



## **Analisis Pengujian Otomatis Antarmuka Web Pada Formulir Jotform Menggunakan Cypress**

**Sigit Wijanarko**

STMIK Antar Bangsa, Indonesia

Email: sgtwijanarko23@gmail.com

### **Abstrak**

Pengujian antarmuka web merupakan tahap penting dalam memastikan fungsionalitas aplikasi berbasis web, termasuk pada platform Software as a Service (SaaS) yang menyediakan layanan pembuatan formulir daring. Jotform merupakan salah satu platform SaaS yang banyak digunakan untuk membangun formulir online, namun karakteristiknya yang bersifat dinamis, penggunaan iframe, serta keterbatasan akses terhadap kode sumber dapat menyulitkan proses pengujian manual. Oleh karena itu, pengujian otomatis berbasis end-to-end menjadi pendekatan yang relevan untuk memverifikasi fungsionalitas antarmuka web formulir tersebut. Penelitian ini bertujuan untuk menganalisis penerapan framework Cypress dalam pengujian otomatis antarmuka web pada formulir Jotform berbasis SaaS. Metode penelitian yang digunakan adalah pendekatan kuantitatif deskriptif dengan objek penelitian berupa formulir simulasi yang dirancang menggunakan fitur multi-page, conditional logic, validasi field, dan prefill URL. Pengujian dilakukan dengan menjalankan satu skenario utama, yaitu skenario submit formulir berhasil, secara berulang pada lingkungan pengujian yang sama. Variabel yang dianalisis meliputi waktu eksekusi pengujian, stabilitas pengujian, dan kompleksitas script pengujian. Hasil penelitian menunjukkan bahwa Cypress mampu menjalankan pengujian otomatis antarmuka web pada formulir Jotform secara konsisten dengan waktu eksekusi yang relatif stabil dan tanpa kegagalan selama eksekusi berulang. Kompleksitas script pengujian meningkat seiring dengan penambahan penanganan elemen dinamis dan validasi kondisi tertentu. Hasil ini menunjukkan bahwa Cypress dapat digunakan sebagai framework pengujian otomatis yang andal untuk pengujian antarmuka web formulir berbasis SaaS.

Kata kunci: Pengujian Otomatis; Cypress; Antarmuka Web; Jotform; Perangkat Lunak sebagai Layanan

### **Abstract**

*Web interface testing is a critical stage in ensuring the functionality of web-based applications, including those delivered through Software as a Service (SaaS) platforms that provide online form services. Jotform is a widely used SaaS platform for building online forms; however, its dynamic characteristics, use of iframes, and limited access to source code can make manual testing inefficient and difficult to reproduce. Therefore, end-to-end automated testing represents a relevant approach for validating the functionality of such web form interfaces. This study aims to analyze the application of the Cypress framework for automated web interface testing on SaaS-based Jotform forms. The research employs a quantitative descriptive approach, with the research object consisting of a simulated form designed using multi-page features, conditional logic, field validation, and URL prefill mechanisms. Testing was conducted by repeatedly executing a single primary scenario, namely successful form submission, under the same testing environment. The analyzed variables include test execution time, test stability, and test script complexity. The results indicate that Cypress is able to execute automated web interface tests on Jotform forms consistently, with relatively stable execution times and no failures observed during repeated test runs. In addition, test script complexity increases as additional handling for dynamic elements and conditional validation is introduced. These findings suggest that Cypress can be considered a reliable automated testing framework for testing web interfaces of SaaS-based online forms.*

*Keywords: Automated Testing; Cypress; Web Interface; Jotform; Software as a Service*

## **PENDAHULUAN**

Seiring meningkatnya pemanfaatan aplikasi berbasis web dalam berbagai domain fungsional seperti bisnis, pendidikan, dan pemerintahan, kebutuhan akan jaminan kualitas perangkat lunak menjadi semakin penting. Pengujian perangkat lunak merupakan proses yang krusial untuk memastikan bahwa aplikasi memenuhi spesifikasi kebutuhan dan bebas dari cacat

sebelum dirilis ke pengguna akhir. Secara umum, pengujian dapat dilakukan secara manual atau otomatis. Pengujian manual melibatkan pemeriksaan langsung setiap elemen dan fungsi oleh penguji, sedangkan pengujian otomatis menggunakan skrip atau alat untuk menjalankan uji secara otomatis tanpa intervensi manusia yang intensif (Fathayanurannisa & Ranaramadhani, 2022).

