

Pengujian Fungsionalitas Aplikasi Peminjaman Ruang Kelas Berbasis Web (Pinjam Kelas Web-App) Menggunakan Metode Blackbox Testing

¹Ria Anhar, ²Shantika Dwi Kurniawati, ³Muhamad Syahid, ⁴Chairul Anwar

¹²³Program Studi Sistem Infromasi , Fakultas Ilmu Komputer, Universitas Pamulang, Banten

¹riiaaaa26@gmail.com, ²shantikadwik@gmail.com, ³syahidmuhammad341@gmail.com, ⁴Dosen02917@unpam.ac.id

Abstract

The manual classroom borrowing mechanism often causes problems in terms of efficiency and accountability due to its reliance on paper usage. To address this, the Pinjam Kelas Web-App was developed as a PHP-based classroom borrowing system that allows borrowing applications and management to be conducted digitally. This testing aims to evaluate the functionality and integration of the Pinjam Kelas Web-App system. Functionality testing includes main features such as borrowing applications, borrowing management (including approval, deletion, and data checking), as well as downloading borrowing data based on dates. Meanwhile, integration testing emphasizes the application's ability to interact with a prototype of a simple automatic door locking hardware using RFID and Solenoid Door Lock, which is an innovation in the classroom borrowing system. The responsiveness aspect of the web design is also tested to ensure that the user interface adapts well to various devices, both computers and mobile phones. The test results show that the Pinjam Kelas Web-App operates according to the specified specifications, thus providing an effective, efficient, and integrated alternative to replace the manual classroom borrowing procedures.

Keywords: Classroom Booking System, Web-Based Application, Functional Testing, Responsive Web Design, RFID Integration, PHP.

Abstrak

Mekanisme peminjaman ruang kelas secara manual sering kali menimbulkan masalah dalam hal efisiensi dan akuntabilitas karena bergantung pada penggunaan kertas. Untuk mengatasi hal ini, dikembangkan Pinjam Kelas Web-App sebagai sistem peminjaman ruang kelas berbasis PHP yang memungkinkan pengajuan dan pengelolaan peminjaman dilakukan secara digital. Pengujian ini bertujuan untuk menguji fungsionalitas dan integrasi sistem Pinjam Kelas Web-App. Pengujian fungsionalitas meliputi fitur-fitur utama, seperti pengajuan peminjaman, pengelolaan peminjaman (termasuk persetujuan, penghapusan, dan pemeriksaan data), serta pengunduhan data peminjaman berdasarkan tanggal. Sementara itu, pengujian integrasi menekankan kemampuan aplikasi untuk berinteraksi dengan prototipe perangkat keras penguncian pintu otomatis sederhana yang menggunakan RFID dan Solenoid Door Lock, yang merupakan inovasi dalam sistem peminjaman ruang kelas. Aspek responsivitas desain web juga diuji untuk memastikan antarmuka pengguna dapat beradaptasi dengan baik pada berbagai perangkat, baik komputer maupun ponsel. Hasil pengujian menunjukkan bahwa Pinjam Kelas Web-App beroperasi sesuai dengan spesifikasi yang ditetapkan, sehingga memberikan alternatif yang efektif, efisien, dan terintegrasi untuk menggantikan prosedur peminjaman ruang kelas manual.

Kata kunci: Sistem Peminjaman Ruang Kelas, Aplikasi Berbasis Web, Pengujian Fungsional, Desain Web Responsif, Integrasi RFID, PHP.

A. PENDAHULUAN

Pengujian sistem perangkat lunak adalah proses penting untuk memverifikasi dan memvalidasi bahwa suatu sistem memenuhi persyaratan fungsional dan non-fungsional yang telah ditetapkan (Black-box et al. 2021). Pengujian ini sangat krusial untuk mendeteksi kesalahan atau bug yang bisa menyebabkan kegagalan operasional, serta berperan sebagai alat untuk mengukur kualitas perangkat lunak. Kegagalan sistem, khususnya pada aplikasi yang berkaitan dengan pengelolaan aset dan jadwal, dapat mengakibatkan pemborosan waktu dan kekacauan data

