

IMPLEMENTASI AUTOMATED TESTING PADA WEBSITE DASHBOARD PENJUALAN DENGAN KATALON DI PT TEMPRINA

IMPLEMENTATION OF AUTOMATED TESTING ON SALES DASHBOARD
WEBSITE WITH KATALON AT PT TEMPRINA

**Eka Rahma Risnawati¹, Alifiasari Zhafira Hendriyetti Esmono², Aisha Nurliana Putri³, Anindo
Saka Fitri⁴**

*E-mail: 22082010218@student.upnjatim.ac.id¹, 22082010250@student.upnjatim.ac.id²,
22082010254@student.upnjatim.ac.id³, anindo.saka.si@upnjatim.ac.id⁴

¹Sistem Informasi, Fakultas Ilmu Komputer, UPN “Veteran” Jawa Timur

²Sistem Informasi, Fakultas Ilmu Komputer, UPN “Veteran” Jawa Timur

³Sistem Informasi, Fakultas Ilmu Komputer, UPN “Veteran” Jawa Timur

⁴Dosen Sistem Informasi, Fakultas Ilmu Komputer, UPN “Veteran” Jawa Timur

Abstrak

Perkembangan teknologi informasi mendorong perusahaan untuk meningkatkan kualitas sistem informasi, termasuk dalam hal pengujian perangkat lunak. Pengujian secara manual cenderung memakan waktu lama dan rentan terhadap kesalahan manusia, terutama pada sistem kompleks seperti dashboard penjualan. Oleh karena itu, PT Temprina Media Grafika Surabaya mengadopsi pendekatan *automated testing* menggunakan Katalon Studio guna meningkatkan efisiensi dan efektivitas pengujian. Project ini bertujuan untuk mengimplementasikan automated testing pada sistem dashboard penjualan, dengan fokus pada pengujian fungsionalitas dari sisi admin yang memiliki akses penuh terhadap berbagai fitur system. Tahapan pengujian mengikuti siklus *Software Testing Life Cycle* (STLC), dimulai dari analisis kebutuhan, perencanaan pengujian, pengembangan test case, pengaturan lingkungan uji, hingga eksekusi pengujian. Test case yang dikembangkan mencakup fitur autentikasi (registrasi, login, logout), manajemen data (produk, kategori, pelanggan, penjualan), serta fitur approval dan profil pengguna. Hasil pengujian menunjukkan bahwa sebagian besar fitur berjalan sesuai dengan spesifikasi, meskipun ditemukan kendala pada beberapa aksi hapus data yang tidak berhasil sepenuhnya. Pengujian ini dilakukan secara otomatis dan terdokumentasi melalui log viewer Katalon Studio yang menampilkan status setiap langkah uji. Implementasi automated testing ini terbukti membantu dalam mempercepat proses pengujian, mengurangi risiko human error, serta memastikan kestabilan sistem dashboard penjualan. Selain memberikan pengalaman praktis kepada mahasiswa, kegiatan ini juga memberikan kontribusi positif bagi perusahaan dalam menjaga kualitas sistem informasi yang digunakan.

Kata kunci: *Automated Testing, Dashboard Penjualan, Katalon Studio, Software Testing Life Cycle.*

Abstract

The development of information technology encourages companies to improve the quality of information systems, including in terms of software testing. Manual testing tends to take a long time and is prone to human error, especially in complex systems such as sales dashboards. Therefore, PT Temprina Media Grafika Surabaya adopts an automated testing approach using Katalon Studio to improve the efficiency and effectiveness of testing. This project aims to

implement automated testing on the sales dashboard system, with a focus on testing functionality from the admin side who has full access to various system features. The testing stages follow the Software Testing Life Cycle (STLC), starting from requirements analysis, test planning, test case development, test environment setup, to test execution. The test cases developed included authentication features (registration, login, logout), data management (products, categories, customers, sales), as well as approval and user profile features. The test results show that most of the features run according to specifications, although there are problems with some data deletion actions that are not fully successful. This test is carried out automatically and documented through the Katalon Studio log viewer which displays the status of each test step. The implementation of automated testing has proven to be helpful in speeding up the testing process, reducing the risk of errors.

