

IMPLEMENTASI QUALITY APPLICATION PADA SOFTWARE TESTING DI INDUSTRI KESEHATAN PT X

IMPLEMENTATION OF QUALITY APPLICATION IN SOFTWARE TESTING IN THE HEALTHCARE INDUSTRY AT PT X

Grace Angelica¹, Jap Tji Beng^{2*}, Intan Ardiani Putri³, Graciella Devina Buntoro⁴,
Sri Tiatri⁵, Vienchenzia Oeyta Dwitama Dinatha⁶, Rahmiyana Nurkholiza⁷

Fakultas Teknologi Informasi, Program Studi Sarjana Sistem Informasi, Universitas Tarumanagara,
Jakarta^{1,2,3,4}

Fakultas Psikologi, Program Studi Psikologi, Universitas Tarumanagara, Indonesia⁵
Faculty of Business & Communications, Doctor of Business Administration Programme, INTI
International University, Malaysia⁶

Binus Graduate Program, Universitas Bina Nusantara, Jakarta, Indonesia⁷
Laboratorium Kognisi Edukasi dan Inovasi Teknologi, Universitas Tarumanagara, Indonesia^{2,5,6,7}
t.jap@untar.ac.id²

ABSTRACT

Software quality is a crucial aspect in the healthcare industry because the failure of support systems can directly impact management efficiency and patient services. This research aims to implement Quality Application through Software Quality Assurance (SQA) strategies using automation testing methods at PT X. The primary issue addressed is the lack of efficiency in manual testing, which is time-consuming and prone to human error. The research method used is a descriptive case study by analyzing the optimization of testing duration through the implementation of the Playwright framework. The results show that the implementation of automation testing is able to reduce the execution duration from 45 minutes to 14,1 minutes. This finding proves an increase in time efficiency of 68,7%. The conclusion of this research is that the transition to automation testing with a module-based structure is effective in improving the speed, consistency, and reliability of healthcare information systems sustainably.

Keywords : Automation Testing, Playwright, Quality Application, Regression Testing, Healthcare Industry.

ABSTRAK

Kualitas *software* merupakan aspek krusial dalam industri kesehatan karena kegagalan sistem pendukung dapat berdampak langsung terhadap efisiensi manajemen dan layanan pasien. Penelitian ini bertujuan untuk mengimplementasikan *Quality Application* melalui strategi *Software Quality Assurance* (SQA) dengan metode *automation testing* di PT X. Masalah utama yang diangkat adalah kurangnya efisiensi pengujian manual yang memakan waktu lama dan rentan terhadap kesalahan manusia. Metode penelitian yang digunakan adalah studi kasus kualitatif deskriptif dengan menganalisis optimalisasi durasi pengujian melalui implementasi *framework* Playwright. Hasil penelitian menunjukkan bahwa implementasi pengujian *automation* mampu mereduksi durasi eksekusi dari 45 menit menjadi 14,1 menit. Temuan ini membuktikan adanya peningkatan efisiensi waktu sebesar 68,7%. Simpulan dari penelitian ini adalah transisi ke pengujian *automation* dengan struktur berbasis modul efektif dalam meningkatkan kecepatan, konsistensi, dan reliabilitas sistem informasi kesehatan secara berkelanjutan.

Kata Kunci: Automation Testing, Playwright, Quality Application, Regression Testing, Industri Kesehatan.

PENDAHULUAN

Perkembangan teknologi informasi yang pesat mendorong perusahaan untuk mengembangkan sistem informasi yang tidak hanya fungsional, tetapi juga memiliki kualitas yang tinggi. Sistem berbasis *web* dan teknologi digital telah diterapkan dalam berbagai bidang untuk meningkatkan efisiensi dan kualitas

pelayanan kepada pengguna (Leovin et al., 2020; Wangi et al., 2020). Oleh karena itu, perancangan aplikasi yang baik menjadi salah satu faktor penting agar sistem dapat memenuhi kebutuhan pengguna secara optimal (Anggie et al., 2023). Perancangan aplikasi yang baik melibatkan penyusunan struktur sistem, alur kerja, dan fitur yang sesuai dengan kebutuhan pengguna agar

dapat mendukung kinerja sistem secara efektif. Namun, perancangan aplikasi yang baik tidak cukup untuk menjamin kualitas sistem secara keseluruhan tanpa adanya proses pengujian yang memadai.

