



Implementasi Otomatisasi Pengujian Fungsional Website Mandala Chain dengan Katalon Studio

Belinda Dwi Sukma Putri^{1*}, Muchlis²

¹⁻² Program Studi Teknik Informatika, STMIK Antar Bangsa, Tangerang, Indonesia

*Penulis Korespondensi: belindaputri90@gmail.com¹

Abstract. *Mandala Chain is the first layer-1 blockchain developed in Indonesia, focusing on interoperability, data security, and ease of adoption. As a blockchain-based platform, Mandala Chain includes key features such as dashboard, connect wallet, project, develop, and mandala academy, which must be verified for proper functionality. This study aims to test these features automatically using Katalon Studio as an automation testing tool. The research adopts the Software Testing Life Cycle (STLC) approach, consisting of Requirement Analysis, Test Planning, Test Case Development, Test Environment Setup, Test Execution, and Test Cycle Closure. This structured method ensures that the testing process is systematic, measurable, and well-documented. A total of 23 test cases were created and organized into seven test suites covering all major features of the Mandala Chain website. The results show that all test cases passed successfully, indicating that the tested features function as specified. The implementation of automated testing using Katalon Studio effectively improves efficiency, ensures result consistency, and minimizes human error risks in verifying the functionality of blockchain-based systems such as Mandala Chain.*

Keywords: *Automation; Blockchain; Katalon; Mandala; Testing.*

Abstrak. Mandala Chain merupakan blockchain layer 1 pertama yang dikembangkan di Indonesia dengan fokus pada interoperabilitas, keamanan data, serta kemudahan adopsi. Sebagai platform berbasis blockchain, Mandala Chain memiliki berbagai fitur penting seperti dashboard, connect wallet, project, develop, dan mandala academy yang harus dipastikan berjalan sesuai spesifikasi. Penelitian ini bertujuan untuk menguji fungsionalitas fitur-fitur tersebut secara otomatis menggunakan *Katalon Studio* sebagai alat bantu pengujian. Metode penelitian yang digunakan adalah *Software Testing Life Cycle (STLC)*, yang mencakup tahapan *Requirement Analysis, Test Planning, Test Case Development, Test Environment Setup, Test Execution*, dan *Test Cycle Closure*. Pendekatan ini memungkinkan pelaksanaan pengujian dilakukan secara sistematis, terukur, dan terdokumentasi dengan baik. Sebanyak 23 *test case* disusun dan dikelompokkan ke dalam tujuh *test suite* yang mencakup seluruh fitur utama pada website Mandala Chain. Hasil pengujian menunjukkan bahwa seluruh *test case* berhasil dijalankan dengan status *pass*, menandakan bahwa sistem telah berfungsi sesuai spesifikasi yang diharapkan. Implementasi otomatisasi pengujian dengan *Katalon Studio* terbukti mampu meningkatkan efisiensi, konsistensi hasil, serta mengurangi risiko kesalahan manual dalam proses verifikasi fungsionalitas sistem berbasis blockchain seperti Mandala Chain.

Kata kunci: Blockchain; Katalon; Mandala; Otomatisasi; Pengujian.

1. LATAR BELAKANG

Mandala Chain adalah *blockchain layer 1* yang dikembangkan pertama di Indonesia, dengan fokus pada interoperabilitas, keamanan data, dan kemudahan adopsi. Platform ini dilengkapi dengan fitur seperti *dashboard, connect wallet, project, develop, mandala academy* dan fitur lainnya yang menjadi objek pengujian dalam penelitian ini (Mandala Chain, 2024).

Pengujian perangkat lunak merupakan tahap penting untuk memastikan bahwa sistem memenuhi kebutuhan pengguna dan berjalan sesuai spesifikasi. Pengujian ini juga bertujuan mengurangi kemungkinan terjadinya *bug* yang dapat mengganggu operasional

sistem (Nugraha, 2022). *Bug* atau *error* dalam perangkat lunak merujuk pada kondisi kesalahan yang menyebabkan sistem tidak berjalan sesuai dengan perintah yang diharapkan. Kesalahan ini dapat berupa kekeliruan logika, ketidaktepatan instruksi, atau kerusakan yang mengakibatkan program mengalami kegagalan fungsi sehingga tidak dapat beroperasi sebagaimana mestinya (Noviansyah, Saiyar, & Sopyan, 2023). Metode pengujian perangkat lunak terbagi menjadi dua, yaitu secara manual dan otomatis. Pengujian manual dilakukan oleh penguji secara langsung dengan mengikuti skenario pengujian yang telah disusun sebelumnya untuk mengevaluasi fungsionalitas sistem. Metode ini sering digunakan untuk menguji skenario yang kompleks dan sulit diotomatisasi, serta untuk mengeksplorasi pengalaman pengguna dalam menggunakan sistem. Sebaliknya, pengujian otomatis merupakan proses pengujian perangkat lunak yang dijalankan dengan memanfaatkan alat atau skrip otomatis, sehingga setiap langkah pengujian tidak memerlukan keterlibatan manual secara langsung (Kamistr, 2025).

Otomatisasi pengujian muncul sebagai solusi untuk meningkatkan akurasi dan efisiensi proses pengujian. Pemanfaatan alat bantu otomatisasi memungkinkan pengujian dilakukan secara berulang dengan konsistensi tinggi serta lebih cepat dibandingkan metode manual (Zulianto et al., 2021). Salah satu alat populer untuk otomatisasi pengujian adalah Katalon Studio, platform pengujian berbasis antarmuka grafis (GUI) yang mendukung pengujian aplikasi web dan *application programming interface (API)* (Azhari & Suyatno, 2024). Proses pembuatan pengujian dapat dilakukan melalui tiga metode, yaitu *record and replay*, *manual mode*, dan *script mode*. Penelitian ini mengombinasikan ketiga metode tersebut dalam pelaksanaannya (Ardi & Putro, 2021).