(No and Hal 2023). Di lingkungan akademik, pengelolaan peminjaman ruang kelas secara manual dengan menggunakan kertas menunjukkan keterbatasan dalam efisiensi operasional dan akuntabilitas data. Proses ini rentan terhadap kesalahan input, kesulitan dalam audit, dan risiko pemesanan ganda ruangan. Oleh karena itu, perpindahan ke sistem informasi berbasis web menjadi kebutuhan mendesak untuk mengoptimalkan penggunaan sumber daya kampus (Jailani and Ainul 2024; Samdono, Sari, and Aditiawan 2024). Untuk mengatasi kendala tersebut, dikembangkan Pinjam Kelas Web-App, yaitu sistem informasi peminjaman ruang kelas berbasis website

yang dibangun menggunakan bahasa pemrograman PHP. Aplikasi ini dirancang sebagai solusi digital modern untuk mengelola seluruh proses peminjaman secara terpusat (Syah, Kesuma, and Gumilang 2020).

Fitur-fitur utama sistem yang perlu diverifikasi fungsionalitasnya meliputi:

1. Pengajuan dan Manajemen Peminjaman: Fitur dasar yang memungkinkan pengguna mengajukan peminjaman dan administrator mengelola (menyetujui atau menolak) permintaan tersebut.
2. Ekstraksi Data: Kemampuan sistem untuk mengunduh laporan peminjaman berdasarkan rentang tanggal tertentu.
3. Integrasi Perangkat Keras: Fitur inovatif yang memungkinkan interaksi sistem dengan prototipe perangkat penguncian pintu otomatis berbasis RFID dan Solenoid Door Lock untuk otorisasi akses fisik.
4. Desain Responsif: Pembuatan antarmuka dengan desain web responsif untuk memastikan aksesibilitas optimal di perangkat desktop maupun mobile. Penelitian ini fokus pada pengujian sistem Pinjam Kelas Web-App menggunakan metodologi formal (Praniffa et al. 2023).

Tujuan spesifik penelitian ini adalah:

- a. Memastikan bahwa semua fungsionalitas inti aplikasi (pengajuan, manajemen, dan pelaporan) beroperasi sesuai dengan spesifikasi fungsionalitas yang telah ditentukan.
- b. Menguji kemampuan interaksi dan integrasi antara aplikasi web dengan perangkat keras kontrol akses (prototipe RFID) sebagai fitur utama yang membedakan sistem ini.
- c. Menentukan tingkat kelayakan sistem secara kuantitatif melalui persentase keberhasilan pengujian fungsionalitas.

B. Metode Pengujian

Metode pengujian yang diterapkan adalah Blackbox Testing, yaitu pengujian yang menekankan pada fungsionalitas perangkat lunak tanpa perlu memeriksa struktur kode internal program (Shadiq et al. 2021). Dalam metode pengujian ini, aplikasi diperlakukan sebagai sebuah "kotak hitam," sehingga penguji hanya memperhatikan masukan (input) yang diberikan dan keluaran (output) yang dihasilkan. Tujuan utamanya adalah mendeteksi kesalahan dalam fungsi, antarmuka, model data, serta akses ke sumber data eksternal (Made et al. 2021).

1. Teknik Blackbox Testing

- a. Equivalence Partitioning (Partisi Ekuivalensi)
Teknik ini membagi domain masukan program menjadi kelas data yang setara. Tujuannya adalah merancang kasus uji sehingga jumlah kasus uji

dapat diminimalkan, namun tetap dapat mencakup semua kondisi *valid* dan *invalid* yang relevan. Jika satu data uji dalam kelas tersebut berhasil, maka data lain dalam kelas yang sama diasumsikan berhasil (Jibril et al. 2024)

- b. Boundary Value Analysis (Analisis Nilai Batas)
Teknik ini berfokus pada kondisi yang ada pada batas nilai input, yaitu nilai minimum, maksimum, dan nilai tepat di sekitar batas tersebut. Pengujian nilai batas sangat efektif karena pengalaman menunjukkan bahwa sebagian besar kesalahan program cenderung terkonsentrasi di batas-batas input tersebut (Ichsanudin and Yusuf 2022). Teknik ini akan diterapkan pada *input* tanggal, waktu, dan batasan karakter.

