

Testing Aplikasi SIAKAD di Perguruan Tinggi XYZ dengan Pendekatan *Exploratory Testing* dan *Black Box Testing*

Joko Purwanto^{1,*}, Adlan Nugroho², Muhammad Abdul Muin³

¹ *D4 Teknologi Rekayasa Perangkat Lunak, Politeknik Negeri Cilacap*
E-mail: jokopurwanto@pnc.ac.id

² *D4 Teknologi Rekayasa Perangkat Lunak, Politeknik Negeri Cilacap*
E-mail: adlannugroho@pnc.ac.id

³ *D3 Teknik Informatika, Politeknik Negeri Cilacap*
E-mail: abdulmuin@pnc.ac.id

*Penulis Korespondensi

Abstrak— Sistem Informasi Akademik (SIAKAD) di Perguruan Tinggi XYZ mempunyai peran penting dalam mendukung kegiatan akademik. Namun, sistem ini belum memiliki dokumentasi pelaporan pengujian yang memadai, yang dapat mempengaruhi kualitas dan keandalan layanan. Penelitian ini bertujuan untuk mengevaluasi kualitas SIAKAD menggunakan pendekatan *exploratory testing* dan *black box testing* serta menghasilkan laporan hasil yang terstruktur. Laporan yang disusun mencakup hasil pengujian tiap modul, skenario pengujian, tingkat keberhasilan dan kegagalan, serta rekomendasi perbaikan sistem. *Exploratory testing* diterapkan untuk mengidentifikasi skenario pengujian tanpa memerlukan dokumentasi rinci, sementara *black box testing* digunakan untuk menguji fungsionalitas berdasarkan spesifikasi tanpa melihat *source code*. Hasil pengujian melibatkan 318 kasus uji pada 11 modul, dengan tingkat keberhasilan 77,36% dan kegagalan 22,64%. Fitur inti seperti *Login*, tahun ajaran, paralel, dan absensi diuji secara komprehensif. Studi ini memberikan rekomendasi untuk memperbaiki validasi *Input* dan penguatan proses bisnis guna meningkatkan keandalan sistem.

Kata Kunci — *black-box testing*, *exploratory testing*, pengujian perangkat lunak, SIAKAD

Abstract— The Academic Information System (SIAKAD) at XYZ University plays a vital role in supporting academic activities. However, this system lacks adequate test reporting documentation, which can affect the quality and reliability of the service. This research aims to evaluate the quality of SIAKAD using *exploratory testing* and *black box testing* approaches and to produce a structured results report. The report includes test results for each module, test scenarios, success and failure rates, and system improvement recommendations. *Exploratory testing* is applied to identify test scenarios without requiring detailed documentation, while *black box testing* is used to test functionality based on specifications without viewing the source code. The test results involve 318 test cases across 11 modules, with a success rate of 77.36% and a failure rate of 22.64%. The essential features such as *Login*, academic year, parallel classes, and attendance were comprehensively tested. This study provides recommendations for improving input validation and strengthening business processes to enhance system reliability.

Keywords — *black-box testing*, *exploratory testing*, software testing, SIAKAD

I. PENDAHULUAN

Perguruan tinggi merupakan suatu organisasi/lembaga yang mendidik mahasiswa dan menghasilkan lulusan yang memiliki kemampuan dan kompetensi yang baik [1]. Perguruan tinggi memiliki banyak sistem informasi untuk menunjang administrasi di kampus masing-masing, baik untuk kegiatan tridharma perguruan tinggi, pembayaran, dan yang lainnya. Salah satunya yang paling sering digunakan di Perguruan Tinggi XYZ untuk administrasi kampus adalah SIAKAD (Sistem Informasi Akademik). SIAKAD merupakan sistem yang mengolah data dan melakukan proses akademik dengan melibatkan mahasiswa, dosen, administrasi akademik, keuangan dan data atribut lainnya [2]. SIAKAD di Perguruan Tinggi XYZ digunakan untuk menunjang kegiatan belajar mengajar, dari pemilihan matakuliah, Input nilai, kartu hasil studi yang ditempuh oleh mahasiswa selama perkuliahan, dan lain sebagainya. Akan tetapi setiap sistem informasi sebelum diimplementasikan perlu adanya pengujian perangkat lunak sehingga semua fitur dapat berjalan dengan semestinya [3]. Pengujian Perangkat Lunak adalah tahap dimana berbagai aspek dari suatu aplikasi, seperti kualitas,