Website Mandala Chain digunakan sebagai objek penelitian karena merupakan *platform* berbasis *blockchain* yang memerlukan kestabilan dalam operasionalnya. *Blockchain* adalah teknologi yang menjadi dasar pengembangan berbagai mata uang kripto digital. Teknologi ini memiliki karakteristik seperti *desentralisasi*, transparansi, imutabilitas, serta kemampuan audit yang menjamin keamanan dan integritas transaksi. Secara sederhana, *blockchain* berfungsi sebagai media penyimpanan data digital yang aman, transparan, dan permanen (Qalam, Keagamaan, & Kunci, 2024).

Pengujian yang dilakukan bertujuan untuk memastikan fitur-fitur penting seperti *dashboard*, *connect wallet*, dan fitur lainnya berjalan sesuai fungsinya. Implementasi otomatisasi diharapkan dapat mempercepat proses pengujian, mengurangi kesalahan manual, dan memberikan hasil yang lebih akurat.

2. KAJIAN TEORITIS

Pengujian Perangkat Lunak

Menurut ANSI/IEEE 1059 sebagaimana dikutip dalam skripsi Nurul Uswatun Hasanah, pengujian perangkat lunak merupakan proses menganalisis untuk mendeteksi perbedaan antara hasil aktual dengan yang diharapkan, serta memastikan fungsi berjalan sesuai spesifikasi (Hasanah, 2022). Metode pengujian dibagi menjadi metode *white-box testing* yang berfokus pada analisis kode program dan metode *black-box testing* yang melakukan pengujian fungsionalitas tanpa melihat struktur internal sistem (Uminingsih, Ichsanudin, Yusuf, & Suraya, 2022). Melalui pengujian, dapat dipastikan bahwa perangkat lunak berfungsi sesuai dengan tujuan yang telah ditetapkan, sekaligus menjaga konsistensi dan kualitas sistem. Proses ini juga menjadi sarana verifikasi untuk menilai apakah perangkat lunak telah memenuhi kriteria kelayakan sehingga dapat digunakan secara optimal oleh pengguna (Albarka Umar, 2023).

Pengujian Fungsional

Pengujian fungsional adalah metode pengujian perangkat lunak yang mendasarkan proses pengujiannya pada spesifikasi yang telah ditetapkan. Metode ini dilakukan dengan memberikan input tertentu dan mengevaluasi output yang dihasilkan tanpa mempertimbangkan struktur internal program. Tujuan pengujian ini adalah memastikan layanan yang diberikan sistem sesuai dengan ketentuan, serta menilai bagaimana sistem merespons masukan dan berperilaku dalam berbagai kondisi (Sianturi et al., 2021).

Otomatisasi Pengujian

Menurut Ryandhita Pietro sebagaimana dikutip dalam skripsi (Rambe, 2022), pengujian otomatis adalah pengujian yang dilakukan dengan menggunakan skrip atau alat bantu untuk mengeksekusi pengujian, memungkinkan pengulangan dengan hasil yang konsisten. Keuntungan utamanya adalah efisiensi waktu, meminimalisir risiko kesalahan manusia, dan kemudahan melakukan pengujian berulang atau regresi ketika aplikasi diperbarui. Rangkuman perbedaan pengujian manual dan otomatis dijabarkan pada Tabel 1 (Zulianto et al., 2021).

Table 1. Perbedaan Pengujian Secara Manual Dan Otomatis

Pengujian Manual	Pengujian Otomatis
Mebutuhkan banyak sumber daya manusia.	Hanya memerlukan sedikit sumber daya manusia.
Proses pengujian memakan waktu lama.	Waktu pelaksanaan lebih singkat.
Berisiko mengalami <i>human error</i> .	Otomatisasi memastikan pengujian dilakukan dengan cara yang sama setiap kali dijalankan.
Mebutuhkan banyak modul pengujian untuk menemukan <i>bug</i> yang dialami pengguna sebenarnya.	Cenderung hanya mendeteksi <i>bug</i> yang sudah diprediksi sebelumnya.
Biaya awal relatif lebih rendah.	Lebih hemat dalam jangka panjang.

Test Case dan Test Suite

Test case merupakan skenario pengujian yang dirancang dengan tujuan mendeteksi kesalahan yang mungkin belum teridentifikasi pada suatu fungsi atau proses dalam sistem. Penyusunannya ditujukan untuk memberikan peluang yang cukup besar dalam menemukan *error* atau *bug*, dengan meminimalkan penggunaan waktu dan usaha, sehingga tetap selaras dengan tujuan pengujian (Hasibuan & Dirgahayu, 2020).

Test suite adalah kumpulan *test case* yang dikelompokkan untuk dijalankan secara terstruktur. Test suite memungkinkan pengujian dieksekusi secara sistematis, mendukung pengujian berulang, dan mempermudah manajemen skenario pengujian pada fitur utama sistem (Ansfridus, Yudi Dwiandiyanta, & Wahyu Rahardjo Emanuel, 2024).

Katalon Studio

Katalon Studio adalah platform otomatisasi pengujian berbasis Antarmuka Grafis (GUI) yang mendukung pengujian web, API, dan mobile. Fitur utamanya mencakup *record and playback*, serta pelaporan hasil pengujian dalam format HTML dan CSV (BuildWithAngga, 2023). Hal ini menjadikannya pilihan yang tepat untuk pengujian *black-box testing* pada website Mandala Chain, di mana pengujian otomatis dilakukan guna memastikan seluruh fungsi utama sistem berjalan sesuai dengan spesifikasi yang telah ditentukan (Rahmat Fauzan & Totok Chamidy, 2025).

