

Analisis Perbandingan *Testing* Manual dan *Automation* terhadap Peningkatan Kestabilan Web *E-commerce* TOCO

Muhammad Amar Mustajab¹, Rissa Nurfitriana Handayani²,

Ignatius Wiseto Prasetyo Agung³

^{1,3}Program Studi Teknik Informatika, Universitas Adhirajasa Reswara Sanjaya

²Program Studi Sistem Informasi, Universitas Adhirajasa Reswara Sanjaya

e-mail: ¹m.amar.mustajab@gmail.com, ²rissa@ars.ac.id, ³wiseto.agung@ars.ac.id

Abstrak

Para pelaku bisnis menciptakan inovasi baru dengan memanfaatkan internet dalam kegiatan e-bisnis. *E-commerce* adalah pasar online untuk membeli dan menjual produk dan jasa. TOCO adalah platform jual beli dan iklan baris berbasis komunitas di Indonesia. Dalam tahapan Requirement Analysis diperoleh ada dua peran penting dalam sistem TOCO, pembeli maupun penjual. Berdasarkan hasil identifikasi kebutuhan, terdapat sejumlah fitur utama yang dianggap krusial untuk diuji secara fungsional guna memastikan kestabilan dan kelancaran pengalaman pengguna. Selain dari *Seller Center* TOCO menunjukkan hasil yang sama, menjadikan fitur-fitur utama TOCO yang diuji telah berjalan sesuai dengan kebutuhan dasar pengguna. Tidak ditemukan *error* fatal atau *crash* selama pengujian dilakukan.

Kata kunci— STLC, pengujian, otomatisasi, *e-commerce*, katalon

Abstract

Business actors are creating new innovations by leveraging the internet in e-business activities. E-commerce serves as an online marketplace for buying and selling products and services. TOCO is a community-based buying, selling, and classified ads platform in Indonesia. During the Requirement Analysis stage, two key roles within the TOCO system were identified the buyer and the seller. Based on the requirements analysis, several core features were considered crucial for functional testing to ensure system stability and a smooth user experience. In addition, TOCO's Seller Center showed similar results, confirming that the key TOCO features tested were performing according to basic user needs. No fatal errors or crashes were encountered during testing.

Keywords— STLC, testing, automation, *e-commerce*, katalon

Corresponding Author:

Rissa Nurfitriana Handayani,

Email: rissa@ars.ac.id

1. PENDAHULUAN

Salah satu cara untuk membangun ekonomi yang stabil dan berjalan lama adalah dengan melakukan bisnis, biasa mulai dari keluarga atau daerah sekitar tempat berbisnis yang dibuktikan dengan banyaknya pelaku bisnis mulai dari skala kecil maupun skala besar yang menggerakkan ekonomi di Indonesia[1]. Melihat perkembangan Koperasi dan Usaha Mikro Kecil dan Menengah (UMKM) masih butuh dukungan aturan-aturan afirmasi terhadap respons perubahan perekonomian pasar yang sangat dinamis [2]. Para pelaku bisnis menciptakan inovasi baru dengan memanfaatkan internet dalam kegiatan e-bisnis. *E-commerce* adalah pasar online untuk membeli

dan menjual produk dan jasa. Ini pertama kali muncul di Indonesia pada permulaan abad ke-20. Di Indonesia, ada empat model berbeda yang muncul: B2B (*Bussines to Business*), B2C (*Bussines to Consumer*), C2B (*Consumer to Business*), C2C (*Consumer to Consumer*) [3].

Di Indonesia, Shopee dan Tokopedia dikenal sebagai dua *marketplace e-commerce* yang memiliki jangkauan pengguna paling luas. Tokopedia, salah satu pemimpin utama dalam industri *e-commerce* di Indonesia, telah menonjol sejak didirikan pada tahun 2009. Pertumbuhan yang luar biasa dan keberhasilan Tokopedia dalam membangun keunggulan kompetitifnya telah menarik perhatian tidak hanya di tingkat nasional, tetapi juga secara global. Sebagai salah satu dari sedikit perusahaan teknologi Indonesia yang telah mencapai status "*unicorn*" (nilai lebih dari 1 miliar dolar), Tokopedia telah menjadi ikon dalam perjalanan transformasi digital negara ini [4]. Menilai kestabilan sistem *e-commerce* dari lapangan secara langsung sangat berkaitan erat dengan kualitas proses pengujian perangkat lunak yang terjadi selama tahap pengembangan sistem *e-commerce*[5]. Untuk meningkatkan efektivitas dalam mendeteksi kesalahan atau *bug* serta efisiensi waktu untuk memenuhi kebutuhan pengguna maka dapat diterapkan pengujian manual dan otomatis[6].

Sebagai pembanding lain dari dua *e-commerce* di atas, peneliti ingin mengangkat pendatang baru *e-commerce* TOCO, untuk memastikan fitur utama ke dalam pengujian, salah satunya seperti register akun baru, *login*, pencarian produk, *checkout*, penambahan produk dan pemindahan produk berjalan stabil di tengah proses integrasi produk dari *marketplace* seperti Tokopedia dan Shopee. Menilai tindakan yang dilakukan oleh dua platform tersebut, penelitian ini bertujuan untuk membandingkan metode pengujian manual dan otomatisasi dalam mendeteksi *bug* selama regresi berlangsung serta memastikan kualitas pengalaman pengguna dalam bertransaksi[7], utamanya di platform TOCO.