fitur/fungsi, keamanan, dan kinerja, diuji untuk memastikan bahwa aplikasi tersebut memenuhi persyaratan sesuai kebutuhan pengguna [4]. Sedangkan menurut [5] pengujian didefinisikan sebagai proses analisis suatu entitas perangkat lunak untuk mendeteksi perbedaan antara kondisi aktual dan kondisi yang diinginkan (cacat/kesalahan/bug) serta mengevaluasi fitur-fitur dari entitas perangkat lunak tersebut (standar ANSI/IEEE 1059). Pengujian kualitas sistem dapat dipertahankan dan didokumentasikan [6], karena pengujian berperan penting untuk mengukur kualitas dari aplikasi sehingga memenuhi harapan pengguna yang diinginkan terhadap sistem selama pengoperasiannya [7][8].

Penelitian sebelumnya menunjukkan berbagai pendekatan dalam pengujian perangkat lunak. Menurut [9] exploratory testing dapat menemukan kesalahan dan mempermudah user untuk memahami apa saja kegagalan yang terjadi pada sistem yang sedang ditinjau. [10] juga menjelaskan bahwa exploratory testing paling baik dikombinasikan dengan metode pengujian yang lainnya. Diantaranya penelitian oleh [11] menguji sistem inventori alat-alat medis menggunakan metode blackbox testing dengan teknik state transition testing untuk menguji Input dan output dari sistem tanpa perlu memahami source code. Pengujian tersebut menghasilkan 100% sesuai dengan yang diharapkan dan layak untuk digunakan. Kemudian oleh [12] pengujian yang dilakukan bertujuan untuk menguji aplikasi database perguruan tinggi menggunakan metode pengujian black box dengan teknik equivalence partition dalam mendeteksi kesalahan aplikasi. Pengujian tersebut berhasil menguji dari 35 kasus uji, 31 berhasil dan 4 tidak sesuai dengan harapan. Selanjutnya penelitian oleh [13] mengevaluasi sistem informasi perpustakaan berbasis web menggunakan metode black box dengan teknik Equivalen Partitions untuk menguji apakah ada kesalahan pada fungsi sistem, interface, struktur data atau akses data, performansi, dan inisialisasi. Hasil perhitungan validitas sistem tersebut diperoleh nilai 75%, menunjukkan bahwa sistem masih memiliki beberapa kesalahan yang perlu diperbaiki.

Pendekatan exploratory testing dipilih karena metode ini memungkinkan fleksibilitas dalam merancang skenario pengujian tanpa memerlukan dokumentasi secara rinci [14][15]. Hal ini relevan dengan kondisi sistem SIAKAD yang masih dalam tahap pengembangan awal dan sering mengalami perubahan. Penguji dapat dengan cepat menyesuaikan skenario pengujian berdasarkan pengamatan langsung selama pengujian [16]. Sementara itu, metode black-box testing digunakan untuk memvalidasi fungsionalitas sistem tanpa perlu mengakses kode sumber [17]. Pendekatan ini cocok untuk memeriksa apakah masukan dan keluaran sistem sesuai dengan spesifikasi yang ditentukan [18]. Black box testing juga bermanfaat untuk mengidentifikasi potensi kelemahan dalam validasi Input dan pengolahan data [19]. Teknik yang digunakan pada metode black box testing ini yaitu boundary value analysis dan equivalent partitioning. Boundary value analysis merupakan teknik yang mengacu pada proses memasukkan data berdasarkan batas-batas sesuai dengan kebutuhan, seperti nomor NPM, nomor telepon, password, email, dan lainnya [6]. Sementara equivalent partitioning merupakan teknik pengujian berdasarkan masukan data pada setiap form yang ada. Setiap masukan akan dilakukan dan dikelompokkan berdasarkan fungsinya baik itu hasilnya valid atau tidak valid [20]. Pada penelitian ini digunakan kombinasi kedua tahap tersebut karena dapat mengidentifikasi masalah pada sistem SIAKAD secara menyeluruh sekaligus memastikan bahwa dokumentasi pengujian yang dihasilkan relevan dan aplikatif dalam pengembangan sistem kedepannya.