Pengujian otomatis (*automation testing*) merupakan pendekatan yang efisien untuk mengurangi waktu eksekusi dan meningkatkan konsistensi pengujian, terutama pada aplikasi yang sering mengalami perubahan rilis (*frequent deployment*) (Abdillah et al., 2025; Yodi & Kristianto, 2024). Pada lingkungan pengembangan modern, terutama *Software as a Service* (SaaS), struktur aplikasi yang dinamis dan kompleksitas antarmuka pengguna menuntut strategi pengujian yang adaptif dan reliable. Jotform, sebagai salah satu platform SaaS yang banyak digunakan untuk *form building*, memberikan tantangan unik dalam pengujian karena struktur DOM yang dinamis dan keterbatasan akses terhadap kode sumber internal.

Dalam konteks *web automation testing*, berbagai framework telah dikembangkan untuk mendukung pengujian end-to-end pada aplikasi web (Bhanuwa & Jannah, 2025; Min et al., 2020; Rahayu et al., 2024). Dari sekian banyak framework yang tersedia, Cypress dipandang sebagai salah satu alat yang modern dan ringan yang dirancang khusus untuk pengujian antarmuka web berbasis JavaScript (Aidil & Jibril, 2025).

Beberapa penelitian dan laporan teknis sebelumnya telah mengevaluasi karakteristik dan efektivitas berbagai framework pengujian otomatis. Misalnya, Mobaraya dan Shahid Ali (2019) melakukan analisis teknis terhadap framework Selenium dan Cypress sebagai alat functional automation untuk pengujian aplikasi web modern. Mereka menunjukkan bahwa meskipun Selenium memiliki dukungan lintas browser yang luas, Cypress menawarkan kemudahan penggunaan dan integrasi yang lebih sederhana untuk pengujian antarmuka web. Selain itu, perbandingan antara Cypress dan Selenium juga dilaporkan secara praktikal oleh Anderson (2021) yang menekankan eksekusi skrip yang lebih cepat dan metode penulisan yang lebih sederhana pada Cypress dibandingkan dengan Selenium, yang masih bergantung pada WebDriver dalam mengontrol browser. Kajian lain oleh Khetarpal (2021) menjelaskan arsitektur dan prinsip kerja Cypress, termasuk pendekatan eksekusi dalam browser yang memberikan hasil konsisten dan penanganan waktu tunggu (*automatic waiting*) secara otomatis. Taky (2021) dalam tesisnya juga mencatat bahwa Cypress memberikan keuntungan berupa real-time reloads, fitur debug yang kuat, dan pelaporan hasil yang interaktif.

Meskipun demikian, mayoritas studi sebelumnya lebih menekankan perbandingan alat atau karakteristik teknis umum, dan belum fokus terhadap penerapan nyata dalam konteks pengujian antarmuka web SaaS berbasis formulir seperti yang disediakan oleh Jotform. Hal ini meninggalkan ruang penelitian yang jelas terkait penerapan Cypress pada objek real-world SaaS form yang memiliki elemen kompleks seperti conditional logic, multi-page flow, dan berbagai tipe input antarmuka. Oleh karena itu, penting untuk menggali lebih dalam mengenai penerapan Cypress dalam pengujian otomatis untuk formulir SaaS seperti Jotform, yang memiliki kompleksitas tersendiri dalam hal elemen antarmuka dan interaksi pengguna.