Dalam sebuah penelitian yang membahas metode pengujian manual dan otomatisasi dalam aplikasi *E-commerce* menyajikan hasil perbandingan waktu yang digunakan selama pengujian dan menilai efektivitas antara pengujian manual dan otomatisasi menggunakan alat bantu otomatisasi Katalon Studio dalam konteks aplikasi *e-commerce*. Persamaan penelitian ini dengan sangat kuat karena membahas kestabilan dan efisiensi sistem berdasarkan metode pengujian. Perbedaan utama penelitian ini dengan penelitian lain terletak pada objek yang diuji, yaitu platform TOCO, sehingga seluruh prosesnya mengikuti kebutuhan sistem ini.

Untuk menjaga agar pengujian perangkat lunak dapat berlangsung teratur dan mengikuti alur kerja yang jelas, penelitian ini menerapkan tahapan dalam *Software Testing Life Cycle* (STLC). Pada konteks penelitian ini, STLC dipakai sebagai kerangka kerja agar setiap aktivitas pengujian berjalan sistematis dan hasilnya dapat dipertanggungjawabkan [8].

1. *Requirement Analysis*

Tahap pertama dalam penelitian ini dilakukan dengan meninjau ulang dokumen kebutuhan yang diberikan oleh *stakeholder* TOCO. Dari analisis tersebut, tim QA mengidentifikasi fitur-fitur inti seperti *login*, *checkout*, dan pengelolaan produk, itu bagian dari contoh fitur yang harus diuji. Proses ini membantu menentukan batas pengujian, baik untuk skenario manual maupun otomatis, sehingga QA dapat memastikan bahwa fitur yang diuji benar-benar sesuai dengan kebutuhan sistem [9].

2. *Test Planning*

Pada tahap perencanaan, QA menyusun strategi pengujian berdasarkan fitur TOCO yang sudah ditentukan sebelumnya. Aktivitasnya mencakup pemilihan alat uji (seperti Katalon Studio), penentuan kebutuhan perangkat, estimasi waktu pelaksanaan, serta pembagian tanggung jawab antar anggota tim [6], [10]. Hasil perencanaan ini kemudian dituangkan ke dalam dokumen test plan yang nantinya menjadi dasar penyusunan test case dan jadwal pengujian [9]. Setelah melakukan dokumentasi *test plan* dan *test case* pengujian dapat dimulai untuk mengembangkan hasil akhir yang diminta oleh pengguna.

3. *Test Case Development*

Setelah rencana pengujian tersusun, QA mulai menuliskan *test case* sesuai kebutuhan TOCO. Setiap *test case* memuat langkah uji, data yang dipakai, dan hasil yang diharapkan [6]. Pembuatan *test case* akan dilakukan manual menggunakan tangan dengan mengetiknya dalam Exel untuk dokumentasi, hasilnya nanti akan dimasukkan ke dalam alat uji perangkat lunak Katalon Studio untuk memudahkan proses dokumentasi *test case* yang dijalankan otomatisasi. Data uji yang dibuat sesuai dengan peluang aksi yang terjadi saat user menggunakan aplikasi [11].

4. Test Environment Setup

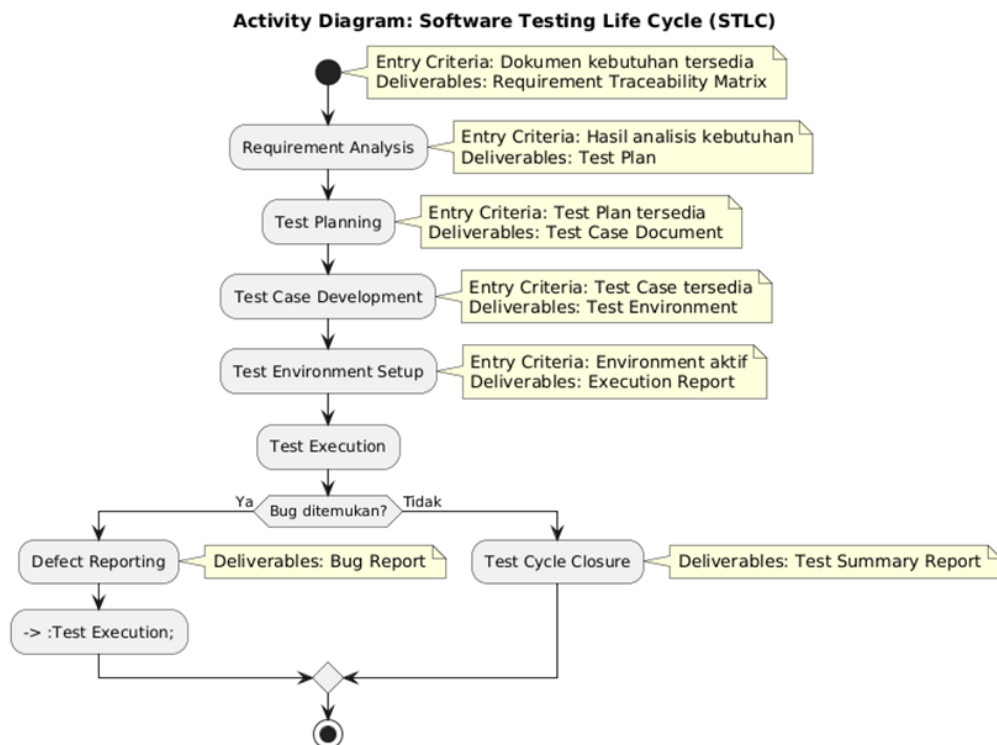
Tahap berikutnya adalah penyiapan lingkungan pengujian. Tim pengembang menyediakan *build* atau versi aplikasi TOCO yang siap diuji, sedangkan QA mempersiapkan data pendukung serta melakukan pengecekan awal berupa *smoke test* untuk memastikan lingkungan tersebut berjalan stabil. Tahap ini memastikan bahwa proses pengujian dapat dilakukan tanpa gangguan dari sisi teknis [10]. Sementara itu, tim QA merencanakan *smoke test* dari *environment* yang telah diberikan, selain itu tim QA juga menyiapkan *test data* yang dimasukkan ke dalamnya.