Meskipun sistem informasi semakin berkembang pesat, kualitas *software* tetap menjadi masalah utama. Seiring dengan berkembangnya teknologi, sistem informasi menjadi semakin kompleks sehingga memerlukan proses pengujian yang lebih optimal untuk memastikan kualitasnya (Lusiana et al., 2020). Pengujian yang dilakukan secara manual seringkali membutuhkan waktu yang lama, rentan terhadap kesalahan manusia, serta kurang efisien dalam menangani pengujian dengan skala yang besar (Kamir, 2025). Pengelolaan *bugs* yang kurang terstruktur dapat mempengaruhi kualitas sistem karena kesalahan yang tidak terdokumentasi dengan baik berpotensi mengganggu fungsi aplikasi (Saputra et al., 2025). Permasalahan ini menjadi semakin krusial pada industri kesehatan, karena kesalahan atau kegagalan sistem pendukung dapat berdampak langsung terhadap efisiensi manajemen dan layanan yang diberikan (Suryandari et al., 2024). Oleh karena itu, diperlukan suatu pendekatan yang mampu meningkatkan efektivitas dan efisiensi dalam proses pengujian *software*.

Salah satu pendekatan yang digunakan untuk menjamin kualitas *software* adalah penerapan *Software Quality Assurance* (SQA) dalam tahapan pengembangan dan pengujian sistem. SQA merupakan serangkaian strategi dan teknik yang digunakan untuk memastikan kualitas *software*, sehingga mampu memberikan tingkat kepercayaan terhadap hasil akhir serta menjamin konsistensi dan reabilitas sistem (Wambua & Maake, 2022). Dalam lingkup yang lebih luas, SQA mencakup *software quality control* yang berfokus pada pemenuhan kebutuhan sistem, serta *software testing* yang bertujuan untuk menemukan *defect* pada aplikasi. Penerapan SQA dilakukan secara terstruktur dalam setiap tahapan

pengembangan sistem untuk memastikan kualitas *software* tetap terjaga. Oleh karena itu, SQA menjadi dasar penting dalam mendukung proses pengujian agar lebih efektif dan efisien.

Dalam penerapannya di PT X, *Software Quality Assurance* diwujudkan dalam bentuk *Quality Application*, yaitu penerapan prinsip-prinsip kualitas secara langsung pada proses pengembangan dan pengujian sistem. *Quality Application* berperan dalam memastikan bahwa aplikasi yang dikembangkan memenuhi standar kualitas melalui pengujian fungsional dan non-fungsional. Pengujian fungsional dilakukan melalui penyusunan *test case*, pelaksanaan pengujian, serta pengelolaan *bugs*, sedangkan pengujian non-fungsional dilakukan untuk menilai aspek kinerja dan keamanan sistem.

Untuk mengatasi keterbatasan pengujian manual, digunakan metode yang lebih efektif, salah satunya melalui *automation testing*. *Automation testing* mampu meningkatkan efisiensi, konsistensi, serta cakupan pengujian, serta menghasilkan proses pengujian yang lebih stabil dibandingkan metode manual. Hal ini sejalan dengan penelitian sebelumnya yang menunjukkan bahwa penggunaan *automation testing* dapat mengurangi waktu eksekusi pengujian serta meningkatkan efisiensi dalam proses pengujian *software* (Wati et al., 2024).

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk menganalisis penerapan *automation testing* dalam proses pengujian *software*. Fokus penelitian ini meliputi identifikasi proses pengujian yang dilakukan serta analisis hasil pengujian untuk menggambarkan pelaksanaan pengujian *software* secara bertahap. Adapun manfaat dari penelitian ini adalah memberikan pemahaman mengenai penerapan *automation testing* dalam praktik nyata serta menjadi referensi dalam pengembangan proses pengujian *software* yang lebih terstruktur dan berkelanjutan.