Berdasarkan tinjauan literatur terdahulu, sebagian besar kajian masih bersifat umum dan teknis pada tingkat framework saja tanpa mengeksplorasi penerapan Cypress dalam konteks pengujian antarmuka web untuk layanan SaaS tertentu seperti Jotform. Kebaruan ilmiah artikel ini terletak pada analisis penerapan Cypress dalam pengujian otomatis antarmuka web pada

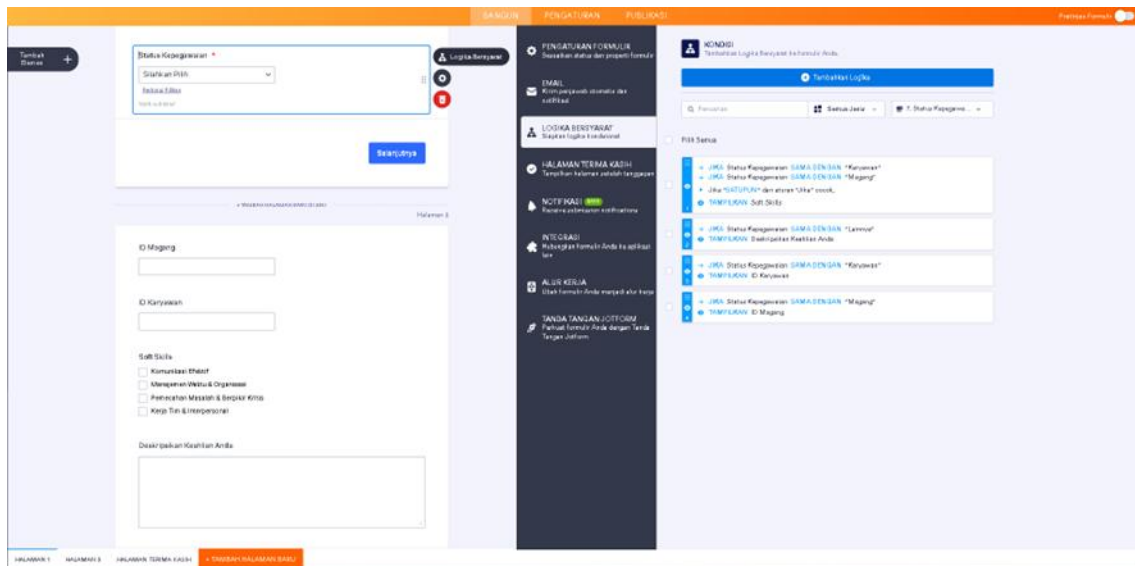
formulir SaaS (Jotform) yang mencakup pengukuran waktu eksekusi, stabilitas pengujian berulang, dan kompleksitas penulisan skrip pada skenario nyata. Oleh karena itu, tujuan penelitian ini adalah untuk menganalisis penerapan Cypress dalam pengujian otomatis antarmuka web pada formulir Jotform, mengukur waktu eksekusi pengujian otomatis, mengevaluasi stabilitas pengujian berulang, dan mengkaji kompleksitas penulisan skrip dalam konteks tersebut.

### **METODE PENELITIAN**

Penelitian ini menggunakan pendekatan kuantitatif deskriptif dengan metode eksperimen terkontrol. Pendekatan kuantitatif digunakan untuk mengukur dan menganalisis nilai numerik berupa waktu eksekusi pengujian, tingkat stabilitas pengujian, dan kompleksitas skrip pengujian otomatis. Metode eksperimen diterapkan dengan menjalankan skenario pengujian otomatis secara berulang pada objek penelitian yang sama dan dalam lingkungan pengujian yang terkontrol. Penelitian ini tidak melibatkan responden manusia karena objek yang diuji berupa sistem perangkat lunak, sehingga data diperoleh sepenuhnya dari hasil eksekusi pengujian otomatis.

