

Penerapan *Automated Black Box Testing* pada Sistem Monitoring Pembangunan Sekolah Menggunakan Katalon Studio

Tiara Iffatunadia*, Rosa Ariani Sukamto, Asep Wahyudin

Ilmu Komputer, FPMIPA, Universitas Pendidikan Indonesia, Bandung, Indonesia

Email: ¹*tiaraiffatunadia@upi.edu, ²rosa.ariani@upi.edu, ³away@upi.edu

Email Penulis Korespondensi: tiaraiffatunadia@upi.edu

Submitted: 27/07/2026; Accepted: 20/09/2026; Published: 30/09/2026

Abstrak—Kualitas perangkat lunak menjadi salah satu aspek penting dalam memastikan sistem dapat berjalan sesuai dengan kebutuhan pengguna. Sistem Informasi Pemantauan Pembangunan Sekolah (SiPantau) yang dikembangkan oleh Dinas Pendidikan Jawa Barat digunakan untuk memantau progres pembangunan sekolah dari aspek fisik dan keuangan. Namun, pengujian pada sistem SiPantau belum dilakukan secara sistematis, sehingga ketidaksesuaian fungsi pada sistem belum dapat diidentifikasi secara menyeluruh. Kondisi tersebut berkaitan dengan keterbatasan waktu dalam proses pengembangan sistem. Penelitian ini bertujuan untuk mengimplementasikan *automated black box testing* pada Sistem SiPantau menggunakan kombinasi teknik *Equivalence Partitioning* (EP), *Boundary Value Analysis* (BVA), dan *Decision Table* (DT), serta melakukan klasifikasi *bug* berdasarkan tingkat *severity* sebagai dasar penyusunan rekomendasi perbaikan. Metode pengujian dilakukan melalui tahapan analisis sistem, identifikasi kebutuhan pengujian, perancangan *test case*, implementasi *automated testing* menggunakan Katalon Studio, eksekusi pengujian, analisis hasil pengujian, dan penyusunan rekomendasi perbaikan. Hasil penelitian menunjukkan bahwa 153 *test case* berhasil dirancang dan dieksekusi pada 7 modul sistem SiPantau, dengan 145 *test case* berstatus *pass* dan 8 *test case* berstatus *fail*. Hasil analisis terhadap *test case* yang gagal mengidentifikasi 8 *bug* yang terdiri dari 4 *bug* dengan *severity* major dan 4 *bug* dengan *severity* minor. *Bug* yang ditemukan menunjukkan ketidaksesuaian antara hasil aktual dengan hasil yang diharapkan, yang berkaitan dengan validasi input, sinkronisasi data antar modul, dan tampilan antarmuka. Penerapan *automated black box testing* menggunakan Katalon Studio berhasil mengidentifikasi *bug* pada fungsi yang diuji dan menghasilkan temuan yang dapat menjadi dasar evaluasi dan rekomendasi perbaikan Sistem SiPantau.

Kata Kunci: *Automated Testing*; *Black Box Testing*; Katalon Studio; *Software Testing*; Sistem Monitoring; Sekolah

Abstract— *Software quality is an important aspect in ensuring that a system operates according to user requirements. The School Development Monitoring Information System (SiPantau), developed by the West Java Education Office, is used to monitor school development progress in terms of physical and financial aspects. However, time constraints during the development process resulted in the absence of a dedicated stage for evaluating system quality through software testing. This study aims to implement automated black box testing on the SiPantau system by combining Equivalence Partitioning (EP), Boundary Value Analysis (BVA), and Decision Table (DT) techniques, as well as classifying detected bugs based on their severity levels as a basis for developing system improvement recommendations. The testing process was conducted through several stages, including system analysis, identification of testing requirements, test case design, implementation of automated testing using Katalon Studio, test execution, result analysis, and formulation of improvement recommendations. The results showed that 153 test cases were successfully designed and executed across seven SiPantau modules, with 145 test cases passing and 8 test cases failing. Analysis of the failed test cases identified 8 bugs, consisting of 4 major-severity bugs and 4 minor-severity bugs. The identified bugs indicated discrepancies between the actual and expected results, particularly in input validation, data synchronization between modules, and user interface presentation. The implementation of automated black-box testing using Katalon Studio successfully identified bugs in the tested functions and produced findings that can serve as a basis for evaluating the system and formulating improvement recommendations for the SiPantau system.*

Keywords: *Automated Testing*; *Black Box Testing*; Katalon Studio; *Software Testing*; Monitoring System; School

1. PENDAHULUAN

Dinas Pendidikan Jawa Barat melalui Unit Pelaksana Teknis Daerah (UPTD) Teknologi Informasi dan Komunikasi Pendidikan (TIKOMDIK) telah mengembangkan Sistem Informasi Pemantauan Pembangunan Sekolah yang dinamakan SiPantau. Sistem ini digunakan untuk memantau progres pembangunan sekolah baik dari aspek fisik maupun keuangan secara terintegrasi.

Sistem monitoring merupakan suatu sistem yang memiliki proses pengumpulan data dari berbagai sumber secara *real-time* dan umumnya tujuan dari penerapan sistem monitoring adalah sebagai bentuk tindakan pencegahan [1]. Sehingga, dengan adanya sistem SiPantau, proses bisnis menjadi lebih transparan dan meminimalisir ketidaksesuaian antara progres pembangunan fisik dengan progres keuangan. Namun, ditemukan bahwa dalam proses pengembangan sistem ini tidak terdapat tahapan khusus untuk mengevaluasi kualitas sistem karena terbatasnya waktu yang diberikan untuk proses pengembangannya. Padahal, suatu proses pengembangan perangkat lunak yang cepat tanpa melibatkan *independent tester* dan hanya fokus pada pengembangan pemenuhan kebutuhan fungsional oleh developer, beresiko memicu penurunan kualitas sistem [2].

Hingga saat ini, dalam menentukan kelayakan suatu perangkat lunak, belum ada kesepakatan akhir mengenai mana standar kualitas yang paling tepat untuk semua jenis perangkat lunak [3][4]. Walaupun demikian,

menurut prinsip ISO 9000, upaya untuk memperbaiki kualitas suatu sistem, dapat dilakukan melalui proses testing atau pengujian guna mendeteksi *bug* yang nantinya akan menjadi landasan komprehensif dalam membuat susunan rekomendasi perbaikan sistem [5]. Selain itu, proses testing juga penting dilakukan untuk mengidentifikasi ketidaksesuaian fungsi pada sistem yang telah dikembangkan dengan kebutuhan sistem [6].

Berdasarkan permasalahan tersebut, penelitian ini mengimplementasikan *automated testing* menggunakan metode *black box* yang dapat digunakan untuk mendeteksi ketidaksesuaian fungsi pada sistem [7], sehingga dapat mendukung proses analisis pembuatan rekomendasi perbaikan dan menyediakan hasil pengujian yang dapat digunakan sebagai informasi pendukung dalam proses pengembangan dan pemeliharaan sistem di masa yang akan datang. Selain itu, metode *black box* dengan fokus pada jenis pengujian fungsional dan integrasi, dipilih karena metode yang sesuai untuk mengukur sistem SiPantau [8].

