p-ISSN : 2406-7733
e-ISSN : 2597-9922

PENGUJIAN SISTEM MANAJEMEN RUMAH SAKIT MENGGUNAKAN METODE *BLACK BOX TESTING*

Ananda Syafika[1], Zulfahmi Indra[2], M. Farel Revansha Buulolo[3], Nayla Luthfiah Hanan[4]
Program Studi Ilmu Komputer, Universitas Negeri Medan
Jl. William Iskandar Ps.V, Kenangan Baru, Kec. Percut Sei Tuan, Kabupaten Deli Serdang
Sumatera Utara 20221
E-mail: anandasyafika402@gmail.com[1], farelrevansha@gmail.com[2], naylaluthfiahh@gmail.com[3], zulfahmi.indra@unimed.ac.id[4]

Abstrak - Layanan kesehatan yang berkualitas membutuhkan sistem informasi yang andal dan bebas dari kesalahan fungsional. Sistem Manajemen Rumah Sakit (SIMRS) berbasis web merupakan solusi terintegrasi yang mengelola proses operasional rumah sakit secara menyeluruh, mulai dari autentikasi pengguna, pengelolaan data dokter dan pasien, hingga pendaftaran rawat inap. Namun, kompleksitas fitur yang dimiliki berpotensi menimbulkan kesalahan yang berdampak pada mutu pelayanan. Penelitian ini bertujuan merancang dan melaksanakan pengujian fungsional pada Sistem Manajemen Rumah Sakit berbasis web menggunakan metode *Black Box Testing* untuk memastikan seluruh fungsi sistem berjalan sesuai kebutuhan pengguna. Sistem dibangun dengan bahasa pemrograman PHP dan basis data MySQL, melibatkan tiga aktor utama yaitu Admin, Dokter, dan Pasien. Metode yang diterapkan adalah *Black Box Testing* dengan teknik *Equivalence Partitioning* dan *Boundary Value Analysis*. Pengujian dilaksanakan terhadap delapan fitur utama melalui tujuh skenario test case yang mencakup modul autentikasi, pendaftaran rawat inap, riwayat rawat inap, dan *logout.* Hasil pengujian menunjukkan seluruh tujuh skenario uji berstatus berhasil, yang berarti setiap fungsi utama sistem menghasilkan keluaran sesuai ekspektasi, termasuk penanganan masukan tidak valid seperti kesalahan nama pengguna dan kata sandi. Penelitian ini membuktikan bahwa metode *Black Box Testing* efektif memvalidasi fungsionalitas SIMRS dari perspektif pengguna dan dapat dijadikan acuan dalam proses penjaminan kualitas perangkat lunak di bidang kesehatan sebelum sistem diimplementasikan secara penuh.
Kata Kunci: *Black Box Testing, Equivalence Partitioning*, Pengujian Perangkat Lunak, Sistem Manajemen Rumah Sakit, *Test Case*

I. PENDAHULUAN

Perkembangan teknologi informasi telah membawa perubahan signifikan di sektor kesehatan (Nurfaidah et al., 2025). Rumah sakit memerlukan sistem yang mampu mengelola proses operasional secara efektif, salah satunya melalui Sistem Informasi Manajemen Rumah Sakit (SIMRS) sistem terintegrasi yang mencakup aspek klinis, keuangan, dan administrasi, serta terbukti meningkatkan efisiensi dan meminimalkan kesalahan medis (Pratama & Purwanto, 2023).

Namun, kompleksitas fitur SIMRS berpotensi menimbulkan kesalahan yang berdampak pada kualitas layanan pasien, sehingga diperlukan pengujian perangkat lunak untuk memastikan setiap fungsi berjalan sesuai spesifikasi (Utami & Setyodewi, 2023).

Pengujian perangkat lunak bertujuan menemukan kesalahan dan memastikan sistem berjalan sesuai kebutuhan pengguna (Syihab et al., 2024). Penelitian ini menggunakan metode *Black Box Testing*, yaitu pengujian fungsionalitas berdasarkan masukan dan keluaran tanpa memperhatikan struktur internal program (Tonggiroh et al., 2024), yang dinilai tepat untuk mengevaluasi fitur SIMRS dari sudut pandang pengguna.

Beberapa penelitian sebelumnya telah menerapkan metode ini pada layanan kesehatan, namun masih memiliki keterbatasan. (Sabirin & Widyarto, n.d.) berhasil mengidentifikasi *bug* pada sistem penjadwalan klinik. (Musyaffa et al., 2024) hanya mencakup satu form login pada SIMRS berbasis web, dan (Syabrullah et al., 2023) meski menguji 120 skenario, fokusnya bukan pada SIMRS secara keseluruhan. Keterbatasan tersebut menunjukkan perlunya pengujian yang lebih komprehensif.

Oleh karena itu, penelitian ini bertujuan merancang pengujian SIMRS menggunakan *Black Box Testing* melalui identifikasi fitur, penyusunan *use case*, dan pembuatan *test case* guna memastikan sistem berfungsi sesuai kebutuhan pengguna.