Keywords: *Automated Testing, Sales Dashboard, Katalon Studio, Software Testing Life Cycle.*

1. PENDAHULUAN

Perkembangan teknologi informasi telah mendorong perusahaan untuk terus berinovasi dan memperkuat sistem informasi guna mendukung operasional dan pengambilan keputusan bisnis [1]. Salah satu aspek penting dalam sistem informasi adalah pengujian perangkat lunak, yang berfungsi untuk memastikan kesesuaian sistem terhadap kebutuhan dan spesifikasi pengguna [2]. Menurut Arifandi et al. [3], pengujian manual cenderung memakan waktu, kurang konsisten, dan rentan terhadap kesalahan manusia, sehingga pendekatan *automated testing* menjadi solusi yang lebih efektif dan efisien.

PT Temprina Media Grafika, perusahaan percetakan yang tergabung dalam Jawa Pos Group, telah menerapkan sistem dashboard penjualan untuk memantau performa penjualan secara real-time. Mengingat pentingnya sistem tersebut dalam mendukung keputusan manajerial, keandalan dan stabilitas sistem harus selalu terjaga. Untuk itu, perusahaan menerapkan automated testing menggunakan Katalon Studio, alat uji berbasis antarmuka pengguna (UI) dan API yang intuitif dan mendukung pengujian terstruktur [4].

Melalui kegiatan Praktik Kerja Lapangan (PKL), dilakukan implementasi automated testing pada website dashboard penjualan PT Temprina. Tujuan utamanya adalah memastikan seluruh fungsi sistem berjalan sesuai harapan, mengidentifikasi potensi error, serta meningkatkan efisiensi proses uji [5]. Penelitian ini difokuskan pada tiga rumusan masalah: teknik pengujian yang diterapkan, tahapan pengujian menggunakan Katalon Studio, dan hasil dari pengujian tersebut.

2. METODOLOGI

Dalam proyek ini digunakan metode *Software Testing Life Cycle (STLC)* yang terdiri dari enam tahap, yaitu requirement analysis, test planning, test case development, environment setup, test execution, dan test cycle closure [6]. Metode ini dipilih untuk memastikan proses pengujian berjalan terstruktur, efektif, dan dapat menghasilkan sistem yang sesuai dengan kebutuhan pengguna. Pengujian dilakukan menggunakan tools Katalon Studio secara otomatis.

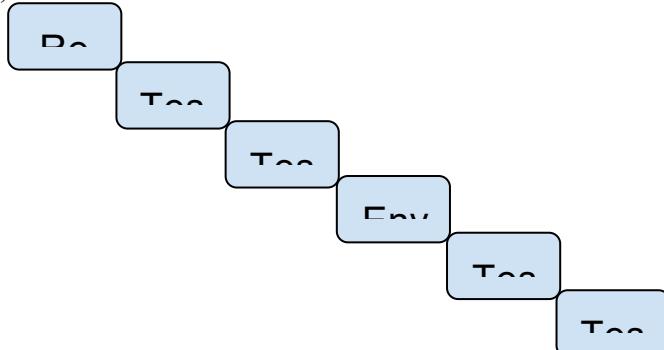
2.1 Automated Testing

Testing adalah proses menganalisa suatu entitas software untuk mendeteksi perbedaan antara kondisi aktual dengan kondisi yang diinginkan (defect / error / bugs) dan mengevaluasi fitur dari entitas software (Rahayu, et. Al) [7]. Automated testing adalah tahapan dalam menyusun sebuah program (test script) yang merepresentasikan langkah-langkah dari test case manual, menggunakan bahasa pemrograman tertentu dan didukung oleh alat bantu otomatisasi eksternal [8]. Metode ini mendukung CI/CD, mempercepat deteksi bug, dan cocok untuk sistem dengan pengujian berulang seperti aplikasi web. Penggunaan tools seperti Katalon Studio membantu

mengurangi waktu uji dan menjaga stabilitas sistem saat terjadi perubahan kode, terutama pada sistem kompleks seperti dashboard penjualan.