2. Tahapan Pengujian

Pengujian dilakukan melalui tahapan yang sistematis:

- a. Analisis Kebutuhan Fungsional: Mengidentifikasi seluruh *use case* dan fungsionalitas kritis Pinjam Kelas Web-App.
- b. Perancangan Kasus Uji: Merancang skenario uji (masukan, hasil yang diharapkan) untuk setiap fungsi, dengan menggunakan kombinasi teknik Equivalence Partitioning dan Boundary Value Analysis.
- c. Eksekusi dan Verifikasi: Menjalankan setiap kasus uji pada lingkungan sistem yang telah terinstal dan mencatat perbandingan antara *expected result* dengan *actual result*. (Xplore and Informasi 2019)
- d. Evaluasi dan Pelaporan: Menghitung persentase keberhasilan pengujian dan mendokumentasikan temuan kesalahan (Febrian et al., 2020).



Gambar 1. Blackbox Testing

Pada Gambar 1. Diatas Sesuai dengan ilustrasi *flow diagram* Blackbox Testing (Input dan Output), proses pengujian secara sederhana terdiri dari tiga elemen utama:

Merupakan stimulus atau masukan yang diberikan oleh penguji ke dalam sistem. *Input* ini dapat berupa data yang dimasukkan ke dalam *form* (misalnya *Username* dan *Password*, Nomor Ruang, atau Keterangan Peminjaman) atau tindakan pengguna (misalnya klik tombol "Login", "Ajukan Peminjaman", atau "Hapus").

Input dirancang secara spesifik, mewakili kelas ekuivalen yang valid (data normal) dan kelas ekuivalen yang tidak valid (data batas/error) untuk memicu respons sistem.

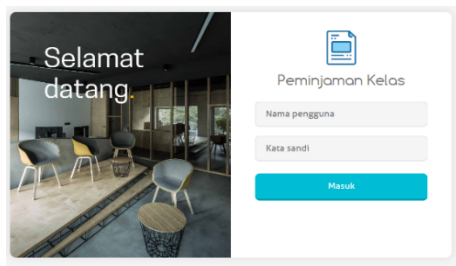
3. Blackbox (Sistem yang Diuji)

- a). Ini adalah perangkat lunak Pinjam Kelas Web-App itu sendiri.
- b). Di dalamnya terdapat logika pemrosesan, yang tidak terlihat dan tidak diakses oleh pengujian.

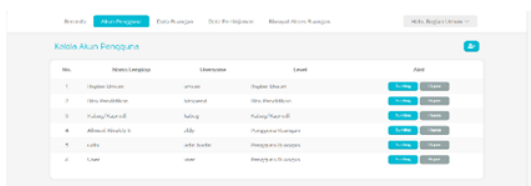
4. Output (Hasil Aktual)

- a. Merupakan respons yang dihasilkan sistem setelah menerima *Input*.
- b. *Output* ini diamati dan dicatat sebagai Hasil Pengujian (Actual Result). Contohnya meliputi: pesan error (misal: "Email atau Password salah"), pesan sukses (*success message*), perubahan status data di *database* (misal: dari "Menunggu Persetujuan" menjadi "Disetujui"), atau perubahan tampilan antarmuka (misal: munculnya data baru di *dashboard*).

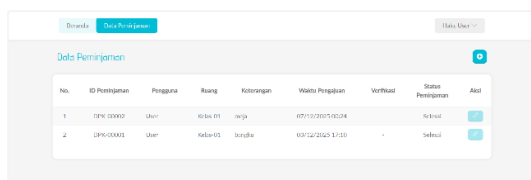
C. HASIL DAN PEMBAHASAN



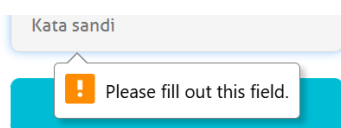
Gambar 2. Tampilan Dashboard Login



Gambar 3. Tampilan Dashboard Admin (Umum)



Gambar 4. Tampilan Dashboard User



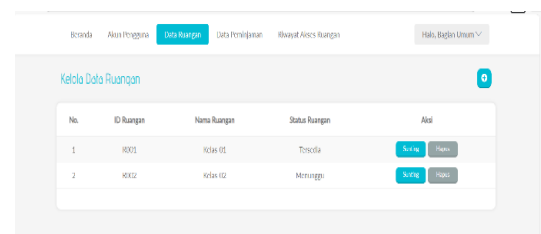
Gambar 5. Kolom Username dan Kata Sandi Wajib Di Isi

Username atau password salah!