Penelitian ini berfokus pada proses pengujian sekaligus penyusunan dokumentasi pengujian untuk memastikan kualitas dan keandalan SIAKAD. Kombinasi metode exploratory testing dan black box testing untuk mendapatkan hasil pengujian yang optimal. Dokumentasi yang dihasilkan mencakup laporan pengujian, tabel hasil uji, serta analisis keberhasilan dan kegagalan pengujian.

Perguruan tinggi merupakan suatu organisasi/lembaga yang mendidik mahasiswa dan menghasilkan lulusan yang memiliki kemampuan dan kompetensi yang baik [1]. Perguruan tinggi memiliki banyak sistem informasi untuk menunjang administrasi di kampus masing-masing, baik untuk kegiatan tridharma perguruan tinggi, pembayaran, dan yang lainnya. Salah satunya yang paling sering digunakan di Perguruan Tinggi XYZ untuk administrasi kampus adalah SIAKAD (Sistem Informasi Akademik). SIAKAD merupakan sistem yang mengolah data dan melakukan proses akademik dengan melibatkan mahasiswa, dosen, administrasi akademik, keuangan dan data atribut lainnya [2]. SIAKAD di Perguruan Tinggi XYZ digunakan untuk menunjang kegiatan belajar mengajar, dari pemilihan matakuliah, *Input* nilai, kartu hasil studi yang ditempuh oleh mahasiswa selama perkuliahan, dan lain sebagainya. Akan tetapi setiap sistem informasi sebelum diimplementasikan perlu adanya pengujian perangkat lunak sehingga semua fitur dapat berjalan dengan semestinya [3].

Pengujian Perangkat Lunak adalah tahap dimana berbagai aspek dari suatu aplikasi, seperti kualitas, fitur/fungsi, keamanan, dan kinerja, diuji untuk memastikan bahwa aplikasi tersebut memenuhi persyaratan sesuai kebutuhan pengguna [4]. Sedangkan menurut [5] pengujian didefinisikan sebagai proses analisis suatu entitas perangkat lunak untuk mendeteksi perbedaan antara kondisi aktual dan kondisi yang diinginkan (cacat/kesalahan/bug) serta mengevaluasi fitur-fitur dari entitas perangkat lunak tersebut (standar ANSI/IEEE 1059). Pengujian kualitas sistem dapat dipertahankan dan didokumentasikan [6], karena pengujian berperan penting untuk mengukur kualitas dari aplikasi sehingga memenuhi harapan pengguna yang diinginkan terhadap sistem selama pengoperasiannya [7][8].

Penelitian sebelumnya menunjukkan berbagai pendekatan dalam pengujian perangkat lunak. Menurut [9] *exploratory testing* dapat menemukan kesalahan dan mempermudah *user* untuk memahami apa saja kegagalan yang terjadi pada sistem yang sedang ditinjau. [10] juga menjelaskan bahwa *exploratory testing* paling baik dikombinasikan dengan metode pengujian yang lainnya. Diantaranya penelitian oleh [11] menguji sistem inventori alat-alat medis menggunakan metode *blackbox testing* dengan teknik *state transition testing* untuk menguji *Input* dan *output* dari sistem tanpa perlu memahami *source code*. Pengujian tersebut menghasilkan 100% sesuai dengan yang diharapkan dan layak untuk digunakan. Kemudian oleh [12] pengujian yang dilakukan bertujuan untuk menguji aplikasi *database* perguruan tinggi menggunakan metode pengujian *black box* dengan teknik *equivalence partition* dalam mendeteksi kesalahan aplikasi. Pengujian tersebut berhasil menguji dari 35 kasus uji, 31 berhasil dan 4 tidak sesuai dengan harapan. Selanjutnya penelitian oleh [13] mengevaluasi sistem informasi perpustakaan berbasis *web* menggunakan metode *black box* dengan teknik *Equivalen Partitions* untuk menguji apakah ada kesalahan pada fungsi sistem, *interface*, struktur data atau akses data, performansi, dan inisialisasi. Hasil perhitungan validitas sistem tersebut diperoleh nilai 75%, menunjukkan bahwa sistem masih memiliki beberapa kesalahan yang perlu diperbaiki.