2.2 Software Testing Life Cycle

Software Testing Life Cycle (STLC) merupakan proses terstruktur yang digunakan dalam pengujian perangkat lunak [9]. *Software Testing Life Cycle* adalah serangkaian tahapan sistematis yang digunakan dalam proses pengujian perangkat lunak untuk memastikan bahwa sistem berfungsi sesuai kebutuhan dan bebas dari kesalahan [10]. STLC terdiri dari enam tahap utama [11]. Pertama, *requirement analysis* dilakukan untuk meninjau kebutuhan sistem, baik fungsional maupun non-fungsional, guna menentukan bagian yang akan diuji. Kedua, *test planning* menyusun strategi pengujian secara menyeluruh, termasuk metode, ruang lingkup, dan risiko pengujian. Ketiga, *test case development* mencakup pembuatan skenario uji dan data uji berdasarkan kebutuhan sistem. Keempat, *environment setup* menyiapkan perangkat keras, lunak, dan data uji untuk memastikan lingkungan pengujian siap digunakan. Kelima, *test execution* adalah pelaksanaan pengujian berdasarkan *test case* yang telah disusun, pencatatan hasil uji, serta pelaporan bug. Terakhir, *test cycle closure* dilakukan untuk mengevaluasi keseluruhan proses pengujian, mendokumentasikan hasil akhir, serta menyusun laporan sebagai dasar perbaikan dan pengembangan sistem lebih lanjut. Keenam tahapan ini mendukung proses pengujian yang terstruktur, efisien, dan terukur.



Gambar 1. Tahapan *Software Testing Life Cycle*.

2.3 Katalon Studio

Katalon Studio merupakan perangkat lunak open source yang dikembangkan oleh Katalon LLC, dirancang untuk mendukung pengembang dan penguji dalam mengelola, merancang, mengeksekusi, serta mengatur pengujian otomatis secara efektif dan terstruktur [12]. Platform ini menyediakan antarmuka yang ramah pengguna serta mendukung fitur *record* dan *playback*, sehingga memudahkan pembuatan *test case* tanpa perlu menulis kode secara manual. Katalon Studio juga mendukung pengujian berbasis data (*data-driven testing*), integrasi dengan alat CI/CD seperti Jenkins dan Git, serta menghasilkan laporan uji secara otomatis dan sistematis [13]. Dengan kemampuan ini, Katalon Studio sangat cocok digunakan dalam proyek pengembangan perangkat lunak yang memerlukan pengujian berulang, efisien, konsisten, seperti pada sistem dashboard penjualan.

3. HASIL DAN PEMBAHASAN

Bab ini menjelaskan hasil implementasi *automated testing* pada dashboard penjualan menggunakan Katalon Studio serta pembahasannya. Fokus utama adalah evaluasi fungsional sistem dan efektivitas pengujian dalam mendeteksi bug.

3.1 Requirement Analysis

Tahap awal dari proses pengujian adalah melakukan analisis kebutuhan yang diperlukan [14]. Sistem dashboard ini memiliki dua jenis pengguna, yaitu admin dan user. Pengujian yang

dilakukan difokuskan pada peran admin, karena admin memiliki akses paling luas dalam sistem, termasuk pengelolaan data dan proses transaksi. Berikut merupakan rincian kebutuhan untuk pengguna admin :

1. Sistem dapat melakukan registrasi.
2. Sistem dapat melakukan autentikasi (login/logout)
3. Sistem dapat menambahkan, mengedit dan menghapus data produk.
4. Sistem dapat menambahkan, mengedit dan menghapus data kategori.
5. Sistem dapat menambahkan, mengedit dan menghapus data pelanggan.
6. Sistem dapat menambahkan, mengedit dan menghapus data penjualan.
7. Sistem dapat melakukan *approve/reject* pada user.
8. Sistem dapat menampilkan laporan penjualan dalam bentuk tabel dan grafik.
9. Sistem dapat menampilkan total penjualan, total biaya, dan total keuntungan
10. Sistem dapat melakukan filtering data berdasarkan tanggal.