II. LANDASAN TEORI

Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses evaluasi sistem untuk menemukan kesalahan dan memastikan perangkat lunak berjalan sesuai kebutuhan. Menurut (Syihab et al., 2024), pengujian merupakan proses verifikasi dan validasi untuk memastikan perangkat lunak memenuhi spesifikasi yang ditetapkan serta bebas dari kesalahan yang dapat memengaruhi kinerja sistem. Pengujian menjadi bagian penting dalam SDLC karena berperan menjaga kualitas, keandalan, dan kinerja sistem sejak tahap awal pengembangan.

p-ISSN : 2406-7733
e-ISSN : 2597-9922

Berdasarkan pendekatannya, pengujian dibagi menjadi *Black Box Testing*, *White Box Testing*, dan *Grey Box Testing*. *Black Box Testing* berfokus pada fungsionalitas melalui input dan output tanpa memperhatikan struktur internal, *White Box Testing* menganalisis struktur internal dan kode sumber, sedangkan *Grey Box Testing* menggabungkan keduanya (Fauzi & Yunial, 2025; Tonggiroh et al., 2024). Selain itu, pengujian juga dibedakan berdasarkan tujuan menjadi *Functional* dan *Non-Functional Testing* (Tonggiroh et al., 2024), serta serta berdasarkan tahapan meliputi *Unit, Integration, System*, dan *Acceptance Testing* (Abdillah et al., 2024; Wicaksono, 2023).

Black Box Testing

Black Box Testing adalah metode pengujian yang berfokus pada fungsionalitas sistem tanpa memperhatikan struktur internal maupun kode sumber program. Pengujian dilakukan dengan memberikan input dan membandingkan output yang dihasilkan dengan hasil yang diharapkan. Menurut (Tonggiroh et al., 2024), metode ini dilakukan dari sudut pandang pengguna sehingga penguji tidak perlu memahami proses internal program. Menurut Jaya (2018) dalam (Syabrullah et al., 2023), menambahkan bahwa penguji dapat menentukan kondisi input yang dirancang untuk menguji seluruh syarat fungsional program. Metode ini banyak digunakan karena mudah diterapkan, tidak memerlukan pemahaman kode, dan efektif dalam menemukan kesalahan pada fungsi sistem, validasi data, serta antarmuka pengguna (Sabirin & Widyarto, n.d.).

Pelaksanaannya diawali dengan mengidentifikasi fitur, menyusun skenario dan *test case*, menjalankan pengujian, lalu membandingkan hasil aktual dengan yang diharapkan. Penyusunan *test case* bertujuan memastikan sistem berfungsi sesuai maksud awal dan menghasilkan keluaran yang tepat untuk data valid maupun tidak valid (Christian dkk., 2019 dalam Musyaffa et al., 2024). Ketidaksesuaian yang ditemukan dicatat dan didokumentasikan sebagai bahan evaluasi.

Teknik yang umum digunakan meliputi *Equivalence Partitioning*, yang membagi data masukan ke dalam kelompok-kelompok agar pengujian lebih efisien, dan Boundary Value Analysis, yang berfokus pada nilai batas minimum dan maksimum input (Abdillah et al., 2025). Kelebihan metode ini adalah tidak memerlukan pengetahuan teknis pemrograman dan efektif mendeteksi kesalahan fungsionalitas, namun kelemahannya adalah tidak dapat mendeteksi kesalahan pada logika internal dan cakupannya bergantung pada kelengkapan skenario uji yang dirancang.

Test Case

Test case adalah pengujian berdasarkan kondisi dan hasil yang telah ditentukan sebelumnya, di mana hasilnya dibandingkan dengan ekspektasi awal dan jika terdapat perbedaan maka dilakukan perbaikan kode. *Test case* menjadi titik awal pengujian setelah memasukkan nilai input ke sistem hingga diperoleh hasil yang definitif pada postcondition eksekusi (Hasibuan & Dirgahayu, 2021).

Atribut yang harus dipenuhi dalam penyusunan test case meliputi fitur, *test case id,* deskripsi, *pre condition, test steps, expected result, actual result, dan status. Test case* yang baik adalah yang memiliki kemungkinan tinggi menemukan kesalahan yang belum ditemukan sebelumnya, bukan sekadar membuktikan program bekerja dengan benar (Hasibuan & Dirgahayu, 2021). Dalam *Black Box Testing,* pengujian menggunakan skenario yang mencakup kasus pengujian, perkiraan, hasil, dan status untuk setiap fitur yang diuji (Utami & Setyodewi, 2023), di mana setiap skenario dibuat berdasarkan alur dan fungsionalitas aplikasi (Luginawati & Wahyu, 2023).

Sistem Manajemen Rumah Sakit

SIMRS adalah sistem komprehensif dan terintegrasi untuk pengelolaan seluruh aspek rumah sakit, mencakup kebutuhan klinis, fasilitas, keuangan, dan administrasi (Pratama & Purwanto, 2023). Berdasarkan Permenkes No. 82 Tahun 2013, setiap rumah sakit wajib menerapkan SIMRS sebagai sistem pengumpul dan pengintegrasian data dari seluruh unit rumah sakit (Sitompul, 2023).

Implementasinya terbukti meningkatkan efisiensi pelayanan, produktivitas tenaga kesehatan, akurasi data, serta meminimalisir kesalahan medis (Nurfaidah et al., 2025; Pratama & Purwanto, 2023). Fitur utamanya mencakup pendaftaran rawat inap, rawat jalan, dan gawat darurat, pengelolaan rekam medis, laporan analisis, manajemen keuangan, serta sistem keamanan akses login per modul (Pratama & Purwanto, 2023).