Objek penelitian berupa formulir web berbasis SaaS yang dibangun menggunakan platform Jotform. Formulir dirancang secara khusus untuk mendukung pengujian otomatis dengan karakteristik sebagai berikut: 1) terdiri dari beberapa halaman (*multi-page form*), 2) memiliki validasi field wajib, 3) menerapkan *conditional logic*, 4) mendukung mekanisme *prefill* melalui parameter URL, 4) dan menghasilkan halaman konfirmasi setelah submit berhasil.

Penelitian dilaksanakan secara lokal pada lingkungan pengujian berbasis komputer personal dengan koneksi internet stabil. Seluruh pengujian dilakukan pada lingkungan yang sama untuk menjaga konsistensi hasil pengukuran. Alat utama yang digunakan dalam penelitian ini meliputi: 1) Framework Cypress versi stabil terbaru, digunakan sebagai alat utama pengujian otomatis antarmuka web. 2) Browser Google Chrome, digunakan sebagai media eksekusi pengujian otomatis. 3) Node.js, digunakan sebagai runtime untuk menjalankan Cypress. 4) Platform Jotform, digunakan untuk membangun dan mengelola formulir sebagai objek pengujian.



**Gambar 1.** Struktur Formulir Jotform dengan Multi-Page dan Conditional Logic

Alat bantu umum seperti editor teks dan sistem operasi tidak dijelaskan secara rinci karena bersifat standar dan telah umum digunakan dalam penelitian pengujian perangkat lunak. Skenario pengujian difokuskan pada satu skenario utama, yaitu skenario pengisian dan pengiriman formulir secara valid (*successful submission*). Pemilihan satu skenario utama bertujuan untuk menjaga konsistensi pengukuran dan meminimalkan variabel eksternal yang dapat memengaruhi hasil pengujian.

Skenario pengujian mencakup tahapan: 1) pemuatan halaman formulir, 2) pengisian field input sesuai ketentuan validasi, 3) aktivasi *conditional logic* berdasarkan pilihan pengguna, 4) navigasi antar halaman formulir, 5) dan pengiriman formulir hingga halaman konfirmasi ditampilkan.

Setiap skenario pengujian dijalankan secara berulang dengan jumlah iterasi yang telah ditentukan untuk memperoleh data waktu eksekusi dan stabilitas pengujian. Pengujian dilakukan menggunakan satu skenario utama, yaitu skenario pengisian dan pengiriman formulir secara valid (*successful submission*). Skenario ini dipilih karena merepresentasikan alur utama penggunaan formulir oleh pengguna akhir dan memungkinkan pengukuran kinerja pengujian secara konsisten.

**Tabel 1.** Skenario Pengujian Otomatis

| Kode | Deskripsi Skenario                    | Tujuan Pengujian                   |
|------|---------------------------------------|------------------------------------|
| S1   | Membuka halaman formulir Jotform      | Memverifikasi halaman dapat dimuat |
| S2   | Mengisi field teks wajib              | Memastikan validasi input          |
| S3   | Mengaktifkan <i>conditional logic</i> | Memastikan elemen dinamis muncul   |
| S4   | Navigasi antar halaman formulir       | Memastikan alur multi-page         |

|    |                             |                              |
|----|-----------------------------|------------------------------|
| S5 | Submit formulir             | Memastikan proses pengiriman |
| S6 | Validasi halaman konfirmasi | Memastikan submit berhasil   |

Skenario S1 hingga S6 dijalankan secara berurutan sebagai satu rangkaian pengujian end-to-end untuk mengukur efektivitas dan efisiensi pengujian menggunakan Cypress. Variabel yang dianalisis dalam penelitian ini mencakup waktu eksekusi pengujian, yang mengukur durasi yang dibutuhkan Cypress untuk menyelesaikan satu siklus pengujian end-to-end dalam satuan milidetik (ms), yang dicatat menggunakan built-in timing pada Cypress dan log hasil eksekusi. Stabilitas pengujian diukur berdasarkan konsistensi hasil eksekusi pada setiap iterasi, dengan indikator stabilitas yang berupa jumlah pengujian yang berhasil dan gagal (flaky rate). Terakhir, kompleksitas skrip pengujian dianalisis secara deskriptif berdasarkan jumlah baris kode, jumlah perintah (commands), serta tingkat kesulitan dalam menangani elemen dinamis seperti conditional logic dan validasi.