3.2 Test Planning

Melakukan penentuan terhadap alat uji yang akan digunakan, merumuskan tujuan dari proses pengujian, menyusun skenario uji, menetapkan output yang diharapkan setelah pengujian, memperkirakan durasi pelaksanaan, menentukan jumlah pengguna yang disimulasikan, serta menetapkan lingkungan tempat pengujian dilakukan [15]. Pengujian dilakukan secara otomatis menggunakan Katalon Studio sebagai *automation testing tool*, dengan tujuan untuk meningkatkan efisiensi proses pengujian. Berikut merupakan rincian butir uji :

1. Pengujian fungsionalitas autentikasi.
2. Pengujian manajemen data produk.
3. Pengujian manajemen data kategori.
4. Pengujian manajemen data pelanggan.
5. Pengujian manajemen data penjualan.
6. Pengujian *approve/reject* pada user.
7. Pengujian tampilan laporan penjualan.

3.3 Test Case Development

Tahap ini mulai merancang skenario uji berdasarkan rencana pengujian [16]. Tabel 1 menunjukkan rangkaian skenario pengujian yang akan digunakan di Katalon Studio. *Test case* yang dibuat adalah hasil dari tahap *requirement analysis*.

Tabel 1. Tabel Daftar Test Case.

ID Test Case	Nama Test Case
RG01	Registrasi
LG01	Login
LO01	Logout
CT01	Manajemen data kategori
PD01	Manajemen data produk
SL01	Manajemen data sales
CM01	Manajemen data pelanggan
PF01	Profil
AP01	<i>Approve user</i>
RJ01	<i>Reject user</i>

3.3.1 Test Case Registrasi

Pengujian mencakup skenario positif dan negatif untuk memastikan validasi input berjalan baik. Hasilnya menunjukkan sistem mampu memproses pendaftaran yang valid dan menolak input yang tidak sesuai, menjaga keamanan dan integritas data pengguna.

Item	Object	Input	Output
1 - Open Browser		"http://127.0.0.1:8000"	
2 - Click	a_User		
3 - Click	a_Daftar di sini		
4 - Set Text	input_Email_email	"mila@gmail.com"	
5 - Set Text	input_Nama_name	"mila"	
6 - Select Option By Val select_Pilih Jabatan		"Staff"; false	
7 - Select Option By Val select_Pilih DivisiStaf ITStaf A_ Staff IT"; false			
8 - Set Encrypted Text	input_Password_password	"RgbbBtsp0BGNiWfCIA-	
9 - Click	button_Register		
10 - Get Text	div_Berhasil register		

Gambar 2. *Test Case Registrasi.*

3.3.2 Test Case Login

Proses login diuji mulai dari input email dan password hingga verifikasi keberhasilan melalui tampilan teks “Dashboard”. Skenario negatif juga diuji, seperti input kosong dan kredensial salah, untuk memastikan sistem menolak akses tidak sah dan menampilkan pesan error. Hasilnya menunjukkan autentikasi berjalan sesuai fungsi.

Item	Object	Input	Output
1 - Open Browser		"http://127.0.0.1:8000"	
2 - Click	a_Admin		
3 - Set Text	input_Email_email	"dina@gmail.com"	
4 - Set Encrypted Text	input_Password_password	"RgbbBtsp0BGNiWfCIA-	
5 - Click	button_Login		
6 - Verify Element Text	a_Dashboard	"Dashboard"	

Gambar 3. *Test Case Login.*

3.3.3 Test Case Manajemen Data Produk

Proses dimulai dari akses halaman dashboard, pengisian data produk seperti nama, kategori, harga, dan stok, hingga klik tombol “Save”. Keberhasilan diuji melalui verifikasi teks “Produk berhasil ditambahkan”, yang menandakan fitur tambah produk berfungsi dengan baik.

Item	Object	Input	Output
1 - Open Browser		"http://127.0.0.1:8000/dashboard"	
2 - Click	a_Sale		
3 - Wait For Element Click a_Product	a_Product	10	
4 - Click	a_Product		
5 - Wait For Element Click button_Add Product	button_Add Product	10	
6 - Click	button_Add Product		
7 - Wait For Element Value input_Product_Name_name	input_Product_Name_name	10	
8 - Execute JavaScript		"document.querySelector('input[name='Product Name']')	
9 - Select Option By Val select_Cat select_Cat		"Majalah"; false	
10 - Set Text	input_Price_In_priceIn	"10000"	
11 - Set Text	input_Price_Out_priceOut	"12000"	
12 - Set Text	input_Stock_stock	"10000"	
13 - Click	button_Save		
14 - Verify Element Text div_Prod berhasil ditambahkan	div_Prod berhasil ditambahkan	"Produk berhasil ditambahkan"	

Gambar 4. *Test Case Manajemen Data Produk.*

3.3.4 Test Case Manajemen Data Penjualan

Proses pengujian mencakup tambah, edit, dan hapus data penjualan, dengan verifikasi melalui notifikasi sistem seperti “*Sales data has been added successfully*” dan “*Sales record deleted success*”. Semua proses berjalan lancar tanpa *error*, menunjukkan fitur transaksi penjualan berfungsi baik dan dapat digunakan.