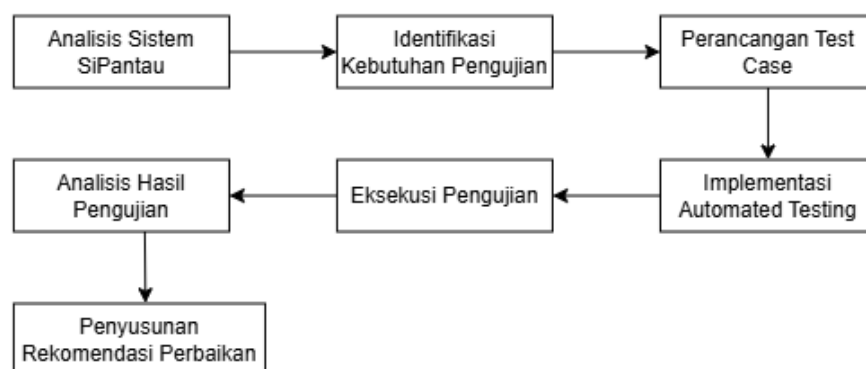
Dengan menggunakan metode *black box*, pengujian dapat dilakukan tanpa perlu mengetahui struktur kodenya. *Black box* juga memiliki teknik *Equivalence Partitioning* (EP), *Boundary Value Analysis* (BVA), dan *Decision Table* (DT) [9] yang akan diterapkan pada penelitian ini. Kombinasi ketiga teknik tersebut digunakan untuk mendukung evaluasi kesesuaian fungsi sistem terhadap kondisi dan aturan pengujian yang telah ditetapkan. *Bug* yang akan ditemukan dari proses pengujian akan diklasifikasikan berdasarkan tingkat keparahan atau *severity*. Klasifikasi ini digunakan untuk menentukan prioritas perbaikan sistem [10].

Dalam lima tahun terakhir, berbagai penelitian telah menerapkan metode *black box* untuk pada pengujian sistem. Beberapa penelitian menggunakan teknik EP dan BVA, namun proses pengujiannya masih dilakukan secara manual [11][12]. Penelitian lain juga menerapkan teknik DT dalam pengujian fungsional, tetapi pelaksanaan pengujiannya masih bersifat manual [13]. Di sisi lain, beberapa penelitian telah melakukan pengujian otomatis menggunakan metode *black box*, namun belum mengadopsi teknik EP, BVA, dan DT dalam proses pengujiannya [14][15]. Selain itu, penelitian yang menggunakan metode *black box* untuk mengevaluasi kualitas sistem juga masih mengandalkan proses pengujian secara manual [16]. Sebagian besar penelitian tersebut berfokus pada kecepatan kebutuhan waktu uji dan validitas hasil pengujian tanpa melakukan pengklasifikasian *bug* berdasarkan *severity* [17][18]. Pada sistem monitoring, penelitian yang menggunakan metode *white box* telah dilakukan [19], sedangkan penelitian yang menerapkan *black box* pada sistem monitoring masih melaksanakan proses pengujian secara manual [20].

Berdasarkan kajian tersebut, masih terdapat kesenjangan penelitian, yaitu belum adanya penelitian yang mengintegrasikan pengujian otomatis menggunakan metode *black box testing* dengan teknik EP, BVA, dan DT pada sistem monitoring sekaligus melakukan klasifikasi *bug* berdasarkan tingkat *severity*. Oleh karena itu, tujuan dari penelitian ini adalah untuk mengimplementasikan *automated black box testing* pada Sistem SiPantau dengan mengkombinasikan teknik EP, BVA, dan DT untuk menguji berbagai karakteristik input, batas nilai, dan aturan bisnis pada fungsi yang diuji serta mengklasifikasikan *bug* berdasarkan tingkat *severity* sebagai dasar penyusunan rekomendasi perbaikan sistem. Dengan demikian, penelitian ini tidak hanya mengevaluasi keberhasilan fungsi sistem, tetapi juga menyediakan informasi mengenai temuan *bug* dan tingkat *severity* sebagai dasar penyusunan rekomendasi perbaikan.

2. METODOLOGI PENELITIAN

Penelitian ini akan dilakukan melalui 7 tahapan yang dapat dilihat pada Gambar 1. Tahapan Penelitian sebagai berikut :



Gambar 1. Tahapan Penelitian

2.1 Tahapan Penelitian

- Analisis sistem SiPantau, pada tahap pertama ini dilakukan analisis pada sistem SiPantau untuk memahami alur proses bisnis, fitur-fitur yang tersedia, dan menentukan modul-modul yang akan menjadi prioritas objek pengujian.
- Identifikasi kebutuhan pengujian, tujuan dari tahap identifikasi ini adalah untuk menghasilkan daftar fitur yang akan diuji dengan menggunakan metode *black box*
- Perancangan *test case*, pada tahap ini *test case* atau skenario pengujian dibuat menggunakan teknik EP atau BVA atau DT untuk menguji kelas input, nilai batas, dan kombinasi kondisi atau aturan bisnis yang relevan dengan fungsi yang diuji.
- Implementasi *automated testing*, seluruh *test case* yang telah dibuat diimplementasikan ke dalam skrip *automated testing* lalu dibuat menjadi *test suite* menggunakan *tools* Katalon Studio
- Eksekusi Pengujian, ditahap ini skrip pengujian dijalankan pada sistem SiPantau menggunakan Katalon Studio dengan *web browser* Chrome untuk mendapatkan hasil berupa *pass* atau *fail* dan juga informasi *error* atau *bug*.
- Analisis hasil pengujian, setelah mendapatkan hasil pengujian dari tahap eksekusi pengujian, dilakukan analisis pada hasil pengujian untuk mengidentifikasi *bug* dan mengklasifikasikan *bug* berdasarkan tingkat *severity*.
- Penyusunan rekomendasi perbaikan, pada tahap ini penyusunan rekomendasi dan prioritas perbaikan dibuat berdasarkan hasil dari klasifikasi *severity bug*.

2.2 Perancangan Pengujian

Perancangan pengujian dilakukan pada Sistem SiPantau dengan fokus pengujian fungsional dan integrasi yang melibatkan *user admin* sekolah dan *user admin* Kantor Cabang Dinas (KCD). Pemilihan modul sebagai objek pengujian didasarkan pada relevansinya terhadap tujuan dan fungsi Sistem SiPantau serta keterkaitannya dengan alur proses bisnis dan kebutuhan pengguna. Informasi mengenai tujuan dan fungsi sistem diperoleh dari dokumen resmi yang diterbitkan oleh Dinas Pendidikan Provinsi Jawa Barat, sedangkan informasi mengenai alur proses bisnis dan kebutuhan pengguna diperoleh melalui wawancara dengan penanggung jawab dan analis sistem SiPantau. Berdasarkan hasil analisis terhadap dokumen resmi dan wawancara tersebut, modul yang dipilih sebagai objek pengujian adalah Login, Dashboard, Realisasi, Progres Realisasi, Pemantauan Laporan, Kelola Pekerjaan, dan Logout. Modul-modul tersebut dipilih karena memiliki keterkaitan dengan proses pengelolaan, pemantauan, dan pelaporan pembangunan sekolah yang menjadi fungsi sistem yang diteliti. Sementara itu, terdapat fungsi lain pada Sistem SiPantau, seperti ganti *password* dan ubah penanggung jawab. Fungsi tersebut tidak termasuk dalam objek pengujian karena berdasarkan hasil analisis dokumen dan wawancara, fungsi tersebut tidak berkaitan langsung dengan proses bisnis pemantauan dan pengelolaan pembangunan sekolah yang menjadi fokus penelitian. Modul Login dan Logout dipilih karena berkaitan dengan proses autentikasi dan pengelolaan sesi pengguna. Modul Dashboard dipilih karena berkaitan dengan pengelolaan informasi sekolah dan pemantauan rekapitulasi data. Modul Realisasi dan Progres Realisasi dipilih karena berkaitan langsung dengan pencatatan dan pelaporan progres pembangunan serta progres realisasi keuangan, termasuk penerapan aturan validasi nilai dan keterkaitan data antar-modul. Modul Pemantauan Laporan dipilih karena digunakan untuk memantau dan memverifikasi informasi hasil pembangunan serta progres keuangan. Sementara itu, modul Kelola Pekerjaan dipilih karena menyediakan fungsi tambah, edit, dan hapus data pekerjaan dengan beberapa aturan validasi input.