3. METODE PENELITIAN

Penelitian ini menerapkan pendekatan *Software Testing Life Cycle* (STLC), yaitu serangkaian tahapan terstruktur yang digunakan untuk memastikan proses pengujian perangkat lunak berjalan sistematis (Ruliansyah, Tukino, Baenil Huda, & April Lia

Hananto, 2023). STLC yang meliputi 5 langkah yaitu *Requirement Analysis*, *Test Planning*, *Test Case Development*, *Test Environment Setup*, *Test Execution*, dan *Test Cycle Closure* (I G. N. P. D. D. Diastama, I. M. Sukarsa, & N. K. A. Wirdiani, 2021). STLC mencakup beberapa tahapan dengan kriteria dan output yang jelas pada setiap fase, yang terdiri dari tahapan:

- a) *Requirement Analysis* – Tahap analisis kebutuhan dilakukan dengan mengidentifikasi fitur utama pada website Mandala Chain, seperti koneksi *wallet*, navigasi menu, *dashboard*, dan fitur lainnya. Hasil analisis digunakan sebagai dasar untuk menentukan cakupan pengujian otomatis sesuai spesifikasi fungsional yang telah ditetapkan.
- b) *Test Planning* – Menyusun rencana pengujian dengan menentukan metode, cakupan, strategi eksekusi, serta lingkungan pengujian menggunakan katalon studio.
- c) *Test Case Development* – Proses pembuatan *test case* dilakukan dengan memanfaatkan fitur *record and replay* pada Katalon Studio, serta dilengkapi dengan penulisan manual *script* pada skenario tertentu. *Test case* kemudian dikelompokkan ke dalam *test suite* agar pengujian dapat dijalankan secara sistematis dan mendukung pengujian berulang.
- d) *Test Environment Setup* – Menyiapkan katalon studio, konfigurasi browser, dan akses ke mandala chain agar pengujian dapat berjalan sesuai rencana.
- e) *Test Execution* – Menjalankan semua *test case* melalui *test suite*. Hasil pengujian dicatat sebagai *pass* atau *fail*. *Bug* yang ditemukan didokumentasikan dan diuji ulang setelah diperbaiki.
- f) *Test Cycle Closure* – Melakukan evaluasi hasil pengujian dan pembuatan laporan otomatis dari eksekusi test suite di Katalon Studio. Laporan ini berisi status pengujian, menjadi referensi untuk perbaikan dan pengujian berikutnya, serta memuat dokumentasi hasil pengujian yang menyimpulkan tingkat kelayakan sistem (Arfan & Hendrik, 2022).

4. HASIL DAN PEMBAHASAN

Requirement Analysis

Tahap pertama adalah identifikasi kebutuhan pengujian. Analisis dilakukan terhadap fitur utama pada website mandala, yaitu *dashboard*, *mandala academy*, *mandala wallet*, *project and develop*, serta *sidebar* dan *navbar*. Fokus pengujian diarahkan pada

fungsi penting seperti koneksi dompet digital, validasi input *mnemonic phrases*, navigasi menu, dan pengaturan tampilan. Hasil analisis ini menjadi dasar dalam menentukan cakupan pengujian otomatis agar sesuai dengan spesifikasi fungsional. Misalnya, pada fitur mandala wallet, pengujian difokuskan pada proses autentikasi dengan berbagai variasi input, seperti *mnemonic phrases* kosong, format tidak sesuai, *password* dengan kriteria tertentu, hingga kondisi validasi ulang *password*.

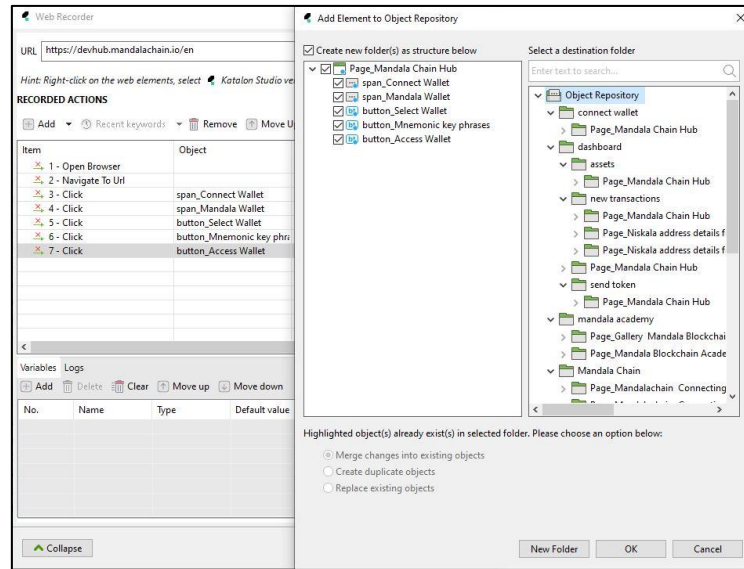
Test Planning

Pada tahap ini dilakukan perencanaan pengujian dengan menetapkan strategi pengujian fungsional berbasis otomatisasi menggunakan Katalon Studio. Jenis pengujian yang digunakan adalah *black-box testing*, yang berfokus pada pengujian output berdasarkan input yang diberikan, tanpa melihat struktur kode. Perencanaan meliputi:

- a) Penentuan cakupan pengujian pada fitur *website* yang terbagi menjadi 6 bagian, diantaranya *dashboard*, *mandala academy*, *mandala wallet*, *project and develop*, serta *sidebar* dan *navbar*.
- b) Penetapan *browser* yang digunakan yaitu *microsoft edge* untuk pengujian.
- c) Persiapan katalon studio versi 9 untuk pengujian otomatisasi.
- d) Penggunaan *test suite* agar pengujian berjalan secara paralel dan mendukung regresi testing.