III. METODOLOGI PENELITIAN

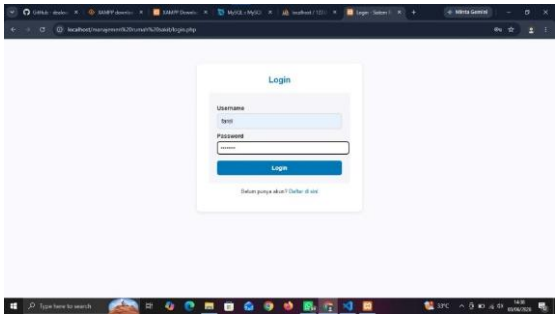
Deskripsi Sistem

Sistem yang menjadi objek pengujian dalam penelitian ini adalah Sistem Manajemen Rumah Sakit berbasis web yang dibangun menggunakan bahasa pemrograman PHP dengan basis data MySQL. Sistem ini dirancang untuk mengelola proses operasional rumah sakit secara terintegrasi dan melibatkan tiga aktor utama, yaitu Admin, Dokter, dan Pasien. Masing-masing aktor memiliki hak akses yang berbeda sesuai dengan peran dan tanggung jawabnya dalam sistem.

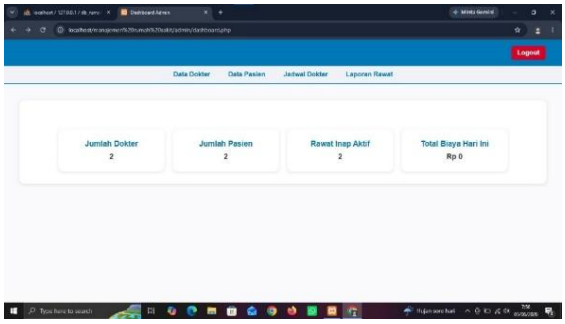
Sistem ini memiliki beberapa modul utama. Modul Autentikasi mencakup fitur *login* dan registrasi yang memungkinkan pengguna masuk ke

p-ISSN : 2406-7733
e-ISSN : 2597-9922

sistem sesuai peran masing-masing. Modul Admin mencakup pengelolaan data dokter, data pasien, jadwal dokter, dan laporan rawat inap. Modul Dokter mencakup fitur melihat data pasien rawat inap serta menginput diagnosa dan biaya perawatan. Modul Pasien mencakup fitur pendaftaran rawat inap dan melihat riwayat rawat inap. Seluruh modul terhubung melalui satu sistem basis data terpusat sehingga data dapat diakses secara *real-time* oleh semua pengguna yang berwenang.



Gambar 1. Tampilan Halaman *Login*



Gambar 2. Tampilan *Dashboard* Admin

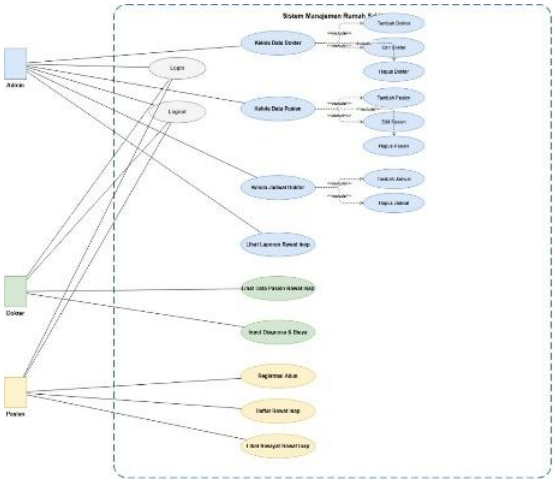
Use Case

Sistem Manajemen Rumah Sakit ini melibatkan tiga aktor utama yang berinteraksi langsung dengan sistem, yaitu Admin, Dokter, dan Pasien. Setiap aktor memiliki hak akses dan *use case* yang berbeda sesuai dengan peran dan tanggung jawabnya dalam sistem. Identifikasi aktor beserta *use case* masing-masing disajikan pada Tabel 1.

Tabel 1. Identifikasi Aktor dan *Use Case* Sistem Manajemen Rumah Sakit

Aktor	*Use Case*
Admin	*Login*, Kelola Data Dokter (tambah/*edit*/hapus), Kelola Data Pasien (tambah/*edit*/hapus), Kelola Jadwal Dokter (tambah/hapus), Lihat Laporan Rawat Inap
Dokter	*Login*, Lihat Data Pasien Rawat Inap, Input Diagnosa & Biaya
Pasien	Login, Registrasi Akun, Daftar Rawat Inap, Lihat Riwayat Rawat Inap

Berdasarkan Tabel 1, Admin memiliki hak akses paling luas, meliputi pengelolaan data dokter, pasien, jadwal dokter, dan laporan rawat inap. Dokter dapat mengakses data pasien yang menjadi tanggung jawabnya serta mengisi diagnosa dan biaya perawatan. Pasien dapat melakukan registrasi akun, pendaftaran rawat inap, dan melihat riwayat perawatan. Ketiga aktor berbagi *use case Login* dan *Logout*. Hubungan interaksi antara aktor dan *use case* digambarkan pada Gambar 3.



