

Software Testing in E-Commerce: A Comparison Between Manual and Automated Testing Using Katalon Studio and Python

Brian Duen Rakly^{1*}, Widyastuti Andriyani²

^{1,2}Magister Teknologi Informasi, MTI, Universitas Teknologi Digital Indonesia, Yogyakarta, 55198, Indonesia

Informasi Artikel

Diterima : 17 Februari 2025
Revisi : 29 Februari 2025
Publikasi : 20 Maret 2025

Kata Kunci:

Pengujian perangkat lunak
Pengujian manual
Pengujian otomatis
Katalon studio
Python

ABSTRAK

Pengujian perangkat lunak adalah elemen penting dalam pengembangan perangkat lunak untuk memastikan kualitas dan keandalan. Penelitian ini membandingkan pengujian manual dan otomatis menggunakan alat seperti Katalon Studio dan Python. Pengujian manual efektif untuk skenario yang memerlukan penilaian manusia, seperti evaluasi pengalaman pengguna (UX). Sebaliknya, pengujian otomatis lebih efisien untuk tugas rutin dan berulang, mengurangi kesalahan manusia dan mempercepat proses. Studi ini mengevaluasi efektivitas, efisiensi, dan biaya kedua metode dalam konteks pengujian perangkat lunak e-commerce. Hasilnya menunjukkan bahwa pengujian manual lebih unggul dalam mendeteksi cacat sebelum rilis, sementara pengujian otomatis lebih hemat biaya dan waktu untuk pengujian berulang. Panduan ini membantu pengembang dalam memilih metode pengujian yang sesuai dengan kebutuhan proyek mereka.

ABSTRACT

Software testing is a crucial element in software development to ensure quality and reliability. This study compares manual and automated testing using tools like Katalon Studio and Python. Manual testing is effective for scenarios requiring human judgment, such as user experience (UX) evaluation. In contrast, automated testing is more efficient for routine and repetitive tasks, reducing human error and speeding up the process. This study evaluates the effectiveness, efficiency, and costs of both methods in the context of e-commerce software testing. The results indicate that manual testing is superior in detecting defects before release, while automated testing is more cost-effective and time-efficient for repetitive testing. This guide assists developer for selecting the appropriate testing method based on their project needs.

This is an open-access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license



*Penulis Koresponden

Email: brianduen@gmail.com

Cara sitasi IEEE::

B. D. Rakly, & W. Andriyani, "Pengujian Perangkat Lunak E-Commerce : Perbandingan Antara Pengujian Manual dan Otomatis dengan Katalon Studio dan Python," *Journal of Artificial Intelligence and Software Engineering (J-AISE)*, vol. 5, no. 1, pp. 156-163, Maret 2025. doi: 10.30811/jaise.v5i1.6448

1. PENDAHULUAN

Pengujian perangkat lunak merupakan tahapan krusial dalam siklus pengembangan perangkat lunak yang bertujuan untuk memastikan produk akhir memenuhi standar kualitas yang diharapkan. Proses ini melibatkan identifikasi cacat, kesalahan, atau bug yang dapat mempengaruhi fungsionalitas, performa, dan keamanan perangkat lunak. Dalam praktiknya, terdapat dua pendekatan utama yang umum digunakan dalam pengujian perangkat lunak: pengujian manual dan pengujian otomatis.

Pengujian otomatis menjadi pilihan populer karena kemampuannya untuk mempercepat proses pengujian, meningkatkan produktivitas tim pengembang, dan mengurangi kesalahan manusia yang mungkin terjadi selama pengujian manual. Alat-alat seperti Katalon Studio dan bahasa pemrograman Python telah banyak digunakan dalam pengujian otomatis untuk mengotomatiskan skenario pengujian rutin dan repetitif. Meskipun demikian, pengujian otomatis memiliki keterbatasan dalam menangani skenario pengujian yang kompleks atau memerlukan validasi manual yang mendalam.