5. Test Execution

Pada proses eksekusi, akan dilakukan eksekusi pengujian berdasarkan *test case* dan *test plan* yang dibuat pada tahap sebelumnya. Selain menjalankan *script*, QA juga melakukan pemeliharaan *test script* apabila terdapat perubahan pada aplikasi serta melaporkan temuan *bug* yang muncul selama proses pengujian berlangsung [10], [11].

6. Test Cycle Closure

Setelah melaksanakan tahap-tahapan pengujian di atas, QA melakukan evaluasi terhadap hasil pelaksanaan pengujian untuk memastikan semua skenario telah dijalankan dan setiap temuan telah dilaporkan. Tim QA akan melaporkan hasil pengujian yang dilakukan pada *environment* yang disiapkan oleh tim pengembang[9]. Kegiatan inti di fase ini adalah menyusun *test summary report*, membersihkan *test environment*, melakukan *knowledge transfer*, dan memberikan rekomendasi serta perbaikan pengembangan.



Gambar 1. Diagram Software Testing Life Cycle (STLC)

A. Pengujian Perangkat Manual (*Manual Testing*)

Pengujian manual merupakan proses ketika QA menjalankan aplikasi secara langsung tanpa bantuan alat otomatisasi. Setiap langkah dilakukan sesuai skenario uji yang telah disusun, kemudian hasilnya diamati untuk memastikan perilaku sistem sesuai dengan yang diharapkan. [12].

Karena melibatkan observasi manusia, pengujian manual sangat efektif dalam menemukan *bug* yang berkaitan dengan logika, kenyamanan pengguna, dan tampilan [13]. Dalam penelitian ini, pengujian manual digunakan terutama pada fase awal pengembangan, saat perubahan fitur masih sering terjadi dan pemeriksaan dari sudut pandang pengguna sangat dibutuhkan. *Manual testing* juga sangat penting untuk mengevaluasi aplikasi dari perspektif penggunaan di lapangan atau dikenal dalam tahap *User Acceptance Testing* (UAT)[6].

Jenis-jenis *Manual Testing* yang Umum[14]:

1. *Black-box testing*: Menguji fungsionalitas dari UI/UX tanpa melihat kode sumber.
2. *White-box testing*: Menguji logika dan struktur kode.
3. *Grey-box testing*: Kombinasi UI/UX dan logika struktur.

Jenis pengujian *Black-box* yang sering dijumpai oleh pengguna aplikasi, peneliti menilai dua jenis pengujian lainnya diluar kemampuan untuk dilakukan peneliti karena keterbatasan akses terhadap struktur kode objek penelitian, jika secara fungsi sudah berjalan sesuai kebutuhan maka struktur kode dan logika sudah teruji.

B. Pengujian Otomatisasi (*Automation Testing*)

Pengujian otomatisasi adalah proses pengujian yang bergantung pada *script* uji yang dibuat oleh penguji dalam perangkat lunak terpisah dan pelaksanaan pengujiannya dilakukan secara otomatis dari perangkat lunak terpisah[15]. Tidak seperti pengujian manual yang dilakukan oleh tangan manusia, *automation testing* memungkinkan eksekusi pengujian dilakukan secara berulang dengan menggunakan skrip yang sudah ditentukan. Maka, pengujian ini dapat meningkatkan frekuensi pengujian dan memungkinkan untuk melakukan pengujian regresi yang berulang tanpa memakan waktu lama [11].

Automation testing adalah metode pengujian perangkat lunak yang menulis skrip atau perintah dalam bahasa tertentu agar sistem dapat menjalankan simulasi pengujian secara otomatis sesuai alur yang telah ditentukan [1].

