

Article history:

Received: Jun 13, 2025

Accepted: Jul 16, 2025

Published: Oct 2, 2025

ANALISIS PERBANDINGAN AUTOMATION TESTING KATALON STUDIO DAN SELENIUM IDE PADA WEBSITE E-COMMERCE MENGGUNAKAN METODE END-TO-END (STUDI KASUS:ENTERKOMPUTER.COM)

¹⁾Muhammad Andhika Linggar Bhanuwa, ²⁾Miftahul Jannah

^{1,2)} Program Studi Magister Manajemen Sistem Informasi, Jurusan Sistem Informasi Bisnis, Universitas Gunadarma

email: [1muhammadandhika1@gmail.com](mailto:muhammadandhika1@gmail.com), [2pi.miftahul@gmail.com](mailto:pi.miftahul@gmail.com)

Abstract

Enterkomputer.com is an E-Commerce website that offers technology and computer products to customers in Indonesia. In facing the rapid growth of online business and tight competition, Enterkomputer.com developers need to ensure that the website operates well, is free from bugs, and provides an optimal shopping experience for users. This can be achieved by conducting regular web testing. This study aims to compare the performance of software testing automation tools, namely Katalon Studio and Selenium IDE, on the E-Commerce website Enterkomputer.com using the End-to-End testing method. The study was conducted to analyze the time efficiency, reliability, and effectiveness of both tools in detecting bugs and ensuring service quality. Testing was carried out by repeating five times for each test case on the main features such as login, product search, product details, favorites list, product ordering, and logout. The test results show that Katalon Studio excels in terms of intuitive user interface and feature integration, while Selenium IDE is more flexible in supporting various programming languages. This study provides relevant recommendations for developers to choose the appropriate tool based on specific E-Commerce.

Keywords: Automation Testing, Katalon Studio, Selenium IDE, End-to-End testing, E-Commerce.

Abstrak

Enterkomputer.com adalah website E-Commerce yang menawarkan produk-produk teknologi dan komputer kepada pelanggan di Indonesia. Dalam menghadapi pertumbuhan bisnis *online* yang sangat pesat dan persaingan ketat, pengembang Enterkomputer.com perlu memastikan bahwa website beroperasi dengan baik, bebas dari *bug*, dan memberikan pengalaman belanja yang optimal bagi pengguna. Hal tersebut dapat dicapai dengan melakukan pengujian website secara berkala. Penelitian ini bertujuan untuk membandingkan performa alat otomatisasi pengujian perangkat lunak, yaitu Katalon Studio dan Selenium IDE, pada website E-Commerce Enterkomputer.com menggunakan metode *End-to-End testing*. Penelitian dilakukan untuk menganalisis efisiensi waktu, keandalan, serta efektivitas kedua alat dalam mendeteksi *bug* dan memastikan kualitas layanan. Pengujian dilakukan dengan pengulangan lima kali untuk setiap *test case* pada fitur utama seperti *login*, pencarian produk, rincian produk, daftar favorit, pemesanan produk, dan *logout*. Hasil pengujian menunjukkan bahwa Katalon Studio unggul dalam hal antarmuka pengguna yang intuitif dan integrasi fitur, sedangkan Selenium IDE lebih fleksibel dalam mendukung berbagai bahasa pemrograman. Studi ini memberikan rekomendasi yang relevan bagi pengembang untuk memilih alat yang sesuai berdasarkan kebutuhan pengujian spesifik E-Commerce.

Kata Kunci: Pengujian Otomatisasi, Katalon Studio, Selenium IDE, End-to-End testing, E-Commerce.

1. PENDAHULUAN

Website E-Commerce di era digital saat ini menjadi salah satu platform utama bagi bisnis untuk menjual produk dan layanan mereka kepada

pelanggan secara *online*. Peran website E-Commerce sangat vital yang menjadi sarana interaksi utama antara penjual dan pembeli dan berfungsi sebagai etalase *virtual*. Website haruslah

berfungsi dengan baik dan memberikan pengalaman pengguna yang menyenangkan, mulai dari pencarian produk hingga proses pembayaran. *Website E-Commerce* harus memiliki tingkat keandalan yang tinggi yang artinya *website* harus selalu tersedia dan dapat diakses dengan cepat tanpa gangguan atau *downtime* yang signifikan. Gangguan teknis, *bug*, atau kesalahan dalam proses transaksi dapat mengakibatkan kerugian finansial dan menurunkan kepercayaan pelanggan.

Untuk memastikan semua aspek dari *website E-Commerce* berjalan dengan baik, pengujian fungsional diperlukan untuk menjaga kualitas layanan dan pengalaman pengguna. Pengujian bertujuan menganalisis aplikasi guna mendeteksi perbedaan antara kondisi aktual dan yang diharapkan, serta mengidentifikasi kegagalan seperti *error* atau *bug* dalam sistem, sekaligus mengevaluasi fungsionalitas perangkat lunak tersebut [1]. Beberapa perusahaan perangkat lunak mengalokasikan hingga 40% dari sumber dayanya untuk proses pengujian [2]. Tanpa pengujian yang memadai, pengembang tidak akan dapat mengetahui kualitas sistem, yang bisa menyebabkan kesalahan yang merugikan sistem secara keseluruhan [3].

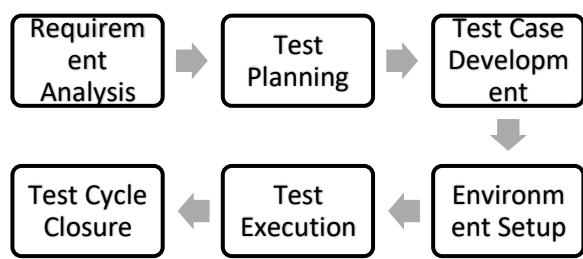
Enterkomputer.com merupakan salah satu *website E-Commerce* yang menawarkan berbagai produk teknologi dan komputer kepada pelanggan di Indonesia. Sejak didirikan pada tahun 2009, Enterkomputer.com fokus pada penjualan komponen komputer dan aksesoris elektronik. Layanan yang tersedia meliputi penjualan *hardware* seperti *prosesor*, *motherboard*, kartu *grafis*, memori, penyimpanan, dan perangkat pendingin serta aksesoris seperti *casing*, *keyboard*, *mouse*, monitor, dan perangkat jaringan. *Website* ini juga menyediakan informasi spesifikasi produk yang lengkap, harga terkini, serta fitur perbandingan produk untuk membantu pengguna memilih komponen yang sesuai dengan kebutuhan. Layanan pemesanan *online* dengan opsi pengiriman ke seluruh Indonesia, dan layanan konsultasi teknis melalui fitur kontak merupakan bagian dari layanan yang ditawarkan.