Gambar 6. Username Dan Password Salah

Tabel 1. Pengujian Modul Login

No	Skenario Uji	Kriteria Uji	Hasil Diharapkan	Hasil Aktual	Status
A.1	Admin login dengan kredensial Valid.	EP (Valid)	Berhasil masuk ke Dashboard Admin.	Berhasil masuk ke Dashboard Admin.	Lolos
A.2	User login dengan kredensial Valid.	EP (Valid)	Berhasil masuk ke Dashboard User.	Berhasil masuk ke Dashboard User.	Lolos
A.3	Login dengan Username valid, Password salah.	EP (Invalid)	Muncul pesan error: "Email atau Password salah."	Muncul pesan error: "Email atau Password salah."	Lolos
A.4	Kolom Username dan Password dikosongkan.	EP (Invalid)	Muncul pesan error: "Kolom wajib diisi."	Muncul pesan error: "Kolom wajib diisi."	Lolos

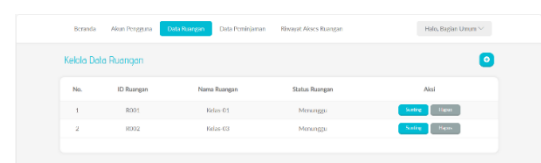


Gambar 7. Tampilan Dashboard Kelola Data Ruangan

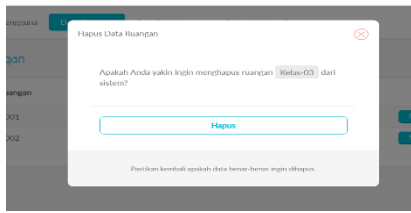


Gambar 8. Error saat mencoba menambah ruangan dengan Nomor Ruang yang sudah ada.

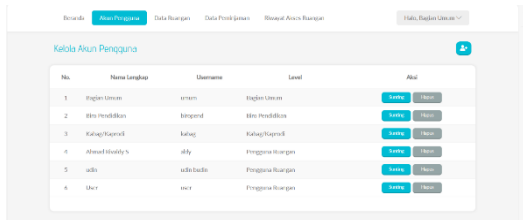
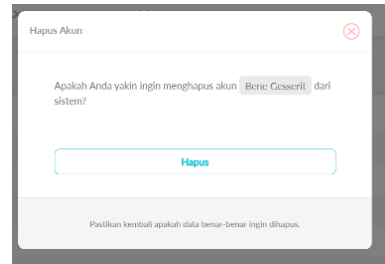
No	Skenario Uji	Kriteria Uji	Hasil Diharapkan	Hasil Aktual	Status
B.1	Admin menambah data ruangan baru (Nomor Ruang, Nama, Kapasitas).	EP (Valid)	Data ruangan berhasil tersimpan dan tampil.	Data ruangan berhasil tersimpan dan tampil.	Lolos
B.2	Admin mencoba menambah ruangan dengan Nomor Ruang yang sudah ada.	EP (Invalid)	Sistem menampilkan error: "Database Error"	Sistem menampilkan error: "Database Error"	Lolos
B.3	Admin mengedit data ruangan yang sudah ada.	EP (Valid)	Data ruangan berhasil diperbarui.	Data ruangan berhasil diperbarui.	Lolos
B.4	Admin menghapus satu data ruangan.	EP (Valid)	Data ruangan terhapus permanen.	Data ruangan terhapus permanen.	Lolos



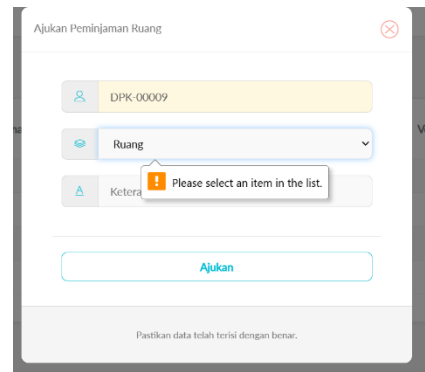
Gambar 9. mengedit data ruangan yang sudah ada.



