

# Jurnal JTik (Jurnal Teknologi Informasi dan Komunikasi)

DOI: <https://doi.org/10.35870/jtik.v10i2.5477>

## Pengujian Kualitas Sistem Informasi Kesehatan Periksa.ID Berbasis Autentikasi Menggunakan *Software Testing Life Cycle*

Andini Dwi Salshabila <sup>1\*</sup>, Imam Sutanto <sup>2</sup>, Hosizah <sup>3</sup>, Tria Saras Pertiwi <sup>4</sup>

<sup>1\*2,3,4</sup> Program Studi Terapan Manajemen Informasi, Fakultas Ilmu Kesehatan, Universitas Esa Unggul, Kota Jakarta Barat, Daerah Khusus Ibukota Jakarta, Indonesia.

### article info

#### Article history:

Received 2 September 2025

Received in revised form

10 September 2025

Accepted 20 October 2025

Available online April 2026.

#### Keywords:

Software Testing Life Cycle;  
Periksa.id; Quality Assurance.

### abstract

The advancement of information technology in the healthcare sector encourages hospitals to implement reliable and secure digital systems, one of which is the Periksa.id application. This application is used by doctors to record inpatient examination results and is now equipped with a fingerprint authentication feature. This study aims to evaluate the quality of the Periksa.id application using the Software Testing Life Cycle (STLC) method, which consists of six stages: requirement analysis, test planning, test case design, test environment setup, test execution, and test closure. The research was conducted at Soeharto Heerdjan Hospital over a period of three months with a total of 200 virtual users involved in the reliability testing. Testing was carried out on both functional and non-functional aspects, including response time, reliability, security, portability, as well as positive and negative testing, using tools such as JMeter, WAPT, and ZAP. The results showed that the application met over 85% of the software quality criteria, although several minor bugs and 12 security vulnerabilities (2 high, 10 medium) were identified and require improvement. This study contributes to the evaluation of web-based hospital application systems to enhance the efficiency of medical services and ensure secure patient data management.

### abstract

Perkembangan teknologi informasi dalam dunia kesehatan mendorong rumah sakit untuk menerapkan sistem digital yang andal dan aman, salah satunya melalui aplikasi Periksa.id. Aplikasi ini digunakan oleh dokter untuk mencatat hasil pemeriksaan pasien rawat inap dan kini dilengkapi fitur autentikasi sidik jari. Penelitian ini bertujuan untuk menguji kualitas aplikasi Periksa.id menggunakan metode Software Testing Life Cycle (STLC) yang terdiri dari enam tahapan: requirement analysis, test planning, test case design, test environment setup, test execution, dan test closure. Penelitian dilakukan di Rumah Sakit Soeharto Heerdjan selama tiga bulan dengan melibatkan 200 pengguna virtual pada uji reliabilitas. Pengujian dilakukan secara fungsional dan non-fungsional, termasuk uji respons waktu, reliabilitas, keamanan, portabilitas, serta pengujian positif dan negatif menggunakan tools seperti JMeter, WAPT, dan ZAP. Hasil pengujian menunjukkan bahwa aplikasi memenuhi lebih dari 85% kriteria kualitas perangkat lunak, meskipun ditemukan beberapa bug minor serta 12 kerentanan keamanan (2 tingkat tinggi, 10 tingkat sedang) yang perlu diperbaiki. Penelitian ini memberikan kontribusi dalam evaluasi kualitas sistem aplikasi rumah sakit berbasis web untuk meningkatkan efisiensi layanan medis dan menjaga keamanan data pasien.

\*Corresponding Author. Email: andinisalsha@gmail.com <sup>1\*</sup>.



Copyright 2026 by the authors of this article. Published by Lembaga Otonom Lembaga Informasi dan Riset Indonesia (KITA INFO dan Riset). This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