Enterkomputer.com dalam menghadapi pertumbuhan bisnis *online* yang sangat pesat dan persaingan yang ketat, perlu memastikan bahwa *website* beroperasi dengan baik, bebas dari *bug*, dan memberikan pengalaman belanja yang optimal bagi pengguna, dan untuk hal tersebut dapat dicapai salah satunya dengan melakukan pengujian web secara berkala. Pengujian dapat

dilakukan dengan pendekatan *automated testing* yaitu metode pengujian perangkat lunak yang memanfaatkan alat otomatis untuk melaksanakan skenario pengujian, membandingkan hasil yang diperoleh dengan hasil yang diharapkan, serta menyajikan laporan mengenai kesesuaian perangkat lunak dengan kriteria pengujian yang telah ditentukan sebelumnya [4]. Alat *Automation Testing* yang paling banyak digunakan khususnya pada aplikasi berbasis *website* adalah Katalon studio dan Selenium [5]. Katalon studio memiliki antarmuka pengguna intuitif dan fitur terintegrasi, memudahkan pengembang melakukan pengujian otomatis dengan cepat, bahkan tanpa harus memiliki keterampilan *coding* yang mendalam, dan Selenium adalah *framework open-source* yang sangat fleksibel dan mendukung berbagai bahasa pemrograman, memungkinkan pengujian yang lebih kompleks.

2. METODE PENELITIAN

Tahapan-tahapan dalam penelitian memastikan proses penelitian berlangsung secara sistematis dan terstruktur. Tahapan penelitian ini merupakan bagian dalam metode *Software Testing Life Cycle* (STLC). Metode ini merupakan kerangka kerja yang sistematis untuk melakukan pengujian perangkat lunak [6].



Gambar 1. Tahapan Penelitian

Diagram alir tahapan penelitian pada gambar 1. terdapat 6 (enam) tahapan dalam STLC yang harus dilalui dalam penelitian ini. Tahapan dimulai dengan *Requirement Analysis*, dimana pengujian dimulai dengan mengumpulkan dan menganalisis persyaratan untuk menentukan apa yang perlu diuji, setelah analisis selesai, langkah berikutnya adalah *Test Planning*, yaitu merencanakan strategi pengujian dan mendefinisikan ruang lingkup serta sumber daya yang diperlukan. Selanjutnya *Test case Development*, yaitu mengembangkan dan mendokumentasikan *test case* yang akan digunakan untuk pengujian. Setelah *test case* selesai dikembangkan, tahap berikutnya adalah *Environment Setup*. Pada tahap ini menyiapkan

lingkungan pengujian yang diperlukan untuk menjalankan *test case*. Setelah lingkungan siap, dilakukan *Test Execution* dengan melaksanakan *test case* yang telah dirancang, termasuk pengujian fungsional untuk memastikan bahwa semua fitur berfungsi sesuai dengan spesifikasi. Setelah semua *test case* dieksekusi, maka tahap terakhir melakukan *Test Closure* yaitu menyusun laporan hasil pengujian dan mendokumentasikan temuan serta perbaikan yang diperlukan [7].

2.1. Requirement Analysis

Analisis kebutuhan (*requirement analysis*) adalah tahap pertama dari STLC. Tahapan ini diawali dengan menentukan obyek yang akan diuji. Enterkomputer.com menjadi obyek dalam pengujian. Pada bab sebelumnya dijelaskan bahwa Enterkomputer.com merupakan web *E-Commerce* yang berfokus pada penjualan yang berkaitan dengan komputer dan komponen-komponennya. Selama lebih dari 10 tahun sejak berdirinya Enterkomputer.com, website ini dilengkapi dengan berbagai fitur untuk memudahkan navigasi pengguna mengakses *webpage* yang disediakan antara lain Katalog Produk, Filter Pencarian, Rincian Produk, Daftar Harga, *Checkout*, Fitur Favorite Barang, Notifikasi Stock, Chat *Online*, Cek Status Pesanan dan Fitur *Login* atau Registrasi Akun.

2.2. Test Planning

Langkah berikutnya adalah *Test Planning*, yaitu merencanakan strategi pengujian dan mendefinisikan ruang lingkup serta sumber daya yang diperlukan. Pengujian web *E-Commerce* Enterkomputer.com menggunakan metode *End-to-End* dan memanfaatkan dua perangkat *Automation Testing* yaitu Katalon Studio dan Selenium IDE. Pengujian secara otomatis dilakukan berdasarkan kasus uji (*test case*) fitur yang telah disiapkan.

Metode *End-to-End* memungkinkan untuk menguji seluruh proses mulai dari pembuatan skrip, eksekusi pengujian, hingga analisis hasil dalam lingkungan yang sesungguhnya [8]. Setiap langkah dalam siklus pengujian dapat dievaluasi secara mendetail, sehingga memberikan data yang komprehensif mengenai efisiensi, efektivitas, dan keandalan kedua alat [9].

2.3. Test case Development

Pada tahapan pembuatan dan pengembangan kasus uji (*Test case Development*), langkah pertama adalah menyiapkan SUT (*Software*

Under Test) untuk pengujian terdiri dari pembuatan dan dokumentasi kasus uji yang akan diuji, termasuk halaman-halaman spesifik dari situs web *E-Commerce* Enterkomputer.com. Fokus pengujian mengacu pada halaman-halaman yang sering diakses oleh pengguna saat melakukan pemesanan produk antara lain Halaman *Login*, Halaman Pencarian Produk, Halaman Rincian Produk, Halaman Daftar Favorite, Halaman Pemesanan Produk, dan *Logout*. Setiap halaman akan diuji menggunakan *Automation Testing* secara terstruktur untuk memastikan fungsionalitas dari setiap halaman web [10].

2.4. Environment Setup

Tahap ini menyiapkan lingkungan pengujian mencakup konfigurasi perangkat keras dan perangkat lunak yang diperlukan untuk menjalankan pengujian. Sistem yang digunakan untuk menjalankan alat pengujian harus memenuhi spesifikasi minimum, termasuk sistem operasi yang kompatibel, jumlah RAM yang cukup, serta ruang penyimpanan yang memadai.