Gambar 3. *Diagram Use Case* Sistem Manajemen Rumah Sakit

Fitur yang Diuji

Tablel 2. Fitr-Fitur yang Diuji pada Sistem Manajemen Rumah Sakit

No	Fitur	Modul	Deskripsi	Aktor	Prioritas
1	**Login**	Autentikasi	Input username & password untuk masuk sistem	Admin, Dokter, Pasien	Tinggi
2	**Registrasi Pasien**	Autentikasi	Pendaftaran akun baru dengan data diri lengkap	Pasien	Tinggi
3	**Kelola Data Dokter**	Admin	Tambah, edit, dan hapus data dokter	Admin	Tinggi
4	**Kelola Data Pasien**	Admin	Tambah, edit, dan hapus data pasien	Admin	Tinggi
5	**Kelola Jadwal Dokter**	Admin	Tambah dan hapus jadwal praktik dokter	Admin	Sedang

p-ISSN : 2406-7733
e-ISSN : 2597-9922

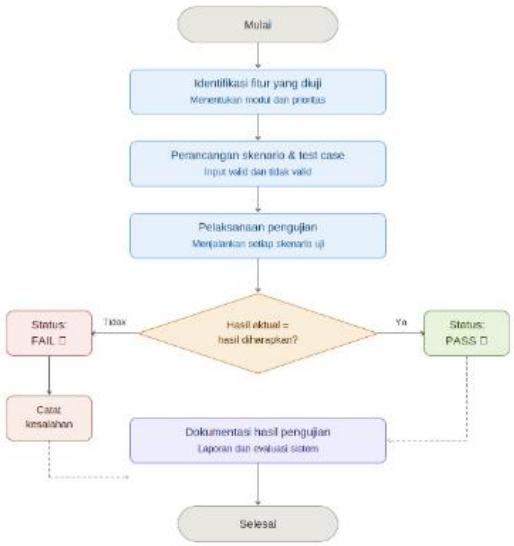
No	Fitur	Modul	Deskripsi	Aktor	Prioritas
6	**Daftar Rawat Inap**	Pasien	Pasien memilih dokter dan tanggal masuk rawat inap	Pasien	Tinggi
7	**Input Diagnosa & Biaya**	Dokter	Dokter menginput diagnosa dan biaya untuk pasien	Dokter	Tinggi
8	**Riwayat Rawat Inap**	Pasien	Pasien melihat riwayat rawat inap beserta diagnosa	Pasien	Sedang

Berdasarkan Tabel 2, terdapat delapan fitur yang menjadi objek pengujian dalam penelitian ini. Pemilihan fitur didasarkan pada tingkat kekritisan fungsi terhadap operasional sistem. Fitur dengan prioritas tinggi merupakan fitur inti yang paling sering digunakan dan berpotensi menimbulkan dampak signifikan apabila terjadi kegagalan, seperti *login*, registrasi, dan pendaftaran rawat inap.

Metode Pengujian

Metode pengujian yang digunakan dalam penelitian ini adalah *Black Box Testing*. Metode ini dipilih karena pengujian dilakukan dari sudut pandang pengguna dengan berfokus pada fungsionalitas sistem tanpa perlu memahami struktur internal kode program, sehingga sesuai dengan tujuan pengujian yang ingin memastikan seluruh fitur sistem berjalan sesuai kebutuhan pengguna. Teknik yang diterapkan adalah *Equivalence Partitioning* dan *Boundary Value Analysis*. *Equivalence Partitioning* digunakan untuk membagi data masukan ke dalam kelompok valid dan tidak valid pada fitur-fitur seperti *login*, registrasi, dan pengelolaan data. *Boundary Value Analysis* diterapkan pada *field* input yang memiliki batasan nilai tertentu, seperti *field* umur pada *form* registrasi dan input biaya pada modul diagnosa.

Proses pengujian dilakukan melalui beberapa tahapan. Pertama, mengidentifikasi fitur-fitur yang akan diuji berdasarkan modul yang tersedia dalam sistem. Kedua, merancang skenario pengujian dan *test case* untuk setiap fitur dengan mempertimbangkan kondisi input valid maupun tidak valid. Ketiga, melaksanakan pengujian dengan menjalankan setiap skenario secara langsung pada sistem. Keempat, membandingkan hasil aktual yang diperoleh dengan hasil yang diharapkan pada setiap *test case*. Terakhir, mendokumentasikan seluruh hasil pengujian sebagai bahan evaluasi sistem.



Gambar 4. Alur Proses Pengujian *Black Box Testing*

Berdasarkan Gambar 4, proses pengujian diawali dengan mengidentifikasi fitur yang diuji, kemudian merancang skenario dan *test case* berdasarkan kondisi input valid maupun tidak valid. Setelah pengujian dilaksanakan, hasil aktual dibandingkan dengan hasil yang diharapkan. Apabila sesuai, pengujian dinyatakan berhasil dengan status *Pass*, sedangkan apabila terdapat ketidaksesuaian, pengujian dinyatakan gagal dengan status *Fail* dan kesalahan dicatat untuk dianalisis. Seluruh hasil pengujian kemudian didokumentasikan sebagai bahan evaluasi sistem.

IV. HASIL DAN PEMBAHASAN

Tabel *Test Case*

Tabel 3. *Test Case*

ID Uji	Skenario Pengujian	Data Input	Expected Result	Actual Result	Status
TC-1	Register akun	Nama lengkap: farel Username: farel Password: 12345678 Umur: 19 Jenis Kelamin: Laki-Laki Alamat: Medan	Pembuatan akun berhasil	Pembuatan akun berhasil	**Pass**