Sementara itu, pengujian manual tetap diperlukan untuk skenario pengujian yang lebih rumit atau kasus-kasus yang memerlukan kecermatan dan penilaian manusia yang tidak dapat digantikan oleh algoritma. Contohnya, validasi pengalaman pengguna (UX) atau situasi-situasi yang memerlukan interpretasi kasus penggunaan yang tidak terduga. Oleh karena itu, penelitian ini bertujuan untuk membandingkan keefektifan dan biaya antara pengujian manual dan otomatis menggunakan alat-alat seperti Katalon Studio dan Python. Rumusan masalah penelitian mencakup pertanyaan seputar kualitas produk yang dihasilkan, efektivitas pengujian, serta kelebihan dan keterbatasan dari kedua metode pengujian ini. Tujuan utama adalah untuk menganalisis dampak biaya dan efektivitas dari pengujian manual dan otomatis, serta mengidentifikasi situasi di mana salah satu metode pengujian lebih cocok atau menguntungkan daripada yang lain.

Beberapa penelitian yang telah ada sebelumnya yang berkaitan dengan Pengujian Perangkat Lunak E-Commerce adalah sebagai berikut. Pertama, tahun 2018, Tri Snadhika Jaya melakukan penelitian untuk aplikasi menggunakan metode blackbox testing. Hasil dari penelitian tersebut menunjukkan bahwa aplikasi mampu menangani data, baik data normal maupun tidak normal dengan presentase keberhasilan 91,67%. Lalu pada tahun 2018, Nurul Rita Mustika dan Novrina melakukan penelitian untuk melakukan pengujian otomatis menggunakan metode Black Box Testing dengan menggunakan Selenium dan Python. Penelitian ini berhasil memberikan hasil pengujian otomatis berupa successful dan failed sesuai dengan kasus pengujian yang digunakan.

Selanjutnya pada tahun 2020, Lutfi Pratama juga melakukan penelitian tentang kurangnya informasi pengukuran kelayakan dari satu perangkat lunak informatika di UIKA Bogor dengan metode pengujian otomatis. Hasil dari penelitian mengatakan bahwa perangkat lunak yang digunakan dalam pengujian dianggap layak untuk digunakan. Pada tahun 2020, Yogesh Singh dan rekan-rekan melakukan penelitian terkait pengukuran pada pengujian perangkat lunak karena kurangnya informasi terkait pengukuran pengujian perangkat lunak secara pasti. Hasil dari penelitian tersebut memberikan pengukuran secara pasti dari pengujian perangkat lunak seperti biaya, efisiensi dan efektivitas.

Kemudian pada tahun 2021, Fityan Ardi dan H. Prihantoro Putro melakukan penelitian untuk melakukan pengujian otomatis dan manual pada aplikasi Mobile ACCPartner dengan menggunakan metode Blacbox Testing. Hasil dari penelitian ini adalah pengujian yang dilakukan secara otomatis menggunakan metode Blackbox Testing lebih efektif pada aplikasi Mobile ACCPartner. Selanjutnya pada tahun 2022, Ilham R. Dhaidullah dan rekan-rekan melakukan penelitian untuk menentukan teknik pengujian yang tepat untuk diterapkan pada aplikasi berbasis website. Hasil dari penelitian tersebut adalah pengujian dengan menggunakan metode Grey Box Testing lebih efisien untuk aplikasi berbasis website. Lalu pada tahun

Dari berbagai penelitian tersebut, dapat dilihat bahwa pengujian perangkat lunak secara otomatis dengan menggunakan alat seperti Katalon Studio dan Python telah menjadi fokus penelitian yang signifikan dalam upaya meningkatkan efisiensi dan efektivitas pengembangan perangkat lunak.

Studi ini menyoroti pentingnya memiliki pemahaman yang lebih mendalam tentang persyaratan pengujian perangkat lunak dan memilih metode yang tepat untuk setiap kasus.