## 1. Pendahuluan

Kemajuan teknologi digital telah mendorong berbagai sektor industri, termasuk kesehatan, untuk memanfaatkan sistem informasi guna meningkatkan kualitas layanan. Di Indonesia, rumah sakit telah mulai mengadopsi teknologi digital, meskipun tantangan yang dihadapi masih cukup besar. Aisyah *et al.* (2024) mencatat bahwa tingkat kematangan Teknologi Informasi dan Komunikasi (TIK) pada layanan kesehatan primer di sembilan provinsi baru mencapai rata-rata 2,74 dari skala 5. Sebanyak 62% fasilitas kesehatan masih kekurangan sistem informasi yang memadai, dengan masalah utama terletak pada literasi SDM (2,71), infrastruktur (2,84), dan perangkat keras (2,59). Transformasi digital ini mendukung visi pemerintah untuk menciptakan layanan kesehatan yang lebih efisien, sesuai dengan amanat UU No. 44 Tahun 2009 tentang Rumah Sakit. Akan tetapi, implementasi sistem informasi kesehatan tidak hanya menghadapi tantangan teknis. Handayani *et al.* (2020) menyoroti bahwa faktor organisasi, sumber daya manusia, dan lingkungan juga berperan besar dalam kesuksesan implementasi. Hambatan serupa juga ditemukan oleh Afrizal *et al.* (2019), yang mencatat adanya resistensi pengguna, keterbatasan infrastruktur, serta minimnya dukungan manajemen.

Widiyanto (2021) mengungkapkan bahwa 54% dari masalah yang muncul dalam penerapan Sistem Informasi Manajemen Rumah Sakit (SIMRS) di rumah sakit publik disebabkan oleh kualitas perangkat lunak yang buruk, seperti adanya bug dan kesalahan sistem. Secara global, kualitas sistem informasi terbukti menjadi faktor utama dalam menentukan keberhasilan implementasi. Ebnehoseini *et al.* (2019) mengembangkan instrumen pengukuran yang menekankan pentingnya kualitas sistem, sedangkan Atashi *et al.* (2016) mengidentifikasi 56 masalah antarmuka dalam sistem rawat inap nasional, yang menunjukkan perlunya pengujian yang lebih mendalam. Menurut standar ISO/IEC 25010:2023, kualitas perangkat lunak dapat diukur melalui delapan aspek utama, seperti fungsionalitas, keandalan, keamanan, dan portabilitas. Beberapa penelitian lain juga menunjukkan manfaat pengujian sistem yang sistematis. Tohirin *et al.* (2019) menggunakan model

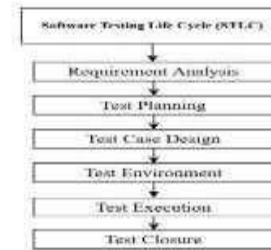
ISO/IEC 9126 untuk menemukan kelemahan yang tidak terdeteksi oleh metode konvensional. Meski demikian, masih terdapat keterbatasan dalam penelitian-penelitian tersebut. Misalnya, Sukums *et al.* (2015) hanya mengandalkan uji penerimaan pengguna (UAT), sementara Yoshiura *et al.* (2017) menggunakan uji ad-hoc tanpa dokumentasi yang memadai. Chukwu *et al.* (2022) menyoroti kegagalan pasca-implementasi akibat pengabaian infrastruktur, sedangkan Odekunle *et al.* (2017) menyatakan bahwa lemahnya pengujian sistematis merupakan salah satu penyebab rendahnya adopsi rekam kesehatan elektronik di Afrika Sub-Sahara. Sayyadi Tooranloo *et al.* (2021) juga mengaitkan kegagalan SIMRS dengan pengujian yang tidak memadai serta kurangnya dokumentasi. Kesenjangan dalam penelitian masih terlihat jelas, di mana sebagian besar uji hanya dilakukan secara parsial, dokumentasi terbatas, dan belum banyak penelitian di Indonesia yang fokus pada sistem kesehatan berbasis biometrik. Oleh karena itu, penelitian ini mengadopsi pendekatan Software Testing Life Cycle (STLC) yang mencakup seluruh tahapan, dari analisis kebutuhan hingga penutupan pengujian, dengan dokumentasi yang lengkap. Berbeda dengan Black Box Testing atau UAT yang bersifat parsial, STLC menawarkan pendekatan yang lebih terstruktur dan dapat diterapkan pada sistem yang sudah beroperasi.