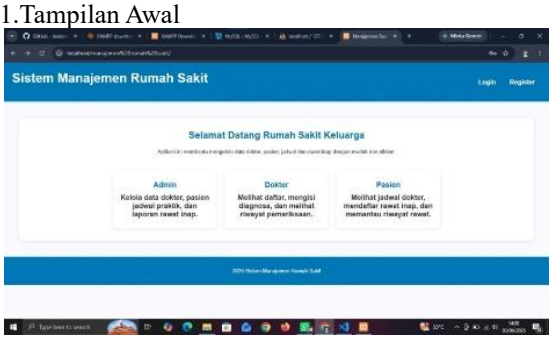
p-ISSN : 2406-7733
e-ISSN : 2597-9922

ID Uji	Skenario Pengujian	Data Input	Expected Result	Actual Result	Status
TC -2	Login Akun	Username: farel Password: 12345678	Login berhasil dan masuk ke halaman dashboard	Login berhasil dan masuk ke halaman dashboard	**Pass**
TC -3	Login dengan username salah	Username: baru Password: 12345678	Login gagal, sistem menampilkan pesan error "username tidak ditemukan"	Login gagal, sistem menampilkan pesan error "username tidak ditemukan"	**Pass**
TC -4	Login dengan password salah	Username: farel Password: qwerty	Login gagal, sistem menampilkan pesan error "Password salah"	Login gagal, sistem menampilkan pesan error "Password salah"	**Pass**
TC -5	Daftar rawat inap	Pilih dokter: Dokter 1 (jantung) Tanggal masuk: 03/06/2026 14.40	Sistem memberikan informasi pendaftaran berhasil	Sistem memberikan notifikasi pendaftaran berhasil	**Pass**
TC -6	Cek riwayat rawat inap	Klik menu "lihat riwayat rawat inap"	Menunjukkan daftar rawat inap yang telah didaftarkan	Menunjukkan daftar rawat inap yang telah didaftarkan	**Pass**
TC -7	Log out dari akun	Klik menu log out	Akun log out dan dialihkan ke tampilan login	Akun log out dan dialihkan ke tampilan login	**Pass**

Tabel *test case* berguna sebagai alur atau panduan untuk pengujian dengan *blackbox testing* dari awal sampai akhir, memudahkan perbandingkan ekpektasi dari fitur dengan hasil sebenarnya apakah sesuai atau tidak bahkan mungkin terjadi *bug* atau *error*.

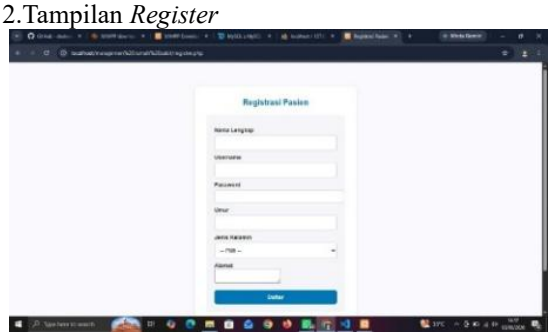
Hasil uji

1.Tampilan Awal



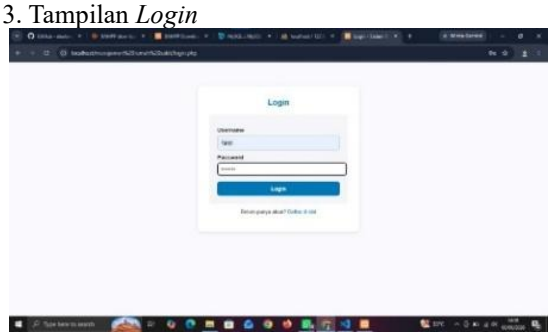
Gambar 5. Tampilan Awal Sistem Manajemen Rumah Sakit

2.Tampilan *Register*



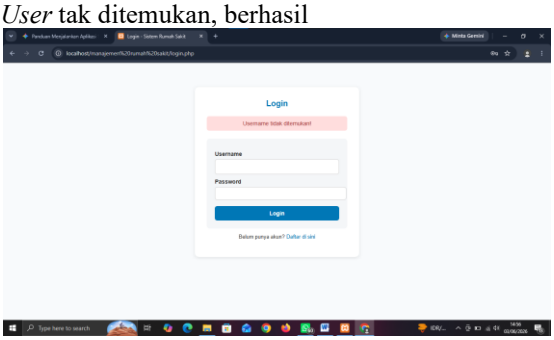
Gambar 6. Tampilan *Register* Sistem Manajemen Rumah Sakit