Pemilihan teknik pengujian pada setiap modul disesuaikan dengan karakteristik fungsi, input, dan aturan bisnis yang perlu diverifikasi. Teknik EP digunakan ketika terdapat kelompok input valid dan tidak valid. Teknik BVA digunakan ketika terdapat nilai batas yang perlu diverifikasi, sedangkan teknik DT digunakan ketika hasil suatu proses ditentukan oleh kombinasi kondisi atau aturan bisnis. Berdasarkan karakteristik tersebut, penerapan teknik pada masing-masing modul ditentukan sesuai kebutuhan pengujiannya, sebagaimana ditunjukkan pada Tabel 1 :

Tabel 1. Modul yang Diuji

Modul	Pengguna	Teknik
Login	Admin Sekolah, Admin KCD	EP
Dashboard	Admin Sekolah, Admin KCD	EP, BVA
Realisasi	Admin Sekolah	EP, BVA, DT
Progres Realisasi	Admin KCD	EP, BVA, DT
Pemantauan Laporan	Admin KCD	EP, BVA, DT
Kelola Pekerjaan	Admin KCD	EP, BVA
Logout	Admin Sekolah, Admin KCD	-

2.3 Penyusunan Test Case

Penyusunan *test case* dilakukan menggunakan teknik *Equivalence Partitioning* (EP), *Boundary Value Analysis* (BVA), dan *Decision Table* (DT) yang disesuaikan dengan karakteristik input dan aturan bisnis pada setiap modul. *Equivalence Partitioning* untuk membagi data inputan valid dan tidak valid ke dalam kelompok input, *Boundary*

Value Analysis untuk menguji inputan yang berada diambang batas nilai dari rentang data, dan *Decision Table* untuk menguji kombinasi kondisi dari alur bisnis sistem SiPantau. Penggunaan teknik-teknik tersebut akan disesuaikan dengan karakteristik modul atau kebutuhan verifikasi masing-masing fitur. Dengan demikian, satu modul dapat menggunakan lebih dari satu teknik apabila memiliki karakteristik pengujian yang berbeda. Contoh dari penyusunan *test case* dapat dilihat pada Tabel 2. *Test Case* sebagai berikut :

Tabel 2. *Test Case*

Modul	Skenario	Input	Expected Result	Teknik
Login	Login Berhasil	Username dan password valid	Login berhasil	EP
Login	Username salah	Username tidak valid, password valid	Login gagal	EP
Realisasi	Tambah realisasi	Progres realisasi = -1, nilai tepat dibawah batas bawah	Tambah realisasi gagal	BVA
Realisasi	Tambah realisasi	Progres realisasi = lebih rendah dari sebelumnya	Tambah realisasi gagal	DT

Pada modul Login, teknik EP digunakan untuk menguji kelompok input *username* dan *password* yang valid dan tidak valid. Pada modul Realisasi, teknik BVA digunakan untuk menguji nilai input pada kolom progres realisasi berdasarkan batas nilai yang diperbolehkan, khususnya nilai di sekitar batas bawah dan atas untuk memastikan sistem menolak input yang berada di luar rentang yang ditentukan. Sementara itu, teknik DT digunakan untuk menguji aturan bisnis yang bergantung pada kondisi data sebelumnya, yaitu nilai pada kolom progres realisasi yang diinput tidak boleh lebih rendah daripada nilai progres sebelumnya. Dengan demikian, pemilihan BVA dan DT didasarkan pada karakteristik kondisi yang diverifikasi pada masing-masing *test case*.

2.4 Implementasi Automated Testing

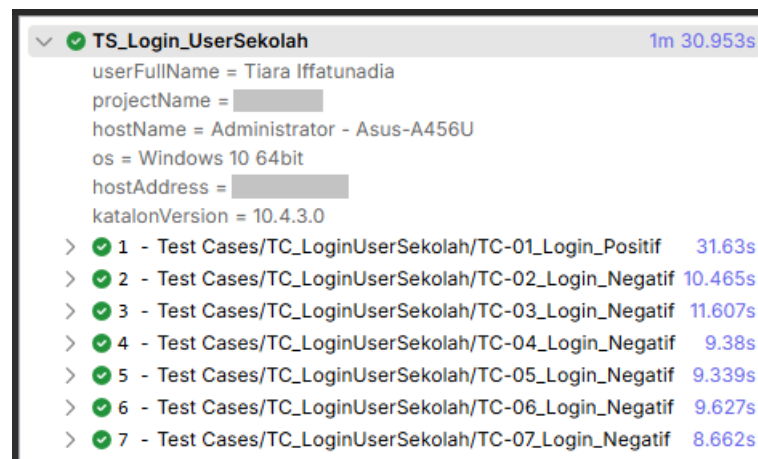
Automated testing berbasis web dilakukan pada lingkungan Sistem SiPantau yang sama dengan yang digunakan oleh user Sekolah dan KCD. Pengujian dilakukan menggunakan akun *dummy* yang disediakan untuk keperluan pengujian. Perangkat keras atau *hardware* yang digunakan berupa laptop dengan spesifikasi *processor* Intel Core i5 dan RAM 8 GB. Perangkat lunak atau *software* yang digunakan untuk *automated testing* ini adalah Katalon Studio versi 10.4.3.0. Katalon Studio dipilih karena memiliki fitur *record web*, *spy web*, dan *object repository* sehingga dapat memudahkan proses pembuatan *test case*. Pengujian otomatis sistem SiPantau dilakukan menggunakan browser Chrome pada sistem operasi Windows 10 64-bit. Implementasi diawali dengan mengubah skenario uji menjadi *test case* menggunakan fitur *record web* dan membuat objek UI menggunakan fitur *spy web* apabila terdapat kegagalan pengambilan komponen menggunakan fitur *record web*. Objek-objek tersebut disimpan ke dalam *object repository* agar dikenali oleh Katalon dan memudahkan apabila terjadi pemanggilan berulang. *Test case* akan berbentuk skrip berbahasa Groovy. Berikut adalah Gambar 2. Implementasi *Test Case* Pada Katalon Studio yang menampilkan contoh *test case* untuk skenario login :



Gambar 2. Implementasi *Test Case* Pada Katalon Studio

Setelah seluruh *test case* selesai dibuat, pengujian dijalankan melalui fitur *test suite* agar dapat mengeksekusi beberapa *test case* sekaligus secara otomatis. Pengelompokan *test suite* dilakukan berdasarkan jenis modul atau alur proses bisnis yang saling berkaitan. Hasil dari menjalankan/*running test suite* ini berupa laporan pengujian

yang berisi status *pass* atau *fail*, waktu eksekusi, dan informasi *error* yang digunakan untuk menganalisis hasil pengujian. Contoh hasil eksekusi dapat dilihat pada Gambar 3. *Log Viewer Running Test Suite* Pada Katalon Studio yang menampilkan hasil, sebagai berikut :



