Maliki Interdisciplinary Journal (MIJ)
Volume 2, Issue 11, 2024 pp. 762-771
eISSN: 3024-8140
http://urj.uin-malang.ac.id/index.php/mij/index

Analisis perbandingan Tools pengujian otomatis pada GUI aplikasi berbasis WEB

Yudha Pramana Putra
Program Studi Teknik Informatika, Universitas Islam Negeri Maulana Malik Ibrahim Malang;
e-mail: yudhapramanaputra58@gmail.com

Kata Kunci:
selenium webdriver; katalon studio; aplikasi quizizz; siklus hidup pengujian perangkat lunak; STLC

Keywords:
selenium webdriver; katalon studio; aplikasi quizizz; software testing life cycle; STLC

A B S T R A K

Penelitian ini membandingkan Selenium Webdriver dan Katalon Studio dalam menguji antarmuka pengguna grafis (GUI) pada aplikasi web Quizizz. Penelitian ini menggunakan metode Software Testing Life Cycle (STLC) untuk membandingkan Selenium Webdriver dan Katalon Studio dalam pengujian antarmuka pengguna grafis (GUI) pada aplikasi web Quizizz. Hasilnya menunjukkan Selenium Webdriver lebih cepat dalam eksekusi test case dan mendukung scrolling, namun memerlukan instalasi tambahan untuk pengujian berbasis gambar. Katalon Studio menonjol dalam dokumentasi hasil eksekusi yang lengkap dan mendukung berbagai bahasa pemrograman. Pemilihan alat harus disesuaikan dengan kebutuhan pengujian spesifik aplikasi yang dituju, dengan memperhatikan karakteristik dan kemampuan masing-masing alat untuk meningkatkan efisiensi dan efektivitas pengujian GUI.

A B S T R A C T

This research compares Selenium Webdriver and Katalon Studio in testing the graphical user interface (GUI) on the Quizizz web application. This research uses the Software Testing Life Cycle (STLC) method to compare Selenium Webdriver and Katalon Studio in testing the graphical user interface (GUI) on the Quizizz web application. The results show Selenium Webdriver is faster in test case execution and supports scrolling, but requires additional installation for image-based tests. Katalon Studio stands out for its complete documentation of execution results and support for multiple programming languages. Tool selection must be tailored to the specific testing needs of the intended application, taking into account the characteristics and capabilities of each tool to increase the efficiency and effectiveness of GUI testing.

Pendahuluan

Pengujian antarmuka pengguna grafis (GUI) pada perangkat lunak sangat penting untuk menjamin kualitas keseluruhan sistem, terutama dalam aplikasi kuis. Elemen-elemen GUI seperti pertanyaan, pilihan jawaban, dan navigasi memainkan peran signifikan dalam pengalaman pengguna dan kinerja aplikasi. Di era perkembangan perangkat lunak yang pesat, pengujian GUI menjadi aspek esensial untuk memastikan kualitas sistem secara keseluruhan, terutama mengingat kompleksitas aplikasi modern seperti aplikasi seluler dan situs web. Pengujian GUI adalah pengujian yang dilakukan pada aplikasi yang memiliki titik koneksi dengan cara mencari beberapa cara untuk

This is an open access article under the CC BY-NC-SA license.
Copyright © 2023 by Author. Published by Universitas Islam Negeri Maulana Malik Ibrahim Malang.

menguji bagian bagian GUI. Karena kualitas sistem secara keseluruhan dipengaruhi oleh fungsionalitas widget GUI (Melia, 2023).

Ada dua metode pengujian yang umum digunakan: pengujian manual dan pengujian otomatis. Dalam pengujian manual, analis atau penguji secara langsung mencoba dan menguji perangkat lunak. Sedangkan pengujian otomatis digunakan untuk mengurangi risiko yang tidak dapat dihindari dalam pengujian manual dan untuk mengeliminasi kesalahan yang mungkin dilakukan oleh manusia. Penguji berinteraksi dengan sistem untuk menguji berbagai fungsionalitas system (Nyamathulla et al., 2021), tetapi pengujian manual tidak mencakup berbagai pengujian dan penguji sering melakukan hal yang sama berulang kali. Selenium Webdriver, Katalon Studio, dan sejumlah alat lainnya adalah beberapa di antara banyak alat pengujian otomatis (Bhagat et al., 2020).