Item	Object	Input	Output
1 - Open Browser	a_Sale	"http://127.0.0.1:8000/dashboard"	
2 - Click	a_Sales		
3 - Click	button_AddSale		
4 - Click	button_AddSale		
5 - Select Option By List select -- Select Customer --	button_Bilma	"Bilma"; false	
6 - Select Option By List select -- Select Product --	button_Tabeloid_Berita	"Tabeloid Berita"; false	
7 - Click	button_Save		
8 - Verify Element Text	div_Sales	"Sales data has been added"; "Sales data has been added"	
9 - Click	button_Edit		
10 - Select Option By List select -- Select Product --	button_Tabeloid_Kuliner	"Tabeloid Kuliner"; false	
11 - Click	button_Update		
12 - Verify Element Text	div_Sales	"Sales updated successfully"; "Sales updated successfully."	
13 - Click	button_Delete		
14 - Accept Alert			
15 - Verify Element Text	div_Sales	"Sales record deleted successfully"; "Sales record deleted successfully"	

Gambar 5. Test Case Manajemen Data Penjualan.

3.3.5 Test Case Approve User

Proses dimulai dari akses menu *approved* hingga verifikasi elemen teks “*Approved*”. Hasil pengujian dinyatakan *passed*, menandakan fitur approval berjalan sesuai fungsi dan memastikan hanya pengguna terverifikasi yang mendapat akses.

Item	Object	Input	Output
1 - Open Browser	a_Admin	"http://127.0.0.1:8000/dashboard"	
2 - Click	a_Approved		
3 - Click	span_Approved		
4 - Verify Element Text	span_Approved	"Approved"	

Gambar 6. Test Case Approve User.

3.3.6 Test Case Reject User

Pengujian memverifikasi teks “*Rejected*” sebagai indikator penolakan, dan hasilnya *passed*. Ini membuktikan fitur *reject* berfungsi sesuai harapan dan memberikan umpan balik yang tepat kepada admin.

Item	Object	Input	Output
1 - Open Browser	a_Admin	"http://127.0.0.1:8000/dashboard"	
2 - Click	a_Approved		
3 - Click	span_Rejected		
4 - Verify Element Text	span_Rejected	"Rejected"	

Gambar 7. Test Case Reject User.

3.4 Environment Setup

Pada tahap ini memastikan lingkungan pengujian berjalan optimal dan sesuai kebutuhan [17]. Tabel 2 menampilkan spesifikasi perangkat yang digunakan, yaitu HP Pavilion Gaming Laptop 15-ec2xx. Spesifikasi ini memungkinkan proses pengujian berjalan lancar, termasuk saat menjalankan Katalon Studio. Selain itu, dukungan grafis dari Radeon Graphics turut menunjang visualisasi elemen antarmuka sistem yang diuji. Kombinasi spesifikasi tersebut menjadikan laptop ini sangat sesuai untuk keperluan pengujian.

Tabel 2. Tabel Spesifikasi Laptop.

Nama Perangkat	HP Pavilion Laptop 15-ec2xx
Operation System (OS)	Windows 11
Memori	16 GB (64-bit)
Processor	AMD Ryzen 5 5600H Radeon Graphics 3.3- GHz

3.5 Test Execution

Dalam tahap ini, pengujian dilakukan terhadap skenario yang telah disiapkan [18]. Setelah *test case* selesai dibuat, langkah selanjutnya adalah menjalankan pengujian. Urutan pengujian dapat dipantau log viewer. Tanda centang hijau di samping *test case* menunjukkan bahwa pengujian

berhasil atau sesuai dengan fungsionalitas yang diharapkan. Sebaliknya, tanda silang merah menandakan bahwa *test case* mengalami kegagalan atau tidak sesuai dengan fungsinya.