Gambar 3. *Log Viewer Running Test Suite* Pada Katalon Studio

2.5 Analisis Hasil Pengujian

Hasil pengujian dari *automated testing* dianalisis berdasarkan status eksekusi setiap *test case*, yaitu *pass* atau *fail*. *Test case* dengan status *pass* menunjukkan bahwa fungsi yang dimiliki oleh sistem telah berjalan lancar dan sesuai dengan hasil yang diharapkan. Sedangkan *test case* dengan status *fail* mengidentifikasi bahwa terdapat ketidaksesuaian antara hasil aktual dengan hasil yang diharapkan. Selanjutnya, setiap terdapat kegagalan pada pengujian dilakukan verifikasi secara manual untuk mengetahui penyebab dan lokasi terjadinya kesalahan pada sistem. Verifikasi dilakukan dengan menjalankan kembali setiap *test case* yang menghasilkan status *fail* untuk memastikan bahwa kegagalan dapat direproduksi. Selanjutnya, hasil aktual dibandingkan dengan *expected result* yang telah ditetapkan pada tahap perancangan *test case*. Apabila terdapat ketidaksesuaian, dilakukan pemeriksaan terhadap *error log*, pesan kesalahan, *screenshot*, input yang digunakan, serta kondisi modul terkait untuk mengidentifikasi penyebab kegagalan. Pemeriksaan juga dilakukan untuk memastikan bahwa kegagalan bukan disebabkan oleh kesalahan input, konfigurasi pengujian, atau kondisi lingkungan pengujian. *Test case* kemudian dikategorikan sebagai *confirmed bug* apabila ketidaksesuaian dapat direproduksi dan penyebabnya berasal dari sistem yang diuji.

Kemudian, *bug* yang ditemukan dilakukan pengklasifikasian berdasarkan tingkat keparahan atau *severity* untuk mengetahui tingkat dampaknya terhadap fungsi dan proses bisnis sistem. Hasil dari pengklasifikasian *bug* ini, digunakan untuk menentukan tingkat prioritas perbaikan sehingga rekomendasi perbaikan yang diberikan dapat disesuaikan dengan tingkat urgensi setiap *bug*. Klasifikasi *severity* yang digunakan dalam penelitian ini mengadaptasi 5 tingkat *severity* yang digunakan oleh Tian *et al.* [21]. Deskripsi masing-masing *severity* disesuaikan dengan dampak *bug* terhadap fungsi dan proses bisnis Sistem SiPantau. Klasifikasi yang digunakan dapat dilihat pada Tabel 3. Klasifikasi *Severity* :

Tabel 3. Klasifikasi *Severity*

Jenis <i>Severity</i>	Deskripsi
<i>Blocker</i>	<i>Bug</i> menyebabkan sistem tidak dapat digunakan atau proses utama mengalami kegagalan sehingga proses pengujian atau proses bisnis tidak dapat dilanjutkan
<i>Critical</i>	<i>Bug</i> menyebabkan fungsi utama tidak dapat digunakan atau kegagalan serius pada proses utama
<i>Major</i>	<i>Bug</i> mengganggu fungsi utama, tetapi masih terdapat alternatif proses atau masih dapat digunakan
<i>Minor</i>	<i>Bug</i> mempengaruhi sedikit fungsi, namun tidak menghambat proses utama
<i>Trivial</i>	<i>Bug</i> dengan dampak sangat kecil, seperti kesalahan tampilan atau pesan sistem, tidak mempengaruhi fungsi utama

Penentuan klasifikasi *severity* dilakukan dengan menganalisis dampak setiap *bug* terhadap fungsi dan proses bisnis Sistem SiPantau. Hasil analisis kemudian dibandingkan dengan deskripsi *severity* pada Tabel 3 untuk menentukan

klasifikasi yang sesuai. Klasifikasi ditetapkan berdasarkan tiga aspek, yaitu fungsi yang terdampak, kemampuan proses untuk tetap dilanjutkan setelah terjadi *bug*, serta dampak terhadap validitas data atau penyajian informasi. Dengan demikian, *severity* setiap *bug* ditentukan berdasarkan kesesuaian antara kondisi aktual dan kriteria *severity* yang telah ditetapkan.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Implementasi Automated Black Box Testing

Berdasarkan tahapan implementasi yang telah dilakukan, seluruh *test case* yang dirancang menggunakan teknik *Equivalence Partitioning* (EP), *Boundary Value Analysis* (BVA), dan *Decision Table* (DT) berhasil diimplementasikan ke dalam Katalon Studio. *Test case* dikelompokkan ke dalam beberapa *test suite* sesuai modul yang diuji sehingga proses eksekusi dapat dilakukan secara otomatis menggunakan *browser* Google Chrome. Hasil eksekusi menghasilkan status *Pass* dan *Fail* beserta informasi waktu eksekusi, log kesalahan, dan *screenshot* yang digunakan sebagai dasar analisis terhadap ketidaksesuaian fungsi yang ditemukan pada sistem.

Tabel 4. Rangkuman Hasil Eksekusi *Test Case*

Modul	Jumlah <i>Test Case</i>	<i>Pass</i>	<i>Fail</i>
Login	14	14	0
Dashboard	50	49	1
Realisasi	36	35	1
Progres Realisasi	19	17	2
Pemantauan Laporan	19	19	0
Kelola Pekerjaan	12	8	4
Logout	3	3	0

Berdasarkan Tabel 4. Rangkuman Hasil Eksekusi *Test Case*, terlihat bahwa sebagian besar *test case* berhasil dijalankan sesuai dengan hasil yang diharapkan. Namun, masih ditemukan beberapa *test case* dengan status *fail* yang menunjukkan adanya ketidaksesuaian fungsi pada beberapa modul sistem.

3.2 Hasil Analisis Failed Test Case

Berdasarkan 8 *test case* yang memperoleh status *fail*, dilakukan verifikasi menggunakan prosedur yang telah dijelaskan pada Bagian 2.5. Hasil verifikasi menunjukkan bahwa seluruh 8 *test case* dapat dikonfirmasi sebagai *bug* karena terdapat ketidaksesuaian antara *actual result* dan *expected result* yang berasal dari sistem yang diuji. Hasil analisis *bug* dapat dilihat pada Tabel 5:

Tabel 5. Hasil Analisis *Bug*

ID <i>Test Case</i>	Modul	<i>Expected Result</i>	<i>Actual Result</i>	<i>Severity</i>
TC-DS-09	Dashboard	Data dengan nilai negatif tidak bisa terinput dan tidak bisa tersimpan	Data dengan nilai negatif bisa terinput dan bisa tersimpan	Major
TC-RS-22	Realisasi	Data terhapus, semua teks persen dan data bar progres tersinkronisasi	Data terhapus, semua teks persen dan bar progres gagal tersinkronisasi	Major
TC-PR-05	Progres Realisasi	Data dengan nilai negatif tidak bisa terinput dan tidak bisa tersimpan	Data dengan nilai negatif bisa terinput dan bisa tersimpan	Major
TC-PR-09	Progres Realisasi	Data dengan nilai lebih dari batas atas tidak bisa terinput dan tidak bisa tersimpan	Data dengan nilai lebih dari batas atas bisa terinput dan bisa tersimpan	Major
TC-KP-04	Kelola Pekerjaan	Muncul pesan validasi dropdown perlu dipilih pada form tambah pekerjaan	Pesan validasi dropdown perlu dipilih tidak muncul	Minor