3. Tampilan *Login*



Gambar 7. Tampilan *Login* Sistem Manajemen Rumah Sakit

User tak ditemukan, berhasil



Gambar 8. *User* tak ditemukan

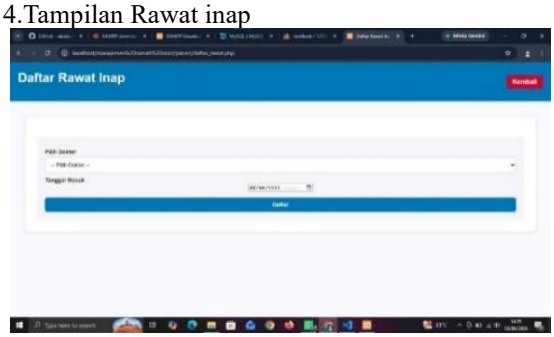
p-ISSN : 2406-7733
e-ISSN : 2597-9922

User password salah, Berhasil



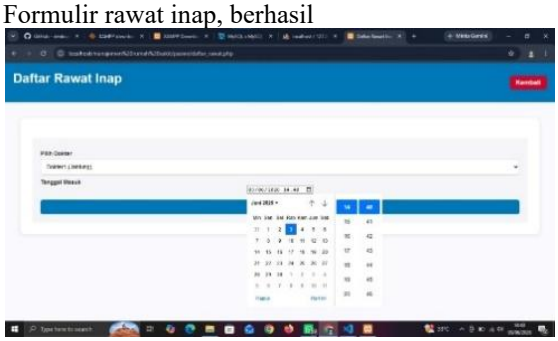
Gambar 9. *User Password* Salah

4.Tampilan Rawat inap



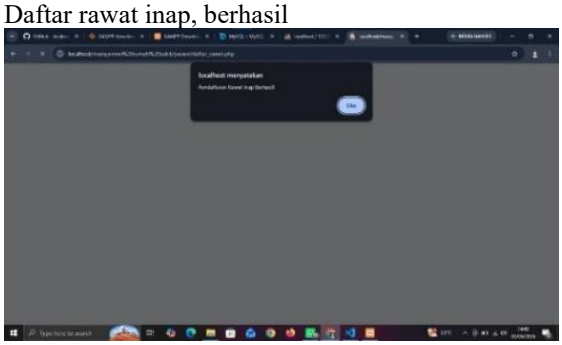
Gambar 10. Tampilan Rawat Inap Sistem Manajemen Rumah Sakit

Formulir rawat inap, berhasil



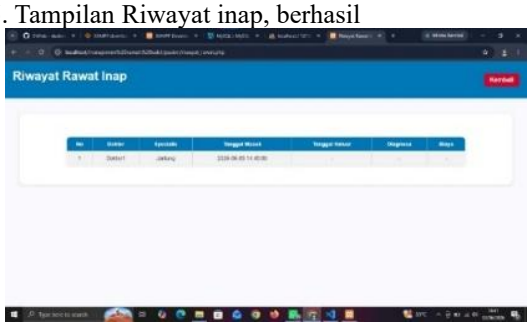
Gambar 11. Formulir Rawat Inap

Daftar rawat inap, berhasil



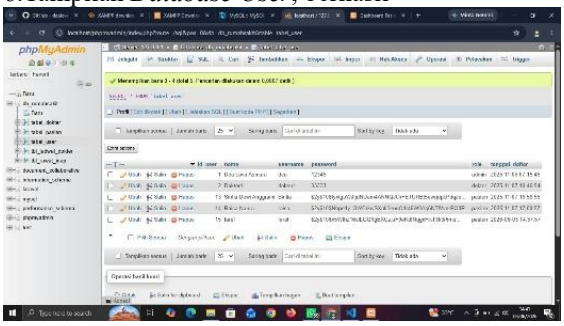
Gambar 12. Daftar Rawat Inap

5. Tampilan Riwayat inap, berhasil



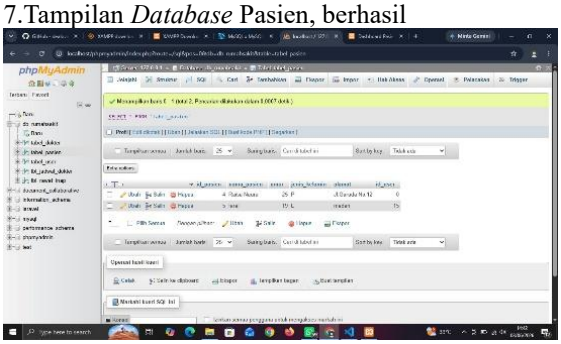
Gambar 13. Tampilan Riwayat Inap Sistem Manajemen Rumah Sakit

6.Tampilan *Database User*, berhasil



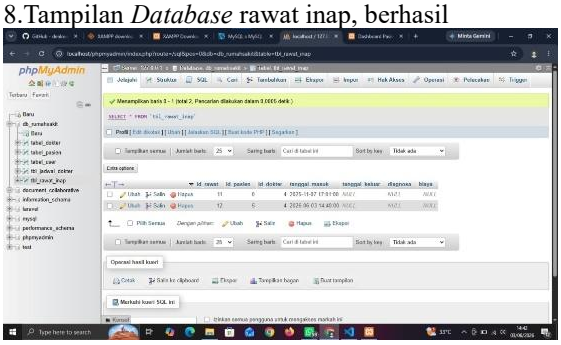
Gambar 14. Tampilan Database User Sistem Manajemen Rumah Sakit

7.Tampilan *Database* Pasien, berhasil



Gambar 15. Tampilan Database Pasien Sistem Manajemen Rumah Sakit

8.Tampilan *Database* rawat inap, berhasil



Gambar 16. Tampilan Database Rawat Inap Sistem Manajemen Rumah Sakit

Database pada *project* ini sangat berguna karena berguna untuk menyimpan data seperti data dokter, pasien, *user*, jadwal dokter, dan rawat inap

p-ISSN : 2406-7733
e-ISSN : 2597-9922

dan fitur seperti *user* atau *password* yang salah dapat berjalan dengan adanya data yang ada di *database*.

Jika tidak ada *database* maka semua data tadi hanya akan jadi data tetap dan tidak bisa ditambah maupun dihapus dan jika register akun maka akun tidak akan tersimpan dan ketika *logout* harus buat akun baru lagi untuk masuk, begitupun untuk data daftar rawat inap juga hilang dan tidak tersimpan.

Pembahasan

Berdasarkan hasil pengujian yang telah dilaksanakan, seluruh tujuh skenario *test case* (TC-1 hingga TC-7) menunjukkan status Pass, yang berarti keluaran aktual sistem sesuai dengan hasil yang diharapkan. Hal ini mengindikasikan bahwa fungsionalitas inti Sistem Manajemen Rumah Sakit telah berjalan dengan baik dan memenuhi kebutuhan pengguna sesuai spesifikasi yang telah ditentukan.