Pendekatan *exploratory testing* dipilih karena metode ini memungkinkan fleksibilitas dalam merancang skenario pengujian tanpa memerlukan dokumentasi secara rinci [14][15]. Hal ini relevan dengan kondisi sistem SIAKAD yang masih dalam tahap pengembangan awal dan sering mengalami perubahan. Penguji dapat dengan cepat menyesuaikan skenario pengujian berdasarkan pengamatan langsung selama pengujian [16]. Sementara itu, metode *black-box* testing digunakan untuk memvalidasi fungsionalitas sistem tanpa perlu mengakses kode sumber [17]. Pendekatan ini cocok untuk memeriksa apakah masukan dan keluaran sistem sesuai dengan spesifikasi yang ditentukan [18]. *Black box testing* juga bermanfaat untuk mengidentifikasi potensi kelemahan dalam validasi *Input* dan pengolahan data [19]. Teknik yang digunakan pada metode *black box testing* ini yaitu *boundary value analysis* dan *equivalent partitioning*. *Boundary value analysis* merupakan teknik yang mengacu pada proses memasukkan data berdasarkan batas-batas sesuai dengan kebutuhan, seperti nomor NPM, nomor telepon, *password*, email, dan lainnya [6]. Sementara *equivalent partitioning* merupakan teknik pengujian berdasarkan masukan data pada setiap form yang ada. Setiap masukan akan dilakukan dan dikelompokkan berdasarkan fungsinya baik itu hasilnya valid atau tidak valid [20]. Pada penelitian ini digunakan kombinasi kedua tahap tersebut karena dapat mengidentifikasi masalah pada sistem SIAKAD secara menyeluruh sekaligus memastikan bahwa dokumentasi pengujian yang dihasilkan relevan dan aplikatif dalam pengembangan sistem kedepannya.

Penelitian ini berfokus pada proses pengujian sekaligus penyusunan dokumentasi pengujian untuk memastikan kualitas dan keandalan SIAKAD. Kombinasi metode *exploratory testing* dan *black box testing* untuk mendapatkan hasil pengujian yang optimal. Dokumentasi yang dihasilkan mencakup laporan pengujian, tabel hasil uji, serta analisis keberhasilan dan kegagalan pengujian.

II. METODE PENELITIAN

Penelitian ini menggunakan pendekatan kombinasi *exploratory testing* dan *black box testing* untuk menguji kualitas SIAKAD. *Exploratory testing* digunakan untuk merancang skenario pengujian secara fleksibel dan adaptif [10][21] berdasarkan pengamatan langsung terhadap sistem. Sedangkan *black box testing* difokuskan untuk menguji validitas fungsi sistem berdasarkan *Input* dan *output* tanpa melihat *source code*.



Gambar 1. Tahapan Penelitian

Merujuk pada Gambar 1 diatas, tahapan penelitian ini dimulai dari identifikasi modul dengan metode *exploratory testing* untuk mengeksplorasi sistem secara fleksibel [10] dengan tujuan menentukan komponen utama yang akan diuji. Setelah modul utama teridentifikasi, tahap selanjutnya melakukan perancangan skenario pengujian, yang disusun berdasarkan fungsionalitas sistem. Skenario ini mencakup pembuatan skenario valid dan tidak valid untuk menguji bagaimana sistem merespon berbagai *input*. Skenario *valid* mengacu pada input yang sesuai dengan aturan, sedangkan skenario tidak *valid* digunakan untuk menguji ketahanan sistem terhadap kesalahan atau input yang tidak terduga. Setelah skenario dibuat, tahap berikutnya adalah pelaksanaan pengujian dengan menggunakan metode *blackbox testing* yang berfokus pada validasi *input* dan *output* sistem. Hasil dari pengujian ini kemudian dicatat secara sistematis dalam tahap pencatatan hasil, yang mencakup kasus uji, tingkat keberhasilan, serta kegagalan yang terjadi. Seluruh data dikompilasi dalam bentuk tabel dan laporan untuk mempermudah analisis lebih lanjut. Informasi ini menjadi dasar dalam tahap penyusunan dokumentasi yang berisi analisis hasil pengujian, rekomendasi perbaikan dan laporan pengembangan sebagai referensi bagi pengembang dalam penyempurnaan sistem.

III. HASIL DAN PEMBAHASAN

Secara keseluruhan, berdasarkan tabel 1 pengujian SIAKAD ini melibatkan 11 modul dengan total 318 *test case*. Dari jumlah tersebut 246 skenario berhasil (77.36%), sementara 72 lainnya gagal (22.64%).