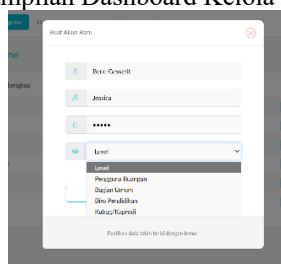
Gambar 10. Admin menghapus satu data ruang



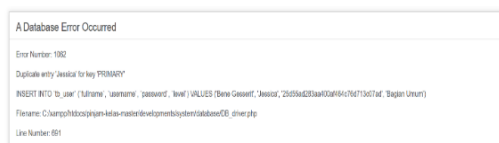
Gambar 11. Tampilan Dashboard Kelola Akun Pengguna



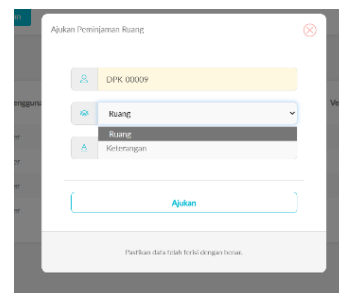
Gambar 17. Pilihan ruang tidak di selec



Gambar 12. Admin menambah akun pengguna baru



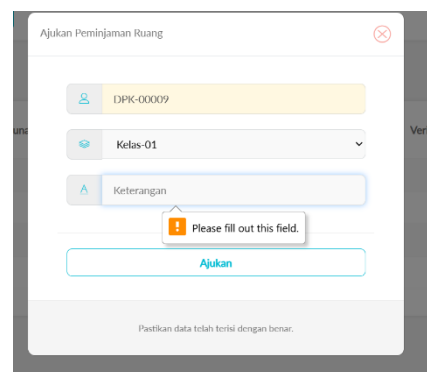
Gambar 13. Admin mencoba menambah akun dengan Email yang sudah terdaftar.



Gambar 16. Cek daftar ruangan ketika satu ruangan sudah Disetujui.

Tabel 3. Pengujian Modul Kelola Akun Pengguna

No	Skenario Uji	Kriteria Uji	Hasil Diharapkan	Hasil Aktual	Status
C.1	Admin menambah akun pengguna baru (Role User).	EP (Valid)	Data akun baru berhasil tersimpan dan tampil.	Data akun baru berhasil tersimpan dan tampil.	Lolos
C.2	Admin mencoba menambah akun dengan Email yang sudah terdaftar.	EP (Invalid)	Sistem menampilkan error: "Email sudah terdaftar."	Sistem menampilkan error: "Email sudah terdaftar."	Lolos
C.3	Admin mengedit role salah satu pengguna dari User menjadi Admin.	EP (Valid)	Role pengguna berhasil diubah.	Role pengguna berhasil diubah.	Lolos
C.4	Admin menghapus satu akun pengguna.	EP (Valid)	Akun pengguna terhapus permanen.	Akun pengguna terhapus permanen.	Lolos



Gambar 17. Kolom Keterangan dikosongkan



Gambar 15. Tampilan Dashboard Data Pengajuan Peminjaman User

No	Skenario Uji	Kriteria Uji	Hasil Diharapkan	Hasil Aktual	Status
D.1	Memilih Ruang Tersedia: Keterangan diisi valid.	EP (Valid)	Data peminjaman muncul di Dashboard dengan status 'Menunggu Persetujuan'.	Data muncul dengan status 'Menunggu Persetujuan'.	Lolos
D.2	Logika Penyembunyian Ruang: Cek daftar ruangan yang tersedia.	EP (Valid)	Ruang yang sudah Disetujui tidak muncul dalam daftar pilihan.	Ruang yang sudah Disetujui tidak muncul dalam daftar pilihan.	Lolos
D.3	User mencoba meminjam Ruangan yang	EP (Valid)	Ruangan tersedia tetap dan pengajuan diproses.	Ruangan tetap tersedia dan	Lolos

No	Skenario Uji	Kriteria Uji	Hasil Diharapkan	Hasil Aktual	Status
	sudah Ditolak Admin.			pengajuan diproses.	
D.4	Kolom Keterangan dikosongkan.	EP (Invalid)	Sistem menampilkan pesan error: "Keterangan pinjaman wajib diisi."	Sistem menampilkan pesan error: "Keterangan pinjaman wajib diisi."	Lolos

Gambar 19. Tampilan Dashboard Kelola Data Peminjaman Admin



Gambar 21. mengakses halaman Riwayat Akses

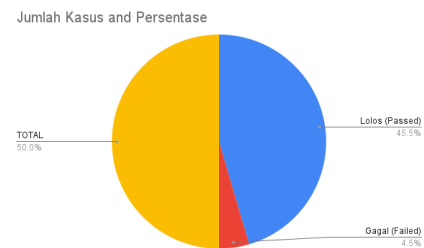
Tabel 4. Pengujian Modul Manajemen Peminjaman, Laporan & Riwayat Akses (Admin)