Rumah Sakit Soeharto Heerdjan, sebagai salah satu institusi kesehatan terkemuka di Jakarta Barat, telah mengimplementasikan sistem informasi berbasis web bernama Periksa.id untuk mendukung operasionalnya. Aplikasi Periksa.id digunakan oleh tenaga medis untuk menginput Catatan Perkembangan Pasien Terintegrasi (CPPT) setelah pemeriksaan pasien rawat inap. Seiring dengan regulasi terbaru dari Kementerian Kesehatan, yang tertuang dalam surat Direktorat TK.04.01/D.IV/2592/2024, kini diwajibkan bagi Dokter Penanggung Jawab (DPJP) untuk melakukan rekam kehadiran pada ruang pelayanan rawat inap menggunakan autentikasi rekam medis berbasis biometrik. Implementasi ini mengharuskan penggunaan sidik jari (fingerprint) saat menginput CPPT ke dalam sistem Periksa.id. Oleh karena itu, Rumah Sakit Soeharto Heerdjan melakukan pengembangan signifikan pada sistem aplikasi tersebut. Aplikasi Periksa.id dikelola oleh vendor

Periksa Solusi Indonesia dan telah digunakan sejak 2019 untuk mendukung kegiatan operasional rumah sakit. Implementasi fitur autentikasi biometrik sidik jari baru dimulai pada 11 Mei 2025 dan hingga saat ini masih beroperasi. Sebelum fitur ini diterapkan secara luas, staf IT Rumah Sakit telah melakukan pengujian awal menggunakan metode Black Box Testing. Meskipun pengujian tersebut bersifat asesmen dan tidak mendalam, hasilnya menunjukkan adanya beberapa bug dan kesalahan pada aplikasi Periksa.id yang perlu segera ditangani. Penelitian ini bertujuan untuk membantu Rumah Sakit Soeharto Heerdjan memastikan kualitas aplikasi Periksa.id melalui metode STLC. Penelitian ini juga memberikan kontribusi dalam melakukan pengujian kualitas perangkat lunak pada sistem Aplikasi Periksa.id, dengan mencakup kerangka uji yang sistematis untuk sistem biometrik yang dapat diterapkan kembali. Penelitian ini akan mengidentifikasi bug secara komprehensif, memberikan rekomendasi perbaikan, serta mendokumentasikan praktik terbaik STLC dalam konteks SIMRS di Indonesia. Tujuan utama dari penelitian ini adalah untuk memastikan implementasi fitur sidik jari berjalan dengan lancar, dan sistem secara keseluruhan memenuhi standar kualitas yang diharapkan, sehingga mendukung efisiensi dan keamanan layanan medis di Rumah Sakit.

## 2. Metodologi Penelitian

Penelitian ini menggunakan pendekatan kualitatif deskriptif dengan studi kasus (*case study research*) yang dilakukan di Rumah Sakit Soeharto Heerdjan. Tujuan utama dari penelitian ini adalah untuk melakukan evaluasi terhadap kualitas aplikasi Periksa.id, khususnya dalam implementasi fitur autentikasi sidik jari (*fingerprint*), dengan menggunakan metode *Software Testing Life Cycle (STLC)*. Metode *STLC* terdiri dari enam tahapan utama, yaitu: analisis kebutuhan (*requirement analysis*), perencanaan pengujian (*test planning*), perancangan kasus uji (*test case design*), pengaturan lingkungan pengujian (*test environment setup*), pelaksanaan pengujian (*test execution*), dan penutupan pengujian (*test closure*).



Gambar 1. Diagram *Software Testing Life Cycle (STLC)*

### **Requirement Analysis**

Tahap awal ini dimulai dengan mengenali kriteria pengujian perangkat lunak dan menganalisis sistem yang akan diuji. Kriteria ini mencakup berbagai spesifikasi penting, seperti kebutuhan fungsional, performa sistem, keamanan aplikasi, dan kemudahan penggunaan. Semua aspek ini perlu dipahami agar pengujian dilakukan secara menyeluruh dan sesuai dengan tujuan sistem.

### **Validitas dan Reliabilitas Instrumen**

*Requirement Traceability Matrix (RTM)* menjamin validitas dengan menguji semua kebutuhan fungsional dan non-fungsional. Uji beban 200 pengguna virtual di *JMeter* yang dijalankan berulang menghasilkan konsistensi.

### **Keterbatasan dan Potensi Bias**

Studi ini terbatas pada satu rumah sakit dengan konfigurasi tertentu, uji performa yang mencakup hingga 200 pengguna virtual, dan analisis keamanan yang terbatas pada deteksi *ZAP* otomatis tanpa uji penetrasi manual.