Tabel 1. Persentase Keberhasilan Pengujian

Fitur	Total Pengujian	Berhasil	Gagal	Persentase Berhasil (%)	Persentase Gagal (%)
Login	11	6	5	54,55	45,45
Master Data Tahun Ajaran	29	16	13	55,17	44,83
Master Data Tingkat	9	8	1	88,89	11,11
Master Data Paralel	10	9	1	90,00	10,00
Master Data Jurusan	6	5	1	83,33	16,67
Master Data Jenjang	12	9	3	75,00	25,00
Master Data Program Studi	9	7	2	77,78	22,22
Master Data Mata Kuliah	12	8	4	66,67	33,33
Master Data MBKM	8	6	2	75,00	25,00
Master Data Mitra	10	7	3	70,00	30,00
Data Mahasiswa	37	22	15	59,46	40,54
Status Pengajuan	21	16	5	76,19	23,81
Absensi	20	17	3	85,00	15,00
Nilai	3	2	1	66,67	33,33
Total				77,36	22,64

Modul yang diuji meliputi *Login* , tahun ajaran, tingkat, paralel, mata kuliah dan fitur administrasi lainnya

1. Fitur *Login* , fitur ini diuji dengan 11 skenario termasuk validasi kredensial yang benar dan salah, *Input* karakter khusus, serta skenario *Input* kosong. Tabel 2 menunjukkan sample hasil pengujian fitur *Login* termasuk validasi kredensial yang benar, salah serta *Input* kosong. Hasilnya mengidentifikasi kebutuhan untuk memperkuat validasi *Input* tertentu dengan tingkat keberhasilan 54,5%.
2. Fitur tahun ajaran, dari 42 skenario yang diuji, tingkat keberhasilan mencapai 71,4% yang merujuk pada tabel 2. Kegagalan terutama terjadi pada input karakter non-numerik, simbol dan nilai batas yang tidak sesuai. Ini mengidentifikasi kelemahan pada pengolahan data *Input* yang tidak valid.
3. Fitur paralel, tingkat, dan jurusan, ketiga fitur ini memiliki keberhasilan diatas 80% dengan sebagian besar kegagalan disebabkan oleh pengolahan data duplikasi dan input yang tidak valid. Tabel 3,4 dan 5 yang menyajikan sample hasil pengujian dari ketiga fitur tersebut.
4. Fitur absensi dan nilai, modul ini diuji dengan skenario *input* data dan validasi, dengan tingkat keberhasilan di atas 85% yang tersaji pada tabel 4. Tantangan utamanya adalah memastikan integritas data ketika terjadi pengeditan atau penghapusan.

Tabel 2. Sample Hasil Pengujian Fitur Login

<i>Test ID</i>	<i>Action</i>	<i>Input</i>	<i>Expected Result</i>	<i>Result</i>
TC_Login_001	Login dengan <i>username</i> dan <i>password</i> valid	<i>Username</i> : admin, <i>Password</i> : correctpass	Pengguna berhasil Login	Berhasil
TC_Login_002	Login dengan <i>username</i> tidak valid	<i>Username</i> : invalid, <i>Password</i> : correctpass	Login gagal dengan pesan <i>Error</i>	Gagal
TC_Login_003	Login dengan <i>password</i> tidak valid	<i>Username</i> : admin, <i>Password</i> : wrongpass	Login gagal dengan pesan <i>Error</i>	Gagal
TC_Login_004	Kosongkan kolom <i>username</i>	<i>Password</i> : correctpass	Login gagal dengan pesan <i>Error</i>	Berhasil
TC_Login_005	Kosongkan kolom <i>password</i>	<i>Username</i> : admin	Login gagal dengan pesan <i>Error</i>	Berhasil