Pemahaman mendalam tentang manfaat dan keterbatasan setiap metode pengujian memungkinkan pengembang dan pengguna perangkat lunak untuk membuat keputusan yang lebih tepat dan mengoptimalkan keseluruhan proses pengembangan perangkat lunak.

2. METODE

Penelitian yang telah ada sebelumnya masih menitik beratkan pada kecepatan pengujian otomatis dibandingkan dengan pengujian manual. Namun belum ada penelitian yang membahas dari sisi biaya, efisiensi dan kualitas produk. Penelitian sebelumnya yang mendekati dengan usulan penelitian diantaranya Analisis Perbandingan Pengujian Manual Dan Automation Testing Pada Website E-Commerce penelitian oleh (Dewandra Sapto Prasetyo, Widya Silfianti, 2023) Survei Teknik Pengujian Software penelitian (Ilham R. Dhaidullah, M Muttanifudin, et al, 2022) dan An Empirical Study of Product Metrics in Software Testing (Yogesh Singh, Arvinder Kaur, Bharti Suri, 2020).

Data yang diperlukan telah dikumpulkan dari berbagai pengujian perangkat lunak yang dilakukan. Data ini menjadi fondasi utama dari seluruh analisis yang akan dilakukan dalam penelitian ini. Data pengujian perangkat lunak telah mencakup berbagai aspek yang dibutuhkan untuk proses pengolahan data.

Pada biaya pengujian digunakan pengambilan data dari rata-rata gaji per tahun dari seorang Quality Assurance. Penyesuaian dilakukan berdasarkan total hari kerja yang dibutuhkan dalam pembangunan atau pengujian perangkat lunak dari metode yang digunakan, serta dihitung rerata pembagian dari hari kerja dan salary.

Untuk waktu pengujian digunakan total hari kerja yang dibutuhkan, waktu pengujian termasuk dalam proses penulisan test case, pembangunan program (pengujian otomatis) dan penyesuaian kasus pengujian dengan metode pengujian. Waktu pengujian tidak berfokus pada kecepatan pada proses pengujian ketika berjalan.

Setelah data tersebut berhasil dikumpulkan, dilakukan proses pengolahan data menggunakan perhitungan yang telah diadaptasi dari penelitian terdahulu yaitu An Empirical Study of Product Metrics in Software Testing (Yogesh Singh, Arvinder Kaur, Bharti Suri, 2020). Data diperoleh melalui metode dari pengujian perangkat lunak. Dari hasil data yang didapat pada masing-masing metode pengujian, selanjutnya data diproses menggunakan perhitungan pada penelitian An Empirical Study of Product Metrics in Software Testing (2020). Untuk mendapatkan hasil berupa biaya, efektivitas, kualitas produk dan kecepatan dari metode pengujian perangkat lunak, terlebih dahulu dilakukan pengambilan data yang dibutuhkan untuk pengolahan data yaitu sebagai berikut.

Tabel 1. Variabel yang digunakan

Variabel	Keterangan
C_T	Total biaya pengujian produk dalam dolar.
KCSI	Jumlah baris kode sumber baru atau yang diubah dalam ribuan
W_T	Jumlah cacat berbobot yang ditemukan oleh tim pengujian selama siklus produk.
W_{TP}	Jumlah cacat berbobot yang ditemukan dalam produk yang diuji (sebelum rilis resmi).
W_F	Jumlah cacat berbobot yang ditemukan dalam produk setelah rilis.
T_T	Jumlah hari kerja yang digunakan untuk pengujian produk.

Setelah data yang dibutuhkan untuk pengolahan terkumpul, selanjutnya dapat dilakukan pengolahan data dengan menggunakan perhitungan metrik pada Tabel III. Metrik Pengukuran. pada tabel ini data yang telah diambil dari hasil pengujian dari setiap metode akan diproses lagi untuk dilakukan perhitungan untuk mendapatkan hasil nilai yang akan digunakan sebagai perbandingan dari metode pengujian yang digunakan penulis.