**Tabel 2. Variabel dan Metrik Pengukuran**

| Variabel           | Indikator                    | Satuan         | Metode Pengukuran          |
|--------------------|------------------------------|----------------|----------------------------|
| Waktu eksekusi     | Durasi pengujian per iterasi | Milidetik (ms) | Log waktu eksekusi Cypress |
| Stabilitas         | Jumlah pengujian berhasil    | Persentase (%) | Rasio <i>pass/fail</i>     |
| Kompleksitas skrip | Jumlah baris & perintah      | Deskriptif     | Analisis struktur skrip    |

Prosedur penelitian dilakukan dengan merancang formulir Jotform sebagai objek pengujian, menulis skrip pengujian otomatis menggunakan Cypress sesuai skenario yang telah ditentukan, serta menjalankan pengujian secara berulang dalam lingkungan yang sama. Setiap hasil eksekusi dicatat untuk dianalisis berdasarkan variabel yang telah ditetapkan. Data hasil pengujian dianalisis menggunakan metode analisis deskriptif kuantitatif. Nilai waktu eksekusi dianalisis berdasarkan nilai rata-rata dan kecenderungan stabilitas antar iterasi. Stabilitas pengujian dianalisis dengan menghitung persentase keberhasilan pengujian terhadap total iterasi yang dilakukan. Kompleksitas skrip dianalisis secara kualitatif deskriptif dengan membandingkan karakteristik skrip sederhana dan skrip kompleks. Hasil analisis disajikan dalam bentuk tabel dan grafik untuk memudahkan interpretasi hasil penelitian.

## HASIL DAN PEMBAHASAN

### Hasil Pengujian Waktu Eksekusi

Pengujian waktu eksekusi dilakukan dengan menjalankan skenario pengujian *end-to-end* pengisian dan pengiriman formulir Jotform secara valid menggunakan Cypress. Skenario yang sama dijalankan secara berulang dalam lingkungan pengujian yang terkontrol untuk memperoleh data waktu eksekusi yang konsisten.

**Tabel 3. Hasil Pengukuran Waktu Eksekusi Pengujian Otomatis**

| Iterasi | Waktu Eksekusi (ms) |
|---------|---------------------|
| 1       | 6.944               |
| 2       | 5.628               |
| 3       | 5.268               |
| 4       | 4.783               |
| 5       | 4.960               |

Berdasarkan data pada Tabel 3, diperoleh bahwa waktu eksekusi pengujian berada pada rentang yang relatif sempit dengan variasi yang kecil antar iterasi. Nilai rata-rata waktu eksekusi berada pada kisaran  $\pm 4,79$  detik. Temuan ilmiah yang diperoleh dari hasil ini adalah bahwa Cypress mampu mengeksekusi pengujian antarmuka web formulir SaaS secara konsisten dalam waktu yang stabil.

Stabilitas waktu eksekusi ini terjadi karena arsitektur Cypress yang menjalankan pengujian langsung di dalam browser, bukan melalui *remote driver* seperti pada Selenium. Pendekatan ini mengurangi latensi komunikasi antar proses dan memungkinkan Cypress melakukan sinkronisasi otomatis terhadap elemen antarmuka (*automatic waiting*), sehingga pengujian tidak bergantung pada penundaan waktu statis (*hard wait*). Fenomena ini sejalan dengan temuan Khetarpal (2021) dan Mobaraya dan Ali (2019) yang menyatakan bahwa Cypress memiliki performa eksekusi yang lebih stabil pada aplikasi web modern dengan DOM dinamis. Dengan demikian, hasil ini menjawab permasalahan penelitian pertama dan kedua terkait efektivitas Cypress dalam menjalankan pengujian otomatis dengan waktu eksekusi yang terukur dan konsisten.