### **Test Planning**

Pada tahap ini, proses pengujian mulai dirancang dengan menyiapkan alat dan sarana yang dibutuhkan. Selain itu, ditentukan juga ruang lingkup pengujian, tujuan pengujian, dan metode atau teknik pengujian yang akan digunakan. Pemilihan alat uji juga disesuaikan dengan jenis pengujian yang akan dilakukan.

### **Kriteria Pemilihan Alat Pengujian**

Menurut *ISO/IEC 25010*, *JMeter* digunakan untuk menguji reliabilitas dan beban, *WAPT* untuk menguji performa karena detail waktu respons dan transaksi, dan *OWASP ZAP* digunakan untuk menguji keamanan karena mampu mendeteksi kerentanan web dengan baik (*OWASP Top 10*).

## Aspek Etika Penelitian

Untuk menjaga kerahasiaan dan privasi, pengujian dilakukan dengan akun dummy dari tim IT rumah sakit. Ini melibatkan data simulasi tanpa data pasien asli.

## Test Case Design

Fase ini berfokus pada pembangunan skenario pengujian dan penentuan prioritas pengujian. Peneliti akan mengidentifikasi berbagai situasi atau kondisi yang perlu diuji, lalu menyusun langkah-langkah pengujian (*test script*) berdasarkan kebutuhan sistem.

### Test Environment

Tahap ini mencakup pengaturan berbagai elemen teknis yang diperlukan untuk menjalankan pengujian, seperti pengaturan protokol jaringan, alamat IP, URL aplikasi, serta gateway.

### *Test Execution*

Setelah seluruh persiapan selesai, pengujian mulai dilakukan. Setiap *test case* dijalankan sesuai dengan rencana, dan hasilnya dicatat untuk dianalisis lebih lanjut.

### *Test Closure*

Setelah semua pengujian dilakukan, tahap selanjutnya adalah mendokumentasikan setiap temuan, baik berupa kesalahan sistem maupun bug. Laporan ini mencakup penjelasan detail mengenai masalah yang ditemukan.

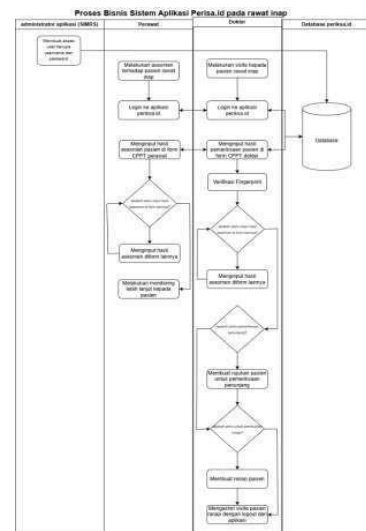
belum dilakukan secara mendalam, sehingga mereka membutuhkan informasi lebih detail mengenai kecepatan respons sistem, ketahanan saat diakses oleh banyak pengguna, keamanan terhadap potensi ancaman, fleksibilitas di berbagai perangkat, kegunaan antarmuka, serta respons sistem terhadap data yang tidak valid. Selain itu, pada tahap ini, dilakukan observasi mendalam untuk memahami alur proses bisnis aplikasi Periksa.id. Observasi ini menggambarkan bagaimana sistem bekerja dari awal hingga akhir, termasuk interaksi antara dokter dan pasien rawat inap. Diagram proses bisnis yang berhasil disusun memberikan gambaran visual yang jelas mengenai setiap langkah yang terlibat dalam proses input CPPT, mulai dari dokter yang mengakses sistem hingga data berhasil disimpan. Diagram ini sangat berguna dalam merancang skenario pengujian yang mencakup seluruh alur kerja. *Use Case Diagram* juga disusun untuk menampilkan interaksi antara dokter sebagai pengguna aplikasi dan pasien rawat inap sebagai objek data. Diagram ini menunjukkan dengan jelas fungsi-fungsi yang dapat diakses oleh dokter, seperti mengisi formulir, memindai sidik jari, dan menyimpan data. *Use Case Diagram* ini membantu dalam mengidentifikasi fungsionalitas utama yang harus diuji dan memastikan bahwa setiap fitur yang penting telah tercakup dalam skenario pengujian. Selain itu, dokumen *Requirement Traceability Matrix* (RTM) disusun untuk memastikan bahwa seluruh kebutuhan yang telah dianalisis benar-benar diuji secara lengkap.