Pada pengujian modul autentikasi (TC-1, TC-2, TC-3, dan TC-4), sistem berhasil menangani berbagai kondisi input, baik yang valid maupun tidak valid. Fitur registrasi akun (TC-1) berhasil menyimpan data pengguna baru ke dalam basis data dengan benar. Fitur *login* (TC-2) berhasil mengautentikasi pengguna dan mengarahkannya ke halaman dashboard yang sesuai. Lebih lanjut, sistem juga terbukti mampu menampilkan pesan kesalahan yang informatif ketika pengguna memasukkan username yang tidak ditemukan (TC-3) maupun password yang salah (TC-4). Kemampuan validasi input ini merupakan aspek keamanan yang penting dalam sebuah sistem informasi kesehatan.

Pada pengujian modul rawat inap, fitur pendaftaran rawat inap (TC-5) berhasil memproses pemilihan dokter dan tanggal masuk dengan memberikan notifikasi keberhasilan kepada pengguna. Fitur riwayat rawat inap (TC-6) juga berhasil menampilkan daftar rekam perawatan pasien yang telah tersimpan di basis data. Kedua fitur ini merupakan komponen kritis dalam operasional rumah sakit karena berkaitan langsung dengan pelayanan dan keselamatan pasien.

Pengujian fitur *logout* (TC-7) membuktikan bahwa sistem mampu mengakhiri sesi pengguna dengan aman dan mengarahkan kembali ke halaman login. Fungsi ini penting untuk menjaga keamanan data, terutama mengingat sistem mengelola informasi sensitif terkait pasien dan rekam medis.

Dari sisi arsitektur, penggunaan basis data MySQL yang terpusat terbukti mendukung integritas data secara menyeluruh. Seluruh operasi seperti registrasi akun, pendaftaran rawat inap, dan pencatatan riwayat perawatan dapat dilakukan secara dinamis karena data tersimpan dan dapat diakses secara *real-time*. Tanpa dukungan basis data yang andal, berbagai fungsi kritis tersebut tidak dapat beroperasi dengan benar.

Hasil penelitian ini konsisten dengan temuan penelitian sebelumnya. Musyaffa dkk. (2024) yang menerapkan *Black Box Testing* pada SIMRS berbasis web di RSUD Otista Soreang juga melaporkan keberhasilan pengujian pada modul yang diuji, meski cakupannya masih terbatas. Penelitian ini memperluas cakupan tersebut dengan menguji delapan fitur yang mencakup seluruh aktor sistem. Syabrullah dkk. (2023) pun menunjukkan bahwa semakin luas cakupan skenario uji, semakin komprehensif evaluasi yang dapat dilakukan terhadap sistem.

Meskipun seluruh skenario menunjukkan hasil yang memuaskan, perlu diakui bahwa penelitian ini memiliki keterbatasan. Pengujian baru mencakup tujuh skenario dari delapan fitur yang diidentifikasi, dan belum menyertakan skenario untuk modul Admin secara lengkap, seperti pengelolaan data dokter, pengelolaan jadwal, dan laporan rawat inap. Selain itu, pengujian non-fungsional seperti uji performa, keamanan, dan skalabilitas sistem belum dilakukan dalam penelitian ini. Pengujian lebih lanjut dengan cakupan skenario yang lebih komprehensif direkomendasikan untuk memastikan kualitas sistem secara menyeluruh sebelum diimplementasikan di lingkungan rumah sakit yang sesungguhnya.

V. KESIMPULAN DAN SARAN

Kesimpulan

1. Seluruh tujuh skenario *test case* (TC-1 hingga TC-7) yang dieksekusi menunjukkan status berhasil, yang berarti semua fungsi utama sistem—meliputi registrasi akun, *login*, pendaftaran rawat inap, riwayat rawat inap, dan *logout*—telah berjalan sesuai dengan hasil yang diharapkan.
2. Metode *Black Box Testing* dengan teknik *Equivalence Partitioning* dan *Boundary Value Analysis* terbukti efektif dalam memvalidasi fungsionalitas sistem dari perspektif pengguna tanpa memerlukan pemahaman terhadap kode internal program. Metode ini mampu mendeteksi perilaku sistem pada kondisi input valid maupun tidak valid, termasuk keberhasilan sistem menampilkan pesan kesalahan yang informatif saat pengguna memasukkan kredensial yang salah.
3. Basis data MySQL yang terpusat terbukti mendukung integritas dan konsistensi data secara menyeluruh. Seluruh operasi dinamis seperti penyimpanan akun baru, pencatatan pendaftaran rawat inap, dan penelusuran riwayat perawatan dapat berjalan dengan baik berkat dukungan basis data tersebut.
4. Penelitian ini memiliki keterbatasan berupa belum tercakupnya pengujian pada seluruh skenario modul Admin, seperti pengelolaan data dokter, kelola jadwal dokter, dan laporan rawat inap. Selain itu, pengujian non-fungsional seperti uji keamanan, performa, dan

p-ISSN : 2406-7733
e-ISSN : 2597-9922

kompatibilitas antarmuka belum dilaksanakan dalam penelitian ini.