2.5. Test Execution

Setelah kasus uji telah disiapkan, langkah berikutnya menjalankan pengujian (*test execution*) untuk setiap halaman. Pada tahap ini menjalankan *test case* yang telah disiapkan termasuk pengujian fungsional untuk memastikan bahwa semua fitur berfungsi sesuai dengan spesifikasi. Semua *test case* diuji sebanyak 5 kali percobaan di masing-masing alat uji yang digunakan. Setiap hasil pengujian akan dibandingkan dengan hasil uji yang diharapkan. Jika hasil uji sesuai dengan yang diharapkan, maka status pengujian akan ditandai Berhasil/Positive. Namun, jika hasilnya tidak sesuai dengan yang diharapkan, maka status uji akan ditandai sebagai Tidak Berhasil/Negatif.

2.6. Test Closure

Setelah semua test dieksekusi, tahapan terakhir yang akan dilakukan yaitu membuat dokumentasi hasil keseluruhan dari pengujian yang dilakukan untuk memastikan bahwa semua tujuan pengujian telah selesai semua dan menarik kesimpulan dari hasil pengujian yang telah dilakukan semuanya.

2.7. Automated Software Testing

Automated Software Testing adalah metode pengujian perangkat lunak dilakukan secara otomatis menggunakan alat atau

program khusus. Perbedaan dengan manual *testing*, penguji (*tester*) menjalankan tes secara manual, langkah demi langkah. *Automated testing* memungkinkan pengujian yang cepat, konsisten, dan lebih efektif terutama pada tugas-tugas pengujian berulang atau yang membutuhkan eksekusi dalam jumlah besar [11].

3. HASIL DAN PEMBAHASAN

3.1. Hasil *Test case Development*

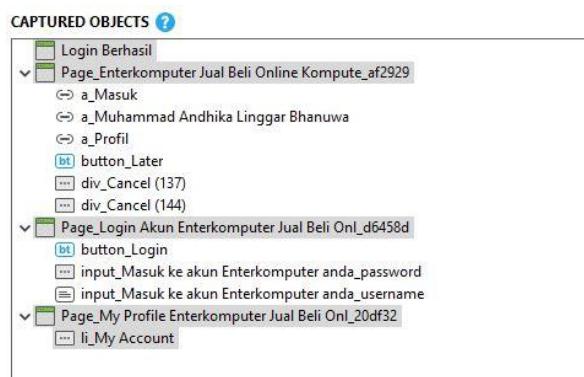
Perancangan dan pengembangan kasus uji dilakukan dengan membuat *test case* secara langsung di dalam perangkat uji. Tabel *scenario test* terlebih dahulu dibuat menggunakan aplikasi Microsoft Excel dengan elemen-elemen uji yang sesuai dengan standart ISO/IEC/IEEE 29119-3:2021 [12]. Pada Tabel 1 ditampilkan 13 (tiga belas) daftar *test case*.

Tabel 1. Daftar *Test case*

Test_ID	Test case Name
TC01	<i>Login Gagal (Email Invalid)</i>
TC02	<i>Login Gagal (Password Invalid)</i>
TC03	<i>Login Berhasil</i>
TC04	Hasil pencarian tidak ada
TC05	Hasil pencarian ditemukan
TC06	Halaman rincian produk ditemukan
TC07	Menambahkan produk ke daftar <i>favorite</i>
TC08	Melihat produk yang didaftarkan ke <i>favorite</i>
TC09	Menghapus barang dari produk <i>favorite</i>
TC10	Membeli produk menggunakan <i>whatsapp</i>
TC11	Membeli produk menggunakan <i>tokopedia</i>
TC12	Membeli produk menggunakan <i>shopee</i>
TC13	<i>Logout</i>

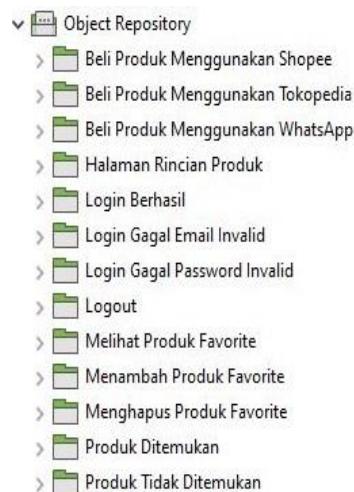
3.2. Hasil *Test case Development* Menggunakan Katalon Studio

Test case automation menggunakan Katalon Studio diawali dengan merekam pengambilan obyek yang diperlukan dalam pembuatan kasus uji menggunakan *record object*. Penggunaan *record object* mempermudah dalam pembuatan kasus uji tanpa menulis kode secara manual dan dapat meminimalisir kesalahan manusia seperti pada Gambar 2.



Gambar 2. *Record Object* di Katalon Studio

Perekaman obyek menggunakan *record object* akan diidentifikasi dan disimpan dalam *object repository* untuk selanjutnya digunakan dalam pengujian tanpa melakukan perekaman kembali. Detail *object repository* ditampilkan pada Gambar 3 berikut ini.



Gambar 3. *Object Repository* di Katalon Studio

Obyek yang telah diambil selanjutnya dibuat *test case*. Obyek diberikan *keyword* sesuai dengan skenario pengujian yang telah dibuat. *Keyword* yang ada di Katalon Studio cukup lengkap dan fleksibel, sehingga mempermudah pembuatan kasus uji.

Pembuatan *test case* dilakukan dengan menggunakan fitur *record object* untuk merekam elemen-elemen pada aplikasi yang diuji, seperti tombol, kolom teks, atau *dropdown*, dan menyimpannya sebagai objek di *repository* Katalon. Gambar 4 menampilkan hasil pembuatan *test case* TC01 *Login Gagal Email Invalid* di Katalon Studio menggunakan *record object*.

Item	Object	Input
→ 1 - Open Browser		""
→ 2 - Navigate To Url		"https://www.enterkomputer.com/
→ 3 - Wait For Page Load		15
→ 4 - Click	div_Cancel (131)	
→ 5 - Click	button_Later	
→ 6 - Click	a_Masuk	
→ 7 - Verify Element Text	input_Masuk ke akun Enterkomputer	""
→ 8 - Set Text	input_Masuk ke akun Enterkomputer	"muhammadandhika@gmail.com"
→ 9 - Verify Element Text	input_Masuk ke akun Enterkomputer	""
→ 10 - Set Encrypted Text	input_Masuk ke akun Enterkomputer	"G1Vut0Huh3B1jY57ssxA+
→ 11 - Click	button_Login	
→ 12 - Verify Element Present	li_AkunEmail tidak ditemukan	0
→ 13 - Close Browser		

Gambar 4. Hasil *Test case* di Katalon Studio

3.3. Pembuatan Test Suite Pada Katalon Studio

Setelah menjalankan *test case* dari TC01 sampai TC13, tahapan berikutnya adalah menjalankan *test suite* untuk mengelompokkan dan menjalankan *test case* dalam satu eksekusi secara bersamaan. *Test suite* memungkinkan pengujian otomatis terhadap berbagai skenario dalam satu rangkaian dan pengembang dapat menjalankan pengujian berulang kali pada berbagai lingkungan atau *platform*, meningkatkan kualitas perangkat lunak sekaligus menghemat waktu dan sumber daya sehingga mempermudah pengelolaan dan pelacakan hasil pengujian.