Dalam proses pengujian ini, diperlukan alat bantu pendukung agar pelaksanaan pengujian lebih sistematis, efisien, dan terdokumentasi dengan baik[10], [16]. Untuk membantu sistematis pengujian *test case* maka perlu alat dokumentasi pengujian digunakan untuk menyusun *test case* lalu menyimpan hasilnya, jika terjadi masalah selama pengujian maka dapat didokumentasikan untuk melacak masalah yang terjadi, selain melacak masalah, ini cukup membantu untuk tim QA menganalisis hasil pengujian dan menyampaikannya kepada tim pengembang[17].

Beberapa alat bantu bahkan dapat terintegrasi dalam proses pengembangan perangkat lunak yang lebih luas, seperti dalam *pipeline* CI/CD (*Continuous Integration/Continuous Deployment*), sehingga pengujian dapat dilakukan setiap kali ada perubahan kode.

a) Alat Bantu Pengujian Manual

Pengujian manual lebih mengandalkan interaksi langsung dari mata penguji terhadap aplikasi dari sisi UI melihat respons sistem. Untuk mendukung kelancaran dan ketelitian proses ini, penguji dapat menggunakan berbagai bentuk alat bantu bersifat dokumentatif, administratif, maupun observasi[10]. Berikut beberapa alat bantu yang umum digunakan dalam *manual testing*:

1. *Microsoft Excel* atau *Google Sheets*

Digunakan untuk menyusun dan mencatat *test case*, *test step*, status pengujian (*pass/fail*), serta catatan uji lainnya. Excel mempermudah dalam membuat daftar hasil uji, mengurutkan skenario pengujian, dan mendokumentasikan hasil uji secara sistematis. Karena fleksibel dan tersedia secara luas, *Excel* atau *Google Sheets* sangat cocok untuk tim QA dengan sumber daya terbatas.

2. *Browser DevTools (Inspect Element)*

Developer *Tools* pada *browser* seperti Chrome, Firefox, atau Edge digunakan untuk memeriksa elemen HTML, menganalisis respons HTTP, melihat console log, serta memeriksa *layout* atau *script error*. *Tools* ini sangat membantu dalam menemukan penyebab *bug* dan memahami perilaku antarmuka.

3. *Screenshot dan Screen Recording Tools*

Tools seperti *Snipping Tool* (Windows), *Lightshot*, atau OBS Studio digunakan untuk mengambil dokumentasi visual dari hasil pengujian. Ini berguna sebagai bukti ketika ada *bug* atau sebagai referensi pengujian ulang..

b) Alat Bantu Pengujian Otomatisasi (*Automation Testing*)

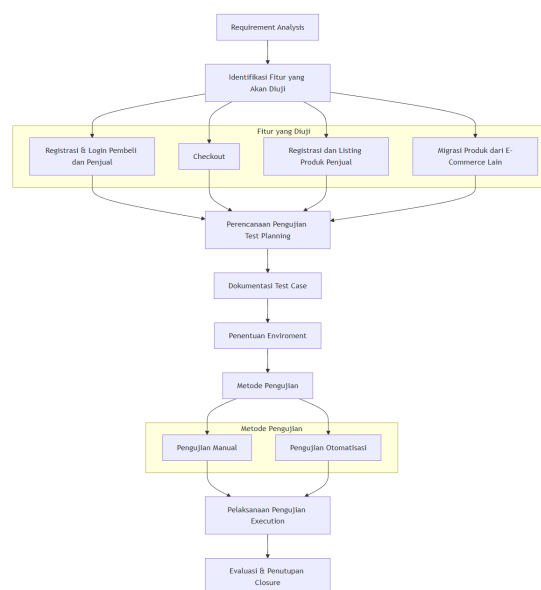
Pengujian otomatis dilakukan dengan menjalankan skenario uji yang ditulis dalam bentuk skrip, sehingga proses eksekusinya tidak lagi mengandalkan interaksi manual. Melalui pendekatan ini, rangkaian pengujian dapat dijalankan berulang dengan hasil yang konsisten[1]. Metode otomatisasi sangat membantu ketika sistem membutuhkan pengujian regresi atau pengujian pada fitur yang dijalankan berkali-kali. Selain itu, *automation* dapat dihubungkan dengan proses pengembangan berkelanjutan (CI/CD), sehingga setiap perubahan pada aplikasi dapat langsung diuji tanpa harus menunggu proses manual. Alat bantu yang dapat digunakan.

1. Katalon Studio

Merupakan platform pengujian otomatis yang mendukung Web, *mobile*, dan API *testing*. Versi gratisnya sudah cukup untuk membuat skrip uji berbasis UI, terutama menggunakan fitur *Record & Playback*, *Script Mode*, dan *Spy Web*. Katalon juga mendukung penggunaan XPath, verifikasi halaman, dan laporan hasil uji.