### Hasil Pengujian Stabilitas

Stabilitas pengujian dianalisis berdasarkan konsistensi hasil eksekusi skenario pengujian yang dijalankan secara berulang. Indikator stabilitas ditentukan dari jumlah keberhasilan dan kegagalan pengujian (*pass* dan *fail*) pada setiap iterasi. Ringkasan hasil stabilitas pengujian disajikan pada Tabel 4.

**Tabel 4. Hasil Stabilitas Pengujian Otomatis**

| Iterasi | Status Eksekusi |
|---------|-----------------|
| 1       | Berhasil        |
| 2       | Berhasil        |
| 3       | Berhasil        |
| 4       | Berhasil        |
| 5       | Berhasil        |

Seluruh iterasi pengujian menunjukkan hasil berhasil tanpa kegagalan, sehingga *flaky rate* yang diperoleh adalah 0%. Temuan ilmiah dari hasil ini adalah bahwa pengujian otomatis menggunakan Cypress pada formulir Jotform menunjukkan tingkat stabilitas yang sangat tinggi. Stabilitas ini dapat dijelaskan secara ilmiah melalui mekanisme *automatic retry* dan *implicit waiting* yang dimiliki Cypress. Framework ini secara otomatis menunggu hingga kondisi elemen terpenuhi sebelum melakukan interaksi, sehingga mampu mengurangi kegagalan yang disebabkan oleh keterlambatan pemuatan elemen atau perubahan DOM secara dinamis. Hal ini sangat relevan pada platform SaaS seperti Jotform yang banyak menggunakan pemuatan elemen berbasis JavaScript.

Temuan ini sejalan dengan penelitian Taky (2021) yang menyatakan bahwa Cypress memiliki tingkat *flakiness* yang lebih rendah dibandingkan framework pengujian berbasis WebDriver. Selain itu, hasil ini juga konsisten dengan Anderson (2021) yang menyebutkan bahwa Cypress unggul dalam menjaga determinisme hasil pengujian pada aplikasi web modern.

Walaupun tidak ditemukan kegagalan pengujian, hasil ini tetap diterima secara ilmiah, karena stabilitas nol kegagalan menunjukkan bahwa konfigurasi pengujian, skenario, dan lingkungan telah dikontrol dengan baik. Namun demikian, hasil ini juga mengindikasikan bahwa pengujian dilakukan pada skenario yang terdefinisi dengan jelas dan dalam kondisi lingkungan yang stabil.

### **Analisis Kompleksitas Skrip Pengujian**

Kompleksitas skrip pengujian dianalisis secara deskriptif dengan membandingkan karakteristik skrip pengujian sederhana dan skrip pengujian kompleks. Skrip sederhana mencakup pengisian field dasar dan pengiriman formulir, sedangkan skrip kompleks mencakup penanganan *conditional logic*, navigasi antar halaman, dan validasi kondisi tertentu. Temuan ilmiah yang diperoleh adalah bahwa kompleksitas skrip pengujian meningkat secara signifikan seiring bertambahnya jumlah kondisi logika dan elemen dinamis yang harus ditangani. Peningkatan kompleksitas ini terlihat dari bertambahnya jumlah baris kode, penggunaan *custom commands*, serta kebutuhan pengelolaan state antar halaman formulir.

Fenomena ini terjadi karena pengujian antarmuka web pada aplikasi SaaS tidak hanya bergantung pada keberadaan elemen, tetapi juga pada kondisi tertentu yang memengaruhi visibilitas dan interaksi elemen tersebut. Meskipun demikian, Cypress menyediakan sintaks yang relatif ringkas dan mekanisme penanganan DOM yang eksplisit, sehingga kompleksitas skrip masih dapat dikelola dengan baik dibandingkan framework lain yang memerlukan konfigurasi tambahan.

Hasil ini mendukung temuan García et al. (2021) yang menyatakan bahwa kompleksitas pengujian otomatis lebih dipengaruhi oleh kompleksitas antarmuka aplikasi dibandingkan oleh framework yang digunakan. Dengan demikian, Cypress dapat dianggap sebagai framework yang cukup efisien dalam menangani kompleksitas pengujian antarmuka web pada formulir berbasis SaaS.