Aplikasi Quizizz merupakan salah satu contoh platform interaktif yang dirancang untuk menyediakan berbagai jenis kuis atau ujian secara online. Dalam konteks pendahuluan, penjelasan tentang aplikasi Quizizz memberikan latar belakang yang kuat tentang mengapa pengujian antarmuka pengguna grafis (GUI) menjadi sangat penting (Shariff, 2019). Dalam proses belajar mengajar modern, teknologi telah menjadi komponen penting dalam penyampaian materi pembelajaran dan penilaian pemahaman siswa. Quizizz dideskripsikan sebagai sebuah web tool untuk membuat permainan kuis interaktif untuk digunakan dalam pembelajaran di kelas anda misalnya saja untuk penilaian formatif (Gonzalez et al., 2022).

Meskipun terdapat banyak alat pengujian otomatis, tidak semuanya dapat mengakomodasi berbagai jenis pengujian yang dibutuhkan, seperti pengujian fungsional, pengujian beban, pengujian kinerja, dan pengujian unit. Oleh karena itu, pemahaman mendalam tentang aplikasi yang akan diuji sangat penting untuk memilih alat pengujian yang sesuai dengan kebutuhan spesifik.

Penelitian ini bertujuan untuk membandingkan dua alat pengujian otomatis, yaitu Selenium WebDriver dan Katalon Studio, berdasarkan beberapa parameter pengujian pada objek yang spesifik, yaitu aplikasi Quizizz. Dengan demikian, penelitian ini memberikan kontribusi penting dalam membantu memilih alat pengujian otomatis yang tepat untuk aplikasi tertentu, serta menekankan pentingnya memilih alat yang sesuai dengan kebutuhan pengujian spesifik.

Pembahasan

Software Testing Life Cycle (STLC) adalah serangkaian proses dan aktivitas yang dirancang untuk mengelola dan mengatur pengujian perangkat lunak secara efektif. Pada penelitian ini, metodologi STLC digunakan untuk menguji dan membandingkan alat. Proses pengujian STLC dibagi menjadi enam tahap: analisis kebutuhan, perencanaan pengujian, pengembangan kasus uji, penyiapan lingkungan, eksekusi pengujian, dan penutupan siklus pengujian. Pada penelitian ini, metode STLC digunakan untuk menjelaskan tahapan kegiatan yang lebih spesifik dengan menggunakan pendekatan terstruktur (Pelivani, 2022). STLC memberikan kerangka kerja yang terstruktur dan terorganisir untuk mengelola seluruh proses pengujian perangkat lunak

dari awal hingga akhir. Ini membantu memastikan bahwa pengujian dilakukan secara menyeluruh dan efisien, sehingga mengurangi risiko dan meningkatkan kualitas perangkat lunak yang dihasilkan. Dengan menggunakan STLC, tim pengujian dapat mengidentifikasi masalah lebih awal dalam siklus pengembangan perangkat lunak dan memberikan umpan balik yang berharga kepada tim pengembangan untuk perbaikan yang diperlukan.

Quizizz adalah platform interaktif yang dirancang untuk menyediakan berbagai kuis dan ujian secara online. Dengan antarmuka yang intuitif dan ramah pengguna, Quizizz memudahkan pengajar atau instruktur dalam membuat kuis atau ujian, serta memungkinkan siswa atau peserta untuk mengikuti kuis atau ujian tersebut dengan nyaman dari perangkat mereka. Hal ini sesuai dengan penelitian Agustina & Rusmana (2019), bahwa Quizizz merupakan aplikasi yang layak digunakan sebagai aplikasi pembelajaran yang mendukung revolusi pembelajaran 4.0 karena penggunaan yang mudah dann proses penilaian yang cepat.

Quizizz cukup layak jika digunakan sebagai media evaluasi khususnya pada saat pembelajaran daring (Amany, 2020).

Eksplorasi Alat yang akan di Uji

Katalon Studio

Katalon Studio adalah aplikasi open source untuk pengujian secara otomatis yang dikembangkan oleh Katalon LLC. Katalon mendukung 3 platform pengujian secara automasi, yaitu Web testing, API testing, dan Mobile testing. Katalon Studio tidak mengharuskan penggunanya untuk menguasai bahasa pemrograman karena katalon studio memiliki user interface yang simpel dan informatif sehingga mudah digunakan bagi pemula hingga yang sudah berpengalaman. Katalon Studio menyediakan antarmuka pengguna yang intuitif dan fitur-fitur yang kuat untuk mengotomatisasi pengujian aplikasi web, mobile, dan API.

