikasi berat lain yang digunakan untuk menjaga kestabilan hasil pengukuran. Agar pengujian tidak bias karena perbedaan engine peramban, Google Chrome versi terbaru digunakan. Script Python untuk Selenium dijalankan menggunakan Chrome WebDriver. Playwright juga menggunakan



JavaScript dan dijalankan melalui Node, sementara Cypress menggunakan runner internalnya js. (Hardianto, Prima Aditiawan, and Nugroho Sihananto 2024) (Rahma et al. 2023).

Karena Cypress dan Playwright dapat mengukur waktu dengan sangat akurat dalam satuan milidetik, kami menggunakan `performance.now()` dan `perf_counter()` untuk Selenium di Python. Pengukuran TTFB dan waktu load total dapat diperoleh melalui API kinerja yang tersedia di masing-masing alat. Pengukuran CPU dan memori dilakukan menggunakan modul `psutil` di Selenium, `pidusage` di Playwright, dan `os-utils` di Cypress. Pendekatan estimasi konsumsi sistem secara real-time digunakan selama proses. (Jamaludin et al. 2024) (Ningtyas, Gumilang, and Hanafi 2023).

Setelah data dikumpulkan, nilai rata-rata masing-masing metrik pengujian digunakan untuk membandingkan semua hasil. Untuk memudahkan analisis dan penarikan kesimpulan, perbandingan dilakukan baik dalam bentuk numerik maupun visual dengan menggunakan grafik. Metodologi ini dimaksudkan untuk menjawab rumusan masalah secara langsung dan memberikan dasar yang kuat untuk menilai efisiensi dan performa alat pengujian otomatisasi dalam pengujian aplikasi web. (Ulfah and Najiah 2023).

3. HASIL DAN DISKUSI

Hasil pengujian terhadap tiga alat pengujian otomatis (Selenium, Cypress, dan Playwright) disajikan secara sistematis pada bab ini. Waktu proses login total, waktu *to first byte* (TTFB), waktu load total, penggunaan CPU, dan penggunaan memori adalah lima parameter utama yang menjadi fokus pemeriksaan. Data diperoleh dari pengujian aktual yang berulang yang mempertimbangkan ketepatan, stabilitas, dan konsistensi metrik teknis.

3.1 Hasil Pengujian

Tabel 1. Hasil Pengujian Perbandingan

Tools	Total Waktu Login (ms)	TTFB (ms)	Total Load Time (ms)	CPU Digunakan (%)	Memori Digunakan (MB)
Selenium	2048.60	652.40	2701.00	12.60	167.32
Cypress	1220.40	168.20	1634.80	8.70	97.45
Playwright	1071.80	144.60	1480.00	6.25	89.73

Playwright menunjukkan bahwa di antara ketiga alat, dia melakukan pekerjaan yang paling efektif, menurut data yang ditunjukkan dalam Tabel 1. Dengan waktu login total 1071.80 milidetik, TTFB (*Time to First Byte*) 144.60 milidetik, dan waktu *load* total 1480.00 milidetik, Playwright unggul dalam kecepatan proses dan *rendering* halaman. Dengan menggunakan hanya 6.25% CPU dan 89.73 MB memori, ini sangat cocok untuk pengujian berskala besar atau otomatisasi paralel. Untuk saat ini, Cypress berada di tengah-tengah dalam hampir semua metrik. Meskipun sedikit di bawah Playwright, kecepatan *login* 1220.40 ms dan *load* 1634.80 ms cukup bagus. Namun, Cypress menggunakan lebih banyak sumber daya, seperti yang ditunjukkan oleh konsumsi CPU 8.70% dan memori 97.45 MB. Ini wajar mengingat Cypress menjalankan tes dalam browser yang sama dengan aplikasi, yang memungkinkan pemantauan DOM secara *real-time*. Sebaliknya, Selenium memiliki performa terbaik. Dengan waktu login total 2048.60 milidetik, TTFB 652.40 milidetik, dan waktu pemuatan halaman penuh 2701.00 milidetik, Selenium menjadi yang paling lambat dalam interaksi awal dan pemuatan halaman penuh. Ini menunjukkan peningkatan beban sistem dengan penggunaan CPU 12.60% dan memori 167.32 MB. Arsitektur Selenium yang menggunakan **WebDriver** dan proses eksternal yang berbeda dari browser menyebabkan latensi komunikasi yang lebih besar.



3.2 Analisis Perbandingan Efisiensi Waktu Proses Login

Untuk waktu *login* total, Playwright paling efisien, dengan rata-rata 1071.80 milidetik, diikuti oleh Cypress dengan 1220.40 milidetik. Selenium, meskipun berfungsi, mencatat waktu *login* yang paling lambat, yaitu 2048.60 milidetik, tetapi ini menunjukkan keunggulan Playwright dalam efisiensi skrip dan eksekusi yang cepat.