Tabel 2. Metrik pengukuran

Pengukuran	Perhitungan	Keterangan
(TCS) Test cost to product size	$C_T/KCSI$	Semakin rendah nilai TCS, semakin rendah biaya yang diperlukan untuk menguji setiap seribu baris kode.
(TE) Test effectiveness	$W_T/(W_{TP}+W_F)*100\%$	Semakin tinggi TE, menunjukkan rasio cacat yang lebih tinggi atau cacat penting terdeteksi sebelum rilis, semakin tinggi pula efektivitas organisasi pengujian untuk menyingkirkan cacat.
(QP) Quality Product	$W_F/KCSI$	Jumlah QP yang rendah menunjukkan lebih sedikit cacat, atau lebih sedikit cacat serius yang dikirimkan, menyiratkan kualitas yang lebih tinggi dari kode yang dikirimkan oleh tim pengujian.

(TT) Test time	$T_T/KCSI$	Semakin rendah nilai TT , semakin sedikit waktu yang dibutuhkan oleh tim penguji untuk menguji produk.
-------------------	------------	--------------------------------------------------------------------------------------------------------

Pada tahap pengujian digunakan skenario testing yang sama untuk setiap metode pengujian agar beban pengujian yang digunakan setara pada setiap metode pengujian.

Tabel 3. Skenario pengujian

Id	Kasus	Hasil
C1	Register	User berhasil mendaftar akun untuk akses web
C2	Gagal Login	User menerima notifikasi jika terjadi gagal login
C3	Sukses Login	User berhasil login dengan akun yang terdaftar
C4	Akses Profil	User dapat mengakses halaman profil
C5	Edit Nama Profil	User dapat melakukan perubahan pada profil
C6	Edit Alamat	User dapat melakukan perubahan pada profil
C7	Tambah Barang	User dapat mengakses fitur tambah barang ketika login
C8	Hapus Barang	User dapat mengakses fitur menghapus barang ketika login
C9	Menampilkan Notifikasi	User dapat melihat semua notifikasi yang masuk ke dalam akun ketika login
C10	Logout	User berhasil melakukan logout

3. HASIL DAN PEMBAHASAN

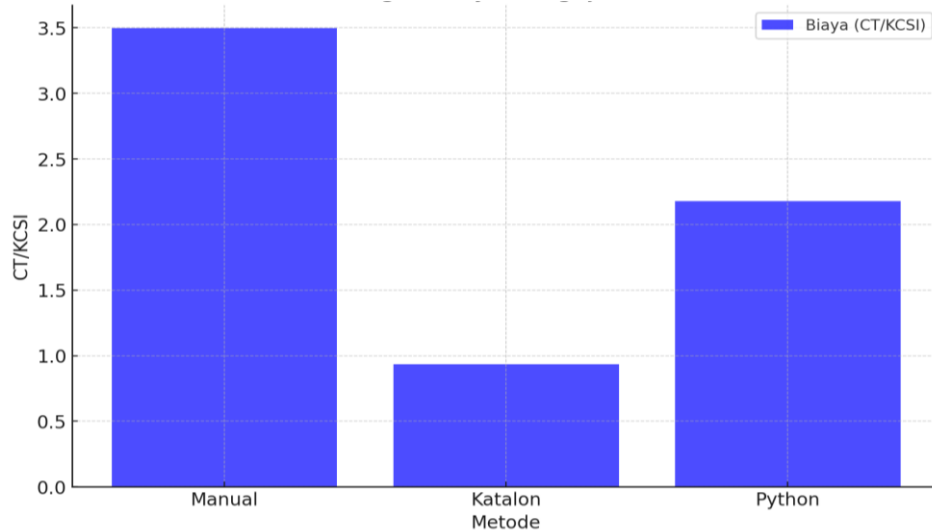
Dari kasus pengujian yang telah dilakukan didapatkan hasil berupa nilai dari setiap variabel yang dibutuhkan untuk proses perhitungan, nilai ini mewakili setiap metode pengujian yang telah dilakukan.