Pengujian perangkat lunak adalah teknik yang paling sering digunakan untuk memverifikasi dan memvalidasi kualitas perangkat lunak. Katalon Studio memiliki kelebihan yaitu katalon mendukung pengujian API, Web, dan Seluler dan tidak mewajibkan penggunanya memiliki kemampuan pemrograman. Kelemahan katalon tidak mendukung pengujian terdistribusi.

Selenium WebDriver adalah salah satu alat pengujian otomatis yang sangat populer dan banyak digunakan. Selenium sendiri terdiri dari empat jenis komponen utama, yaitu Selenium Grid, Selenium IDE, Selenium RC (Remote Control), dan Selenium WebDriver. Masing-masing komponen ini menawarkan jenis konten uji yang berbeda, yang dapat disesuaikan dengan berbagai kasus pengujian spesifik yang dihadapi. Dengan fleksibilitas dan kemampuan yang beragam ini, Selenium memberikan solusi pengujian yang komprehensif untuk berbagai kebutuhan pengujian perangkat lunak, memastikan bahwa setiap aspek dari aplikasi dapat diuji secara efektif.Selenium juga menyediakan dukungan multi-browser. Selain itu, selenium juga mendukung untuk beberapa sistem operasi dan bahasa pemrograman yang berbeda. Selenium Webdriver menikmati manfaat sebagai alat selenium tercepat di antara berbagai bagian

kompartemen alat Selenium. Pengujian lintas-browser dan pengujian dinamis halaman situs didukung oleh Selenium WebDriver.

Metode Penelitian

Menggunakan metode Softwate Testing Life Cycle (STLC) yang meliputi:

1. Requirement Analisis
2. Test Planning
3. Test Case Development
4. Environtment Setup
5. Test Execution
6. Test Cycle Closure

Requirement Analysis

Aplikasi Quizizz dirancang dengan arsitektur yang modular dan skalabel, yang memungkinkan pengembang untuk menambahkan fitur baru dengan mudah seiring dengan perubahan dan perkembangan kebutuhan pengguna. Struktur modular ini tidak hanya memudahkan dalam hal penambahan fungsionalitas baru, tetapi juga memastikan bahwa setiap komponen aplikasi dapat diperbarui atau diganti tanpa mempengaruhi keseluruhan sistem. Skalabilitas arsitektur ini juga memungkinkan aplikasi untuk menangani peningkatan jumlah pengguna dan data dengan efisien, memastikan kinerja yang optimal meskipun terjadi pertumbuhan yang signifikan dalam basis pengguna. Arsitektur ini terdiri dari komponen-komponen berikut:

1. Antarmuka Pengguna (GUI): Menyediakan antarmuka interaktif untuk pengguna, termasuk halaman registrasi, login, pembuatan dan pengeditan kuis, serta halaman hasil ujian.
2. Backend Server: Bertanggung jawab untuk memproses permintaan pengguna, menyimpan data pengguna dan kuis, serta menyediakan layanan penyimpanan dan pengambilan data.
3. Database: Menyimpan semua informasi terkait pengguna, kuis, jawaban, dan statistik.
4. Integrasi Pihak Ketiga: Memungkinkan aplikasi untuk berintegrasi dengan layanan eksternal seperti sistem manajemen pembelajaran (LMS) atau layanan otentikasi.

Berdasarkan metode diatas didapat langkah-langkah menggunakan metode STLC (Software Testing Life Cycle) adalah sebagai berikut:

1. Pada tahap ini, peneliti akan melakukan analisis mendalam terhadap kebutuhan aplikasi, baik yang bersifat fungsional maupun non-fungsional. Proses analisis kebutuhan ini sangat penting untuk memastikan bahwa semua aspek yang diperlukan untuk aplikasi Quizizz dapat diidentifikasi dan dipenuhi dengan tepat.

   Kebutuhan fungsional mencakup berbagai fitur dan fungsi yang harus dimiliki oleh aplikasi untuk memenuhi tujuan dan persyaratan pengguna. Misalnya, kemampuan

untuk membuat kuis, fitur penilaian otomatis, kemampuan untuk melacak kemajuan siswa, dan berbagai opsi untuk umpan balik dan laporan. Adapun kebutuhan fungsional dan non-fungsional yang digunakan pada aplikasi quizizz yaitu:

Tabel 1. kebutuhan web quizizz

Fungsional	Non-Fungsional
• Sistem bisa melakukan registrasi ke aplikasi.	• Aplikasi harus memiliki kecepatan akses yang cepat.
• Sistem bisa melakukan login ke aplikasi.	• Aplikasi harus dapat diakses dari berbagai perangkat dan OS. • Aplikasi harus stabil dan dapat menangani jumlah pengguna yang besar.
• Sistem harus menyediakan pembuatan dan pengeditan kuis.	