Time to First Byte (TTFB)

Playwright kembali unggul dalam TTFB, dengan rata-rata 144.60 ms, sedikit lebih cepat dari Cypress (168.20 ms). Selenium memiliki TTFB tertinggi, 652.40 ms, yang menunjukkan bahwa waktu respons awal server lebih lambat saat diakses melalui Selenium.

Total Load Time

Playwright kembali mencatat waktu load tercepat secara keseluruhan sebesar 1480.00 milidetik, dengan Cypress menempati posisi kedua pada 1634.80 milidetik, dan Selenium menempati posisi terakhir pada 2701.00 milidetik. Hasil ini mendukung kesimpulan bahwa Playwright melakukan pengoptimalan *rendering* dan penanganan halaman.

Penggunaan CPU

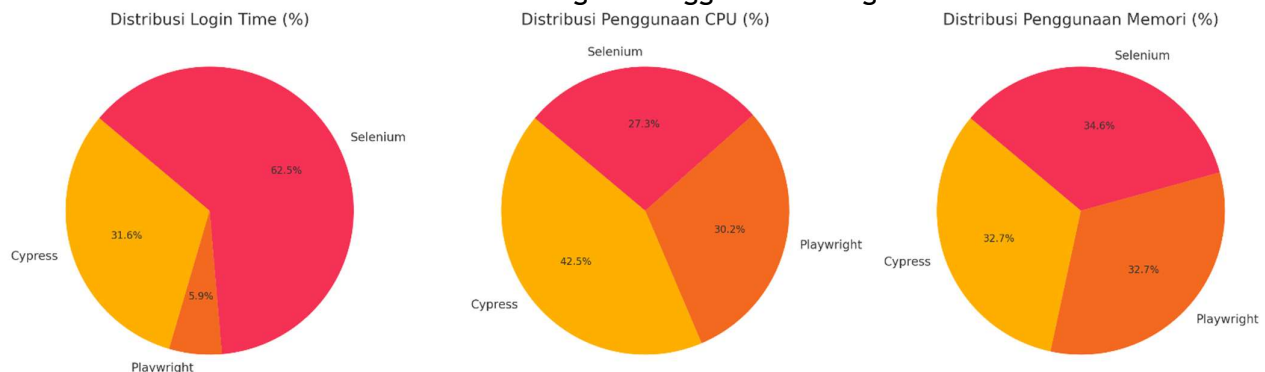
Dengan konsumsi CPU rata-rata hanya 6,25%, Playwright paling efisien, diikuti oleh Cypress sebesar 8,70%, dan Selenium sebesar 12,60%. Ini menunjukkan bahwa arsitektur Playwright lebih ringan dan efisien dalam mengelola proses pengujian otomatis.

Penggunaan Memori

Playwright menggunakan jumlah memori paling sedikit, rata-rata 89.73 MB, sedangkan Cypress menggunakan 97.45 MB, dan Selenium menggunakan jumlah memori tertinggi, 167.32 MB. Ini menunjukkan bahwa Playwright cocok untuk pengujian dengan sumber daya terbatas.

Visualisasi Perbandingan

Grafik 1 : Perbandingan Menggunakan Diagram Pie.



Menurut Grafik 1, Perbandingan Menggunakan Diagram Pie, masing-masing alat uji otomatisasi *web* memiliki perbandingan waktu login, penggunaan CPU, dan penggunaan memori. Data ini menunjukkan efisiensi dan beban kerja Cypress, Playwright, dan Selenium.

1. Distribusi *Login Time* (%):

Dari data asli, waktu login yang dibutuhkan oleh masing-masing alat uji adalah:

- Cypress: 3024.20 ms
- Playwright: 565.17 ms
- Selenium: 5990.00 ms

Dari total waktu *login* otomatis yang tercatat, Selenium membutuhkan waktu paling lama (sekitar 62.5%), karena arsitektur berbasis *WebDriver*-nya yang mengandalkan komunikasi eksternal, menjadikannya lebih lambat dalam merespons dan berinteraksi dengan UI. Sebaliknya, Playwright tampil sebagai yang tercepat dengan hanya memakan 5.9% dari total



waktu karena dibangun langsung di atas protokol *browser*, memungkinkan interaksi yang sangat cepat dan efisien. Sementara itu, Cypress berada di tengah-tengah dengan 31.6%, meskipun cepat, ia berjalan di dalam browser yang sama dengan aplikasi sehingga menimbulkan *overhead* tertentu yang membuatnya sedikit lebih lambat dari Playwright namun tetap jauh lebih baik dari Selenium.