2. METODE PENELITIAN

TOCO adalah platform jual beli dan iklan baris berbasis komunitas di Indonesia [18]. Platform ini menyediakan tempat untuk berbelanja, menjual, dan terhubung dengan komunitas pembeli dan penjual di TOCO. [19] Peneliti menilai TOCO sebagai *e-commerce* yang cocok untuk dijadikan objek penelitian, pasar digital umumnya menerapkan pemasangan iklan gratis sehingga dapat melahirkan pelaku usaha baru, yang menghidupkan roda perekonomian.



Gambar 2. Alur Penerapan STLC Terhadap Situs Jual-Beli TOCO

Pada tahap ini dilakukan identifikasi terhadap fitur-fitur utama yang akan diuji dalam situs TOCO, baik dari sisi pengguna maupun penjual. Ini dilakukan untuk memahami kebutuhan pengguna serta menentukan skenario pengujian, mengambil contoh seperti melakukan registrasi, *login*, pengelolaan produk, migrasi data, hingga transaksi pembelian. Analisis dilakukan dengan mengakses langsung situs TOCO dan *Seller Center*, serta mengamati proses interaksi umum yang dijalankan pengguna.

Setiap skenario pengujian yang dirancang kemudian dituangkan dalam bentuk *test case*. Dokumentasi ini ditulis dan disimpan dalam Microsoft Excel, dengan format dokumentasi yang memuat nama *test case*, urutan langkah pengujian, serta hasil yang diharapkan. Penamaan *test case* disesuaikan dengan alur fitur yang diuji, misalnya menggunakan awalan *Verify* diikuti dengan nama proses pengujian. Di tahap ini juga disusun struktur *framework* otomatisasi berdasarkan observasi elemen HTML melalui fitur *Inspect* pada *browser*, ini dilakukan untuk rancangan dan implementasi *framework* pengujian otomatisasi yang sebelumnya dirancang dalam Katalon Studio[20]. *Framework* ini mencakup struktur penyimpanan *test case* otomatis, pemetaan *selector* XPath yang stabil secara fitur, serta parameter lain dari data uji agar pengujian berjalan secara konsisten dan dapat diulang.

Pengujian ini dilakukan pada lingkungan *production* karena peneliti tidak memiliki akses terhadap *environment* pengembangan (*staging* atau QA). Seluruh proses pengujian dijalankan pada situs TOCO yang digunakan oleh pengguna aktif, dengan tetap memperhatikan batasan pengujian agar tidak memengaruhi keamanan data maupun aktivitas operasional sistem[7].

Penelitian ini menerapkan dua pendekatan pengujian, yaitu *manual testing* dan *automation testing*. Pengujian manual difokuskan pada pemeriksaan alur yang melibatkan tampilan visual, kompleksitas interaksi, serta validasi antarmuka pengguna secara langsung. Sementara itu, pengujian otomatis digunakan untuk skenario yang dijalankan berulang dan bersifat regresi, sehingga dapat menghemat waktu eksekusi dan menjaga konsistensi hasil[10]. Melalui penerapan tahapan ini, pengujian otomatis dapat dijalankan secara konsisten dan selanjutnya dimanfaatkan untuk membandingkan waktu pelaksanaan antara pengujian manual dan otomatis pada fitur yang sama.

Pengujian semua *test case* yang terdokumentasi dieksekusi dalam pengujian manual dan otomatis; *framework* otomatisasi dibuat berbasis Katalon Studio dengan arsitektur modular sehingga setiap *test case* dapat dijalankan secara independen. Penggunaan XPath disesuaikan dengan elemen halaman TOCO yang dianalisis sebelumnya, dan disiapkan *script* otomatisasi untuk mencerminkan langkah-langkah pada *test case* manual.

Setelah mendapat hasil pengujian *pass* atau *fail* yang didokumentasikan di dalam Excel dan laporan hasil uji otomatisasi Katalon Studio, peneliti kemudian melakukan analisis terhadap hasil pengujian tersebut untuk melihat perbedaan waktu pelaksanaan antara metode manual dan otomatis.

3. HASIL DAN PEMBAHASAN

Melihat kebutuhan dari pengguna yang dituliskan dalam *Requirement Analysis* diperoleh ada dua peran penting dalam sistem TOCO, pembeli maupun penjual. Berdasarkan hasil identifikasi kebutuhan tersebut, terdapat sejumlah fitur utama yang dianggap krusial untuk diuji secara fungsional guna memastikan kestabilan dan kelancaran pengalaman pengguna[10].

Fitur-fitur yang menjadi fokus pengujian dalam penelitian ini meliputi:

Tabel 1. Daftar Fitur Pengujian Pembeli

No.	Fitur yang Diuji	Deskripsi Singkat
1	Registrasi Pembeli	Pengguna membuat akun baru sebagai pembeli melalui halaman registrasi TOCO
2	<i>Login</i> Pembeli	Pengguna masuk ke dalam sistem TOCO menggunakan email dan <i>password</i>

No.	Fitur yang Diuji	Deskripsi Singkat
3	Buka Detail Produk	Pengguna mengakses halaman detail dari produk yang ditampilkan di halaman utama
4	Tambah ke Keranjang	Pengguna menambahkan produk ke keranjang belanja
5	Checkout Produk	Pengguna melakukan pembelian produk dan memilih metode pembayaran