Test Case Development

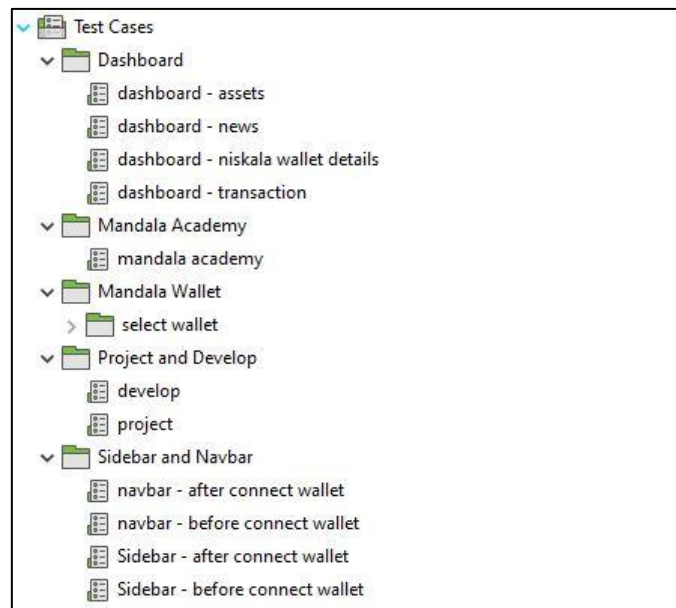
Pembuatan *test case* dilakukan dengan cara mengambil *object* melalui *record web* menggunakan Katalon Studio. Gambar 1 menunjukkan proses pengambilan objek yang dilakukan melalui fitur *record web*.



Gambar 1. Record Web

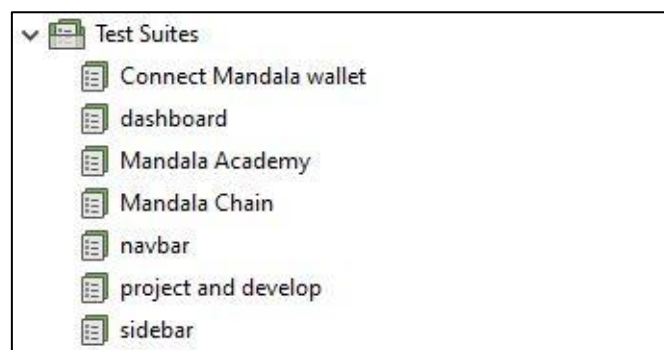
Object yang diperoleh menggunakan fitur *record web*, akan tersimpan di dalam *object repository*. Saat eksekusi dilakukan, hasil rekaman tersebut dijalankan sesuai dengan langkah-langkah yang ada pada *test case*. Selain melalui *record web*, penyusunan *test case* juga dilakukan secara manual dengan menambahkan skrip pada *test case* tertentu, serta menggunakan fitur *spy* di katalon studio untuk mengambil *object* tambahan yang diperlukan.

Sebanyak 23 *test case* disusun berdasarkan hasil analisis kebutuhan pada fitur utama website mandala, meliputi *dashboard*, *mandala academy*, *mandala wallet*, *project and develop*, serta *sidebar* dan *navbar*. Penyusunan *test case* ini mempertimbangkan skenario positif maupun negatif, sehingga pengujian tidak hanya memverifikasi fungsi normal, tetapi juga menguji bagaimana sistem menangani input yang salah atau tidak valid. Gambar 2 menampilkan daftar *test case* yang telah dikelompokkan ke dalam folder-folder tertentu untuk menjaga kerapian dan mempermudah pengelolaan.



Gambar 2. Test Case List

Seluruh *test case* yang telah dibuat kemudian dikelompokkan ke dalam *test suite*. Dari total 23 *test case*, dibentuk 7 *test suite* yang masing-masing mewakili fitur utama pada website mandala, seperti *dashboard*, *mandala wallet*, *sidebar* dan fitur lainnya. Cara ini memungkinkan pengujian dilakukan lebih efisien dibandingkan dengan menjalankan *test case* satu per satu, sekaligus mendukung pengujian berulang ketika terjadi pembaruan sistem. Gambar 3 memperlihatkan daftar *test suite* yang telah disusun.



Gambar 3. Test Suite List

Test Environment Setup

Pada tahap ini mencakup persiapan lingkungan yang digunakan untuk menjalankan otomatisasi pengujian dengan Katalon Studio. Persiapan meliputi pemenuhan prasyarat instalasi, konfigurasi perangkat keras dan perangkat lunak, serta pengaturan *browser* sebagai media eksekusi. Katalon Studio menyediakan dukungan bawaan untuk menjalankan pengujian berbasis web melalui integrasi dengan browser populer seperti Chrome, Edge, dan Firefox. Pada penelitian ini digunakan Katalon Studio versi 9.7.6

dengan *browser microsoft edge* sebagai media eksekusi pengujian. Tidak diperlukan instalasi tambahan karena Katalon sudah menyertakan *WebDriver* yang kompatibel dengan browser tersebut.

Tabel 2. menunjukkan spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam penelitian ini.

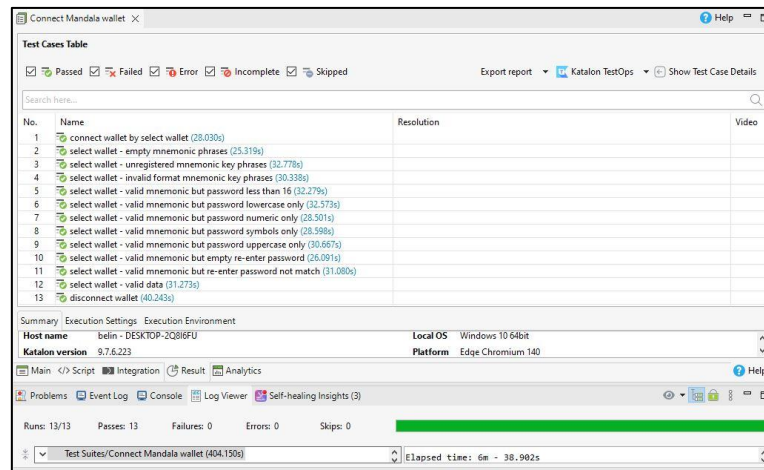
Table 2. Spesifikasi Perangkat

Komponen	Spesifikasi
Sistem Operasi	Windows 10 64-bit
Prosesor	intel celeron n4100
RAM	8
Tools Pengujian	Katalon Studio 9.7.6
Browser Eksekusi	<i>Microsoft Edge</i> (Driver bawaan Katalon)

Setelah spesifikasi perangkat keras dan perangkat lunak ditentukan, langkah selanjutnya adalah menetapkan Microsoft Edge sebagai *default browser* pada Katalon Studio. Pengaturan ini dilakukan melalui menu *Project > Settings*, kemudian pada bagian *Executions*, ubah bagian *default execution* menjadi *Microsoft Edge*. Konfigurasi ini dipilih agar seluruh proses pengujian berjalan konsisten pada satu *browser*.