3.5.1 Log Viewer Register

Gambar 8 menampilkan hasil pengujian otomatis untuk *test case* register (RG01) dengan status dan waktu eksekusi 9.218 detik. Pengujian mencakup pengisian registrasi lengkap hingga pesan sukses. Seluruh langkah berjalan lancar tanpa error, menandakan fitur registrasi berfungsi sesuai harapan.



Gambar 8. Log Viewer Register.

3.5.2 Log Viewer Login

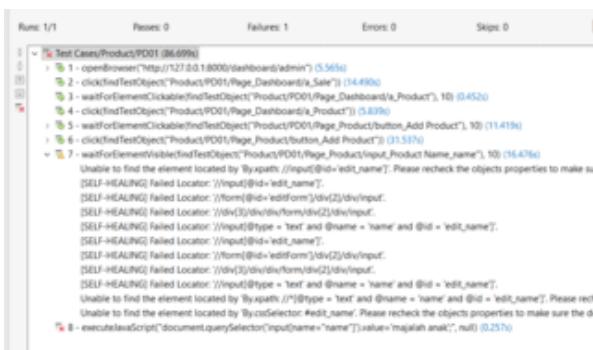
Gambar 9 menampilkan log viewer untuk *test case* login (LG01). Seluruh langkah pengujian ditandai centang hijau, menandakan proses login berhasil dijalankan dari awal hingga verifikasi tampilan dashboard tanpa *error*.



Gambar 9. Log Viewer Login.

3.5.3 Log Viewer Manajemen Data Produk

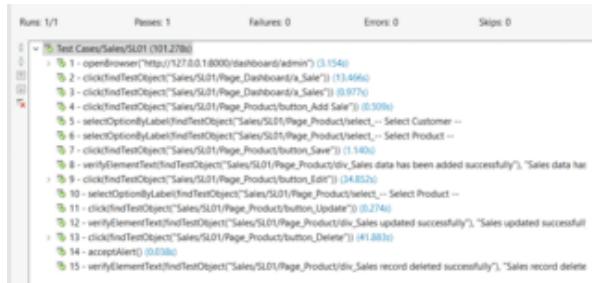
Pengujian mengalami kendala saat mendeteksi elemen input “Product Name”, ditandai error pada log viewer dan dotifikasi [SELF-HEALING]. Meski demikian, sistem mengatasi kendala dengan menjalankan perintah `executeJavaScript` untuk mengisi data secara manual. Secara keseluruhan, alur pengujian tetap dijalankan.



Gambar 10. Log Viewer Manajemen Data Produk.

3.5.4 Log Viewer Manajemen Data Penjualan

Pengujian mencakup tambah, edit, dan hapus data penjualan. Setiap langkah berhasil dijalankan, ditandai dengan notifikasi dan centang hijau pada log viewer. Hasilnya membuktikan bahwa fitur CRUD pada modul penjualan berjalan stabil dan sesuai ekspektasi.



Gambar 10. Log Viewer Manajemen Data Penjualan.

3.5.5 Log Viewer Approve User

Proses dimulai dari navigasi ke menu *approve* hingga verifikasi teks “*Approved*” menggunakan `verifyElementText`. Seluruh langkah berjalan sukses, ditandai centang hijau di log viewer tanpa *error*. Hasil ini membuktikan fitur *approve* berfungsi dengan baik, cepat, dan akurat.



Gambar 11. Log Viewer Approve User.

3.5.6 Log Viewer Reject User

Sistem memverifikasi teks “*Rejected*” menggunakan `verifyElementText`, menandakan data berhasil ditolak. Seluruh langkah berjalan lancar tanpa *error*, dengan waktu eksekusi sekitar 22 detik. Hasil ini membuktikan fitur *reject* berfungsi sesuai skenario dan telah teruji secara otomatis.



Gambar 11. Log Viewer Reject User.

3.6 Test Cycle Closure

Seluruh rangkaian pengujian telah dilaksanakan, mulai dari tahap perencanaan hingga pelaksanaan. Berdasarkan hasil dari tahap pelaksanaan pengujian, dapat disimpulkan bahwa dari sepuluh fitur yang diuji, hanya delapan di antaranya yang berfungsi dengan baik sesuai harapan. Ditemukan bug pada fitur produk terutama pada button tambah produk dan fitur sales.