Tabel 2. Daftar Fitur Pengujian Penjual

No.	Fitur yang Diuji	Deskripsi Singkat
1	Registrasi Penjual	Pengguna membuat akun toko sebagai penjual melalui halaman pendaftaran <i>seller</i>
2	Login Seller Center	Penjual masuk ke <i>dashboard Seller Center</i> untuk mengelola toko dan produk
3	Tambah Produk Manual	Penjual menambahkan produk secara manual melalui <i>form input</i> produk
4	Terima Pesanan	Penjual memproses pesanan yang masuk hingga siap untuk dikirimkan

Lima fitur untuk skenario pembeli dan penjual ini dipilih berdasarkan tingkat penggunaan oleh pengguna nyata, selain itu fitur-fitur ini memiliki potensi risiko yang dapat terjadi apabila terjadi kegagalan fungsional. Oleh karena itu, setiap fitur akan diuji baik secara manual maupun otomatisasi untuk mendapatkan hasil evaluasi yang objektif dan komprehensif..

Berdasarkan rencana pengujian telah disusun pada tahap sebelumnya, seluruh test case yang mencakup fitur-fitur pembeli dan penjual dalam platform TOCO telah didokumentasikan dalam format tabel dan dimasukkan ke dalam platform dokumentasi pengujian. Dokumentasi tersebut memuat urutan pengujian, langkah-langkah uji, serta hasil yang diharapkan dari setiap skenario pengujian:

Tabel 3. Daftar Dokumentasi *Test Case* Untuk Pembeli

No	Flow	Test Case	Deskripsi Uji
1	Login	Verify user login in login page	Memastikan halaman <i>login</i> dapat diakses oleh pengguna
2	Login	Verify user login with correct email and password	Memastikan pengguna dapat masuk menggunakan kombinasi email dan kata sandi yang valid
3	Register	Verify user register in register page	Memastikan halaman pendaftaran dapat diakses oleh pengguna baru
4	Register	Verify user register with correct email	Memastikan pengguna dapat melanjutkan pendaftaran dengan email valid
5	Register	Verify user register with correct password	Memastikan pengguna dapat melanjutkan pendaftaran dengan <i>password</i> sesuai aturan sistem
6	Membuka Product Detail Page	Verify user able to click product card from homepage	Memastikan pengguna dapat mengklik produk dari halaman utama

No	Flow	Test Case	Deskripsi Uji
7	Membuka Product Detail Page	Verify direct user to pdp after click on product card	Memastikan setelah klik produk, pengguna diarahkan ke halaman detail produk (PDP)
8	Membuka Product Detail Page	Verify user arrive in pdp	Memastikan elemen pada halaman PDP ditampilkan dengan benar

Sedangkan untuk fitur penjual, peneliti memilih proses registrasi toko penjual, melakukan login sebagai penjual, menambahkan produk satu per satu, melakukan integrasi dari *e-commerce* lain dan terakhir melakukan terima pesanan dari pembeli. Berikut bentuk dokumentasi *test case* untuk penjual:

Tabel 4. Daftar Dokumentasi Test Case Untuk Penjual

No	Flow	Test Case	Deskripsi Uji
1	Login	Verify seller login in login page	Memastikan halaman login seller dapat diakses
2	Login	Verify seller login with correct email and password	Memastikan seller dapat masuk menggunakan email dan password yang valid
3	Register	Verify store registration page	Memastikan halaman pendaftaran toko tampil dengan benar
4	Register	Verify user fill store profile	Memastikan seller dapat mengisi dan menyimpan profil toko
5	Menambahkan Produk	Verify seller open form add product	Memastikan seller dapat membuka halaman tambah produk
6	Menambahkan Produk	Verify seller fill product detail	Memastikan seller dapat mengisi detail produk secara lengkap
7	Menambahkan Produk	Verify seller saved product info by click button [Simpan]	Memastikan informasi produk tersimpan setelah klik tombol simpan

Setelah mendaftar fitur dan langkah uji yang akan dijalankan pada tahap perencanaan pengujian, proses selanjutnya adalah menuliskannya ke dalam Microsoft Excel sebagai media dokumentasi. Tujuan dari penulisan ini adalah untuk memudahkan pengorganisasian dan penyuntingan *test case* secara manual.

Setelah seluruh *test case* tersusun, langkah selanjutnya adalah memindahkannya ke dalam format digital menggunakan Microsoft Excel untuk kemudian diunggah ke dalam Testmo, sebuah platform manajemen pengujian. Proses ini meliputi penyusunan struktur *test suite* serta pencatatan nama *test case*, langkah pengujian, dan hasil yang diharapkan, dalam bentuk dokumentasi digital berfungsi sebagai dasar pelaksanaan tes dan referensi saat penilaian.

Dalam pelaksanaan pengujian, peneliti menggunakan dua *environment* utama untuk mewakili dua peran pengguna, yaitu sebagai pembeli dan sebagai penjual. Pemisahan ini bertujuan agar skenario pengujian untuk masing-masing peran dapat dijalankan secara terpisah sesuai dengan fungsi pengguna dalam sistem.