Test Planning

Berikut adalah langkah-langkah perancangan pengujian dari yang pertama ke terakhir dalam bentuk list:

A. Menentukan alat yang akan dibandingkan: Selenium Webdriver dan Katalon Studio.

B. Memilih parameter-parameter yang cocok untuk pembanding antara kedua alat, termasuk:

   1) Waktu Eksekusi Test Case
   2) Penundaan (Delay)
   3) Pengujian berbasis gambar
   4) Scrolling
   5) Dokumentasi Hasil Eksekusi
   6) Memilih objek penelitian web quiziz

C. Merancang proses pengujian dengan menggabungkan langkah-langkah di atas secara terstruktur dan berurutan.

Test Case Development

Pada tahap ini akan dibuat beberapa situasi atau kasus pengujian untuk mencoba fungsi sistem sesuai dengan rencana yang telah dibuat sebelumnya.

Tabel 2. Test Case Development

Parameter	Test Case Name
P1 (Waktu Eksekusi Test Case)	Register
P2 (Penundaan atau Delay)	Login
P3 (Image based Testing)	Cover Post
P4 (Scrolling)	Scroll Position
P5 (Dokumentasi Hasil Eksekusi)	Login
P6 Supported Languages	Languages Pemrograman

Environment Setup

Pada tahap ini, penguji akan melakukan evaluasi menyeluruh terhadap lingkungan pengujian yang akan digunakan. Langkah ini melibatkan persiapan yang cermat dari semua aspek yang diperlukan, termasuk menyusun daftar perangkat lunak dan perangkat keras yang akan digunakan selama proses pengujian. Proses ini mencakup pemilihan dan penyiapan peralatan teknis, seperti server, komputer, dan perangkat tambahan lainnya, serta menentukan bahasa pemrograman dan alat pengembangan yang relevan untuk eksperimen ini. Dengan memastikan bahwa semua elemen teknis telah siap dan terkonfigurasi dengan benar, penguji dapat menjamin bahwa lingkungan pengujian akan mendukung pelaksanaan tes dengan efisien dan efektif.

1) Hardware: Intel(R) Core (TM) i5-12500H 16 GB RAM. 512 GB SSD.
2) Network 100Mbps.
3) Software: Selenium Webdriver, Katalon Studio dan Google Chrome

Test Execution

Pada tahap ini, penguji akan melaksanakan proses pengujian yang telah direncanakan berdasarkan skenario uji yang telah disusun sebelumnya. Proses ini melibatkan penerapan berbagai kasus uji yang dirancang untuk mengevaluasi kinerja, fungsionalitas, dan stabilitas aplikasi sesuai dengan spesifikasi yang telah ditentukan. Penguji akan menjalankan setiap skenario uji secara sistematis, mengumpulkan data yang diperlukan, dan memantau hasil pengujian untuk memastikan bahwa aplikasi berfungsi seperti yang diharapkan. Tahap ini juga mencakup identifikasi dan dokumentasi masalah atau cacat yang ditemukan selama pengujian, serta melakukan verifikasi terhadap perbaikan yang telah diterapkan.

Test Cycle Closure

Tahap penutupan siklus pengujian adalah langkah terakhir dalam proses pengujian. Pada tahap ini, penguji akan menyimpulkan hasil dari seluruh rangkaian pengujian yang telah dilakukan. Proses ini melibatkan analisis menyeluruh dari temuan-temuan selama pengujian, penilaian terhadap kualitas dan kinerja aplikasi, serta

pembuatan laporan akhir yang mencakup ringkasan hasil pengujian, masalah yang ditemukan, dan rekomendasi untuk perbaikan lebih lanjut. Penutupan siklus pengujian juga mencakup evaluasi terhadap keseluruhan proses pengujian, termasuk efektivitas dari skenario uji yang digunakan, serta identifikasi area untuk perbaikan dalam pengujian di masa depan.

Analisis Parameter Perbandingan

Dua alat yang akan dibandingkan dalam penelitian ini adalah Katalon Studio dan Selenium WebDriver. Parameter-parameter perbandingan yang akan digunakan telah ditetapkan berdasarkan analisis mendalam dan tinjauan literatur yang relevan. Pengujian akan dilakukan pada sebuah aplikasi website yang dikenal dengan nama Aplikasi DiaryMe, yang berfungsi mirip dengan platform blog pada umumnya.