TC-KP-06	Kelola Pekerjaan	Muncul pesan validasi nilai harus lebih dari 0 pada form tambah pekerjaan	Pesan validasi nilai harus lebih dari 0 tidak muncul	Minor
TC-KP-10	Kelola Pekerjaan	Muncul pesan validasi dropdown perlu dipilih pada form edit pekerjaan	Pesan validasi dropdown perlu dipilih tidak muncul	Minor
TC-KP-12	Kelola Pekerjaan	Muncul pesan validasi nilai harus lebih dari 0 pada form edit pekerjaan	Pesan validasi nilai harus lebih dari 0 tidak muncul	Minor

a. *Bug* pada modul dashboard

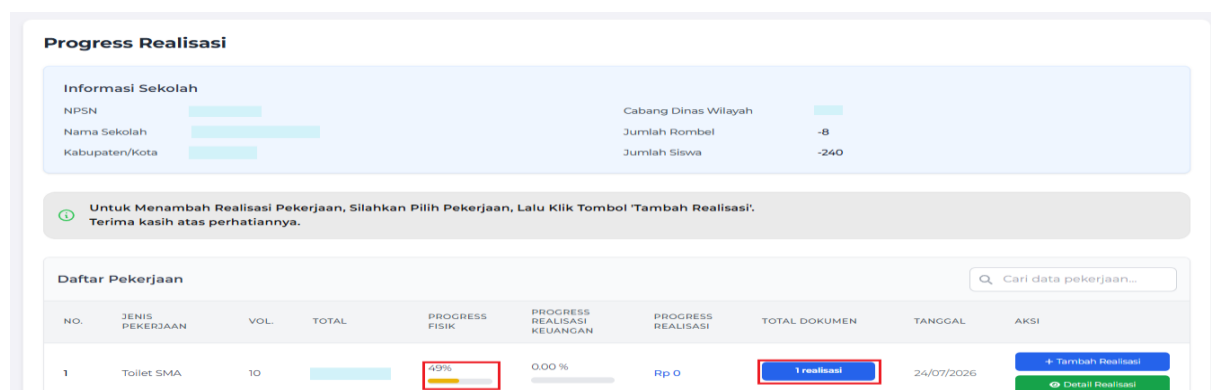
Pada modul dashboard admin sekolah, ditemukan bahwa kolom jumlah rombongan belajar dan jumlah siswa pada form edit info sekolah menerima inputan nilai negatif serta menyimpan nilai negatif, seperti yang ditunjukkan pada Gambar 4. Informasi Sekolah. Berdasarkan aturan bisnis, kedua kolom tersebut seharusnya hanya menerima input nilai numerik non-negatif agar informasi sekolah yang tersimpan adalah informasi yang valid. Kondisi ini menunjukkan bahwa validasi input belum diterapkan dengan optimal. Oleh karena itu, *bug* ini dikategorikan sebagai *major severity* karena mempengaruhi keakuratan data yang ditampilkan pada sistem, namun tidak menyebabkan fungsi utama dan sistem berhenti beroperasi.



Gambar 4. Informasi Sekolah

b. *Bug* pada modul realisasi

Pada pengujian antara modul realisasi sekolah dan dashboard *user* sekolah serta modul pemantauan laporan *user* KCD, ditemukan bahwa penghapusan data realisasi pekerjaan fisik tidak memperbarui nilai progres yang ditampilkan pada tabel daftar pekerjaan dan daftar pekerjaan (CPCL) atau dapat dikatakan tidak tersinkronisasi. Berdasarkan aturan bisnis, perubahan data pada modul realisasi *user* sekolah seharusnya mempengaruhi perhitungan progress sehingga informasi yang ditampilkan pada tabel daftar pekerjaan dan tabel daftar pekerjaan (CPCL) selalu sesuai dengan kondisi data terbaru. Namun, hasil pengujian menunjukkan bahwa nilai progres masih menampilkan data sebelum penghapusan dilakukan, seperti yang ditunjukkan oleh Gambar 5, 6, 7, dan 8. *Bug* ini dikategorikan sebagai *major severity* karena menyebabkan ketidaksesuaian informasi antar modul dan dapat menghambat proses pengelolaan realisasi berikutnya.



Gambar 5. Kondisi Progres Fisik Sebelum Penghapusan Data Realisasi Pada Modul Realisasi Sekolah

Progress Realisasi

Informasi Sekolah

NPSN		Cabang Dinas Wilayah	
Nama Sekolah		Jumlah Rombel	-8
Kabupaten/Kota		Jumlah Siswa	-240

Untuk Menambah Realisasi Pekerjaan, Silahkan Pilih Pekerjaan, Lalu Klik Tombol 'Tambah Realisasi'.
Terima kasih atas perhatiannya.

Daftar Pekerjaan

Cari data pekerjaan...

NO.	JENIS PEKERJAAN	VOL.	TOTAL	PROGRESS FISIK	PROGRESS REALISASI KEUANGAN	PROGRESS REALISASI	TOTAL DOKUMEN	TANGGAL	AKSI
1	Toilet SMA	10		49%	0.00 %	Rp 0	0 realisasi Belum ada status	24/07/2026	+ Tambah Realisasi Detail Realisasi

Gambar 6. Kondisi Progres Fisik Sesudah Penghapusan Data Realisasi Pada Modul Realisasi Sekolah

Informasi Sekolah

[Edit Info Sekolah](#)

Detail Sekolah

NPSN		Cabang Dinas Wilayah	
Nama Sekolah		Jumlah Rombel	-8
Kabupaten/Kota		Jumlah Siswa	-240

Daftar Pekerjaan

Cari data pekerjaan...

NO.	JENIS PEKERJAAN	VOL.	TOTAL	PROGRESS FISIK	PROGRESS REALISASI	PROGRESS REALISASI KEUANGAN	TANGGAL	AKSI
1	Toilet SMA	10		49%	0.00 %	Rp 0	24/07/2026	Detail Realisasi

Gambar 7. Kondisi Progres Fisik Sesudah Penghapusan Data Realisasi Pada Modul Dashboard Sekolah

Informasi Sekolah

Nama Sekolah		Cabang Dinas Wilayah	
NPSN		Jumlah Rombel	-8
Alamat		Jumlah Siswa	-240

Daftar Pekerjaan (CPCL)

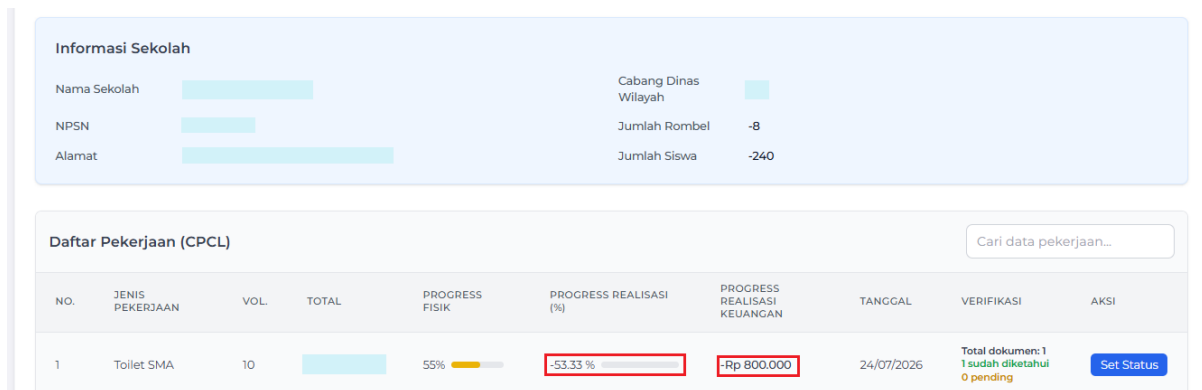
Cari data pekerjaan... [+ Tambah Pekerjaan](#)