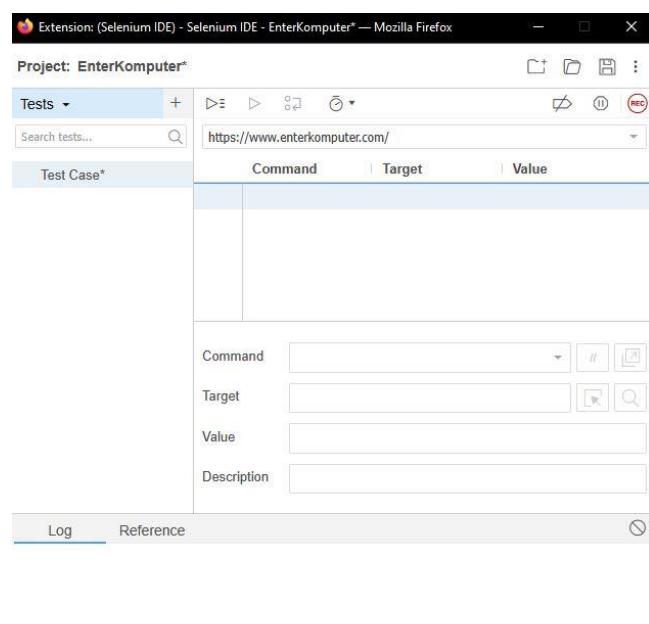
EnterKomputer X			
Execution Information			
Add Delete Move Up Move Down			
Enter text to search...			
No.	ID	Description	Flaki
1	Test Cases/TC-01 Login Gagal Email Invalid		0.00
2	Test Cases/TC-02 Login Gagal Password Invalid		0.00
3	Test Cases/TC-03 Login berhasil		0.00
4	Test Cases/TC-04 Hasil Pencarian Tidak Ada		0.00
5	Test Cases/TC-05 Hasil Pencarian Muncul		0.00
6	Test Cases/TC-06 Halaman Rincian Produk		0.00
7	Test Cases/TC-07 Menambahkan Produk Ke Daftar Favorite		0.00
8	Test Cases/TC-08 Melihat Produk Favorite		0.00
9	Test Cases/TC-09 Menghapus Barang Dari Favorite		0.00
10	Test Cases/TC-10 Beli Produk Menggunakan WhatsApp		0.00
11	Test Cases/TC-11 Beli Produk Menggunakan Tokopedia		0.00
12	Test Cases/TC-12 Beli Produk Menggunakan Shopee		0.00
13	Test Cases/TC-13 Logout		0.00

Gambar 5. *Test Suite* Pada Katalon Studio

Gambar 5 menampilkan 13 *test case* yang akan dijalankan di dalam *test suite*. Setelah semua test dimasukkan, pengguna mengatur urutan eksekusi sesuai dengan kebutuhan. Hasil pengujian akan ditampilkan secara real time di *Log viewer* yang mencatat langkah-langkah pengujian dan menampilkan status keberhasilannya.

3.4. Hasil *Test Case Development* Menggunakan Selenium IDE

Test case automation menggunakan Selenium IDE menggunakan langkah *record and playback* dalam melakukan pengujian. Selenium IDE menyimpan secara otomatis tanpa menulis kode secara manual.

Gambar 6. *Record and Playback* di Selenium IDE

Record and playback berkerja dengan cara mengaktifkan mode perekaman di Selenium IDE. Setiap tindakan yang dilakukan di *browser* dicatat sebagai skrip uji dalam bentuk perintah. Setelah selesai, skrip tersebut disimpan dan diputar ulang untuk mengulangi pengujian dengan langkah yang sama. Selama pemutaran ulang, Selenium menjalankan langkah-langkah yang tercatat secara otomatis untuk memverifikasi bahwa fungsi aplikasi berjalan sesuai harapan.

Pembuatan *test case* menggunakan fitur *Record and Playback* cukup merekam langkah-langkah interaksi dengan aplikasi *website*, seperti mengklik tombol atau mengisi formulir, lalu memutar ulang rekaman tersebut untuk otomatisasi pengujian. Hasil pembuatan *test case* TC 01 *Login Gagal Email Invalid* di Selenium IDE menggunakan *Record and Playback* ditampilkan pada Gambar 7. berikut ini.

Command	Target	Value
1. open	https://www.enterkomputer.co m/	
2. set window size	1324x880	
3. click	linkText=Masuk	
4. click	name=username	
5. type	name=username	muhammadandhika@gm m
6. click	name=password	
7. type	name=password	SENJU1995
8. click	name=login	
9. click	xpath=//li[contains(.,'Akun/Email tidak ditemukan')]	
10. close		

Gambar 7. Test case di Selenium IDE

3.5. Pembuatan Test Suite Pada Selenium IDE

Pembuatan *test suite* pada Selenium IDE memiliki tahapan yang sama dengan tahapan pada katalon studi. Setelah menjalankan *test case* dari TC 01 sampai TC 13, tahapan berikutnya adalah menjalankan *test suite* untuk mengelompokkan dan menjalankan *test case* dalam satu eksekusi secara bersamaan. Pada Gambar 8, ditampilkan 13 *test case* yang akan dijalankan dimasukkan ke dalam *test suite*.