2. Distribusi Penggunaan CPU (%):

Penggunaan CPU oleh masing-masing alat saat uji login:

- Cypress: 7.78%
- Playwright: 5.52%
- Selenium: 5.00%

Ketika berbicara tentang penggunaan CPU, Cypress paling banyak memakan sumber daya, menyumbang sekitar 42.5% dari total 18.3% penggunaan CPU. Ini terjadi karena Cypress menjalankan aplikasi dan pengujian dalam satu proses *browser* yang sama, yang menyebabkan konsumsi CPU lebih tinggi saat melakukan pemantauan dan validasi secara *real-time*. Sebaliknya, Playwright dan Selenium jauh lebih hemat dalam penggunaan CPU. Playwright efisien karena ia langsung menjalankan perintah ke *browser engine*, sementara Selenium, meskipun lebih lambat dalam waktu, berinteraksi melalui *driver* eksternal yang relatif ringan untuk CPU.

3. Distribusi Penggunaan Memori (%):

Memori yang digunakan masing-masing alat saat uji login:

- Cypress: 127.67 MB
- Playwright: 127.48 MB
- Selenium: 135.01 MB

Dari total konsumsi memori sebesar 390.16 MB, distribusinya cukup merata di antara ketiga *tools* pengujian otomatis ini. Selenium sedikit lebih boros, menyumbang 34.6% dari total memori, karena arsitekturnya yang memerlukan proses *browser* independen dan *driver* terpisah untuk setiap sesi. Sementara itu, Cypress dan Playwright menunjukkan efisiensi yang sangat mirip dalam penggunaan memori, masing-masing 32.7%. Keduanya terbukti hemat karena efisiensi mereka dalam memproses *rendering* dan *assertion* yang dilakukan langsung dalam konteks *browser* atau melalui protokol yang cepat, membuat penggunaan memori mereka sangat seimbang.

KESIMPULAN

Penelitian ini telah melakukan analisis perbandingan terhadap tiga alat populer dalam pengujian otomatisasi aplikasi *web*, yaitu Selenium, Cypress, dan Playwright, berdasarkan lima parameter utama:

1. Efisiensi Waktu Eksekusi

Playwright selalu menunjukkan waktu yang lebih efisien daripada dua alat lainnya. Playwright terbukti mampu menyelesaikan pengujian dengan cepat dan efektif dengan waktu *login* dan load time total tercepat. Cypress menempati posisi kedua dengan performa waktu yang cukup kompetitif. Sebaliknya, Selenium memiliki kecepatan eksekusi terendah, dengan waktu *login* dan load hampir dua kali lebih lama daripada Playwright.

2. Respons Awal dari Server (TTFB)

Playwright dan Cypress menunjukkan performa yang sangat baik dalam hal TTFB, dengan respons server yang cepat dan stabil. TTFB yang rendah menunjukkan proses komunikasi jaringan dan *load* awal yang optimal, dan Selenium mencatat TTFB tertinggi, yang menunjukkan adanya *overhead* atau latensi pada engine browser atau driver yang digunakan.



3. Penggunaan Sumber Daya Sistem

Dalam pengujian otomatisasi, penggunaan CPU dan memori sangat penting, terutama dalam skenario berskala besar atau berjalan paralel. Playwright adalah pilihan terbaik untuk eksekusi di lingkungan dengan sumber daya terbatas karena mencatat penggunaan CPU dan memori yang paling efisien. Cypress berada di tengah karena konsumsinya masih wajar. Sebaliknya, Selenium menunjukkan kinerja terburuk dengan konsumsi CPU dan memori tertinggi, yang dapat mempengaruhi skalabilitas pengujian.

4. Kesimpulan Umum

Secara keseluruhan, Playwright adalah alat pengujian otomatisasi yang paling hemat dan efisien dalam penelitian ini, baik dari segi kecepatan eksekusi maupun jumlah sumber daya yang digunakan. Terutama untuk integrasi modern berbasis JavaScript, Cypress menjadi alternatif yang layak. Pengujian ini menunjukkan bahwa selenium, meskipun paling matang dan mendapatkan dukungan luas, kurang efektif dan efisien.

5. Implikasi Praktis

Penelitian ini diharapkan dapat membantu pengembang perangkat lunak, *engineer* QA, dan tim *DevOps* memilih alat pengujian otomatisasi yang paling cocok untuk kebutuhan proyek mereka. Playwright dapat menjadi pilihan utama dalam proyek yang menuntut cepat. Namun, Selenium masih memiliki tempat tersendiri untuk proyek yang membutuhkan kompatibilitas luas dan integrasi dengan ekosistem *legacy*.