### 3. Hasil dan Pembahasan

## Hasil

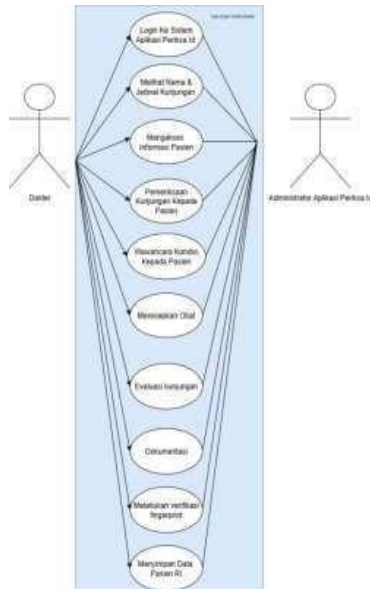
## Requirements Analysis

Tahap *Requirements Analysis* dalam penelitian ini berhasil mengidentifikasi dan mendokumentasikan kebutuhan pengujian aplikasi Periksa.id, dengan fokus utama pada halaman input Catatan Perkembangan Pasien Terintegrasi (*CPPT*). Berdasarkan hasil wawancara dengan staf IT Rumah Sakit Soeharto Heerdjan, diketahui bahwa pengujian perlu difokuskan pada halaman input *CPPT*, karena di sinilah fitur autentikasi sidik jari diimplementasikan. Staf IT juga mengungkapkan bahwa pengujian yang dilakukan sebelumnya menggunakan metode *blackbox* bersifat asesmen dan



Gambar 2. Proses Bisnis Aplikasi Periksa.Id





Gambar 3. Use Case Diagram Aplikasi Periksa.Id

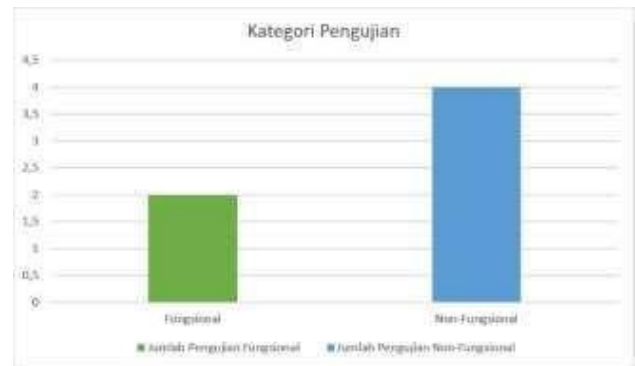


Gambar 4. Network/System Architecture Diagram

### Test Planning

Pada tahap *Test Planning*, peneliti berhasil menyusun rencana pengujian yang terstruktur dan komprehensif. Rencana ini disusun berdasarkan kebutuhan yang telah diidentifikasi pada tahap *Requirement Analysis*. Rencana pengujian mencakup berbagai jenis pengujian, komponen yang akan diuji, pendamping pengujian, dan jadwal pelaksanaan yang rinci. Secara keseluruhan, pengujian dibagi menjadi dua kategori utama, yaitu pengujian fungsional dan non-fungsional. Pengujian fungsional mencakup *Positive Testing* dan *Negative Testing* untuk memastikan bahwa sistem dapat memproses input yang valid dan menolak input yang tidak valid pada kolom input data pasien, serta memverifikasi sidik jari. Selain itu, pada pengujian non-fungsional, dilakukan *Performance Testing* untuk mengukur waktu respons aplikasi, *Reliability Testing* untuk mengevaluasi stabilitas aplikasi ketika diakses oleh banyak pengguna, dan

*Security Testing* untuk menguji ketahanan aplikasi terhadap potensi ancaman. *Portability Testing* juga dilaksanakan untuk memverifikasi kompatibilitas tampilan aplikasi di berbagai perangkat dan browser. Seluruh pengujian ini dilaksanakan dengan pendampingan dari staf IT rumah sakit.