NO.	JENIS PEKERJAAN	VOL.	TOTAL	PROGRESS FISIK	PERSENTASI REALISASI KEUANGAN (%)	PROGRESS REALISASI KEUANGAN	TOTAL DOKUMEN	TANGGAL	AKSI
1	Toilet SMA	10		49%	0.00 %	Rp 0	0 dokumen	24/07/2026	Detail

Gambar 8. Kondisi Progres Fisik Sesudah Penghapusan Data Realisasi Pada Modul Pemantauan Laporan KCD

c. Bug pada modul progres realisasi

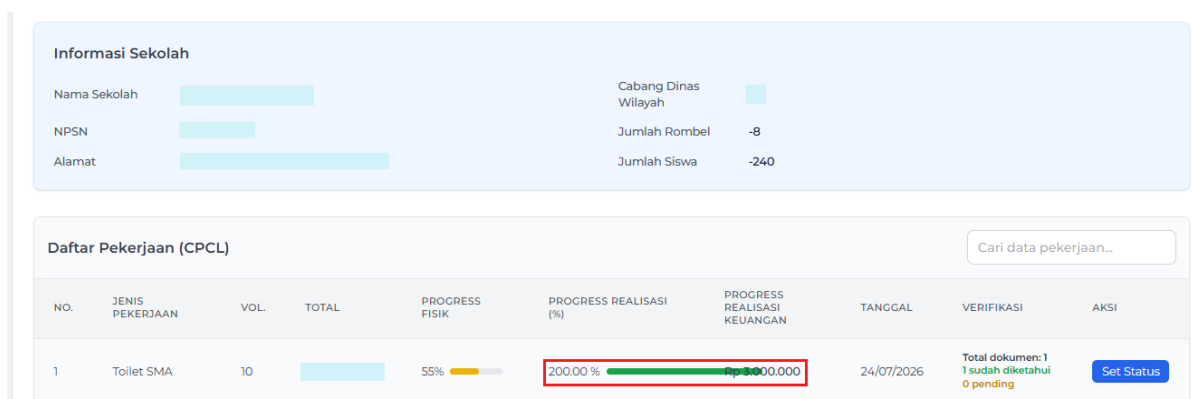
Pada pengujian modul progres realisasi *user* KCD, ditemukan bahwa kolom progres realisasi keuangan pada form edit detail realisasi keuangan masih menerima dan menyimpan nilai negatif, seperti yang ditunjukkan oleh Gambar 9. Progres Realisasi Keuangan. Berdasarkan aturan bisnis, kolom tersebut seharusnya hanya menerima nilai numerik non-negatif. Kondisi ini menunjukkan bahwa validasi input belum diterapkan secara optimal sehingga memungkinkan tersimpannya data progres keuangan yang tidak valid. *Bug* ini dikategorikan sebagai *major severity* karena mempengaruhi validitas data pada proses bisnis utama serta berpotensi menghasilkan informasi progres keuangan yang tidak akurat, meskipun sistem masih dapat dioperasikan.



NO.	JENIS PEKERJAAN	VOL.	TOTAL	PROGRESS FISIK	PROGRESS REALISASI (%)	PROGRESS REALISASI KEUANGAN	TANGGAL	VERIFIKASI	AKSI
1	Toilet SMA	10		55%	-53.33 %	-Rp 800.000	24/07/2026	Total dokumen: 1 1 sudah diketahui 0 pending	Set Status

Gambar 9. Progres Realisasi Keuangan

Selanjutnya, ditemukan bahwa kolom progres realisasi keuangan pada form edit detail realisasi keuangan dapat menerima dan menyimpan nilai yang lebih dari batas atas, yaitu lebih besar dari 100%, seperti yang ditunjukkan oleh Gambar 10. *Bar Progress* Realisasi Keuangan. Berdasarkan aturan bisnis, inputan untuk kolom tersebut seharusnya memiliki batas atas bernilai 100% atau sama dengan total permintaan anggaran. Kondisi ini menunjukkan bahwa validasi input batas atas (*upper bound validation*) belum diterapkan secara optimal sehingga memungkinkan tersimpannya data progres keuangan yang tidak valid. *Bug* ini dikategorikan sebagai *major severity* karena mempengaruhi validitas data pada proses bisnis utama serta berpotensi menghasilkan informasi progres keuangan yang tidak akurat, meskipun sistem masih dapat dioperasikan.

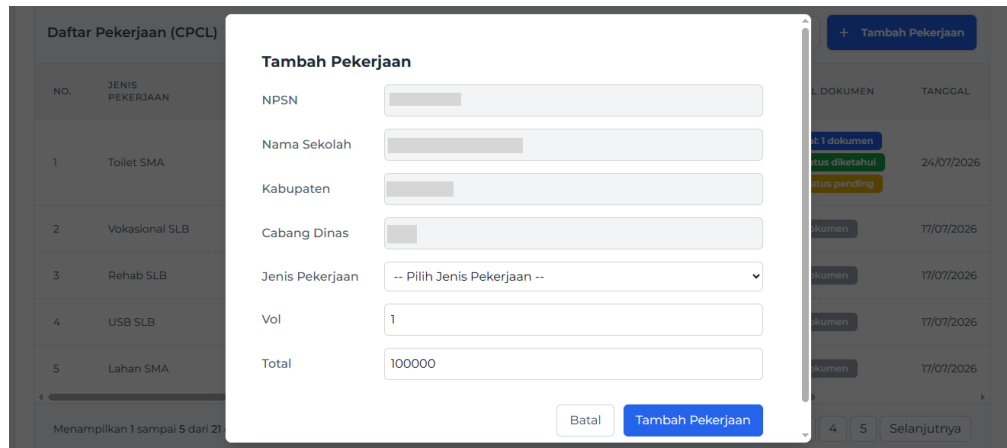


NO.	JENIS PEKERJAAN	VOL.	TOTAL	PROGRESS FISIK	PROGRESS REALISASI (%)	PROGRESS REALISASI KEUANGAN	TANGGAL	VERIFIKASI	AKSI
1	Toilet SMA	10		55%	200.00 %	-Rp 800.000	24/07/2026	Total dokumen: 1 1 sudah diketahui 0 pending	Set Status

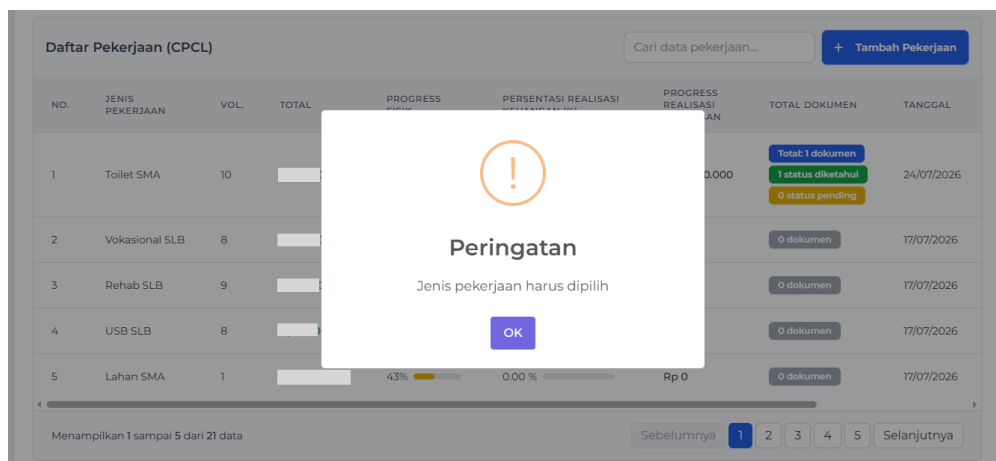
Gambar 10. Bar Progress Realisasi Keuangan