METODE

Penelitian ini menggunakan metode kualitatif deskriptif dengan pendekatan studi kasus untuk menganalisis penerapan *automation testing* dalam proses pengujian *software*. Objek penelitian berupa *website meeting* yang digunakan di PT X. Data penelitian diperoleh dari dokumentasi proses pengujian yang mencakup jadwal pelaksanaan *testing* dan aktivitas pengujian, serta dari implementasi *automation testing* pada beberapa modul yang dipilih. Pemilihan modul dilakukan berdasarkan tingkat penggunaan, kompleksitas pengujian, serta kemampuannya dalam merepresentasikan proses pengujian secara umum. Data tersebut yang kemudian dijadikan dasar dalam pelaksanaan analisis pada penelitian ini.

Berdasarkan data yang telah diperoleh, penelitian ini menggunakan beberapa bahan dan peralatan untuk mendukung proses pengujian. Bahan penelitian meliputi *test case*, *script automation testing*, serta data hasil pengujian berupa jumlah skenario pengujian, status hasil (*pass/fail*), dan durasi eksekusi. Peralatan yang digunakan dalam penelitian ini berupa *software automation testing* yaitu Playwright untuk menjalankan skenario pengujian pada aplikasi berbasis *web*, yang dipilih karena mendukung pengujian *end-to-end* serta mampu mengeksekusi pengujian secara efisien (Talakola, 2024).

Tahapan penelitian dilakukan secara sistematis yang mencakup pemilihan modul, penyusunan *test case*, pengembangan *script automation testing*, pelaksanaan pengujian, serta pengumpulan dan analisis data hasil pengujian. Analisis data dilakukan untuk menggambarkan hasil pengujian serta memberikan pemahaman terhadap pelaksanaan pengujian *software* yang dilakukan secara bertahap. Alur tahapan penelitian secara keseluruhan disajikan pada Gambar 1.



Gambar 1. Alur Tahapan Penelitian

Pada tahap penyusunan *test case*, skenario pengujian dibuat berdasarkan kebutuhan dan fungsi sistem yang akan diuji. Penyusunan skenario pengujian pada penelitian ini menggunakan metode *Black Box Testing* dengan teknik *Equivalence Partitioning*, karena pengujian difokuskan pada fungsionalitas sistem tanpa memperhatikan struktur kode internal aplikasi. Melalui teknik tersebut, data input dikelompokkan ke dalam beberapa partisi untuk mengidentifikasi kondisi valid dan invalid sehingga pengujian dapat dilakukan secara lebih terstruktur dan efisien (Aurelius, 2026; Maspupah, 2024).

HASIL DAN PEMBAHASAN

Pengujian dilakukan secara bertahap dari bulan Agustus 2025 hingga Januari 2026. Dalam prosesnya, pengujian dilaksanakan melalui beberapa tahap pengujian parsial yang kemudian dilanjutkan dengan pengujian menyeluruh (*full testing*) sebelum sistem diimplementasikan. Proses pengujian dimulai dari tahap *QA Test* hingga *QA Full Test* untuk memastikan setiap modul berjalan sesuai kebutuhan sistem. Pelaksanaan pengujian secara berulang selama pengembangan aplikasi menunjukkan adanya kebutuhan terhadap proses *regression testing* untuk memastikan perubahan sistem tidak mempengaruhi fungsi yang telah berjalan sebelumnya (Banjarnahor et al., 2025). Dalam kondisi tersebut, penerapan *automation testing* membantu proses *regression testing* menjadi lebih efisien karena skenario pengujian dapat dijalankan kembali secara konsisten tanpa harus melakukan pengujian manual secara berulang (Putri et al., 2025).

Berdasarkan proses pengujian yang sudah dilakukan, *automation testing*

diterapkan pada beberapa modul utama dalam aplikasi, yaitu *Home*, *MoM*, *Download/Upload*, *Delegation*, *Change PIC*, *Change Scripter*, dan *FAQ*. Hasil pengujian *automation testing* pada setiap modul ditunjukkan pada Tabel 1.