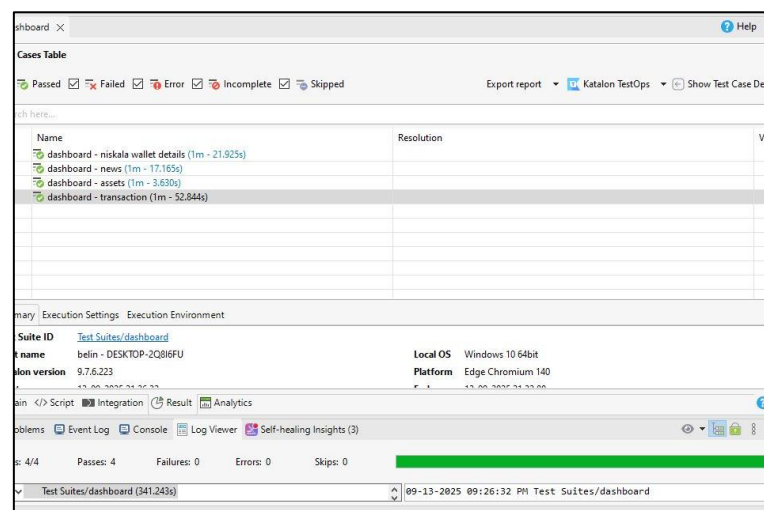
Test Execution

Pelaksanaan pengujian dilakukan dengan menjalankan *test suite* yang telah disusun sebelumnya di Katalon Studio. Hasil pengujian ditampilkan pada *result log*, di mana tanda centang hijau menunjukkan *test suite* berstatus *pass* atau sesuai dengan fungsionalitas yang diharapkan, sedangkan tanda silang merah menandakan status *failed* atau adanya ketidaksesuaian antara hasil aktual dan hasil yang diharapkan. Berikut merupakan hasil dari *test suite connect mandala wallet* pada gambar 4.



Gambar 4. Result Log Test Suite Connect Mandala Wallet

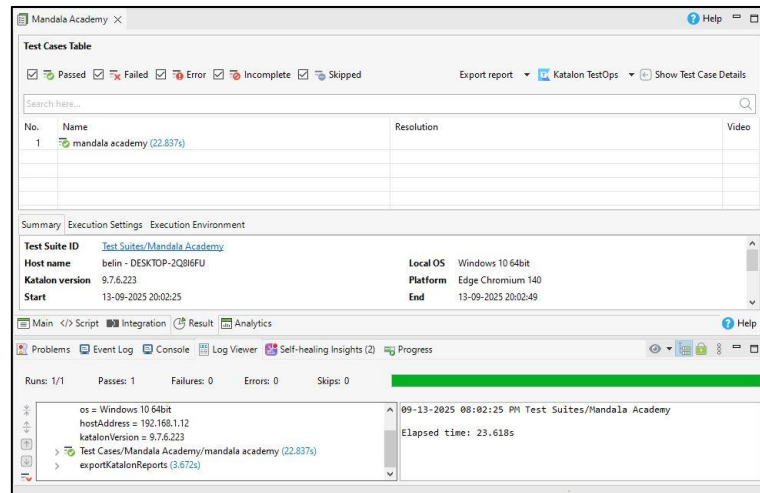
Gambar 4 memperlihatkan hasil eksekusi *test suite connect mandala wallet* yang terdiri dari 13 *test case*. Seluruh skenario berhasil dijalankan dengan status *pass*. Pengujian mencakup koneksi dompet digital, validasi input *mnemonic phrases* (kosong, format tidak sesuai, maupun tidak terdaftar), serta verifikasi *password* dengan berbagai variasi (kurang dari 16 karakter, hanya huruf kecil, hanya huruf besar, hanya angka, atau hanya simbol). Keberhasilan seluruh pengujian ini mengindikasikan bahwa fitur *mandala wallet* telah berjalan sesuai dengan spesifikasi fungsional pada saat pengujian dilakukan. Gambar 5 merupakan hasil eksekusi dari *test suite dashboard*.



Gambar 5. Result Log Test Suite Dashboard

Gambar 5 menunjukkan hasil eksekusi *test suite dashboard* yang terdiri dari 4 *test case*, yaitu akses *Niskala Wallet Details*, *News*, *Assets*, dan *Transaction*. Semua *test case* berstatus *pass*, yang berarti *dashboard* mampu menampilkan data sesuai dengan spesifikasi fungsional.

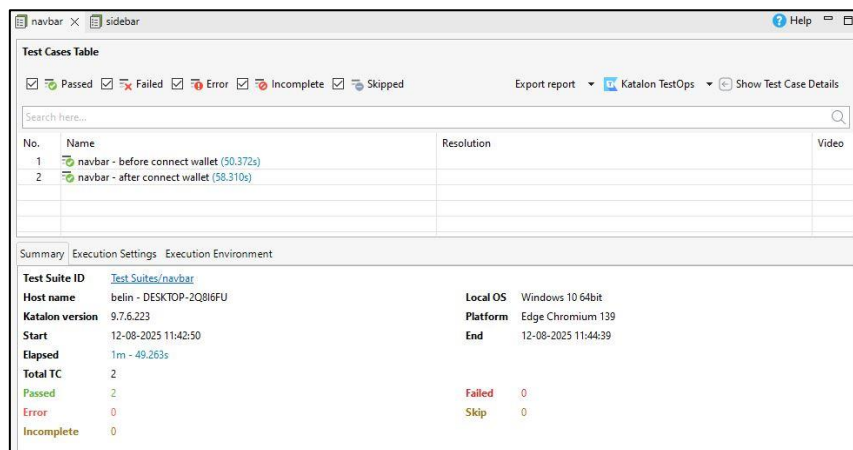
Gambar 6 merupakan hasil eksekusi dari *test suite mandala academy*.