Saran

Berdasarkan temuan dan keterbatasan penelitian yang telah dilakukan, terdapat beberapa hal yang perlu diperhatikan untuk penelitian selanjutnya yaitu:

1. Cakupan pengujian perlu diperluas dengan menambahkan skenario *test case* untuk modul Admin secara lengkap, meliputi fitur pengelolaan data dokter, kelola jadwal dokter, serta laporan rawat inap.
2. Menambahkan pengujian non-fungsional seperti uji performa, uji keamanan, serta uji kompatibilitas pada berbagai perangkat dan peramban web guna memastikan kualitas sistem secara menyeluruh.
3. Penerapan metode *White Box Testing* atau *Grey Box Testing* secara paralel dapat dipertimbangkan untuk melengkapi *Black Box Testing,* sehingga kesalahan pada logika internal program yang tidak terdeteksi melalui *Black Box Testing* dapat diidentifikasi.
4. Setiap penambahan fitur lanjutan pada sistem seperti modul keuangan atau integrasi sistem farmasi perlu disertai perancangan skenario pengujian yang komprehensif sebelum diimplementasikan.
5. Dokumentasi hasil pengujian sebaiknya mengacu pada standar yang lebih formal seperti IEEE 829-2008 untuk meningkatkan keandalan laporan sebagai bahan evaluasi bagi pengembang sistem.

DAFTAR PUSTAKA

Abdillah, R., Hermawan, R., Hermawansyah, W., Adkha, I., & Arifin, H. (2024). Pengujian Perangkat Lunak Sistem Informasi Inventori pada Usaha Jasa Pengiriman Paket. *Polygon : Jurnal Ilmu Komputer Dan Ilmu Pengetahuan Alam*, *2*(4), 166–175. https://doi.org/10.62383/polygon.v2i4.199

Abdillah, R., Hermawan, R., Hermawansyah, W., & Agustin, D. P. (2025). Analisis dan Pengujian Perangkat Lunak Sistem Informasi Pembayaran Sekolah dengan Metode Pengukuran Kualitas SQuaRE. *Jurnal Penelitian Rumpun Ilmu Teknik*, *4*(2), 36–45.

Fauzi, A., & Yunial, A. H. (2025). Testing dan QA Perangkat Lunak: Memahami Konsep, Metode, Teknik, dan Pendekatan dalam Pengujian. In *Eureka Media Aksara* (pp. 1–221).

Hasibuan, A. N., & Dirgahayu, T. (2021). Pengujian dengan Unit Testing dan Test case pada Proyek Pengembangan Modul Manajemen Pengguna. *Automata (Ajang Unjuk Tugas Akhir Oleh Mahasiswa Informatika)*.

Luginawati, S., & Wahyu, A. P. (2023). Pengujian Perangkat Lunak Menggunakan Metode Regression Testing Pada Aplikasi Pembelajaran Matematika Untuk Siswa Tingkat SMP Berbasis Android. *Jurnal Syntax Idea*, *53*(1), 1–19.

Musyaffa, I. F., Prastyo, N. A., Mahendra, R., Pradifta, A., Wicaksana, & Susanto. (2024). Pengujian Black Box pada Aplikasi SIMRS Berbasis Web di RSUD Otista Soreang Kabupaten Bandung. *Seminar Nasional Inovasi Dan Tren Teknologi (SINATTI)*, 375–380.

Nurfaidah, Hafizha, Y. C., & Yeni, H. (2025). Transformasi Efisiensi Layanan Kesehatan dengan Sistem Informasi Manajemen Rumah Sakit (SIMRS) di RSUD Provinsi Sulawesi Barat. *Jurnal Ekonomi Manajemen Sistem Informasi*, *6*(3), 1929–1941. https://doi.org/10.38035/jemsi.v6i3.4072

Pratama, I. F., & Purwanto, E. (2023). Sistem Informasi Manajemen Rumah Sakit Dalam Meningkatkan Efisiensi. *COMSERVA : Jurnal Penelitian Dan Pengabdian Masyarakat*, *3*(07), 2571–2576. https://doi.org/10.59141/comserva.v3i07.1044

Sabirin, & Widyarto. (n.d.). Metode Blackbox Dalam Pengujian Sistem Penjadwalan Janji Klinik / Dokter Berbasis. *Proceeding of the Informatics Conference*, *10*(21), 37–42.

Sitompul, I. S. D. (2023). Studi Literatur: Sistem Informasi Manajemen Rumah Sakit. *Journal of Healthcare Technology and Medicine*, *9*(1), 2615–109.

Syabrullah, A., Fibrian, I. D., & Syaikhuddin, M. M. (2023). Perancangan Sistem Informasi Manajemen Rumah sakit berbasis web. *Jurnal Teknik Informatika*, *8*(3). https://doi.org/10.35793/jti.6.1.2015.9968

Syihab, L. N., Janniethia, & Sugiarti, Y. (2024). Pengujian Strategi Perangkat Lunak: Tinjauan Literatur Sistematis. *Informatech : Jurnal Ilmiah Informatika Dan Komputer*, *1*(2), 112–118. https://doi.org/10.69533/agmegm93

Tonggiroh, M., Pardosi, V. B. A., Basiroh, & Nugroho, F. (2024). Rakayasa Perangkat Lunak. In *PT. Mafy Media Literasi Indoneisa* (pp. 1–97).

Utami, D. F. N., & Setyodewi, R. H. (2023). Dokumentasi Software Testing Untuk Aplikasi Dafbin Berstandar IEEE 829-2008. *Jurnal Restikom : Riset Teknik Informatika Dan Komputer*, *5*(2), 107–117.

Wicaksono, S. R. (2023). Pengujian Perangkat Lunak: Strategi, Metode, Dan Implementasi. In *CV. Seribu Bintang* (p. 226). CV. Seribu Bintang.