Tabel 1. Hasil Pengujian Automation Testing

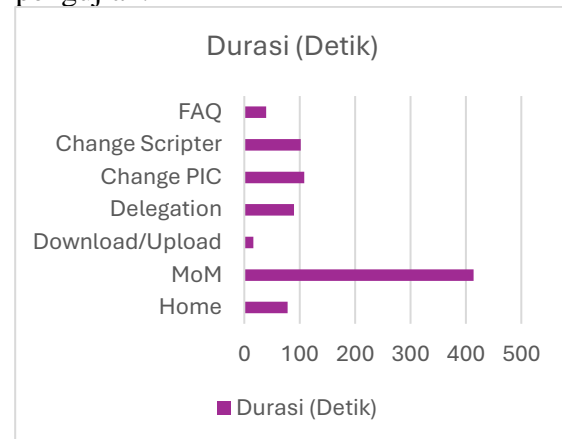
| No. | Modul | Jumlah Test Case | Pass | Fail | Durasi (detik) |
|-----|-----------------|------------------|------|------|----------------|
| 1 | Home | 4 | 4 | 0 | 78 detik |
| 2 | MoM | 19 | 19 | 0 | 414 detik |
| 3 | Download/Upload | 2 | 2 | 0 | 16.5 detik |
| 4 | Delegation | 5 | 5 | 0 | 90 detik |
| 5 | Change PIC | 8 | 8 | 0 | 108 detik |
| 6 | Change Scripter | 7 | 7 | 0 | 102 detik |
| 7 | FAQ | 2 | 2 | 0 | 39.25 detik |

Berdasarkan Tabel 1, seluruh 47 skenario *test case* yang dijalankan menghasilkan status *pass* tanpa ditemukan *fail*. Hal ini menunjukkan bahwa *script automation* mampu berjalan dengan baik dan sesuai dengan alur sistem yang diuji. Hasil pengujian (*pass*) secara keseluruhan tidak secara otomatis menunjukkan kualitas sistem yang sempurna, melainkan mengindikasikan bahwa skenario yang diuji sesuai dengan kondisi atau *requirements* yang ditulis dalam *test case*.

Jumlah *test case* pada setiap modul menunjukkan variasi yang cukup signifikan. Modul *MoM* memiliki jumlah *test case* terbanyak yaitu 19 skenario atau sekitar 40% dari total pengujian, yang mengindikasikan bahwa modul tersebut memiliki kompleksitas dan cakupan fungsi yang lebih tinggi dibandingkan modul lainnya. Sementara itu, modul *Change PIC* memiliki 8 dan *Change Scripter* memiliki 7 skenario, sedangkan modul dengan cakupan fungsi yang lebih sederhana seperti *Download/Upload* dan *FAQ* hanya memiliki 2 skenario. Variasi ini menunjukkan bahwa kebutuhan pengujian sangat dipengaruhi oleh kompleksitas fitur serta peran modul dalam sistem secara keseluruhan.

Selain jumlah skenario pengujian, durasi eksekusi *automation testing* pada

setiap modul juga dianalisis untuk melihat efisiensi proses pengujian, seperti yang ditunjukkan pada Gambar 2. Durasi eksekusi yang disajikan pada Tabel 1 dan Gambar 2 diperoleh dari hasil pengujian yang dijalankan sebanyak 2–3 kali pada setiap modul. Nilai yang ditampilkan merupakan rata-rata dari beberapa kali eksekusi tersebut untuk mengurangi variasi waktu yang terjadi selama proses pengujian.



Gambar 2. Durasi Eksekusi Automation Testing

Berdasarkan Gambar 2, modul *MoM* memiliki waktu eksekusi paling lama yaitu 414 detik, sedangkan modul *Download/Upload* memiliki waktu eksekusi paling singkat yaitu 16,5 detik. Modul lain seperti *Change PIC* dan *Change Scripter* memiliki durasi yang relatif berdekatan yaitu masing-masing 108 detik dan 102 detik. Umumnya, modul dengan jumlah *test case* yang lebih banyak cenderung memiliki waktu eksekusi yang lebih lama, namun hubungan tersebut tidak selalu sebanding. Hal ini menunjukkan bahwa waktu eksekusi tidak hanya dipengaruhi oleh jumlah *test case*, tetapi juga oleh kompleksitas alur proses dalam setiap modul.