Tabel 4. Hasil pengujian

Variabel	Manual	Python	Katalon
C_T	269	933	2.177
KCSI	0.013	0.001	0.001
W_T	13	1	1
W_{TP}	13	1	1
W_F	10	0	0
T_T	1	3	7

3.1. Biaya

Setelah data yang dibutuhkan untuk pengolahan terkumpul, selanjutnya dapat dilakukan pengolahan data dengan menggunakan perhitungan metrik pada Tabel 2. Metrik Pengukuran. pada tabel ini data yang telah diambil dari hasil pengujian dari setiap metode akan diproses lagi untuk dilakukan perhitungan untuk mendapatkan hasil nilai yang akan digunakan sebagai perbandingan dari metode pengujian yang digunakan penulis. Pada tahap pengujian digunakan skenario testing yang sama untuk setiap metode pengujian agar beban pengujian yang digunakan setara pada setiap metode pengujian.



Gambar 1. Bar chart perbandingan biaya

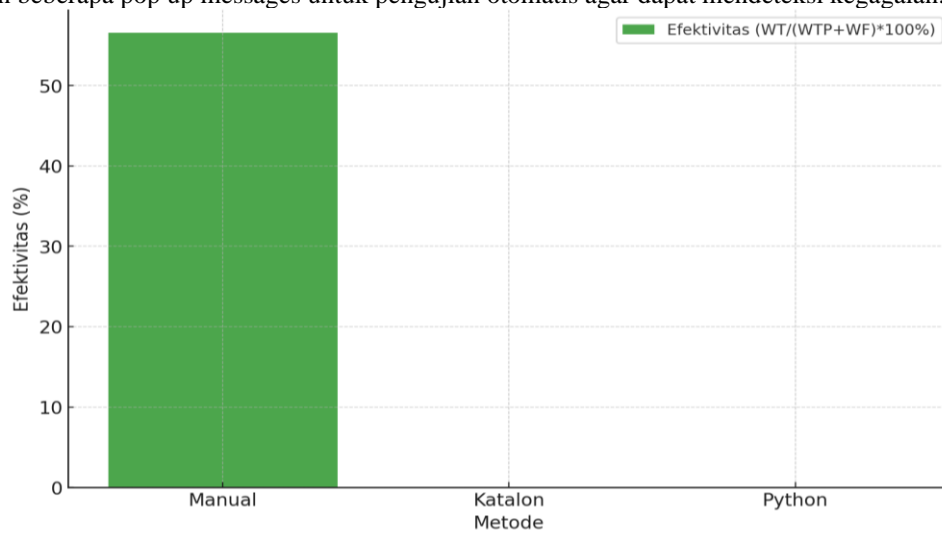
Tabel 5. Hasil perhitungan biaya

Metode	C_T	KCSI	$C_T / KCSI$
Manual	269	0.013	3.497
Python	933	0.001	0.933
Katalon	2177	0.001	2.177

Pada perbandingan biaya didapatkan nilai pada pengujian manual lebih tinggi dibandingkan dengan pengujian otomatis menggunakan katalon atau python. Ini menunjukkan jika pada kasus pengujian yang dilakukan, biaya yang digunakan untuk pengujian manual dibagi dengan baris kode yang dirubah dalam ribuan lebih banyak dibandingkan dengan pengujian otomatis.

3.2. Efektivitas

Pada perbandingan efektivitas diperoleh nilai manual lebih tinggi pada kasus yang digunakan, pada perbandingan ini variabel cacat berbobot sebelum rilis cukup berpengaruh dimana pengujian manual bisa mendeteksi cacat sebelum rilis lebih banyak dibandingkan pengujian otomatis. sedangkan pada fase setelah rilis, developer menyesuaikan beberapa pop up messages untuk pengujian otomatis agar dapat mendeteksi kegagalan.