3.7 Hasil Pengujian

Beberapa skenario pengujian dirancang berdasarkan fitur utama, seperti registrasi, login, logout, manajemen data produk, customer, penjualan, hingga proses approval user. Tabel 3 menunjukkan hasil pengujian.

Table 3. Tabel Hasil Pengujian Pada Website Dashboard Penjualan.

Test Case ID	Type	Test Skenario	Expected Result
--------------	------	---------------	-----------------

RG01	Valid	Melakukan registrasi dengan memasukkan data yang valid	Registrasi berhasil
LG01	Valid	Melakukan login dengan memasukkan data yang valid	Login berhasil dan menampilkan halaman dashboard
LO01	Valid	Melakukan logout dari sistem	Berhasil logout dari sistem dan menampilkan halaman login
CT01	Valid	Melakukan CRUD data kategori	Berhasil melakukan <i>create, read, update, delete</i> data kategori
PD01	Invalid	Melakukan CRUD data produk	Berhasil melakukan <i>create, read, update</i> , namun sistem tidak dapat melakukan <i>delete</i> data produk
SL01	Valid	Melakukan CRUD data penjualan	Berhasil melakukan <i>create, read, update, delete</i> data penjualan
CM01	Invalid	Melakukan CRUD data pelanggan	Berhasil melakukan <i>create, read, update</i> , namun sistem tidak dapat melakukan <i>delete</i> data pelanggan
PF01	Valid	Menampilkan data profil	Berhasil menampilkan data profil
AP01	Valid	<i>Approve</i> registrasi data user	Berhasil melakukan <i>approve</i>
RJ01	Valid	<i>Reject</i> registrasi data user	Berhasil melakukan <i>reject</i>

4. KESIMPULAN DAN SARAN

Berdasarkan kegiatan Praktik Kerja Lapangan (PKL) yang dilakukan di PT Temprina Media Grafika Surabaya, dapat disimpulkan bahwa implementasi automated testing menggunakan Katalon Studio memberikan dampak positif terhadap proses pengujian sistem dashboard penjualan. Pengujian dilakukan secara otomatis mengikuti tahapan Software Testing Life Cycle (STLC), mulai dari analisis kebutuhan hingga penutupan siklus pengujian. Fokus utama diarahkan pada fitur-fitur admin seperti autentikasi (login dan register), manajemen data (produk, pelanggan, penjualan), serta fitur approval dan rejection user. Penggunaan Katalon Studio terbukti mempercepat proses pengujian, mengurangi risiko human error, serta meningkatkan konsistensi hasil pengujian. Test case yang dijalankan menunjukkan status “Pass” tanpa error, menandakan bahwa sebagian besar fitur telah berjalan sesuai fungsinya. Meskipun ditemukan kendala seperti kegagalan dalam mendeteksi elemen, hal tersebut dapat diatasi melalui fitur *self-healing* dan perintah *JavaScript*, menunjukkan fleksibilitas tools yang digunakan. Selain mendukung kualitas sistem, kegiatan ini juga memberikan pengalaman langsung kepada mahasiswa dalam menerapkan konsep software testing secara nyata di lingkungan kerja.

Jika dalam perancangan dan penulisan laporan ini terdapat kekurangan, saya sangat membuka diri terhadap kritik dan saran yang membangun. Harapan saya, laporan ini dapat menjadi bahan referensi dan pembelajaran bagi mahasiswa lain yang ingin melakukan penelitian atau implementasi serupa, khususnya dalam bidang pengujian perangkat lunak menggunakan Katalon Studio. Semoga apa yang telah disampaikan dalam laporan ini dapat bermanfaat dan dikembangkan lebih baik lagi di masa yang akan datang.