Dalam proses perbandingan ini, parameter-parameter tersebut akan berperan sebagai pedoman utama untuk menguji berbagai fitur dari aplikasi DiaryMe. Dengan menetapkan parameter ini, pengujian bertujuan untuk memastikan bahwa semua fitur aplikasi berfungsi dengan baik dan sesuai dengan spesifikasi yang telah ditentukan sebelumnya. Pengujian ini akan mencakup evaluasi terhadap berbagai aspek seperti fungsionalitas, kinerja, dan keandalan fitur aplikasi. Selain itu, hasil dari pengujian ini akan digunakan untuk menentukan efektivitas masing-masing alat dalam konteks aplikasi yang diuji, serta untuk memberikan wawasan tentang kekuatan dan kelemahan dari Katalon Studio dan Selenium WebDriver dalam memenuhi kebutuhan pengujian aplikasi yang bersangkutan.

Tabel 3. Parameter perbandingan

Parameter	Keterangan
P.1 Waktu Eksekusi Test Case:	Parameter ini digunakan untuk membandingkan berapa lama setiap alat pengujian memerlukan waktu dalam menjalankan pengujian sesuai dengan skenario yang telah ditentukan.
P.2 Penundaan (Delay)	Parameter ini mengevaluasi kemampuan alat pengujian untuk menghentikan proses sementara selama pengujian sesuai dengan waktu yang telah ditetapkan.
P.3 Image Based Testing	Ini menilai kemampuan alat pengujian untuk menguji elemen-elemen berbasis gambar dalam kasus uji yang telah ditentukan.
P.4 Scrolling	Ini mengevaluasi apakah alat pengujian mampu melakukan pergerakan scroll saat pengujian sedang dilakukan.

P.5 Dokumentasi Hasil Eksekusi:	Parameter ini mengevaluasi dan membandingkan dokumen yang dihasilkan dari setiap alat pengujian setelah menjalankan kasus uji tertentu, misalnya kasus uji login.
P.6 Supported Languages	Parameter ini digunakan untuk membandingkan apa saja bahasa pemrograman yang didukung saat pengujian berlangsung.

Tabel 4. Perbandingan Hasil akhir Selenium Webdriver dan Katalon Studio

Parameter	Selenium Webdriver	Katalon Studio
P1	15.573 s	21.288 s
P2	✓	✓
P3	Bisa tetapi membutuhkan instalasi tambahan	Sudah bisa dari bawaannya
P4	✓	✓
P5	Tidak ada laporan	Html, EXCEL, dan XML
P6	Java, C#, Perl, Python, JavaScript, Ruby, PHP	Java/Groovy, JavaScript

1) Hasil Eksekusi dari P1 (Waktu Eksekusi TestCase)**:** Berdasarkan hasil pengujian, Selenium WebDriver terbukti lebih efisien dalam mengeksekusi test case dibandingkan dengan Katalon Studio. Rata-rata waktu yang diperlukan oleh Selenium WebDriver adalah 15,573 detik, sedangkan Katalon Studio memerlukan waktu yang lebih lama, yaitu 21,288 detik.
2) Hasil Eksekusi dari P2 (Penundaan)**:** Kedua alat pengujian mendukung fitur penundaan selama proses pengujian, tetapi penerapannya berbeda. Selenium WebDriver memerlukan impor modul tambahan bernama time untuk menerapkan penundaan, sedangkan Katalon Studio sudah memiliki fitur penundaan (delay) bawaan yang dapat digunakan langsung tanpa konfigurasi tambahan.
3) Hasil Eksekusi dari P3 (Pengujian Berbasis Gambar)**:** Dalam hal pengujian berbasis gambar, baik Selenium WebDriver maupun Katalon Studio mampu melaksanakan tugas ini. Namun, Selenium WebDriver memerlukan library atau instalasi tambahan untuk dapat melakukan pengujian gambar, sementara Katalon Studio sudah menyediakan dukungan bawaan untuk pengujian gambar secara langsung tanpa memerlukan tambahan perangkat lunak.