Tabel 3. Sample Hasil Pengujian Fitur Tahun Ajaran

<i>Test ID</i>	<i>Action</i>	<i>Input</i>	<i>Expected Result</i>	<i>Result</i>
TC_TA_001	<i>Input</i> rentang tahun valid	Tahun: 2023-2024	Tahun berhasil ditambahkan	Berhasil
TC_TA_002	<i>Input</i> rentang tahun tidak valid	Tahun: ABCD	<i>Error</i> : Format tahun tidak valid	Gagal
TC_TA_003	<i>Input</i> rentang tahun dengan tahun awal kosong	Tahun: -2024	<i>Error</i> : Format tahun tidak valid	Gagal
TC_TA_004	<i>Input</i> rentang tahun <i>overlapping</i>	Tahun: 2023-2023	<i>Error</i> : Rentang tahun tumpang tindih	Gagal
TC_TA_005	Kosongkan kolom tahun	Tahun:	<i>Error</i> : Tahun tidak boleh kosong	Berhasil

Tabel 4. Sample Hasil Pengujian Fitur Paralel

<i>Test ID</i>	<i>Action</i>	<i>Input</i>	<i>Expected Result</i>	<i>Result</i>
TC_PR_001	Tambah paralel baru dengan data valid	Paralel: A1	Paralel berhasil ditambahkan	Berhasil
TC_PR_002	Kosongkan kolom nama paralel	Paralel:	<i>Error</i> : Nama paralel tidak boleh kosong	Gagal
TC_PR_003	<i>Input</i> nama paralel dengan karakter khusus	Paralel: @!#	<i>Error</i> : Nama paralel tidak valid	Gagal
TC_PR_004	Tambah nama paralel yang sudah ada	Paralel: A1	<i>Error</i> : Nama paralel sudah ada	Berhasil
TC_TA_005	Kosongkan kolom tahun	Tahun:	<i>Error</i> : Tahun tidak boleh kosong	Berhasil

Tabel 5. Sample Hasil Pengujian Fitur Tingkat

<i>Test ID</i>	<i>Action</i>	<i>Input</i>	<i>Expected Result</i>	<i>Result</i>
TC_TK_001	Tambah tingkat dengan data valid	Tingkat: Diploma 3	Tingkat berhasil ditambahkan	Berhasil
TC_TK_002	Kosongkan nama tingkat	Tingkat:	<i>Error</i> : Nama tingkat tidak boleh kosong	Gagal
TC_TK_003	Tambah tingkat dengan karakter khusus	Tingkat: @Diploma	<i>Error</i> : Nama tingkat tidak valid	Gagal
TC_TK_004	Tambah Tingkat	Nama tingkat yang sudah ada	Pesan <i>Error</i> : "Nama Tingkat sudah ada"	Berhasil
TC_TK_005	Edit Tingkat	Nama tingkat valid	Nama tingkat berhasil diperbarui	Berhasil
TC_TK_007	Edit Tingkat	Nama tingkat duplikat	Pesan <i>Error</i> : "Nama Tingkat sudah ada"	Berhasil
TC_TK_008	Hapus Tingkat	Nama tingkat tersedia	Tingkat berhasil dihapus	Berhasil

Tabel 6. Sample Hasil Pengujian Fitur Jurusan

<i>Test ID</i>	<i>Action</i>	<i>Input</i>	<i>Expected Result</i>	<i>Result</i>
TC_JR_001	Tambah Jurusan	Nama jurusan valid	Jurusan baru berhasil ditambahkan	Berhasil
TC_JR_002	Tambah Jurusan	<i>Field</i> kosong	Pesan <i>Error</i> : " <i>Field</i> tidak boleh kosong"	Berhasil
TC_JR_003	Edit Jurusan	Nama jurusan valid	Data jurusan berhasil diperbarui	Berhasil
TC_JR_004	Edit Jurusan	<i>Field</i> kosong	Pesan <i>Error</i> : " <i>Field</i> tidak boleh kosong"	Berhasil
TC_JR_005	Hapus Jurusan	Nama jurusan tersedia	Jurusan berhasil dihapus	Berhasil
TC_JR_006	Hapus Jurusan	Jurusan terkait data lain	Pesan <i>Error</i> : "Jurusan tidak dapat dihapus"	Berhasil