Gambar 6. Result Log Test Suite Mandala Academy

Gambar 6 menampilkan hasil eksekusi *test suite mandala academy* dengan 1 *test case*. Hasil pengujian berstatus *pass*, sehingga halaman pembelajaran *mandala academy* dapat ditampilkan sesuai harapan.

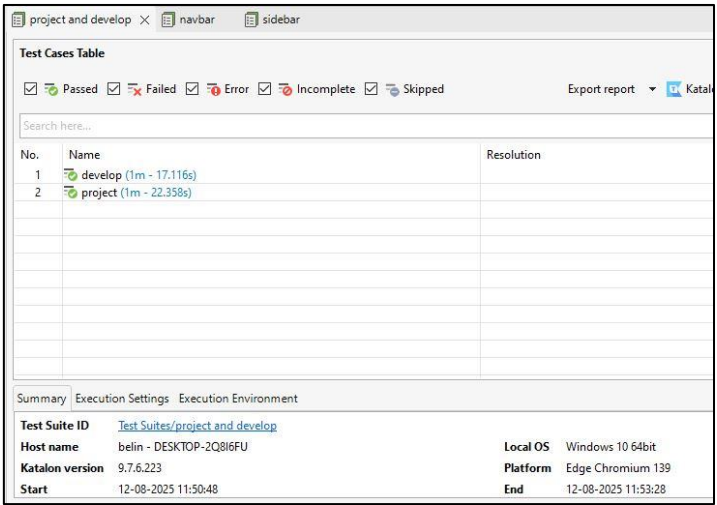
Gambar 7 merupakan hasil eksekusi dari *test suite navbar*.



Gambar 7. Result Log Test Suite Navbar

Gambar 7 menunjukkan hasil eksekusi *test suite navbar* yang terdiri dari 2 *test case*, yaitu sebelum dan sesudah *wallet* terhubung. Keduanya berjalan dengan status *pass*, sehingga dapat disimpulkan bahwa tampilan navigasi atas sistem menyesuaikan kondisi koneksi *wallet* sebagaimana dirancang.

Gambar 8 merupakan hasil eksekusi dari *test suite project and develop*.



The screenshot shows the Katalon Studio interface with the 'Test Cases Table' and 'Summary' tabs. The 'Test Cases Table' shows two test cases, both with a 'Passed' status. The 'Summary' tab provides details about the test suite, including the host name, Katalon version, and execution times.

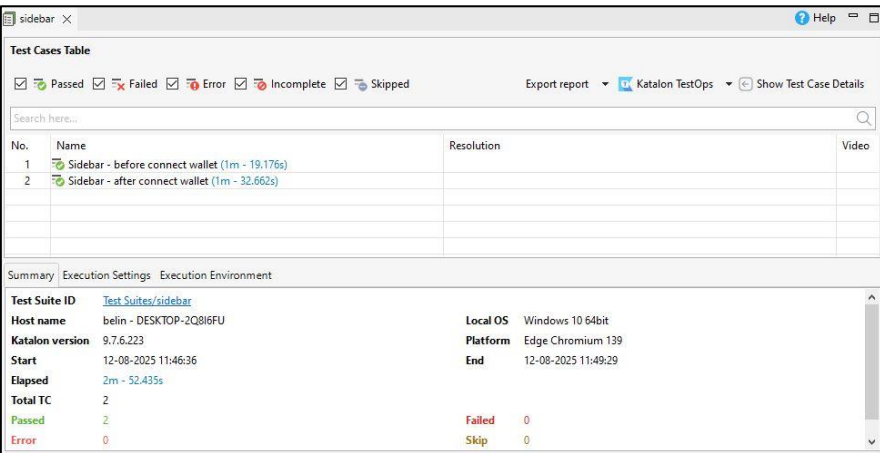
No.	Name	Resolution
1	develop (1m - 17.116s)	
2	project (1m - 22.358s)	

Test Suite ID		Host name		Local OS
Test Suites/project and develop		belin - DESKTOP-2Q8I6FU		Windows 10 64bit
Katalon version		9.7.6.223		Platform
				Edge Chromium 139
Start		12-08-2025 11:50:48		End
				12-08-2025 11:53:28

Gambar 8. Result Log Test Suite Project and Develop

Gambar 8 memperlihatkan hasil eksekusi *test suite project and develop* yang mencakup 2 *test case*, yaitu pengujian halaman *project* dan halaman *develop*. Kedua pengujian berhasil dengan status *pass*, memastikan fitur pengelolaan proyek dan pengembangan dapat diakses tanpa kendala.

Gambar 9 merupakan hasil eksekusi dari *test suite sidebar*.



The screenshot shows the Katalon Studio interface with the 'Test Cases Table' and 'Summary' tabs. The 'Test Cases Table' shows two test cases, both with a 'Passed' status. The 'Summary' tab provides details about the test suite, including the host name, Katalon version, and execution times.