Test suites	+	⟳	⟳	⟳	⟳	⟳
Search tests...	🔍	https://www.enterkomputer.com/	Command	Target	Value	①
EnterKomputer*						
TC-01 Login Gagal Karena Email Invalid						
TC-02 Login Gagal Password Invalid						
TC-03 Login Berhasil						
TC-04 Hasil Pencarian Tidak Ada						
TC-05 Hasil Pencarian Muncul						
TC-06 Halaman Rincian Produk						
TC-07 Menambahkan Produk Ke Daftar Favorit						
TC-08 Melihat Produk Favorite						
TC-09 Menghapus Barang Dari Favorite						
TC-10 Beli Produk Menggunakan Whatsapp						
TC-11 Beli Produk Menggunakan Tokopedia			Command			
TC-12 Beli Produk Menggunakan Shopee			Target			
TC-13 Logout						

Gambar 8. Test Suite Pada Selenium IDE

Gambar 8 menampilkan 13 *test case* yang akan dijalankan di dalam *test suite*. Setelah semua test dimasukkan, pengguna mengatur urutan eksekusi sesuai dengan kebutuhan. Hasil pengujian akan ditampilkan secara *real time* di *Log viewer* yang mencatat langkah-langkah pengujian dan menampilkan status keberhasilannya

3.6. Hasil Test Execution Menggunakan Katalon Studio

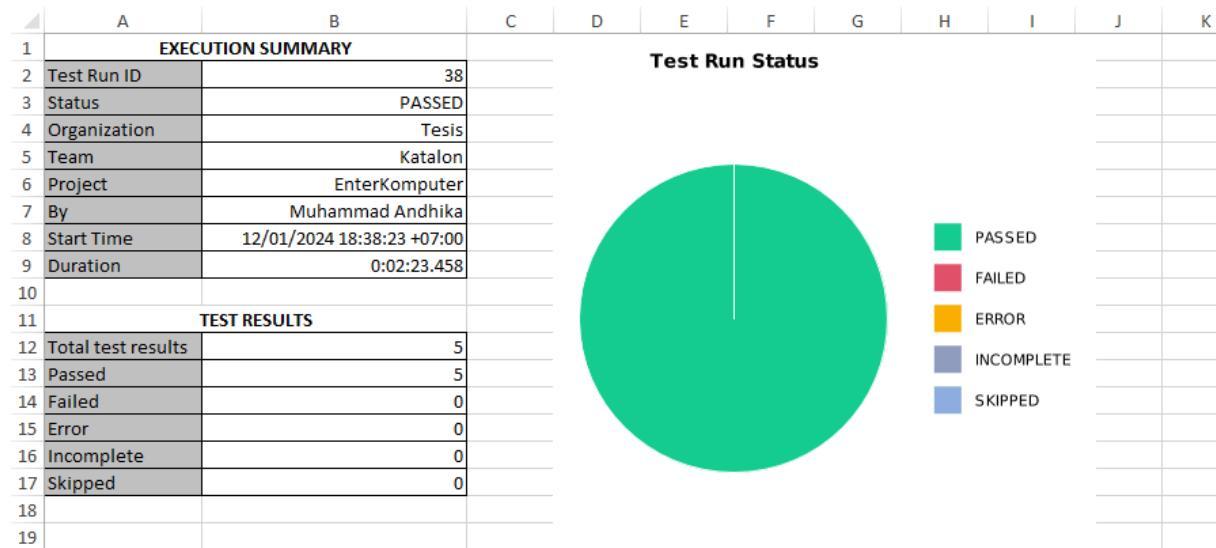
Proses pengujian *test case Login Gagal Email Invalid* berjalan sesuai dengan scenario yang telah dibuat dan ditampilkan pada Gambar 9 dan Gambar 10.

Problems	Event Log	Console	Log Viewer	Self-healing Insights
Runs: 5/5	Passes: 5	Failures: 0	Errors: 0	Skips: 0
<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> ✓ Test Cases/TC-01 Login Gagal Email Invalid (13.802) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 1 - openBrowser("") (5.088s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 2 - navigateToUrl("https://www.enterkomputer...") (0.493s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 3 - waitForPageLoad(15) (0.493s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 4 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 5 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 6 - verifyElementText(findTestObject("Object Re..."), "Email") (0.041s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 7 - setText(findTestObject("Object Repository/Log..."), "SENJU1995") (0.041s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 8 - verifyElementText(findTestObject("Object Re..."), "SENJU1995") (0.041s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 9 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 10 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 11 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 12 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 13 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 14 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 15 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 16 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 17 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 18 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 19 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 20 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 21 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 22 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 23 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 24 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 25 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 26 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 27 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 28 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 29 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 30 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 31 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 32 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 33 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 34 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 35 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 36 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 37 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 38 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 39 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 40 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 41 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 42 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 43 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 44 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 45 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 46 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 47 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 48 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 49 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 50 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 51 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 52 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 53 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 54 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 55 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 56 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 57 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 58 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 59 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 60 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 61 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 62 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 63 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 64 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 65 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 66 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 67 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 68 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 69 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 70 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 71 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 72 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 73 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 74 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 75 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 76 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 77 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 78 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 79 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 80 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 81 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 82 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 83 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 84 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 85 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 86 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 87 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 88 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 89 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 90 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 91 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 92 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 93 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 94 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 95 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 96 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 97 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 98 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 99 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 100 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 101 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 102 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 103 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 104 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 105 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 106 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 107 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 108 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 109 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 110 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 111 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 112 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 113 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 114 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 115 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 116 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 117 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 118 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 119 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 120 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 121 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 122 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 123 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 124 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 125 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 126 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 127 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 128 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 129 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 130 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 131 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 132 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 133 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 134 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 135 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;"> ✓ 136 - click(findTestObject("Object Repository/Log...")) (0.167s) </div> <div style="border: 1px solid #ccc; padding: 2px				

Test Results	Passed	Failed	Error	Incomplete	Skipped
5	5	0	0	0	0

Gambar 11. Katalon TestOps Hasil Pengujian Form *Login Gagal Email Invalid*

A	B	C	D	E	F	G	H	I	J	K	L	M	N
Test Result ID	Test Case	Path	Test Suite	Profile	OS	OS Version	Browser	Browser Version	Status	Error Message	Start Time	End Time	Duration
2	226589635 TC-01 Login Gagal Email Invalid	Test Cases	EnterKomputer	default	Windows 10	Windows 10 64bit	Firefox	133.0	PASSED		12/01/2024 18:38:23 +07:00	12/01/2024 18:38:54 +07:00	0:00:30.838
3	226589636 TC-01 Login Gagal Email Invalid	Test Cases	EnterKomputer	default	Windows 10	Windows 10 64bit	Firefox	133.0	PASSED		12/01/2024 18:38:54 +07:00	12/01/2024 18:39:28 +07:00	0:00:34.736
4	226589637 TC-01 Login Gagal Email Invalid	Test Cases	EnterKomputer	default	Windows 10	Windows 10 64bit	Firefox	133.0	PASSED		12/01/2024 18:39:28 +07:00	12/01/2024 18:39:55 +07:00	0:00:26.165
5	226589638 TC-01 Login Gagal Email Invalid	Test Cases	EnterKomputer	default	Windows 10	Windows 10 64bit	Firefox	133.0	PASSED		12/01/2024 18:39:55 +07:00	12/01/2024 18:40:22 +07:00	0:00:27.045
6	226589639 TC-01 Login Gagal Email Invalid	Test Cases	EnterKomputer	default	Windows 10	Windows 10 64bit	Firefox	133.0	PASSED		12/01/2024 18:40:22 +07:00	12/01/2024 18:40:46 +07:00	0:00:24.674

Gambar 12. Hasil Pengujian Sheet Test Results Form *Login Gagal Email Invalid*Gambar 13. Hasil Pengujian Sheet Overview Form *Login Gagal Email invalid*

Pada gambar 13 menampilkan hasil *test case* berjalan lancar tanpa ditemukan *error* pada saat melakukan pengujian sebanyak 5 kali pengujian dan membutuhkan waktu selama 2 menit 23 detik.