- Untuk pengujian sebagai pembeli, *environment* yang digunakan adalah situs bajatsu.toco.id, yang merepresentasikan tampilan dan alur transaksi dari sisi *front-end* pengguna biasa.
- Untuk pengujian sebagai penjual, *environment* yang digunakan adalah seller.toco.id, yang merupakan halaman *backend* (*seller center*) tempat penjual melakukan aktivitas seperti registrasi toko, menambah produk, mengelola pesanan, dan melakukan integrasi produk dari *marketplace* lain.

Pengujian dilaksanakan menggunakan dua metode, yaitu manual dan otomatisasi, terhadap fitur-fitur utama yang tersedia pada situs TOCO baik sebagai pembeli maupun penjual. Setiap pengujian dijalankan berdasarkan *test case* dan skenario uji yang telah direncanakan sebelumnya, dengan mencatat waktu eksekusi untuk pembandingan efisiensi waktu pengujian serta menuliskan hasil status uji dengan tanda *pass* atau *fail*. Berikut ini adalah ringkasan hasil pengujian berdasarkan waktu eksekusi dan status:

Tabel 5. Hasil Pengujian Sebagai Pembeli – Manual dan Otomatisasi

No.	Fitur yang Diuji	Waktu Manual (detik)	Status	Waktu Otomatisasi (detik)	Status
1	Registrasi Pembeli	58	<i>Pass</i>	38	<i>Pass</i>
2	<i>Login</i> Pembeli	23	<i>Pass</i>	15	<i>Pass</i>
3	Buka Detail Produk	15	<i>Pass</i>	9	<i>Pass</i>
4	Tambah ke Keranjang	20	<i>Pass</i>	13	<i>Pass</i>
5	Checkout Produk	80	<i>Pass</i>	35	<i>Pass</i>

Tabel 6. Hasil Pengujian Sebagai Penjual – Manual dan Otomatisasi

No.	Fitur yang Diuji	Waktu Manual (detik)	Status	Waktu Otomatisasi (detik)	Status
1	Registrasi Penjual	59	<i>Pass</i>	40	<i>Pass</i>
2	<i>Login Seller Center</i>	27	<i>Pass</i>	19	<i>Pass</i>
3	Tambah Produk Manual	129	<i>Pass</i>	57	<i>Pass</i>
4	Terima Pesanan	20	<i>Pass</i>	16	<i>Pass</i>

Berdasarkan paparan tabel hasil pengujian tersebut, seluruh skenario telah dijalankan menggunakan metode manual maupun otomatisasi. Selanjutnya, peneliti melakukan evaluasi terhadap efektivitas pengujian dengan mempertimbangkan waktu pelaksanaan serta tingkat keberhasilan eksekusi setiap *test case*, serta kestabilan sistem TOCO saat dijalankan dari sisi pembeli dan penjual. Hasil paparan tersebut menunjukkan hasil yang memuaskan dengan status *Pass* pada kedua metode pengujian, manual maupun otomatisasi.

4. KESIMPULAN

Hasil pengujian memperlihatkan bahwa semua skenario yang diuji, baik melalui proses manual maupun otomatis, berjalan sesuai perilaku yang diharapkan dan menghasilkan status *Pass*. Tidak ditemukan gangguan fungsional pada fitur yang diuji selama proses tersebut berlangsung. Proses pendaftaran penjual di *Seller Center* TOCO berjalan dengan baik. Selain dari *Seller Center* TOCO menunjukkan hasil yang sama, menjadikan fitur-fitur utama TOCO yang diuji telah berjalan sesuai dengan kebutuhan dasar pengguna. Tidak ditemukan *error* fatal atau *crash* selama pengujian dilakukan.

Selama proses pengujian manual terdapat jeda yang lebih besar disebabkan oleh keterbatasan manusia dalam membaca tampilan dan memproses interaksi UI dibandingkan dengan *script* otomatisasi. Hasil pengujian melalui metode otomatisasi memberikan efisiensi waktu lebih tinggi, skenario pengujian untuk pembeli menghasilkan waktu rata-rata 22 detik dan

skenario penjual memiliki rata-rata selama 33 detik, menunjukkan efisiensi waktu yang cukup memuaskan. Membandingkan dengan pengujian manual selisih hasilnya sekitar 40 persen lebih lama yaitu selama 39 detik untuk skenario pembeli dan skenario penjual selama 59 detik. Hasil kedua pengujian ini menunjukkan otomatisasi unggul dalam waktu pelaksanaan dan mendukung pengujian berulang atau regresi yang memakan waktu cukup lama jika dilakukan secara manual. Lalu pengujian manual unggul memastikan pengalaman pengguna dari UI yang digunakan dalam situs TOCO.

Temuan ini menunjukkan bahwa penerapan tahapan STLC dalam penelitian ini mampu mendukung proses pengujian pada platform *e-commerce* TOCO, terutama dalam memastikan konsistensi fungsi serta membantu mengevaluasi efisiensi sistem secara lebih terstruktur. Penelitian ini juga memberikan kontribusi dalam memberikan gambaran efektivitas penggunaan *automation* testing untuk pengujian regresi serta pembuktian bahwa proses *onboarding*, penambahan produk, migrasi, dan transaksi dapat diuji secara sistematis.