Tabel 7. Sample Hasil Pengujian Fitur Absensi

<i>Test ID</i>	<i>Action</i>	<i>Input</i>	<i>Expected Result</i>	<i>Result</i>
TC_ABS_001	Tampilkan data absensi mahasiswa	Kelas: TI-A	Data absensi ditampilkan	Berhasil
TC_ABS_002	Tambah absensi dengan data valid	Tanggal: 2024-11-19, Status: Hadir	Absensi berhasil ditambahkan	Berhasil
TC_ABS_003	Tambah absensi dengan status kosong	Tanggal: 2024-11-19, Status:	<i>Error</i> : Status tidak boleh kosong	Gagal
TC_ABS_004	Edit data absensi	Status: Terlambat	Data absensi berhasil diperbarui	Berhasil
TC_ABS_005	Hapus data absensi	Tanggal: 2024-11-19	Data absensi berhasil dihapus	Berhasil

Dari hasil pengujian diatas menunjukkan bahwa sistem SIAKAD secara umum telah memenuhi kebutuhan fungsional, terutama pada skenario *valid*. Namun, kelemahan signifikan ditemukan pada validasi *Input* tidak valid, baik di fitur *Login*, tahun ajaran, maupun data paralel. Masalah ini berpotensi meningkatkan risiko keamanan seperti *SQL Injection* atau manipulasi data.

IV. KESIMPULAN

Hasil pengujian ini menggarisbawahi keandalan sistem SIAKAD pada fitur inti dengan tingkat keberhasilan 77,36%. Namun, untuk meningkatkan kualitas sistem, perlu diberikan penguatan pada validasi *input* dan proses bisnis. Penelitian ini tidak hanya memastikan keandalan SIAKAD melalui pengujian tetapi juga menghasilkan laporan hasil yang berfungsi sebagai panduan untuk perbaikan sistem selanjutnya. Laporan hasil mencakup skenario pengujian, laporan keberhasilan dan kegagalan, serta rekomendasi perbaikan. Rekomendasi perbaikan sistem ini dapat memberikan layanan yang lebih aman, andal, dan sesuai kebutuhan perguruan tinggi XYZ.

Penelitian selanjutnya dapat difokuskan pada *usability testing* untuk mengevaluasi aspek pengalaman pengguna efisiensi sistem. Selain itu, pengujian dengan pendekatan otomatisasi dapat diterapkan untuk meningkatkan konsistensi hasil pengujian.