DAFTAR PUSTAKA

- [1] Amalia, Anisa, and Andhik Budi Cahyono. 2019. "Analisis Pemanfaatan Playwright Untuk Pengujian Otomatis Aplikasi Berbasis Web (Studi Kasus: Sistem Manajemen Jaringan)." *Jurnal Universitas Islam Indonesia* 2 (2): 1-8.
- [2] Fairuz, Ratu, and Hasna Sofani. 2024. "Pemanfaatan Selenium WebDriver Untuk Pengujian Regresi Aplikasi Berbasis Web." *Jurnal Informatika Dan Teknologi Komputer* 05 (01): 36-42. <https://ejurnalunsam.id/index.php/jicom/>.
- [3] Haidar, Muhammad, Afif Mufid, Muhammad Aqil Zidane, Shopi Attika Putri, Rintan Wirnanti, and Ali Ibrahim. 2025. "PENGUJIAN WEBSITE MENGGUNAKAN SELENIUM IDE PADA JDIH PESAWARAN MENGGUNAKAN METODE EQUIVALENCE PARTITIONING" 9 (4): 6055-61.
- [4] Hardianto, Eragradiansyah, Firza Prima Aditiawan, and Andreas Nugroho Sihananto. 2024. "Pemanfaatan Selenium Ide Dalam Otomatisasi Pengujian Black Box Pada Website Umkm Kota Madiun Dengan Teknik Equivalence Partitioning." *JATI (Jurnal Mahasiswa Teknik Informatika)* 8 (3): 3471-77. <https://doi.org/10.36040/jati.v8i3.9729>.
- [5] Hartono, Budi. 2021. *Cara Mudah Dan Cepat Sistem Informasi*.
- [6] Ismanto, Edi, Pratama Benny Herlandy, and Renita Rahmadani. 2024. "Pengembangan Pengembangan Learning Management System (LMS) Dengan Pendekatan Self Directed Learning (SDL) Untuk Sekolah Menengah Kejuruan (SMK) Di Kota Pekanbaru." *Jurnal Fasikom* 14 (1): 66-74. <https://doi.org/10.37859/jf.v14i1.6882>.
- [7] Jamaludin, ahmad, Aliv Rivaldi, Faizal Maulana, and Riyan Kristian. 2024. "Otomatisasi Pengujian Aplikasi Bibli Menggunakan IDE." *Jurnal BATIRSI, Vol.8, No.1, Juli 2024* 8 (1): 9-11.
- [8] Jevon, Eliazer, Carlos Loelan, I Gede Suardika, and Riza Wulandari. 2025. "Implementasi Automation Testing Menggunakan Playwright Pada Project Di PT Itsavirus" 2 (1): 883-88.
- [9] Lingard, Lorelei. 2023. "Writing with ChatGPT: An Illustration of Its Capacity, Limitations & Implications for Academic Writers." *Perspectives on Medical Education* 12 (1): 261-70. <https://doi.org/10.5334/pme.1072>.



- [10] Ningtyas, Kinanti Tiara, Soni Fajar Surya Gumilang, and Ridha Hanafi. 2023. "Perancangan Arsitektur Sistem Pemerintahan Berbasis Elektronik Pada Urusan Sosial Di Pemerintah Provinsi Jawa Barat Berbasis Konsep Enterprise Architecture Menggunakan Kerangka Kerja Togaf Adm." *JUPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)* 8 (2): 355-69. <https://doi.org/10.29100/jupi.v8i2.3454>.
- [11] Prasetyo, Gilang Aries, and Novi Setiani. 2025. "Perbandingan Kinerja Framework Automation Ui Testing Menggunakan Metode the Distance to the Ideal Alternative 1." 10 (1): 224-37.
- [12] Putri, Sajidatun Nufus Rani, Muchlas Muchlas, and Ishafit Ishafit. 2022. "Development of Fluid Virtual Laboratory for Online." *Jurnal Pendidikan Fisika* 10 (1): 36-42.
- [13] Rahma, Nabilatur, Yusuf Amrozi, Nur Diana Fahma Salsabila, and M.Hanif Miqdad G. 2023. "Telaah Kajian Pustaka Pemodelan Sistem Pendukung Keputusan Pada Usaha Mikro Kecil Dan Menengah." *Jurnal Simantec* 11 (2): 185-90. <https://doi.org/10.21107/simantec.v11i2.9725>.
- [14] Ulfah, Amanny, and Ina Najiah. 2023. "Implementasi Web Scraping Pada Situs Jurnal Sinta Menggunakan Framework Selenium Webdriver Python." *JIKA (Jurnal Informatika)* 7 (1): 29. <https://doi.org/10.31000/jika.v7i1.7037>.
- [15] Utomo, Ary, Ganang Wijaya, and Yanto Setiawan. 2023. "Implementation of Crowdfunding Web Application Using AWS Amplify Architecture with End-to-End Testing Using Playwright." *Indonesian Journal of Multidisciplinary Science* 2 (11): 3888-3904. <https://doi.org/10.55324/ijoms.v2i11.620>.