5. DAFTAR RUJUKAN

- [1] Wahab, S. A. (2022). *Analisis Kebijakan: Dari Formulasi ke Implementasi Kebijakan Publik*. Jakarta: Rajawali Pers.
- [2] Sutabri, T. (2019). *Analisis Sistem Informasi*. Yogyakarta: Andi.
- [3] Pressman, R. S. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill.
- [4] Y. Kosasih & A. B. Cahyono (2021) "Automation Testing Tool Dalam Pengujian Aplikasi The Point Of Sale", AUTOMATA, Vol. 2 No. 1, 2021
- [5] Arfan, A. (2022). Hendrik, "Penerapan STLC dalam Pengujian Automation Aplikasi Mobile (Studi kasus: LMS Amikom Center PT. GIT Solution)," Jur. Inform. Univ. Islam Indones, 3(2), 1-6.
- [6] Shalsabilla, S. Y., Wahyuni, E. D., & Wibowo, N. C. (2024). Implementasi Blackbox Automation Testing Pada Aplikasi Donor Menggunakan Framework Stlc Dalam Lingkup Pengembangan Agile Scrum. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(2), 2261-2269.
- [7] Rahayu, S. S., Wahyudin, W., Firmansyah, D. A., & Susanti, S. (2024). Analisis Penggunaan Tools Automation Testing pada Aplikasi: Systematic Literature Review. *REMIK: Riset dan E-Jurnal Manajemen Informatika Komputer*, 8(1), 106-116.
- [8] Min, J. L., Istiqomah, A., & Rahmani, A. (2020). Evaluasi Penggunaan Manual Dan Automated Software Testing Pada Pelaksanaan End-To-End Testing. *JTT (Jurnal Teknol. Ter.*, vol. 6, no. 1, p. 18, 2020, doi: 10.31884/jtt. v6i1. 256.
- [9] Utami, D. F. N., & Setyodewi, R. H. (2023). Documentation Of Software Testing For Dafbin Aplication With IEEE 829-2008 Standar. *Jurnal RESTIKOM: Riset Teknik Informatika dan Komputer*, 5(2), 107-117
- [10] Mutiara, M. (2023). *TA: IMPLEMENTASI AUTOMATION TESTING PADA SISTEM INFORMASI INDO TOWING MENGGUNAKAN KATALON STUDIO* (Doctoral dissertation, Politeknik Negeri Lampung).
- [11] Ruliansyah, R., Huda, B., & Hananto, A. L. (2023). Penerapan software testing life cycle pada pengujian otomatisasi platform Dzikra. *CSRID (Computer Science Research and Its Development Journal)*, 15(1), 01-11.
- [12] Azhari, F. W., & Suyatno, D. F. (2024). Pengujian Otomatis GUI dengan Katalon Studio pada Situs Lowongan Kerja Jobstreet dan Glints. *Journal of Emerging Information System and Business Intelligence (JEISBI)*, 5(3), 205-213.
- [13] 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.
- [14] Mahrozi, N., & Yaqin, M. A. (2024). Pengujian aplikasi dengan metode blackbox testing: analisis boundary value dan equivalence partitioning pada aplikasi sistem pakar kucing. *J. Ilm. Sain dan Teknol*, 2985(6), 257-265.
- [15] Sentosa, S. A., Subyantoro, E., & Asrowardi, I. (2024). Pengukuran Kinerja Pada Aplikasi Video Pembelajaran UMKM Berbasis Web Dengan Metode Pengujian Beban. *ROUTERS: Jurnal Sistem dan Teknologi Informasi*, 95-102.

- [16] Ardan, T. S., Zahra, D. F., Junaedi, F. R., & Widianto, S. R. (2021). Dokumentasi Software Testing Berstandar IEEE 829-2008 untuk Learning Management System Fakultas Ilmu Komputer Universitas Subang. *Jurnal Multimedia Networking Informatics (Multinetics)*, 6, 179-191.
- [17] Chasanah, N. I., Al Ataya, M. A., Yeza, M. P., Imron, M. A., Saputra, F. A., Juliansyah, R., ... & Mindara, G. P. (2024). BLACK BOX TESTING WEBSITE RIMA BATIK DENGAN TEKNIK BOUNDARY VALUE DAN EQUIVALENCE PARTITIONING. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(6), 12814-12820.
- [18] Sumiartini, L., Rachmawati, R., & Mahendra, D. (2024). Evaluasi Kualitas Sistem Informasi Menggunakan Pengujian Black Box dan White Box. *Jurnal Sistem dan Teknologi Informasi*, 9(1), 21–30.