d. *Bug* pada modul kelola pekerjaan

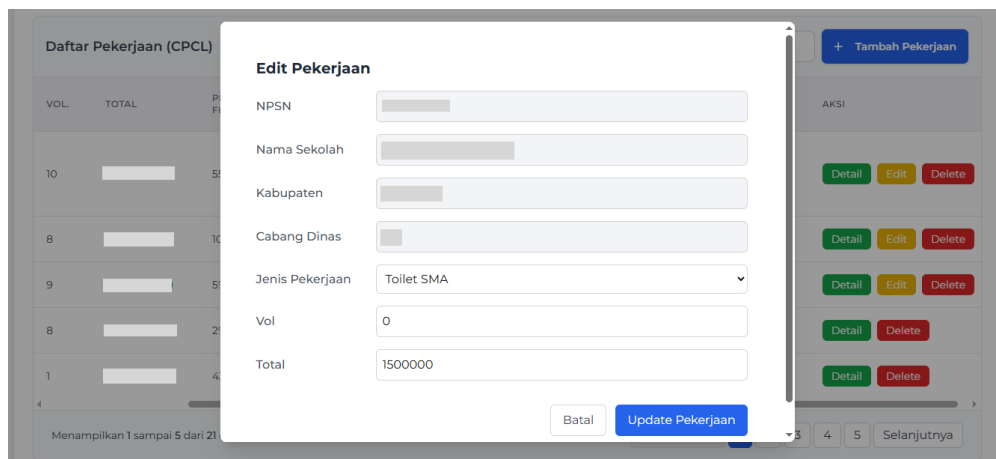
Pada pengujian modul kelola pekerjaan *user* KCD, ditemukan bahwa notifikasi peringatan atau kesalahan pada form tambah dan edit pekerjaan ditampilkan di belakang form sehingga tidak dapat langsung dilihat oleh pengguna, seperti yang ditunjukkan oleh Gambar 11-14. Notifikasi seharusnya muncul di atas form agar pesan validasi dapat segera diketahui. *Bug* ini dikategorikan sebagai *minor severity* karena hanya mempengaruhi aspek antarmuka dan pengalaman pengguna tanpa mengganggu fungsi utama sistem.



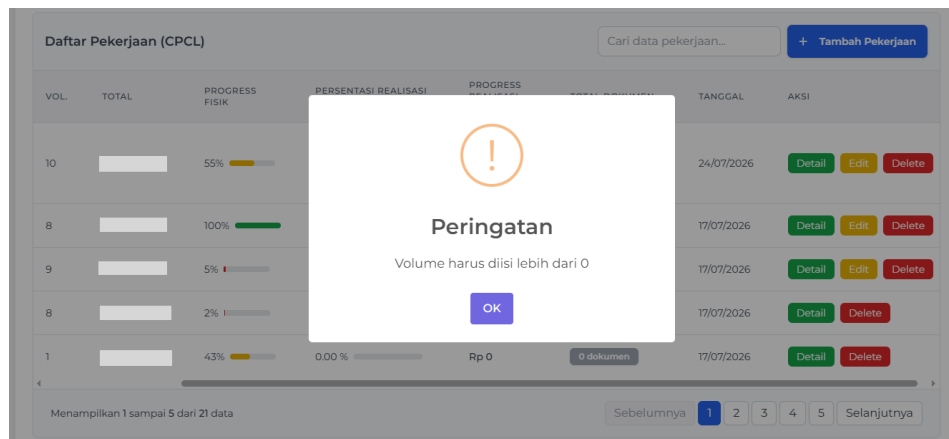
Gambar 11. Kondisi Saat Form Tambah Pekerjaan Masih Terbuka



Gambar 12. Kondisi Saat Form Tambah Pekerjaan Ditutup



Gambar 13. Kondisi Saat Form Edit Pekerjaan Masih Terbuka



Gambar 14. Kondisi Saat Form Edit Pekerjaan Ditutup

3.3 Klasifikasi Bug dan Rekomendasi Perbaikan

Setelah seluruh *bug* diidentifikasi, dilakukan klasifikasi berdasarkan tingkat *severity* untuk menentukan prioritas perbaikan. Pengelompokan dilakukan mengacu pada kriteria yang telah ditetapkan pada Tabel 3. Klasifikasi *Severity*. Hasil pengklasifikasian *bug* disajikan pada Tabel 6. Klasifikasi *Bug* dan rekomendasi perbaikan disajikan pada Tabel 7. Rekomendasi Perbaikan :

Tabel 6. Klasifikasi Bug

Severity	Jumlah Bug
Blocker	0
Critical	0
Major	4
Minor	4
Trivial	0

Tabel 7. Rekomendasi Perbaikan

ID Bug	Modul	Severity-Prioritas	Prioritas
BG-PR-01	Realisasi	Major-Tinggi	Memperbaiki mekanisme sinkronisasi data setelah proses penghapusan realisasi dengan memperbarui nilai persentase dan <i>progress bar</i> pada modul realisasi maupun dashboard secara otomatis berdasarkan data terbaru
BG-PR-02	Progres Realisasi	Major-Tinggi	Menambahkan validasi input pada sisi klien (<i>client-side</i>) dan sisi server (<i>server-side</i>) agar kolom progres realisasi keuangan hanya menerima nilai numerik non-negatif (≥ 0) sebelum data disimpan ke <i>database</i>
BG-PR-03	Progres Realisasi	Major-Tinggi	Menambahkan validasi batas atas sehingga nilai progres realisasi keuangan tidak dapat melebihi 100% atau melebihi total anggaran keuangan sebelum proses penyimpanan dilakukan
BG-DS-04	Dashboard	Major-Tinggi	Menambahkan validasi input pada sisi klien (<i>client-side</i>) dan sisi server (<i>server-side</i>) agar kolom jumlah siswa dan jumlah rombel hanya menerima nilai numerik non-negatif sebelum data disimpan
BG-KP-05, BG-KP-07	Kelola Pekerjaan	Minor-Sedang	Memperbaiki mekanisme penampilan pesan validasi pada komponen <i>dropdown</i> di form tambah dan edit pekerjaan agar ditampilkan secara langsung dan mudah terlihat oleh pengguna ketika belum memilih data yang diperlukan
BG-KP-06, BG-KP-08	Kelola Pekerjaan	Minor-Sedang	Memperbaiki mekanisme penampilan pesan validasi pada kolom input di form tambah dan edit pekerjaan agar pesan "nilai harus

lebih dari 0" ditampilkan secara langsung ketika pengguna memasukkan nilai yang tidak valid

Hasil penelitian menunjukkan bahwa penerapan *automated black box testing* menggunakan Katalon Studio mampu mengotomatisasi proses pengujian pada modul-modul sistem SiPantau serta mengidentifikasi *bug* yang belum ditemukan pada proses pengembangan sebelumnya. Sebanyak 153 *test case* berhasil diimplementasikan dan dieksekusi pada 7 modul sistem. Dari seluruh *test case* tersebut, 145 *test case* memperoleh status *pass*, sedangkan 8 *test case* memperoleh status *fail*. Hasil tersebut menunjukkan bahwa sebagian besar fungsi sistem yang menjadi objek pengujian telah berjalan sesuai dengan kebutuhan fungsional, namun masih ditemukan beberapa ketidaksesuaian yang berkaitan dengan validasi input, sinkronisasi data antar modul, dan tampilan antarmuka. Penerapan kombinasi teknik *Equivalence Partitioning* (EP), *Boundary Value Analysis* (BVA), dan *Decision Table* (DT) memungkinkan pengujian dilakukan terhadap beberapa karakteristik yang berbeda, yaitu menguji validitas masukan, batas nilai, dan kombinasi aturan bisnis pada setiap modul/fungsi yang diuji. *Bug* yang ditemukan kemudian diklasifikasikan berdasarkan tingkat *severity* sehingga dapat digunakan sebagai dasar dalam menentukan prioritas perbaikan. Dengan demikian, penerapan *automated black box testing* tidak hanya menghasilkan temuan *bug*, tetapi juga menyediakan informasi yang mendukung proses evaluasi dan pemeliharaan sistem di masa mendatang.