Automation Testing menggunakan Katalon Studio menghasilkan data pengujian *test case* pertama (TC01) sampai *test case* terakhir (TC13). Hasil pengujian dengan percobaan sebanyak 5 (lima) kali (P1 – P5) yang ditampilkan dalam satuan detik ditampilkan pada Tabel 2.

Tabel 2. Hasil Pengujian Menggunakan Katalon Studio

Test Case	P1	P2	P3	P4	P5	Rata-Rata
TC01	30	34	26	27	24	28,2
TC02	31	28	26	26	26	27,4
TC03	38	36	29	29	29	32,2
TC04	40	37	35	37	34	36,6
TC05	36	29	29	30	31	31
TC06	41	35	36	37	36	37
TC07	41	35	35	34	36	36,2
TC08	36	32	36	34	33	34,2
TC09	39	36	34	35	34	35,6

Test Case	P1	P2	P3	P4	P5	Rata-Rata
TC10	46	44	48	42	40	44
TC11	51	41	39	40	41	42,4
TC12	58	54	53	55	51	54,2
TC13	35	33	32	32	33	33
TOTAL	522	474	458	458	448	472

Ket. P: Percobaan (detik)

Tabel 2 menunjukkan bahwa hasil Automation Testing menggunakan Katalon

Log	Reference	
Running 'TC-01 Login Gagal Karena Email Invalid'		
1. open on https://www.enterkomputer.com/ OK		20:38:27
2. setWindowSize on 1324x880 OK		20:38:27
3. click on linkText=Masuk OK		20:38:30
4. click on name=username OK		20:38:34
5. type on name=username with value muhammadandhika@gmail.com OK		20:38:37
6. click on name=password OK		20:38:41
7. type on name=password with value SENJU1995 OK		20:38:44
8. click on name=login OK		20:38:47
9. click on xpath=/li[contains(.,'Akun/Email tidak ditemukan')] OK		20:38:50
10. close OK		20:38:54
'TC-01 Login Gagal Karena Email Invalid' completed successfully		

Gambar 14. Log Viewer Proses Pengujian Form Login Gagal Email Invalid

Pada gambar 14 menampilkan tampilan log viewer sedang melakukan proses pengujian *test case* secara hierarki pengujian dari skrip *test case* *Login Gagal Email Invalid* yang telah dibuat.

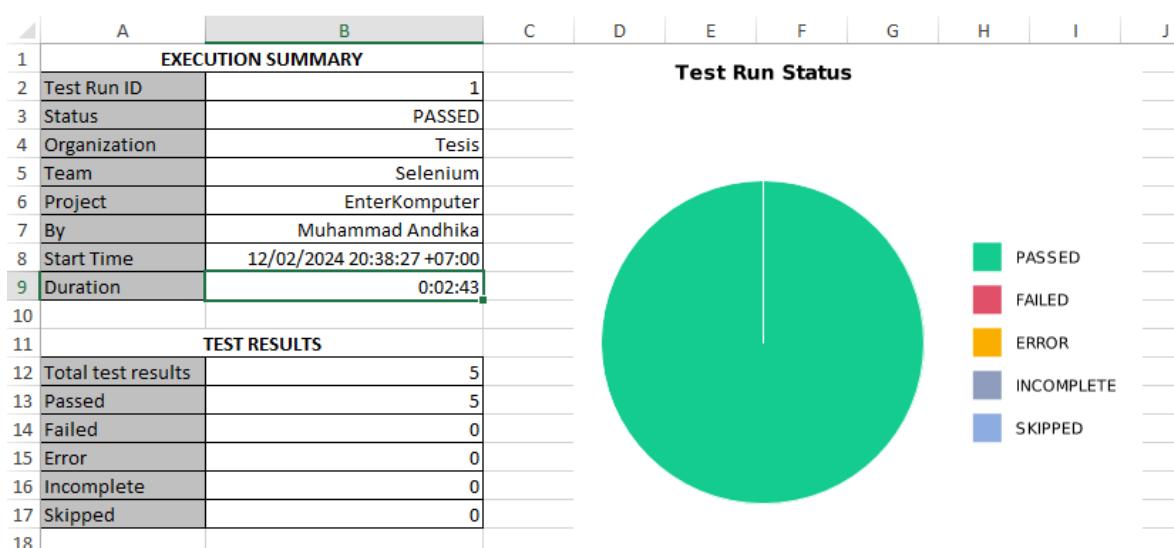
Studio TC01 sampai dengan TC13 berhasil dilaksanakan sesuai dengan skenario dan menghasilkan *test case* berhasil (*PASSED*). Total waktu rata-rata yaitu 472 detik atau 1 jam 31 menit.

3.7. Hasil Test Execution Menggunakan Selenium IDE

Proses pengujian *test case login* gagal karena *email invalid* berjalan sesuai dengan scenario yang telah dibuat ditampilkan pada Gambar 14.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	
1	Test Result ID	Test Case	Path	Test Suite	Profile	OS	OS Version	Browser	Browser Version	Status	Error Message	Start Time	End Time	Duration
2	1 TC-01 Login Gagal Email Invalid	Test Cases	EnterKomputer	default	Windows 10	Windows 10	64bit	Firefox	133.0	PASSED		12/02/2024 20:38:27 +07:00	12/02/2024 20:38:59 +07:00	0:00:33
3	2 TC-01 Login Gagal Email Invalid	Test Cases	EnterKomputer	default	Windows 10	Windows 10	64bit	Firefox	133.0	PASSED		12/02/2024 20:44:02 +07:00	12/02/2024 20:44:34 +07:00	0:00:32
4	3 TC-01 Login Gagal Email Invalid	Test Cases	EnterKomputer	default	Windows 10	Windows 10	64bit	Firefox	133.0	PASSED		12/02/2024 20:45:19 +07:00	12/02/2024 20:45:50 +07:00	0:00:32
5	4 TC-01 Login Gagal Email Invalid	Test Cases	EnterKomputer	default	Windows 10	Windows 10	64bit	Firefox	133.0	PASSED		12/02/2024 20:46:22 +07:00	12/02/2024 20:46:54 +07:00	0:00:32
6	5 TC-01 Login Gagal Email Invalid	Test Cases	EnterKomputer	default	Windows 10	Windows 10	64bit	Firefox	133.0	PASSED		12/02/2024 20:48:22 +07:00	12/02/2024 20:48:53 +07:00	0:00:32

Gambar 15 Hasil Pengujian Sheet Test Results Form Login Gagal Email Invalid



Gambar 16. Hasil Pengujian Sheet Overview Form Login Gagal Email invalid

Pada gambar 16 menampilkan hasil *test case* berjalan lancar tanpa ditemukan *error* pada saat melakukan pengujian sebanyak 5 kali pengujian dan membutuhkan waktu selama 2 menit 47 detik.