No.	Name	Resolution	Video
1	Sidebar - before connect wallet (1m - 19.176s)		
2	Sidebar - after connect wallet (1m - 32.662s)		

Test Suite ID		Host name		Local OS
Test Suites/sidebar		belin - DESKTOP-2Q8I6FU		Windows 10 64bit
Katalon version		9.7.6.223		Platform
				Edge Chromium 139
Start		12-08-2025 11:46:36		End
				12-08-2025 11:49:29
Elapsed		2m - 52.435s		
Total TC		2		
Passed		2		Failed
				0
Error		0		Skip
				0

Gambar 9. Result Log Test Suite Sidebar

Gambar 9 menampilkan hasil eksekusi *test suite sidebar* yang berisi 2 *test case*. Seluruh pengujian berstatus *pass*, mengindikasikan bahwa menu *sidebar* dapat berfungsi sesuai dengan kondisi koneksi *wallet*.

Test Cycle Closure

Seluruh rangkaian pengujian telah dijalankan melalui tujuh *test suite* yang mencakup fungsionalitas utama pada *website* Mandala. Berdasarkan hasil eksekusi, seluruh 23 *test case* dinyatakan berhasil (*passed*) tanpa ditemui kegagalan. Hal ini

menunjukkan bahwa fitur-fitur yang diuji, seperti koneksi dompet digital, validasi *input mnemonic phrases*, navigasi menu, hingga tampilan konten, berjalan sesuai dengan spesifikasi yang ditetapkan.

Sebagai bagian dari penutupan siklus pengujian, Katalon Studio menghasilkan laporan otomatis dalam format HTML dan CSV. Laporan ini berisi ringkasan status pengujian, detail hasil setiap *test case*, waktu eksekusi, serta informasi lingkungan pengujian. Dokumentasi tersebut tidak hanya menjadi bukti pelaksanaan pengujian, tetapi juga dapat digunakan sebagai referensi untuk proses evaluasi maupun pengujian ulang pada pengembangan berikutnya.

Gambar 10 menampilkan contoh laporan hasil pengujian dalam format HTML.

dashboard Test Log	
Execution Environment Host name: belin - DESKTOP-2Q8IGFU Local OS: Windows 10 64bit Katalon version: 9.7.6.223 Browser: Edge Chromium 140	
Test Execution Log TEST SUITE: dashboard Full Name: dashboard Start / End / Elapsed: 2025-09-13 21:26:32.384 / 2025-09-13 21:32:08.685 / 00:05:36.301 Status: 4 test total, 4 passed, 0 failed, 0 error, 0 incomplete, 0 skipped TEST CASE: Test Cases/Dashboard/dashboard - niskala wallet details TEST CASE: Test Cases/Dashboard/dashboard - news TEST CASE: Test Cases/Dashboard/dashboard - assets TEST CASE: Test Cases/Dashboard/dashboard - transaction	

Gambar 10. Laporan hasil pengujian dalam format HTML

Gambar 11 memperlihatkan laporan hasil pengujian dalam format CSV yang diekspor langsung dari Katalon Studio.

Suite/Test/Step Name								
A	B	C	D	E	F	G	H	I
1	Suite/Test	Browser	Descriptive Tag	Start time	End time	Duration	Status	
2	dashboard	Edge Chromium 140		13-09-2021	13-09-2021	5m - 36.30	PASSED	
3								
4	Test Case:	Edge Chromium 140		13-09-2021	13-09-2021	1m - 21.92	PASSED	
5	openBrow	Edge Chromium 140		13-09-2021	13-09-2021	9.671s	PASSED	
6	setViewPi	Edge Chromium 140		13-09-2021	13-09-2021	0.855s	PASSED	
7	navigateT	Edge Chromium 140		13-09-2021	13-09-2021	2.737s	PASSED	
8	delay(2)	Edge Chromium 140		13-09-2021	13-09-2021	2.308s	PASSED	
9	click(find	Edge Chromium 140		13-09-2021	13-09-2021	1.196s	PASSED	
10	delay(1)	Edge Chromium 140		13-09-2021	13-09-2021	1.035s	PASSED	
11	click(find	Edge Chromium 140		13-09-2021	13-09-2021	0.851s	PASSED	
12	delay(1)	Edge Chromium 140		13-09-2021	13-09-2021	1.037s	PASSED	
13	click(find	Edge Chromium 140		13-09-2021	13-09-2021	0.632s	PASSED	
14	delay(1)	Edge Chromium 140		13-09-2021	13-09-2021	1.056s	PASSED	
15	click(find	Edge Chromium 140		13-09-2021	13-09-2021	0.716s	PASSED	
16	delay(1)	Edge Chromium 140		13-09-2021	13-09-2021	1.021s	PASSED	
17	setText(fi	Edge Chromium 140		13-09-2021	13-09-2021	1.963s	PASSED	
18	setText(fi	Edge Chromium 140		13-09-2021	13-09-2021	1.098s	PASSED	
19	setEncryp	Edge Chromium 140		13-09-2021	13-09-2021	0.813s	PASSED	
20	setEncryp	Edge Chromium 140		13-09-2021	13-09-2021	0.782s	PASSED	
21	click(find	Edge Chromium 140		13-09-2021	13-09-2021	0.543s	PASSED	
		20250913_212613						

Gambar 11. Laporan hasil pengujian dalam format CSV

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengimplementasikan otomatisasi pengujian fungsional pada *website* Mandala Chain menggunakan Katalon Studio dengan pendekatan *Software Testing Life Cycle* (STLC). Sebanyak 23 *test case* yang dikelompokkan ke dalam tujuh *test suite* telah dijalankan, dan seluruh hasil pengujian menunjukkan status pass. Hal ini membuktikan bahwa fungsionalitas utama sistem, seperti koneksi dompet digital, validasi *input mnemonic phrases*, navigasi menu, hingga tampilan konten, berjalan sesuai spesifikasi yang telah ditetapkan. Selain itu, pemanfaatan Katalon Studio mempermudah proses pengujian dengan menyediakan mekanisme otomatisasi, pengelolaan test suite yang terstruktur, serta laporan hasil pengujian dalam format HTML dan CSV yang dapat digunakan sebagai dokumentasi maupun referensi evaluasi di tahap pengembangan selanjutnya.