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, penerapan *automated black box testing* menggunakan Katalon Studio pada Sistem SiPantau berhasil mengotomatisasi proses pengujian fungsional pada 7 modul sistem, yaitu Login, Dashboard, Realisasi, Progres Realisasi, Pemantauan Laporan, Kelola Pekerjaan, dan Logout. Sebanyak 153 *test case* berhasil dirancang dan dieksekusi menggunakan kombinasi teknik *Equivalence Partitioning* (EP), *Boundary Value Analysis* (BVA), dan *Decision Table* (DT), dengan 145 *test case* berstatus *pass* dan 8 *test case* berstatus *fail*. Hasil analisis terhadap *test case* yang gagal mengidentifikasi 8 *bug* yang berkaitan dengan validasi input, sinkronisasi data antar modul, dan tampilan antarmuka. Berdasarkan klasifikasi *severity*, ditemukan 4 *bug* dengan tingkat *major* dan 4 *bug* dengan tingkat *minor*, sedangkan *bug* dengan tingkat *blocker*, *critical*, dan *trivial* tidak ditemukan. Temuan *bug* dan hasil klasifikasi *severity* tersebut digunakan sebagai dasar dalam penyusunan rekomendasi perbaikan Sistem SiPantau.

REFERENCES

- [1] M. N. Abduh *et al.*, *Mitigasi Bencana*. Makassar: Tohar Media, 2023.
- [2] I. Salman, B. Turhan, R. Ramac, and V. Mandic, "Confirmation Bias and Time Pressure: A Family of Experiments in Software Testing," *IEEE Trans. Softw. Eng.*, vol. 49, no. 12, pp. 5203–5222, 2023.
- [3] M. Andryan, W. Saputra, W. A. Rioditama, and H. Setyowati, "Survei Teknik-Teknik Pengukuran Kualitas Perangkat Lunak," *J. Ilm. Komput. dan Inf.*, vol. 3, no. 1, pp. 2715–2731, 2021.
- [4] W. Min, W. Zhichao, Z. Yan, X. Yu, E. W. Kenali, and K. Amaliah, *Software Testing*. Jakarta: Penerbit Utama, 2026.
- [5] W. Behutiye, P. Karhapää, X. Franch, P. Rodríguez, and M. Oivo, "Quality Measurement in Agile and Rapid Software Development: A Systematic Mapping," *J. Syst. Softw.*, vol. 186, p. 111187, 2022, doi: 10.1016/j.jss.2021.111187.
- [6] R. A. Sianturi *et al.*, "Perancangan Pengujian Fungsional dan Non Fungsional Aplikasi," *J. Ilmu Komput.*, vol. 9, no. 2, pp. 133–141, 2021, doi: 10.35508/jicon.v9i2.4706.
- [7] B. Hardika *et al.*, "Pengujian Blackbox Testing Website Garuda Farm Menggunakan Teknik Equivalence Partitioning," *J. Teknol. Inf.*, vol. 06, no. 02, pp. 740–753, 2024.
- [8] Y. Salim, "Penerapan Metode Black Box Sebagai Pengukuran Kinerja Aplikasi Surat Perintah Perjalanan Dinas," *Ganesha J.*, vol. 30, no. 1, 2024, doi: 10.36309/goi.v30i1.261.
- [9] A. I. Zidny, Nazilah Marzukoh, and C. Anwar, "Aplikasi Sistem Pencatatan Pengaduan Masyarakat Berbasis Standar ISO/IEC 29119:2022," *J. Rekayasa Perangkat Lunak*, vol. 1, no. 4, pp. 137–142, 2025.
- [10] A. Maemunah, S. Al-Bawazir, and C. Anwar, "Penerapan ISO/IEC 29119-3 dalam Analisis dan Pengujian Sistem Informasi Perpustakaan Berbasis CodeIgniter 3 (Studi Kasus : Perpustakaan Goldsop)," *J. Penguji. Sist.*, vol. 1, no. 3, pp. 55–66, 2025.
- [11] S. D. Pratama, Lasimin, and M. N. Dadaprawira, "Pengujian Black Box Testing Pada Aplikasi Edu Digital Berbasis Website Menggunakan Metode Equivalence Dan Boundary Value," *J. Teknol. Sist. Inf. dan Sist. Komput. TGD*, vol. 6, pp. 560–569, 2023.
- [12] R. Santika, S. Febrianti, S. Hakeki, Y. Supriyanto, and Z. Fajar, "Analisis Kinerja Fungsional pada Aplikasi Mobile JKN Melalui Pengujian Black Box Testing," *J. Sist. Inf.*, vol. 3, no. 1, pp. 59–68, 2025.
- [13] F. J. Putri, D. Febriyanti, D. Wulansari, K. Ramadhani, A. Wicaksono, and G. P. Mindara, "Analisis Penerapan Black Box Testing Berbasis Decision Table," *J. Rekayasa Komput.*, vol. 10, no. 1, pp. 585–591, 2026.
- [14] N. Made, D. Febriyanti, A. A. K. Oka, and I. N. Piarsa, "Implementasi Black Box Testing pada Sistem Informasi Manajemen Dosen," *J. Ilm. Teknol. Inf.*, vol. 2, no. 3, 2021.
- [15] D. Adisty, Y. Putri, N. C. Wibowo, P. Karunia, and F. Ananto, "Implementasi Pengujian Blackbox Menggunakan Automation," *J. Teknol. Inf. dan Komput.*, vol. 9, no. 3, pp. 4278–4285, 2025.



- [16] M. D. Andika *et al.*, “Implementasi Black Box Testing Dalam Pengujian Fungsionalitas Website Leo Gym,” *J. Apl. Komput.*, vol. 1, no. 2, pp. 55–62, 2025.
- [17] 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.
- [18] M. Haidar, A. Mufid, M. A. Zidane, S. A. Putri, R. Wirnanti, and A. Ibrahim, “Pengujian Website Menggunakan Selenium IDE pada JDIH Pesawaran Menggunakan Metode Equivalence Partitioning,” *J. Inform.*, vol. 9, no. 4, pp. 6055–6061, 2025.
- [19] A. P. Putra and I. Nuryasin, “Pengujian Sistem Informasi Monitoring Dava Kebab Menggunakan White Box Testing dengan Teknik Basis Path,” *INFOTECH J. Inform. Teknol.*, vol. 5(1), pp. 63–75, 2024, doi: 10.37373/infotech.v5i1.1147.
- [20] Y. Cahyono, A. Voutama, and N. Heryana, “Evaluasi Stabilitas Performa Sistem Expired Tracker,” *J. Rekayasa Sist.*, vol. 10, no. 3, pp. 3961–3966, 2026.
- [21] Y. Tian, D. Lo, and C. Sun, “Information Retrieval Based Nearest Neighbor Classification for Fine-Grained Bug Severity Prediction,” in *2012 19th Working Conference on Reverse Engineering*, 2012, pp. 215–224. doi: 10.1109/WCRE.2012.31.