Automation Testing menggunakan Selenium IDE menghasilkan data pengujian *test case* pertama (TC01) sampai *test case* terakhir (TC13). Hasil pengujian dengan percobaan sebanyak 5 (lima) kali (P1 – P5) yang ditampilkan dalam satuan detik ditampilkan pada Tabel 3.

Tabel 3. Hasil Pengujian Menggunakan Selenium IDE

Test Case	P1	P2	P3	P4	P5	Rata-Rata
TC01	33	33	32	33	32	32,6
TC02	36	34	33	32	32	33,4
TC03	40	39	39	39	40	39,4
TC04	60	54	56	55	55	56
TC05	57	56	55	58	55	56,2
TC06	62	63	62	61	61	61,8
TC07	76	68	68	68	69	69,8
TC08	54	53	53	53	54	53,4
TC09	64	65	67	66	65	65,4
TC10	74	73	72	71	71	72,2
TC11	73	73	73	73	72	72,8
TC12	73	72	73	73	74	73
TC13	60	60	60	60	59	59,8
TOTAL	762	743	743	742	739	745,8

Ket. P: Percobaan (detik)

Tabel 3 menunjukkan bahwa hasil *Automation Testing* menggunakan Selenium IDE, TC01 sampai dengan TC13 berhasil dilaksanakan sesuai dengan skenario dan menghasilkan *test case* berhasil (*PASSED*). Total waktu rata-rata yaitu 745,8 detik atau 2 jam 1 menit.

3.8. Hasil Perbandingan Pengujian Katalon Studio dan Selenium IDE

Semua *test case* telah selesai dijalankan menggunakan *Automation Testing* Katalon Studio dan Selenium IDE. Perbandingan hasil pengujian dari dua *automation test* berdasarkan pada nilai rata-rata 13 (tiga belas) *test case* yang telah dilakukan. Nilai setiap *test case* diambil nilai terkecil dari 5 kali percobaan yang dilakukan. Perbandingan hasil pengujian ditampilkan pada Tabel 4.

Tabel 4. Hasil Perbandingan Pengujian

Test Case	Rata-Rata (detik)		Hasil
	Katalon Studio	Selenium IDE	
TC01	28,2	32,6	Katalon Studio
TC02	27,4	33,4	Katalon Studio
TC03	32,2	39,4	Katalon Studio
TC04	36,6	56	Katalon Studio
TC05	31	56,2	Katalon Studio
TC06	37	61,8	Katalon Studio
TC07	36,2	69,8	Katalon Studio
TC08	34,2	53,4	Katalon Studio
TC09	35,6	65,4	Katalon Studio
TC10	44	72,2	Katalon Studio
TC11	42,4	72,8	Katalon Studio
TC12	54,2	73	Katalon Studio
TC13	33	59,8	Katalon Studio
TOTAL	472	745,8	Katalon Studio

Pada Tabel 4 menampilkan hasil pengujian 13 (tiga belas) *test case* yang dilakukan yaitu TC01 - TC13 menggunakan Katalon Studio dan Selenium IDE. Secara keseluruhan total waktu yang dibutuhkan Katalon Studio melakukan *automation test* pada situs enterkomputer.com yaitu 472 detik atau 1 jam 31 menit dan Selenium IDE 745,8 detik atau 2 jam 1 menit artinya Katalon Studio lebih cepat 30 menit dibandingkan Selenium IDE.

3.8. Analisis Hasil Pengujian

Interpretasi data diperoleh dari *test case* yang dilakukan menggunakan 2 (dua) *automation test* yaitu Katalon Studio dan Selenium IDE. Pengujian pada 13 (tiga belas) *test case* yang meliputi fitur utama antara lain *login*, pencarian produk, rincian produk, daftar favorit, pembelian melalui platform berbeda, dan *logout* masing-masing diuji sebanyak 5 (lima) kali percobaan dan hasil pengujian yang didapatkan menunjukkan 13 *test*

case berhasil dijalankan tanpa kendala di kedua alat uji tersebut.

Pengujian menggunakan Katalon Studio menunjukkan keunggulan dari segi kemudahan penggunaan berkat antarmuka grafis yang intuitif dan menunjukkan fleksibilitas yang lebih tinggi dalam mendukung skenario pengujian kompleks dengan fitur *Record Object*. Fitur ini memungkinkan pengguna untuk menulis skrip pengujian secara manual dan menjadikannya pilihan yang baik untuk pengembang berpengalaman yang memerlukan tingkat kontrol lebih tinggi. Selain itu, pengelolaan kasus uji melalui *test suite* memungkinkan pelaksanaan pengujian secara terstruktur dan efisien. Di sisi lain, Selenium IDE mempunyai fitur *Record and Playback* yang dapat disesuaikan dengan kebutuhan pengguna dalam mempermudah pembuatan kasus uji tanpa perlu menulis kode secara manual. Fitur ini sangat mendukung pengguna yang tidak memiliki latar belakang pemrograman yang mendalam.

Hasil pengujian pada sistem *E-Commerce* enterkomputer.com yang dilakukan dalam penelitian ini menunjukkan bahwa Katalon Studio dan Selenium IDE mampu memberikan performa yang andal dalam mendeteksi kesalahan atau *bug* pada fitur-fitur yang diuji. Kedua alat uji berhasil menampilkan hasil yang konsisten di semua skenario, baik untuk kasus positif maupun negatif. Setiap langkah pengujian dapat dijalankan tanpa terjadi kesalahan (*error*). Keunggulan ini menunjukkan bahwa kedua alat dapat diandalkan untuk kebutuhan pengujian otomatis pada sistem *E-Commerce* dengan kompleksitas tinggi seperti Enterkomputer.com.