Analisis efisiensi dalam penelitian ini dilakukan dengan membandingkan durasi pengujian manual dan pengujian *automation*. Pendekatan ini merujuk pada metode *Time Cost Trade Off* yang mengukur efisiensi berdasarkan persentase pengurangan durasi dari kondisi normal (Mochdara et al., 2025). Dalam konteks

pengujian *software* di PT X, waktu normal direpresentasikan sebagai durasi pengujian manual yang menjadi tolak ukur standar pengerjaan saat ini. Sementara itu, waktu percepatan direpresentasikan sebagai durasi eksekusi pengujian *automation*. Efisiensi waktu dihitung menggunakan rumus sebagai berikut:

$$\text{Efisiensi Waktu (\%)} = \frac{\text{Waktu Normal} - \text{Waktu Percepatan}}{\text{Waktu Normal}} \times 100\%$$

Berdasarkan data yang diperoleh, waktu normal (manual) ditetapkan sebesar 45 menit (2.700 detik) dan waktu percepatan (otomatis) sebesar 14,1 menit (847,75 detik). Maka, perhitungan efisiensinya adalah sebagai berikut:

$$Et = \frac{45 - 14,1}{45} \times 100\% = 68,7\%$$

Hasil perbandingan durasi eksekusi serta tingkat efisiensi waktu yang dihasilkan disajikan pada Tabel 2.

Tabel 2. Hasil Perbandingan Durasi Eksekusi

| Metode Pengujian | Total Durasi Eksekusi | Keterangan |
|-----------------------------|-----------------------|---------------------|
| Pengujian Manual | 45,0 Menit | Waktu Normal |
| Pengujian <i>Automation</i> | 14,1 Menit | Waktu Percepatan |
| Total Efisiensi Waktu | 30,9 Menit | 68,7% Lebih Efisien |

Berdasarkan Tabel 2, terlihat bahwa penggunaan Playwright mampu menghemat waktu pengujian secara signifikan sebesar 68,7%. Penghematan hampir 31 menit pada setiap siklus pengujian menunjukkan bahwa *automation testing* mampu membantu proses pengujian menjadi lebih efisien, khususnya pada pengujian yang dilakukan secara berulang.

Efisiensi waktu tersebut didukung oleh pendekatan pengujian berbasis modul yang digunakan dalam penelitian ini. Struktur pengujian berbasis modul memberikan fleksibilitas dalam pelaksanaan *automation testing* karena setiap modul dapat diuji secara independen

sesuai kebutuhan pengembangan sistem. Pendekatan ini memudahkan proses identifikasi kesalahan pada bagian tertentu dari aplikasi serta mendukung pelaksanaan *regression testing* tanpa harus menjalankan seluruh skenario pengujian secara keseluruhan.

Temuan dalam penelitian ini menunjukkan bahwa penerapan *automation testing* yang dikombinasikan dengan struktur pengujian berbasis modul dapat membantu proses pengujian *software* menjadi lebih terstruktur dan efisien. Selain mempercepat durasi eksekusi, pendekatan tersebut juga mendukung pelaksanaan *regression testing* secara konsisten selama proses pengembangan sistem berlangsung.

SIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa penerapan *Quality Application* melalui strategi *Software Quality Assurance* (SQA) yang terstruktur di PT X terbukti efektif dalam menjaga konsistensi dan reliabilitas sistem informasi kesehatan. Implementasi *automation testing* menggunakan *framework* Playwright memberikan peningkatan efisiensi waktu yang sangat signifikan, di mana total durasi pengujian berkurang drastis dari 45 menit menjadi hanya 14,1 menit, atau mencapai tingkat efisiensi sebesar 68,7%. Penggunaan struktur pengujian berbasis modul juga memberikan fleksibilitas tinggi dalam mendeteksi kesalahan secara independen dan terorganisir. Secara keseluruhan, transisi dari metode manual ke *automation* mampu meminimalkan risiko *human error* serta memberikan kepastian kualitas *software* yang berkelanjutan, yang sangat krusial bagi industri kesehatan dengan tingkat kompleksitas sistem yang tinggi.