Berdasarkan hasil dan analisis yang telah dilakukan, dapat disimpulkan bahwa seluruh permasalahan penelitian yang dirumuskan pada bagian pendahuluan telah terjawab. Cypress terbukti dapat diterapkan secara efektif untuk pengujian otomatis antarmuka web pada formulir Jotform, dengan waktu eksekusi yang stabil, tingkat stabilitas pengujian yang tinggi, serta kompleksitas skrip yang masih dapat dikelola. Secara keseluruhan, temuan ilmiah dalam penelitian ini menunjukkan bahwa Cypress merupakan solusi yang andal untuk pengujian otomatis antarmuka web pada aplikasi berbasis SaaS, khususnya pada layanan formulir daring dengan karakteristik dinamis seperti Jotform.

### **KESIMPULAN**

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa penerapan framework Cypress dalam pengujian otomatis antarmuka web pada formulir Jotform berbasis Software as a Service (SaaS) dapat dilakukan secara efektif dan andal. Cypress mampu menjalankan pengujian end-to-end pada formulir dengan karakteristik

dinamis, seperti multi-page form dan conditional logic, tanpa memerlukan akses langsung terhadap kode sumber aplikasi.

Temuan ilmiah utama dalam penelitian ini menunjukkan bahwa Cypress memberikan waktu eksekusi pengujian yang stabil pada pengujian berulang, yang mengindikasikan bahwa mekanisme sinkronisasi otomatis dan eksekusi langsung di dalam browser mampu meminimalkan variabilitas waktu dan kegagalan pengujian. Selain itu, tingkat stabilitas pengujian yang tinggi tanpa kegagalan menunjukkan bahwa Cypress dapat mengurangi potensi flaky test pada pengujian antarmuka web berbasis SaaS.

Dari sisi pengembangan pengujian, penelitian ini juga menemukan bahwa kompleksitas skrip pengujian meningkat seiring dengan bertambahnya elemen dinamis dan logika kondisional pada formulir. Namun demikian, kompleksitas tersebut masih dapat dikelola dengan baik karena struktur sintaks Cypress yang ringkas dan dukungan bawaan terhadap penanganan elemen antarmuka yang dinamis.

Secara keseluruhan, hasil penelitian ini menjawab tujuan penelitian dengan menunjukkan bahwa Cypress merupakan framework pengujian otomatis yang layak digunakan untuk pengujian antarmuka web pada formulir daring berbasis SaaS. Sebagai arah pengembangan penelitian selanjutnya, disarankan untuk melakukan perbandingan langsung dengan framework pengujian otomatis lain, serta menguji variasi skenario pengujian yang lebih kompleks, seperti pengujian validasi negatif dan pengujian lintas peramban, guna memperoleh gambaran kinerja yang lebih komprehensif.

#### **DAFTAR PUSTAKA**

- Abdillah, R., Hermawan, R., Hermawansyah, W., Agustin, D. P., Adkha, I., & Alam, N. (2025). Analisis dan pengujian perangkat lunak sistem informasi pembayaran sekolah dengan metode pengukuran kualitas SQuaRE. *Jurnal Penelitian Rumpun Ilmu Teknik*, 4(2), 36–45.
- Aidil, M., & Jibril, M. (2025). Implementasi pengujian otomatis end-to-end pada website konseling mahasiswa (SiKosa) menggunakan Cypress. *TEKNOFILE: Jurnal Sistem Informasi*, 3(12), 929–939.
- Alégroth, E., Nass, M., & Olsson, H. H. (2016). JUnit test smells as indicators of poor test quality. *Software Quality Journal*, 24(3), 545–577. <https://doi.org/10.1007/s11219-016-9320-5>
- Anderson, S. (2021). *Selenium vs Cypress in 2022*. Charlotte Stories. <https://www.charlottestories.com/selenium-vs-cypress-in-2022/>
- Bhanuwa, M. A. L., & Jannah, M. (2025). Analisis perbandingan automation testing Katalon Studio dan Selenium IDE pada website e-commerce menggunakan metode end-to-end (Studi kasus: Enterkomputer.com). *POSITIF: Jurnal Sistem dan Teknologi Informasi*, 11(1).
- Cypress.io. (2024). *Cypress documentation and official guides*. <https://www.cypress.io/>
- Fathayanurannisa, & Ranaramadhani. (2022). *Menggunakan Cypress untuk web testing*. Medium. <https://medium.com/@fathayanurannisa/menggunakan-cypress-untuk-web-testing-6b5eea1888d0>
- García, B., Serrano, M. A., & Piattini, M. (2021). Software testing automation: A systematic literature review. *IEEE Access*, 9, 473–497.