4. PENUTUP

Kesimpulan

Website Enterkomputer.com sebagai web *E-Commerce* harus memiliki tingkat keandalan yang tinggi yaitu harus selalu tersedia dan dapat diakses dengan cepat tanpa gangguan atau *downtime* yang signifikan yang dapat menurunkan kepercayaan pelanggan. Pengujian *E-Commerce* Enterkomputer.com menggunakan *Automation Testing* yaitu Katalon Studio dan Selenium IDE dilakukan untuk membantu dalam pengambilan keputusan terkait penggunaan alat yang lebih efektif digunakan. Berdasarkan hasil analisis perbandingan pengujian yang telah dilakukan sebanyak 5 kali percobaan untuk

masing-masing alat uji pada website Enterkomputer.com disimpulkan bahwa 13 *test case* yang digunakan dalam skenario *test* berhasil dilakukan tanpa ada kendala ataupun *error* pada setiap skenario yang dilakukan.

Perbandingan berdasarkan waktu eksekusi yang digunakan rata-rata waktu eksekusi Katalon Studio pada *test case* TC 01 sampai TC 13 lebih unggul dibandingkan Selenium IDE. Secara keseluruhan Katalon Studio lebih cepat 30 menit dibandingkan Selenium IDE. Berdasarkan hasil tersebut menunjukkan bahwa Katalon memiliki rata-rata waktu eksekusi yang dibutuhkan lebih cepat sehingga dapat digunakan dalam melakukan pengujian web *E-Commerce* enterkomputer.com.

Saran

Analisis perbandingan *Automation Testing* Katalon Studio dan Selenium IDE pada website *E-Commerce* enterkomputer.com menggunakan metode *End-to-End* pada penelitian ini dilakukan berdasarkan 13 (tiga belas) *test case* yang direncanakan. Pengujian pada website *E-Commerce* Enterkomputer dapat ditingkat lagi dengan memperluas lingkup *test case* secara menyeluruh mencakup pengujian terhadap semua fitur utama, seperti pencarian produk, proses *checkout*, pengelolaan keranjang belanja, *autentifikasi* pengguna, hingga validasi diskon dan promosi, dan pengujian untuk skenario *edge case*, seperti pembayaran gagal atau kesalahan input data, untuk memastikan aplikasi tahan terhadap kondisi ekstrem.

Pengujian secara manual dapat dilakukan sebagai pelengkap untuk mengidentifikasi potensi masalah yang mungkin terlewatkan oleh skrip otomatisasi, terutama pada elemen antarmuka pengguna dan pengalaman pengguna secara keseluruhan.

Perbandingan pengujian versi *mobile* dan *desktop* dapat dilakukan untuk memastikan *responsivitas* dan konsistensi fungsi di berbagai perangkat. Hal ini penting untuk menjaga kualitas pengalaman pengguna lintas *platform*, mengingat banyak pelanggan yang menggunakan perangkat berbeda untuk

5. REFERENSI

- [1] A. C. Barus and L. Siburian, "STUDI PERBANDINGAN ALAT PENGUJIAN OTOMATIS UNTUK APLIKASI ANDROID," vol. 6, no. 6, pp. 645–654, 2019, doi: 10.25126/jtiik.20196953.
- [2] A. Zulianto, A. Purbasari, N. Suryani, A. I. Susanti, F. R. Rinawan, and W. G. Purnama, "Pemanfaatan Katalon Studio untuk Otomatisasi Pengujian Black-Box pada Aplikasi iPosyandu," *J. Edukasi dan Penelit. Inform.*, vol. 7, no. 3, p. 370, 2021, doi: 10.26418/jp.v7i3.46954.
- [3] E. Vaskita Nugraha, R. Yanwastika Ariyana*, and E. Kumalasari Nurnawati, "Uji Black Box Tes Aplikasi Software Development System Information (Sodevi) Pt. Dimata Sora Jayate Menggunakan Katalon Studio," *Pros. Snast*, no. November, pp. E60-65, 2022, doi: 10.34151/prosdingsnast.v8i1.4172.
- [4] S. Gupta, S. Rani, and A. Dixit, "Recent Trends in Automation-A study of RPA Development Tools," *2019 3rd Int. Conf. Recent Dev. Control. Autom. Power Eng. RDCAPE 2019*, pp. 159–163, 2019, doi: 10.1109/RDCAPE47089.2019.8979084.
- [5] S. Sri Rahayu, D. Aris Firmansyah, S. Susanti, and U. Adhirajasa Reswara Sanjaya Bandung, "Analisis Penggunaan Tools Automation Testing pada Aplikasi : Systematic Literature Review," *Remik Ris. dan E-Jurnal Manaj. Inform. Komput.*, vol. 8, no. 1, pp. 106–116, 2024, [Online]. Available: <http://doi.org/10.33395/remik.v8i1.13241>
- [6] I. R. Dhaifullah, M. Muttanifudin H, A. Ananda Salsabila, and M. Ainul Yaqin, "Survei Teknik Pengujian Software," *J. Autom. Comput. Inf. Syst.*, vol. 2, no. 1, pp. 31–38, 2022, doi: 10.47134/jacis.v2i1.42.
- [7] M. Mutiara, N. Qomariyah, and D. K. Widyawati, "Automated Testing of Indo Towing Information System with Katalon Studio," *ROUTERS J. Sist. dan Teknol. Inf.*, vol. 2, no. 1, pp. 35–41, 2023, doi: 10.25181/rt.v2i1.3277.
- [8] M. T. Taky, "Automated Testing with Cypress," p. 40, 2021.
- [9] J. Huoponen, "Web-Sovelluksen Päästä Päähän -Testaaminen Cypress-Testaustyökalulla," 2020.
- [10] M. Anugraha, A. E. Tjandra, and R. P. Kristianto, "Pengujian Kualitas Perangkat Lunak menggunakan Pendekatan Manual dan Automation Testing pada Aplikasi 'Nuzantara' Menggunakan Katalon Software Quality Testing through Manual and Automation Testing Approaches on the 'Nuzantara' Application with Katalon," 2024.
- [11] D. Absharina, F. Agustina, and Fahirah, "Survey Paper: Software Automated Testing Tool Using Systematic Literature Review Method," *J. PILAR Nusa Mandiri*, vol. 16, no. 2, pp. 175–182, 2020, [Online]. Available: <https://scholar.google.com/>.
- [12] T. S. Preview, "INTERNATIONAL STANDARD ISO / IEC / IEEE Software and systems engineering — iTeh STANDARD PREVIEW iTeh STANDARD PREVIEW," vol. 2022, 2022.