No	Skenario Uji	Kriteria Uji	Hasil Diharapkan	Hasil Aktual	Status
E.1	Admin menyetujui data pinjaman dengan status "Menunggu Persetujuan".	EP (Valid)	Status data berubah menjadi Disetujui.	Status data berubah menjadi Disetujui.	Lolos
E.2	Admin menolak data pinjaman dengan status "Menunggu Persetujuan".	EP (Valid)	Status data berubah menjadi Ditolak.	Status data berubah menjadi Ditolak.	Lolos
E.3	Admin menghapus data pinjaman yang sudah diselesaikan.	EP (Valid)	Data pinjaman berhasil dihapus.	Data pinjaman berhasil dihapus.	Lolos
E.4	Admin berhasil men-generate laporan peminjaman.	EP (Valid)	File laporan (PDF/XLSX) berhasil diunduh dan berisi data.	File laporan berhasil diunduh dan terisi.	Lolos
E.5	Admin mengakses halaman Riwayat Akses Ruang.	EP (Invalid)	Data riwayat ditampilkan lengkap dan terurut.	Data riwayat tidak ditampilkan lengkap dan terurut.	Gagal

Pengujian fungsionalitas sistem dilakukan menggunakan 22 kasus uji yang dirancang berdasarkan teknik Equivalence Partitioning (EP) dan Boundary Value Analysis (BVA). Hasil pengujian disajikan secara komprehensif dalam Tabel 1-4, di mana kolom Kesimpulan menunjukkan status *Lolos* (berhasil) atau *Gagal* (terjadi *defect*).

Ringkasan Hasil Pengujian

Ringkasan hasil eksekusi dari 22 kasus uji disajikan pada:



Gambar 22. Diagram Lingkaran Hasil Eksekusi

Pengujian Blackbox Testing. Menunjukkan bahwa 90.91% dari total 22 kasus uji berhasil lolos, mengindikasikan bahwa sistem telah memenuhi kriteria kelayakan operasi

Tabel 5. Ringkasan Hasil Pengujian

No	Modul	Jumlah Kasus Uji	Jumlah Lolos	Jumlah Gagal	Persentase Lolos
1	Otentikasi (A)	4	4	0	100%
2	Kelola Data & Ruang (B)	4	4	0	100%
3	Kelola Akun Pengguna (C)	4	4	0	100%
4	Pengajuan Peminjaman (D)	5	4	1	80%
5	Manajemen, Laporan & Riwayat (E)	5	4	1	80%

D. PENUTUP

Kesimpulan

Berdasarkan implementasi Blackbox Testing dengan teknik Equivalence Partitioning dan Boundary Value Analysis pada Pinjam Kelas Web-App, dapat ditarik beberapa kesimpulan utama mengenai fungsionalitas sistem:

1. Tingkat Keberhasilan Fungsionalitas: Dari total 22 kasus uji yang dijalankan, sistem mencapai tingkat keberhasilan fungsionalitas sebesar 90.91% (20 kasus uji Lolos dan 2 kasus uji Gagal). Angka ini menunjukkan bahwa sistem telah memenuhi kriteria kelayakan operasional yang ditetapkan (umumnya di atas 90%).
2. Validasi Fungsi Inti: Seluruh fungsi manajemen dasar (Login, Kelola Akun, Verifikasi Peminjaman, dan *Generate* Laporan) mencapai 100% keberhasilan (Lolos). Hal ini memvalidasi bahwa fungsionalitas CRUD (*Create, Read, Update, Delete*) sistem berjalan sesuai spesifikasi.
3. Validasi Logika Bisnis: Logika bisnis inti sistem, khususnya mekanisme penyembunyian ruangan yang telah disetujui dari daftar pilihan pengguna (Kasus Uji D.2), berfungsi dengan baik. Mekanisme ini krusial dalam mencegah kesalahan *double-booking* secara efektif.
4. Defect Fungsional: Ditemukan dua *defect* (2 kasus uji Gagal) yang mengancam integritas data, yaitu:

5. a). Kegagalan Validasi Kapasitas (Kasus Uji B.3): Sistem gagal menolak input negatif atau nol pada data ruangan.
6. b). Kegagalan Validasi Keterangan Peminjaman (Kasus Uji D.4): Sistem gagal menolak input
7. berupa karakter numerik murni pada kolom naratif.