DAFTAR PUSTAKA

- [1] D. S. Prasetyo and W. Silfianti, "Analisis Perbandingan Pengujian Manual Dan Automation Testing Pada Website E-Commerce," *Jurnal Ilmiah Teknik*, vol. 2, no. 2, pp. 127–131, 2023.
- [2] R. Rachman, R. N. Handayani, and I. Artikel, "Klasifikasi Algoritma Naive Bayes Dalam Memprediksi Tingkat Kelancaran Pembayaran Sewa Teras UMKM," *J. Inform*, vol. 8, no. 2, pp. 111–122, 2021.
- [3] M. I. Syarif, M. Hannum, and S. Wahyuni, "Potensi Perkembangan E-Commerce Dalam Menunjang Bisnis di Indonesia," *Journal of Computers and Digital Business*, vol. 2, no. 1, pp. 11–14, 2023.
- [4] U. S. Sulistyawati, "Membangun Keunggulan Kompetitif melalui Platform E-Commerce: Studi Kasus Tokopedia," *Jurnal Manajemen Dan Teknologi*, vol. 1, no. 1, pp. 43–56, 2024.
- [5] "Ecommerce Testing - Website Testing Guide for 2025." Accessed: Oct. 16, 2025. [Online]. Available: <https://bugbug.io/blog/software-testing/ecommerce-testing/>
- [6] "Manual Testing: Effective Techniques for Beginner QA Engineers." Accessed: Oct. 20, 2025. [Online]. Available: <https://sky.pro/wiki/profession/osnovy-ruchnogo-testirovaniya/>
- [7] "10 Solutions to Improve Interface Analyst Efficiency." Accessed: Oct. 20, 2025. [Online]. Available: <https://it-vacancies.ru/blog/10-resenii-dlia-povyseniia-effektivnosti-analitikov-interfeisov/>
- [8] I. R. Dhaifullah, A. A. Salsabila, and M. A. Yaqin, "Survei Teknik Pengujian Software," *Journal Automation Computer Information System*, vol. 2, no. 1, pp. 31–38, 2022.
- [9] A. Arfan and H. Hendrik, "Penerapan STLC dalam Pengujian Black Box dengan Automation Testing Tool (Studi kasus: PT. GIT Solution)," *AUTOMATA*, vol. 3, no. 2, 2022.
- [10] R. Santos, I. Santos, C. Magalhaes, and R. De Souza Santos, "Are We Testing or Being Tested? Exploring the Practical Applications of Large Language Models in Software Testing," *Proceedings - 2024 IEEE Conference on Software Testing, Verification and Validation, ICST 2024*, pp. 353–360, 2024, doi: 10.1109/ICST60714.2024.00039.
- [11] S. Y. Shalsabilla, E. D. Wahyuni, and N. C. Wibowo, "IMPLEMENTASI BLACKBOX AUTOMATION TESTING PADA APLIKASI DONOR MENGGUNAKAN FRAMEWORK STLC DALAM LINGKUP PENGEMBANGAN AGILE SCRUM," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 2, pp. 2261–2269, 2024.
- [12] T. Palmér and M. Waltré, "Automated end-to-end user testing on single page web applications," 2015.

- [13] M. M. Muhtadi, M. D. Friyadi, and A. Rahmani, “Analisis GUI testing pada aplikasi e-commerce menggunakan katalon,” in *Prosiding Industrial Research Workshop and National Seminar*, 2019, pp. 1387–1393.
- [14] M. D. Gustinov *et al.*, “Analysis of web-based e-commerce testing using black box and white box methods,” *International Journal of Information System and Innovation Management (IJISIM)*, vol. 1, no. 1, pp. 20–31, 2023.
- [15] I. Dobles, A. Martínez, and C. Quesada-López, “Comparing the effort and effectiveness of automated and manual tests,” in *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, IEEE, 2019, pp. 1–6.
- [16] M. Kaur, “Software Testing and Quality Assurance,” 2012.
- [17] S. Najihi, S. Elhadi, R. A. Abdelouahid, and A. Marzak, “Software Testing from an Agile and Traditional view,” *Procedia Comput Sci*, vol. 203, pp. 775–782, 2022.
- [18] PT Toco Ramai Digital, “Tentang | Toco,” PT Toco Ramai Digital. Accessed: Jun. 22, 2025. [Online]. Available: <https://toco.id/about>
- [19] E. M. Rahayu and T. S. Pambudi, “Dari Tokobagus ke Toco; Perjalanan Arnold Sebastian di Bisnis e-Commerce.” Accessed: Jun. 23, 2025. [Online]. Available: <https://swa.co.id/read/449341/dari-tokobagus-ke-toco-perjalanan-arnold-sebastian-di-bisnis-e-commerce>
- [20] Ş. Sağiroğlu *et al.*, “Siber güvenlik ve savunma: Problemler ve çözümler,” 2019.