- <https://doi.org/10.1109/ACCESS.2020.3047278>
- Garousi, V., Felderer, M., & Mäntylä, M. V. (2020). The need for multivocal literature reviews in software engineering. *Information and Software Technology*, 124, 106–127. <https://doi.org/10.1016/j.infsof.2020.106307>
- Khetarpal, A. (2021). *What is Cypress: Introduction and architecture*. ToolsQA. <https://www.toolsqa.com/cypress/what-is-cypress/>
- Min, J. L., Istiqomah, A., & Rahmani, A. (2020). Evaluasi penggunaan manual dan automated software testing pada pelaksanaan end-to-end testing. *JTT (Jurnal Teknologi Terapan)*, 6(1), 18–26. <https://doi.org/10.31884/jtt.v6i1.256>
- Mobaraya, F., & Ali, S. (2019). Technical analysis of Selenium and Cypress as functional automation framework for modern web application testing. *Computer Science & Information Technology (CS & IT)*, 9(3), 21–30. <https://doi.org/10.5121/csit.2019.90303>
- Nguyen, T. H., Nguyen, V. T., & Pham, D. H. (2021). A study on automated web application testing using modern testing frameworks. *International Journal of Advanced Computer Science and Applications*, 12(6), 410–418. <https://doi.org/10.14569/IJACSA.2021.0120648>
- Rahayu, S. S., Wahyudin, W., Firmansyah, D. A., & Susanti, S. (2024). Analisis penggunaan tools automation testing pada aplikasi: Systematic literature review. *REMIK: Riset dan E-Jurnal Manajemen Informatika Komputer*, 8(1), 106–116.
- Rahman, M. M., & Gao, J. (2019). Automated testing of web applications: A survey. *International Journal of Software Engineering and Its Applications*, 13(2), 1–15.
- Roeva, O. (2012). Real-world applications of genetic algorithm. In *Proceedings of the International Conference on Chemical and Material Engineering*. Department of Chemical Engineering, Diponegoro University.
- Santos, E., Esteves, M., & Faria, J. P. (2020). End-to-end testing of web applications: A comparative study of testing frameworks. *Journal of Web Engineering*, 19(6), 553–580. <https://doi.org/10.13052/jwe1540-9589.1964>
- Taky, M. T. (2021). *Automated testing with Cypress* (Undergraduate thesis). Haaga-Helia University of Applied Sciences.
- Wyrich, M., & Bogner, J. (2021). Automation testing challenges in modern web applications. *Software Engineering Notes*, 46(4), 15–22. <https://doi.org/10.1145/3470770.3470775>
- Yodi, A., & Kristianto, R. P. (2024). Analysis of early childhood quiz application evaluation through manual and automation testing using black box and Espresso. *Jurnal Sistem Informasi dan Teknologi*.
- Zhang, Z., Jiang, Y., & Zhang, C. (2022). Reliability analysis of automated UI testing in web-based systems. *Journal of Systems and Software*, 187, 111–126. <https://doi.org/10.1016/j.jss.2022.111126>



**This work is licensed under a Creative Commons Attribution-ShareAlike 4.0**

**International License**