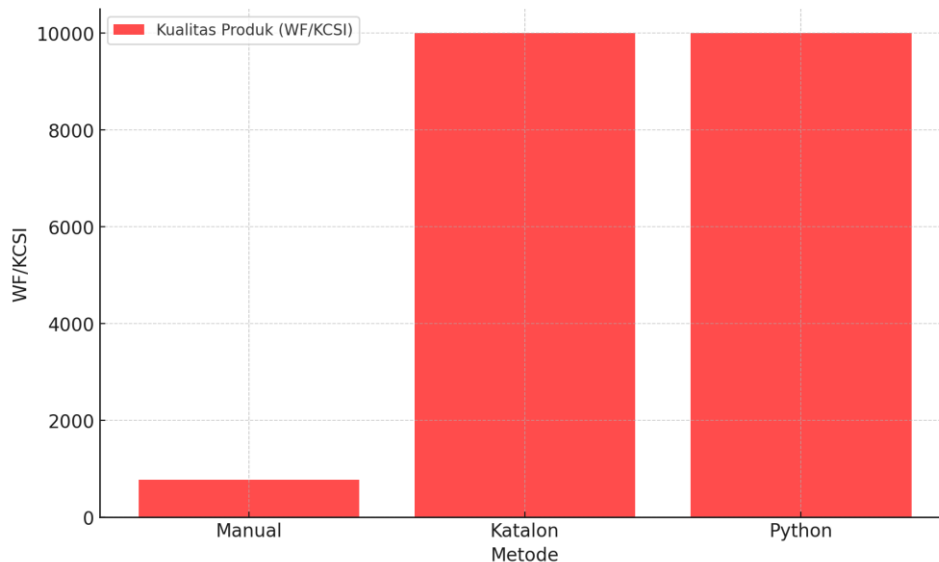
Gambar 2. Bar chart perbandingan efektivitas

Tabel 6. Hasil perhitungan efektivitas

Metode	W_T	W_{TP}	W_F	$W_T/(W_{TP}+W_F)*100\%$
Manual	13	3	0	56.56
Python	0	0	0	0
Katalon	0	0	0	0

3.3. Kualitas Produk

Pada perbandingan kualitas produk, didapatkan nilai yang rendah pada pengujian manual, nilai yang rendah ini menunjukkan lebih sedikitnya cacat atau cacat serius yang dikirimkan saat fase produksi. ini mengindikasikan kualitas produk yang lebih baik. Sedangkan pada pengujian otomatis dengan kasus yang diterapkan sekarang masih cukup banyak cacat yang dikirimkan saat fase produksi.



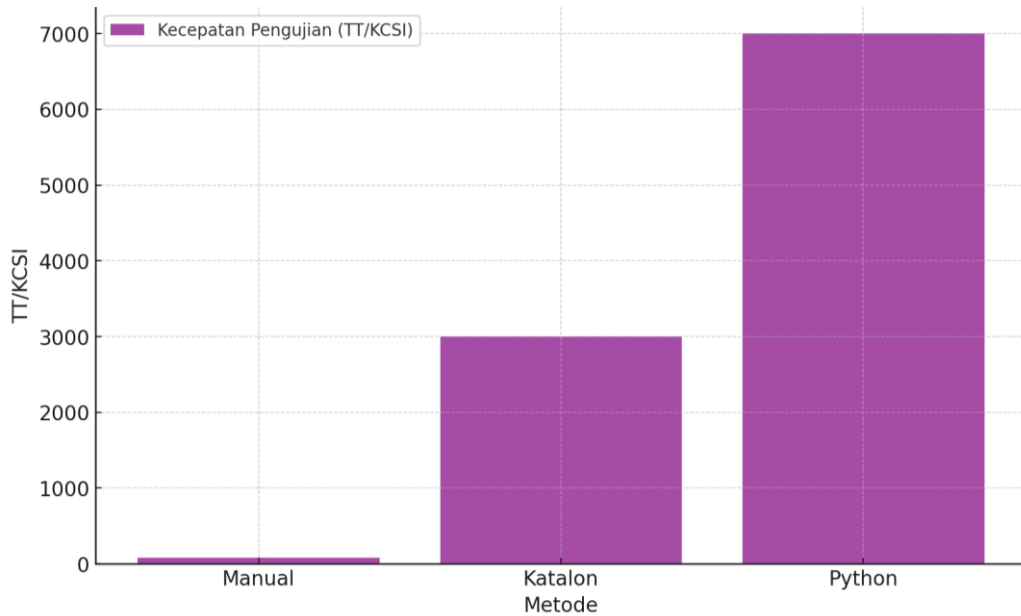
Gambar 3. Bar chart perbandingan kualitas produk