Sebagai tindak lanjut, pengujian ini dapat diperluas dengan menambahkan skenario baru apabila terdapat pembaruan fitur pada sistem Mandala Chain, sehingga keberlangsungan kualitas perangkat lunak dapat tetap terjaga secara konsisten.

DAFTAR REFERENSI

- Albarka Umar, M. (2023). *A study of software testing: Categories, levels, techniques, and types comprehensive study of software testing: Categories, levels, techniques, and types*. 5(6), 32–40. <https://doi.org/10.36227/techrxiv.12578714.v2>
- Ansfridus, M., Yudi Dwianiyanta, B., & Wahyu Rahardjo Emanuel, A. (2024). *Black box testing dengan metode equivalence partitioning pada Techno Expertise Academy (TEA) Astra Credit Companies*. **KONSTELASI: Konvergensi Teknologi dan Sistem Informasi**, 4(2). <https://doi.org/10.24002/konstelasi.v4i2.8996>
- Ardi, F., & Putro, H. P. (2021). *Pengujian black box aplikasi mobile menggunakan Katalon Studio*. **Jurnal UII**.
- Arfan, A., & Hendrik, H. (2022). *Penerapan STLC dalam pengujian black box dengan automation testing tool (Studi kasus: PT. GIT Solution)*. **Automata**, 3. Retrieved from <https://journal.uui.ac.id/AUTOMATA/article/view/24127>
- Azhari, F. W., & Suyatno, D. F. (2024). *Pengujian otomatis GUI dengan Katalon Studio pada situs lowongan kerja Jobstreet dan Glints*. **JEISBI (Journal of Emerging Information Systems and Business Intelligence)**, 5(3), 205–213. <https://doi.org/10.26740/jeisbi.v5i3.63134>

- BuildWithAngga. (2023). *Katalon Studio for automation test*. Retrieved April 8, 2025, from BuildWithAngga website: <https://buildwithangga.com/tips/Katalon-Studio-for-Automation-Test>
- Hasanah, N. U. (2022). *Kajian otomatisasi pengujian GUI: Selenium IDE, UiPath Studio, Katalon Studio*. In *Repository UIN Jakarta*. Retrieved from <https://repository.uinjkt.ac.id/dspace/handle/123456789/66772>
- Hasibuan, A. N., & Dirgahayu, T. (2020). *Pengujian dengan unit testing dan test case pada proyek pengembangan modul manajemen pengguna*. **Jurnal Informatika Universitas Islam Indonesia**, 2(1), 103–109.
- I G. N. P. D. D. Diastama, I. M. Sukarsa, & N. K. A. Wirdiani. (2021). *Pengembangan test script untuk load testing web dengan metode software testing life cycle*. **Jurnal Ilmiah Teknologi dan Komputer**, 2(1), 1–8.
- Kamistr, R. A. (2025). *Studi komparatif pengujian manual dan pengujian otomatis dengan Cypress pada website*. **EMPIRIS: Jurnal Sains, Teknologi dan Kesehatan**, 2(2), 354–364. <https://doi.org/10.62335/empiris.v2i2.1354>
- Mandala Chain. (2024). *Mandala Chain docs*. Retrieved April 12, 2025, from Mandala Chain website: <https://docs.mandalachain.io/docs/getting-started/>
- Noviansyah, M., Saiyar, H., & Sopyan, S. (2023). *Sistem pakar pendeteksi error dan bug berbasis web pada Adobe Premier Pro*. **Jurnal Minfo Polgan**, 12(1), 1139–1149. <https://doi.org/10.33395/jmp.v12i1.12613>
- Nugraha, W. A. (2022). *Pengujian white box berbasis path pada form autentikasi berbasis mobile*. **Jurnal Siliwangi Seri Sains dan Teknologi**, 8(2), 42–47. <https://doi.org/10.37058/jssainstek.v8i2.4098>
- Qalam, A., Keagamaan, J. I., & Kunci, K. (2024). *Penerapan teknologi blockchain pada transaksi online shop*. **ABS**, 18(1), 654–670.
- Rahmat Fauzan, & Totok Chamidy. (2025). *Pengujian blackbox pada web e-course menggunakan Katalon Studio*. **Jurnal Perangkat Lunak**, 7(2), 111–121. <https://doi.org/10.32520/jupel.v7i2.3663>
- Rambe, A. R. (2022). *Pengujian otomatis aplikasi mobile dengan teknik black-box menggunakan Appium*.
- Ruliansyah, Tukino, Baenil Huda, & April Lia Hananto. (2023). *Penerapan software testing life cycle pada pengujian otomatisasi platform Dzikra*. **CSRID (Computer Science Research and Its Development Journal)**, 15(1), 1–11. <https://doi.org/10.22303/csrid.15.1.2023.01-11>
- Sianturi, R. A., Sinaga, A. M., Pratama, Y., Simatupang, H., Panjaitan, J., & Sihotang, S. (2021). *Perancangan pengujian fungsional dan non fungsional aplikasi Siappara di Kabupaten Humbang Hasundutan*. **Jurnal Komputer dan Informatika**, 9(2), 133–141. <https://doi.org/10.35508/jicon.v9i2.4706>

- Uminingsih, Ichsanudin, M. N., Yusuf, M., & Suraya. (2022). *Perpustakaan dengan metode black box testing bagi pemula*. **STORAGE: Jurnal Ilmiah Teknik dan Ilmu Komputer**, 1(2), 1–8. <https://doi.org/10.55123/storage.v1i2.270>
- Zulianto, A., Purbasari, A., Suryani, N., Susanti, A. I., Rinawan, F. R., & Purnama, W. G. (2021). *Pemanfaatan Katalon Studio untuk otomatisasi pengujian black-box pada aplikasi iPosyandu*. **Jurnal Edukasi dan Penelitian Informatika (JEPIN)**, 7(3), 370. <https://doi.org/10.26418/jp.v7i3.46954>