DAFTAR PUSTAKA

- [1] I.G.P. Asto Buditjahjanto, “Analisis Layanan Sistem Informasi Akademik Perguruan Tinggi Berbasis Fuzzy Service Quality,” *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 9, no. 3, pp. 225–232, 2020, doi: 10.22146/v9i3.264.
- [2] H. Anwar, “Penerapan Sistem Informasi Akademik (Siakad) Dalam Meningkatkan Layanan Akademik di Prodi Manajemen Pendidikan Islam Fakultas Ilmu Tarbiyah Dan ...,” *Tadbir J. Manaj. Pendidik. Islam*, vol. 4, no. 1, pp. 6–20, 2016, [Online]. Available: <https://journal.iaingorontalo.ac.id/index.php/tjmpi/article/view/1129%0Ahttps://journal.iaingorontalo.ac.id/index.php/tjmpi/article/download/1129/862>
- [3] M. A. H. Anwar and Y. Kurniawan, “Dokumentasi Software Testing Berstandar Ieee 829-2008 Untuk Sistem Informasi Terintegrasi Universitas,” *Kurawal - J. Teknol. Inf. dan Ind.*, vol. 2, no. 2, pp. 118–125, 2019, doi: 10.33479/kurawal.v2i2.261.
- [4] Rahman Abdillah, Rudi Hermawan, Wawan Hermawansyah, Ibnu Adkha, and Heri Arifin, “Pengujian Perangkat Lunak Sistem Informasi Inventori pada Usaha Jasa Pengiriman Paket,” *Polyg. J. Ilmu Komput. dan Ilmu Pengetah. Alam*, vol. 2, no. 4, pp. 166–175, Jul. 2024, doi: 10.62383/polygon.v2i4.199.
- [5] S. K. S. A. Singh, *software testing*. 2019.
- [6] M. Sholeh, I. Gifas, Cahiman, and M. A. Fauzi, “Black Box Testing on ukmbantul.com Page with Boundary Value Analysis and Equivalence Partitioning Methods,” *J. Phys. Conf. Ser.*, vol. 1823, no. 1, 2021, doi: 10.1088/1742-6596/1823/1/012029.
- [7] F. C. Ningrum, D. Suherman, S. Aryanti, H. A. Prasetya, and A. Saifudin, “Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions,” *J. Inform. Univ. Pamulang*, vol. 4, no. 4, p. 125, Dec. 2019, doi: 10.32493/informatika.v4i4.3782.
- [8] T. Ahmad, J. Iqbal, A. Ashraf, D. Truscan, and I. Porres, “Model-based testing using UML activity diagrams: A systematic mapping study,” *Comput. Sci. Rev.*, vol. 33, pp. 98–112, Aug. 2019, doi: 10.1016/j.cosrev.2019.07.001.
- [9] D. Pfahl, H. Yin, M. V. Mäntylä, and J. Münch, “How is exploratory testing used? A state-of-the-practice survey,” in *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, New York, NY, USA: ACM, Sep. 2014, pp. 1–10. doi: 10.1145/2652524.2652531.
- [10] T. Rahman, “Exploratory Testing: A Dynamic Approach to Software Quality Assurance,” 2024. doi: 10.2139/ssrn.4991527.
- [11] A. Setiawan, H. Gunawan, A. Hidayatullah, M. Aprinaldi, and S. Putra, “Black Box Testing Dengan Teknik State Transition Testing Pada Inventori Alat-Alat Medis,” *J. J. Sains Dan Teknol.*, vol. 2, no. 2, pp. 104–111, 2022.
- [12] H. Raihan and A. Voutama, “Pengujian Black Box Pada Aplikasi Database Perguruan Tinggi dengan Teknik Equivalence Partition,” *Antivirus J. Ilm. Tek. Inform.*, vol. 17, no. 1, pp. 1–18, 2023, doi: 10.35457/antivirus.v17i1.2501.
- [13] B. M. Haqqoni, I. Winarno, M. N. Musthofa, M. Sakdi, and A. Saifudin, “Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan Metode Black Box Testing Bagi Pemula,” *STORAGE J. Ilm. Tek. dan Ilmu Komput.*, vol. 1, no. 2, pp. 1–8, 2022, doi: 10.55123/storage.v1i2.270.
- [14] R. Kimla and R. Czerwinski, “A Methodical Approach to Functional Exploratory Testing for Embedded Systems,” *Appl. Sci.*, vol. 12, no. 19, p. 10016, Oct. 2022, doi: 10.3390/app121910016.
- [15] Bernard Homès, *Fundamental Software Testing*, 2nd ed., vol. 11, no. 1. 2024.
- [16] J. Leveau, X. Blanc, L. Réveillère, J. Falleri, and R. Rouvoy, “Fostering the diversity of exploratory testing in web applications,” *Softw. Testing, Verif. Reliab.*, vol. 32, no. 5, pp. 1–20, Aug. 2022, doi: 10.1002/stvr.1827.
- [17] A. D. Frayudha, I. R. Pande, and M. B. Juwita, “Implementation of Black Box Testing with the Application of Equivalence Partitioning Techniques in the M-Magazine Android Application at Semen Gresik High School,” vol. 9, no. 1, pp. 134–143, 2024.
- [18] S. R. Wicaksono, *Black Box Testing Teori Dan Studi Kasus*, no. February. 2022. doi:

10.5281/zenodo.7659674.

- [19] N. M. D. Febriyanti, A. Sudana, and ..., "Implementasi Black Box Testing pada Sistem Informasi Manajemen Dosen," *J. Ilm. ...*, vol. 2, no. 3, 2021, [Online]. Available: [http://download.garuda.kemdikbud.go.id/article.php?article=3457876&val=30165&title=Implementasi Black Box Testing pada Sistem Informasi Manajemen Dosen](http://download.garuda.kemdikbud.go.id/article.php?article=3457876&val=30165&title=Implementasi%20Black%20Box%20Testing%20pada%20Sistem%20Informasi%20Manajemen%20Dosen)
- [20] Y. D. Wijaya and M. W. Astuti, "Pengujian Blackbox Sistem Informasi Penilaian Kinerja Karyawan Pt Inka (Persero) Berbasis Equivalence Partitions," *J. Digit. Teknol. Inf.*, vol. 4, no. 1, p. 22, 2021, doi: 10.32502/digital.v4i1.3163.
- [21] V. Dahiya, "Testing Competence : Scripted Testing Versus Exploratory Testing," *Int. J. Adv. Res. Sci. Eng. Technol.*, vol. 1, no. 4, pp. 330–332, 2014.