Tabel 7. Hasil perhitungan kualitas produk

Metode	W_F	KCSI	$W_F/KCSI$
Manual	10	0.013	769.2
Python	10	0.001	10000
Katalon	10	0.001	10000

3.4. Kecepatan

Lalu pada perbandingan kecepatan didapatkan nilai yang rendah pada manual, nilai yang rendah ini menunjukkan waktu yang ditempuh dalam hari kerja. semakin kecil nilai yang didapat maka semakin singkat waktu yang diperlukan untuk melakukan pengujian. dari hasil data yang ditunjukkan diatas waktu pengerjaan pengujian manual lebih singkat dibandingkan dengan pengujian otomatis seperti katalon dan python dengan contoh kasus yang digunakan sekarang.



Gambar 4. Bar chart perbandingan kecepatan

Tabel 8. Hasil perhitungan kecepatan

Metode	T_r	KCSI	$T_r/KCSI$
Manual	1	0.013	76.9
Python	3	0.001	3000
Katalon	7	0.001	7000

4. KESIMPULAN

Berdasarkan hasil analisis diperoleh kesimpulan bahwa Biaya, Katalon memiliki nilai biaya terendah, menunjukkan bahwa Katalon membutuhkan biaya paling sedikit untuk menguji setiap seribu baris kode dibandingkan Manual dan Python karena untuk pengujian katalon meskipun memiliki biaya kerja yang lebih besar, perbandingan dengan baris kode yang dirubah setelah pengujian lebih kecil dibandingkan dengan manual. Python berada di posisi tengah dengan biaya dan jam kerja yang paling tinggi tetapi sama seperti katalon, jumlah baris kode yang dirubah lebih sedikit dibandingkan dengan manual, sedangkan Manual membutuhkan biaya tertinggi karena perbandingan biaya dengan jumlah baris kode yang dirubah cukup banyak dengan ditemukannya kecacatan yang lebih banyak dari pengujian otomatis.

Efektivitas Pada pengujian manual menunjukkan nilai efektivitas tertinggi, sehingga pengujian Manual mampu mendeteksi lebih banyak cacat sebelum rilis dibandingkan dengan Katalon dan Python, yang dimana pada proses pengembangan perangkat lunak pengujian manual selalu diutamakan pada fase produksi sedangkan pengujian otomatis lebih sering digunakan pada case tertentu saja yang membutuhkan banyak pengulangan dan input data yang masif. Baik Katalon maupun Python memiliki efektivitas 0, yang menunjukkan ketidakmampuan untuk mendeteksi cacat dalam pengujian ini, karena penggunaannya dalam pengembangan perangkat lunak hanya digunakan pada case yang sudah tidak memiliki kecacatan serta beberapa kasus yang membutuhkan pengulangan.

Kualitas Produk Manual memiliki nilai kualitas produk terbaik (nilai terendah), menunjukkan lebih sedikit cacat yang ditemukan dalam kode, sehingga kualitas kode lebih tinggi. Katalon dan Python memiliki nilai sama (10,000), menunjukkan kualitas produk yang lebih rendah dibandingkan Manual karena pengujian otomatis pada pengembangan proyek perangkat lunak lebih sering digunakan pada kasus yang sudah terbebas dari kecacatan. Maka pada kasus sebelum rilis, pengujian manual lebih menunjang untuk menilai kualitas suatu produk.