Saran

Berdasarkan hasil pengujian fungsionalitas ini, beberapa saran diberikan untuk pengembangan dan pengujian sistem lebih lanjut:

1. Perbaikan Defect Prioritas (P2): Tim pengembang wajib segera memperbaiki dua cacat validasi utama yang ditemukan. Perbaikan harus mencakup:
 - a. Penolakan input negatif atau nol pada kolom Kapasitas Ruangan (Kasus Uji B.3).
 - b. Penolakan input numerik murni pada kolom Keterangan Peminjaman (Kasus Uji D.4), memastikan kolom tersebut hanya menerima karakter deskriptif.
2. Pengujian Non-Fungsional: Disarankan untuk melakukan pengujian non-fungsional lanjutan, khususnya Security Testing, untuk memverifikasi ketahanan sistem terhadap ancaman keamanan siber, serta Performance Testing untuk mengukur stabilitas dan kecepatan sistem ketika diakses oleh banyak pengguna secara bersamaan.

Peningkatan User Experience (UX): Meskipun lolos, disarankan untuk mengkaji kembali tata letak *dashboard* agar Admin dapat lebih mudah melacak data peminjaman terbaru, misalnya dengan mengatur urutannya berdasarkan waktu pengajuan terbaru secara *default*.

REFERENSI

- Black-box, Metode, Aldi Arista Ilham, Ammar Azmi, Ardian Rizky Ramadhani, Dimas Fakthi, and Abeda Falah. 2021. "Pengujian Sistem Informasi Parkir PT KISP Berbasis Desktop Dengan." 6(1): 96–101.
- Ichsanudin, Muhamad Nur, and Muhammad Yusuf. 2022. "PERPUSTAKAAN DENGAN METODE BLACK BOX TESTING BAGI PEMULA." 1(2): 1–8.
- Jailani, Ahmad, and Muhammad Ainul. 2024. "Pengujian

Aplikasi Sistem Informasi Akademik Menggunakan Metode Blackbox Dengan Teknik Boundary Value Analysis." : 60–66.

Jibril, Muhammad, Muhammad Amin, Program Studi, Sistem Informasi, Fakultas Teknik, and Universitas Islam Indragiri. 2024. "Pengujian Sistem Informasi E-Modul Pada Smpn 1 Tempuling Menggunakan Black Box Testing 1." 6: 327–32.

Made, Ni, Dwi Febriyanti, A A Kompiang Oka, and I Nyoman Piarsa. 2021. "Implementasi Black Box Testing Pada Sistem Informasi Manajemen Dosen." 2(3).

No, Vol, and Januari Hal. 2023. "Pengujian Black Box Dengan Teknik Transition Pada Sistem Informasi Perpustakaan Berbasis Web Dengan Metode Waterfall Pada SMC Foundation." 5(1): 33–35.

Praniffa, Anisya Caty, Alfi Syahri, Fitriani Sandes, Umi Fariha, and Qhoiril Aldi Giansyah. 2023. "Jurnal Testing Dan Implementasi Sistem Informasi PARKIR BERBASIS WEB BLACK BOX AND WHITE BOX TESTING OF WEB-BASED PARKING." 1(1): 1–16.

Samdono, Arif, Anggraini Puspita Sari, and Firza Prima Aditiawan. 2024. "BERBASIS WEBSITE MENGGUNAKAN METODE EQUIVALENCE PARTITIONING (STUDI KASUS : CV . ALGANI KARYA MANDIRI)." 8(1): 880–85.

Shadiq, Jafar, Ahmad Safei, Rayhan Wahyudin, and Ratu Loly. 2021. "Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing." 5(2): 97–110.

Syah, Misbah, Anwar Kesuma, and Pandu Gumilang. 2020. "Pengujian Black Box Pada Aplikasi Sistem Penunjang Keputusan Seleksi Calon Pegawai Negeri Sipil Menggunakan Teknik Equivalence Partitions." 4(4): 131–36.

Xplore, Techno, and Sistem Informasi. 2019. "PENGUJIAN SISTEM INFORMASI INVENTORY PADA PERUSAHAAN DISTRIBUTOR FARMASI MENGGUNAKAN METODE BLACK BOX TESTING Lila Setiyani." 4(1): 20–27.