DAFTAR PUSTAKA

- Anggie, A. Y., & Beng, J. T. (2023). Perancangan Aplikasi Berbasis Web Untuk Pemesanan Produk Eksterior dan Interior pada Bengkel Las

- Krisna. *Jurnal Ilmu Komputer dan Sistem Informasi*, 11(1).
- Aurelius, Z. F. J. (2026). Evaluasi Kualitas Perangkat Lunak dengan Metode Black Box Testing dan System Usability Scale (Sus) pada Siuber Universitas Palangka Raya. *Jurnal Informatika dan Teknik Elektro Terapan*, 14(1).
<https://doi.org/10.23960/jitet.v14i1.8386>
- Banjarnahor, R., Bachtiar, Y., & Sarwandianto, A. (2025). Digitalisasi Proses Pengujian Aplikasi Berbasis Analisis Log Error Dengan Metode Regression Testing Di Pt Datacaraka Solusindo. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(3), 4824-4830.
- Leovin, A., Beng, J. T., & Dewayani, E. (2020). Business to business e-commerce sales system using web-based quotation: A case study on company x. *IOP Conference Series: Materials Science and Engineering*, 1007(1).
<https://doi.org/10.1088/1757-899X/1007/1/012156>
- Lusiana, F., Beng, J. T., & Wasino. (2020). Grouping of tourism objects using geotagged photo with hierarchical clustering method in Bantul and Sleman. *IOP Conference Series: Materials Science and Engineering*, 852(1). <https://doi.org/10.1088/1757-899X/852/1/012166>
- Maspupah, A. (2024). Literature Review: Advantages And Disadvantages Of Black Box And White Box Testing Methods. *Jurnal Techno Nusa Mandiri*, 21(2), 151–162.
<https://doi.org/10.33480/techno.v21i2.5776>
- Mochdara, I., Adjam, I., Altarans, I., & Muhammad, A. H. (2025). Analisis Efisiensi Waktu Dan Biaya Pada Pelaksanaan Proyek Konstruksi Dengan Penambahan Jam Kerja Dan Penambahan Tenaga Kerja Menggunakan Metode Time Cost Trade Off. *Jurnal Ilmiah Wahana Pendidikan*, 11(11), 227–243.
- Putri, Y. F., Irianto, A. B. P., & Sharma, S. (2025). Comparison of automatic and manual regression testing on mobile application health technology with black box testing method. *Indonesian Journal of Information Systems*, 7(2), 218-230.
<https://doi.org/10.24002/ijis.v7i2.6850>
- Kamisr, R. A. (2025). *Studi Komparatif Pengujian Manual Dan Pengujian Otomatis Dengan Cypress Pada Website*. In *EMPIRIS: Jurnal Sains, Teknologi dan Kesehatan*, 2(2), 354–364. <https://doi.org/10.62335>
- Saputra, E., Jemakmun, J., Wydyanto, W., & Erlansyah, D. (2025). Optimasi Manajemen Bug dan Desain UI/UX Studi Kasus Payoprint Support Apps. *Jurnal Teknologi Sistem Informasi*, 6(1), 36–47.
<https://doi.org/10.35957/jtsi.v6i1.9271>
- Suryandari, P. I., Hariyanto, S., Roeswara, E. R., & Sanjaya, G. Y. (2024). Implementasi Kebutuhan dan Uji Kelayakan Aplikasi SMARTHEALTH di Klinik Dokter Keluarga Korpagama. *Journal of Information Systems for Public Health*, 9(2), 74-83.
[10.22146/jisph.72280](https://doi.org/10.22146/jisph.72280)
- Talakola, S. (2024). Automated end to end testing with Playwright for React applications. *International Journal of Emerging Research in Engineering and Technology*, 5, 38–47.
<https://doi.org/10.63282/3050-922x.ijeret-v5i1p106>
- Wambua, A. W., & Maake, B. M. (2022). *Characterizing Software Quality Assurance Practices in Kenya*. 8, 22–28.
<https://doi.org/10.15282/ijsecs.8.1.2022.3.0093>
- Wangi, V. H., Beng, J. T., & Wasino. (2020). Start to end: Recommended travel routes based on tourist

preference. *IOP Conference Series: Materials Science and Engineering*, 852(1). <https://doi.org/10.1088/1757-899X/852/1/012163>

Wati, N. L. P. M., Asana, I. M. D. P., Putri, N. W. S., Atmaja, K. J., & Sudipa, I. G. I. (2024). Comparison of Automation Testing On Card Printer Project Using Playwright And Selenium Tools. *Architecture and High Performance Computing*, 6(3). <https://doi.org/10.47709/cnipc.v6i3.4362>