Kecepatan Pada pengujian manual didapatkan waktu uji tercepat dalam menguji produk, sedangkan Katalon dan Python membutuhkan waktu jauh lebih lama, dengan Python menjadi yang paling lambat. Pada kasus ini pengujian manual diunggulkan dengan pengerjaan pengujian hanya 1 hari kerja karena kasus uji yang digunakan berfokus pada end to end testing dan tidak banyak pengulangan yang dilakukan pada kasus yang digunakan, sedangkan untuk membangun pengujian otomatis seperti katalon dibutuhkan 3 hari kerja dengan dukungan user interface yang

memudahkan user untuk membangun pengujian otomatis dan dibutuhkan 7 hari kerja untuk membangun pengujian otomatis.

REFERENSI

- [1] Ardi, F., & Putro, H. P. (2020). Pengujian Black Box Aplikasi Mobile Menggunakan Katalon Studio.
- [2] Cahyono, A. B. (2020). Automation Testing Tool dalam Pengujian Aplikasi the Point of Sale.
- [3] Dewi, F. K. S., Adithama, S. P., & Suhardi, A. T. (2023). Pengujian Aplikasi Doctor to Doctor Menggunakan Metode Black Box Testing. *KONSTELASI: Konvergensi Teknologi Dan Sistem Informasi*, 3(1), 61–72.
- [4] Hasanah, N. U. (2023). Kajian otomatisasi pengujian gui: selenium ide, Uipath Studio, Katalon studio. Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta.
- [5] Jaya, T. S. (2018). Pengujian aplikasi dengan metode blackbox testing boundary value analysis (studi kasus: kantor digital Politeknik Negeri Lampung). *Jurnal Informatika: Jurnal Pengembangan IT*, 3(1), 45–48.
- [6] Luthan, M. A. (2024). Tingkatkan Efisiensi Validasi Data Bill Of Quantity (Boq) Dengan Penerapan Otomasi Capture Banner. Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
- [7] Prasetyo, D. S., & Silfianti, W. (2023). Analisis Perbandingan Pengujian Manual Dan Automation Testing Pada Website E-Commerce. *Jurnal Ilmiah Teknik*, 2(2), 127–131.
- [8] Prasetyo, G. A. (2024). Analisis Perbandingan UI Test Automation Framework dengan Metode the Distance to the Ideal Alternative. Universitas Islam Indonesia.
- [9] Pratama, L. (2020). Pengukuran Kualitas Perangkat Lunak Prodi Teknik Informatika Uika Bogor Menggunakan Iso 9126. *Jurnal Inovatif: Inovasi Teknologi Informasi Dan Informatika*, 3(1), 49–60.
- [10] Putra, Y. W. S., Dawis, A. M., Novi, N., Natsir, F., Fitria, F., Widhiyanti, A. A. S., Hasan, F. N., Somantri, S., & Maniah, M. (2023). *Pengantar Aplikasi Mobile*. Penerbit Widina.
- [11] Y. Singh, A. Kaur, and B. Suri, "An Empirical Study of Product Metrics in Software Testing," in *Innovative Techniques in Instruction Technology, E-learning, E-assessment and Education*, 2008.
- [12] Sudipa, I. G. I., Ariantini, M. S., Pomalingo, S., Ridwan, A., Primasari, D., Ariana, A. A. G. B., Ibrahim, R. N., Ilham, R., Arsana, I. N. A., & Irmawati, I. (2023). *Buku Ajar Rekayasa Perangkat Lunak*. PT. Sonpedia Publishing Indonesia.
- [13] Suryawinata, M. (2019). *Buku Ajar Mata Kuliah Pengembangan Aplikasi Berbasis Web*. Umsida Press, 1–144.
- [14] Zhang, X., Wang, T., Yu, Y., Zhang, Y., Zhong, Y., & Wang, H. (2022). The Development and Prospect of Code Clone. *ArXiv Preprint ArXiv:2202.08497*.