4) Hasil Eksekusi dari P4 (Scrolling)**:** Kedua alat ini juga dapat menangani fitur scrolling selama pengujian, sesuai dengan skenario uji yang telah ditentukan. Kemampuan ini memungkinkan kedua alat untuk mengelola tampilan halaman yang lebih besar dengan melakukan scrolling sesuai kebutuhan pengujian.
5) Hasil Eksekusi dari P5 (Dokumentasi Hasil Eksekusi)**:** Dalam hal dokumentasi hasil eksekusi, Selenium WebDriver tidak menyediakan dukungan langsung untuk pembuatan laporan. Namun, laporan dapat dihasilkan dengan menggunakan framework atau library tambahan seperti pytest. Sebaliknya, Katalon Studio secara langsung mendukung pembuatan laporan dengan kemampuan untuk menghasilkan dokumen dalam format HTML, EXCEL, dan XML tanpa memerlukan instalasi tambahan.
6) Hasil Eksekusi dari P6 (Bahasa Pemrograman yang Didukung)**:** Mengenai bahasa pemrograman, Selenium WebDriver mendukung tujuh bahasa pemrograman, yaitu Java, C#, Perl, Python, JavaScript, Ruby, dan PHP. Sementara itu, Katalon Studio mendukung tiga bahasa pemrograman, yaitu Java/Groovy dan JavaScript.

Kesimpulan dan Saran

Studi ini memberikan analisis perbandingan antara dua alat pengujian otomatis, yaitu Selenium Webdriver dan Katalon Studio, dalam konteks pengujian antarmuka pengguna grafis (GUI) pada aplikasi web Quizizz. Dengan menggunakan metode Software Testing Life Cycle (STLC), penelitian ini merancang dan melaksanakan serangkaian pengujian berdasarkan parameter-parameter tertentu, seperti waktu eksekusi test case, penundaan, pengujian berbasis gambar, scrolling, dan dokumentasi hasil eksekusi. Hasil pengujian menunjukkan bahwa Selenium Webdriver memiliki keunggulan dalam beberapa aspek, seperti waktu eksekusi test case yang lebih cepat dan kemampuan melakukan scrolling, namun memerlukan instalasi tambahan untuk pengujian berbasis gambar. Di sisi lain, Katalon Studio menonjol dalam penyediaan dokumentasi hasil eksekusi yang lengkap dan mendukung berbagai bahasa pemrograman. Kesimpulannya, pemilihan alat pengujian otomatis harus disesuaikan dengan kebutuhan pengujian spesifik dari aplikasi yang dituju. Dengan memahami karakteristik dan kemampuan masing-masing alat, pengembang dapat meningkatkan efisiensi dan efektivitas pengujian GUI, sehingga dapat menghasilkan aplikasi yang lebih berkualitas dan dapat diandalkan bagi pengguna.

Daftar Pustaka

Agustina, L., & Rusmana, I. M. (2019). Pembelajaran matematika menyenangkan dengan aplikasi kuis online quizizz. *Prosiding Sesiomadika*, 2(1a). https://journal.unsika.ac.id/sesiomadika/article/view/2249

Amany, A. (2020). Quizizz sebagai media evaluasi pembelajaran daring pelajaran matematika. *Buletin Pengembangan Perangkat Pembelajaran*, 2(2), 1-11. https://doi.org/10.23917/bppp.v2i2.13811.

Bhagat, B., Bhattacharjee, S., Ratre, S., & PHCET, R. (2020). Software testing techniques & automation tools. *Mukt Shabd Journal*, 9(5), 5957-5962.

Erol, E., & Senan, S. (2022). A comparative study for evaluating automated software testing tools. *Bilişim Teknolojileri Dergisi*, *15*(3), 301-316. https://doi.org/10.17671/gazibtd.1057380.

Gonzalez, R. S. S., Urribarri, D. K., & Larrea, M. L. (2022). Automation tools for web testing. beyond selenium. *JAIIO, Jornadas Argentinas de Informática*, *8*(3), 49-62.

Melia, S. M. S. (2023). Analisis perbandingan tools pengujian otomatis pada GUI aplikasi berbasis Web: Comparative analysis of automated testing tools on GUI Web-based applications. In *Sentimas: Seminar Nasional Penelitian dan Pengabdian Masyarakat*, 1(1), 267-273.

Nyamathulla, S., Ratnababu, P., & Shaik, N. S. (2021). A review on selenium web driver with python. *Annals of the Romanian Society for Cell Biology*, 25(4), 16760–68. http://annalsofrscb.ro.

Pelivani, E., Besimi, A., & Cico, B. (2022). An empirical study of user interface testing tools. *International Journal on Information Technologies and Security*, *14*(1), 37-48. https://www.researchgate.net/publication/358913408.

Shariff, S. M. (2019). Investigating selenium usage challenges and reducing the performance overhead of selenium-based load tests. *Master's thesis, Queen's University, Canada.*