Gambar 5. Test Planning Results Chart

### Test Case Design

Tahap *Test Case Design* merupakan kelanjutan dari *Test Planning*, di mana peneliti merancang skenario pengujian yang rinci dan terukur. Skenario ini disusun berdasarkan *Requirement Traceability Matrix* (RTM) dan *Test Planning* yang telah disusun sebelumnya, serta hasil diskusi dengan staf IT. Secara keseluruhan, terdapat 9 skenario pengujian yang dirancang, yang terdiri dari 3 *test case* untuk pengujian positif, 2 untuk pengujian negatif, dan 4 untuk pengujian non-fungsional. Setiap *test case* dilengkapi dengan ID yang berurutan, nama *test case*, kebutuhan terkait, prasyarat, langkah-langkah pengujian, jenis pengujian, status, dan hasil yang diharapkan. Tujuan dari *test case design* ini adalah untuk memastikan bahwa setiap kebutuhan (*requirement*) dapat diuji secara sistematis, meminimalkan kesalahan selama eksekusi, serta memudahkan pelacakan jika terjadi kegagalan (*defect*).



Gambar 6. Test Case Design Results Chart

### Test Environment Setup

Pada tahap *Test Environment Setup*, peneliti mempersiapkan lingkungan yang diperlukan untuk melaksanakan pengujian dengan lancar. Lingkungan

ini mencakup perangkat keras dan perangkat lunak yang spesifik. Perangkat keras yang digunakan adalah:

Tabel 1. Test Environment Halaman Input CPPT

Komponen	Spesifikasi / Detail
Aplikasi	Periksa.Id
Jenis Aplikasi	<i>Website Application</i>
Tipe Pengujian	<i>Performance Testing, Reability Testing, UI Testing, Negative Testing</i>
Alat Pengujian ( <i>Software</i> )	Jmeter 5.6.3, Java 24.0.2, WAPT 10.1, ZAP 2.16.1
URL Pengujian	<a href="http://rsjsh.periksa.id/index.php/">http://rsjsh.periksa.id/index.php/</a>
Browser	<i>Chrome, Firefox, Edge</i>
Sistem Operasi	Windows 11

Untuk perangkat lunak, peneliti menggunakan akun yang dibuat khusus oleh tim IT Rumah Sakit dengan nama "Dokter Testing". Akun ini digunakan untuk mengakses dan menguji aplikasi. Selain itu, alat pengujian yang relevan juga dipersiapkan, yaitu:

- 1) *JMeter* 5.6.4 untuk *Reliability Testing*
- 2) *WAPT* 10.1 untuk *Performance Testing*
- 3) *ZAP* 2.16.1 untuk *Security Testing*



Gambar 7. Halaman Login Aplikasi Periksa.Id

Pada gambar 7 terlihat aplikasi berhasil diakses menggunakan jaringan rumah sakit.

```
C:\Users\Andini>java --version
java 24.0.2 2025-07-15
Java(TM) SE Runtime Environment (build 24.0.2+12-54)
Java HotSpot(TM) 64-Bit Server VM (build 24.0.2+12-54, mixed mode,
```

Gambar 8. Java

Pada gambar 8 Java yang digunakan adalah versi 24.0.2, Modul ini di instalasi agar Aplikasi Jmeter dapat berjalan. JDK 24, dapat di *download* di situs resmi milik Oracle.



Gambar 9. Versi Jmeter

Pada gambar 9 Java yang digunakan adalah versi 24.0.2, Modul ini di instalasi agar Aplikasi Jmeter dapat berjalan. JDK 24, dapat di *download* di situs resmi milik Oracle.



Gambar 10. Versi WAPT

Gambar 10 diatas merupakan versi dari aplikasi WAPT yaitu 10.1 yang sudah berhasil terinstalasi di laptop peneliti. Aplikasi ini membantu dalam proses *Performance Testing*.



Gambar 11. Versi ZAP

Pada gambar 11 peneliti menggunakan ZAP versi 2.16.1 yang sudah berhasil terinstalasi di laptop peneliti untuk membantu melakukan *testing*.



Gambar 12. Spesifikasi Laptop Pengujian

Pada gambar 12 terlihat spesifikasi detail dari laptop pribadi peneliti yang digunakan selama proses pengujian.



Gambar 13. Alat Fingerprint U Are U 4500 SDK Only

Sensor sidik jari yang digunakan memiliki resolusi 512 dpi dengan area tangkap 14,6 mm x 18,1 mm dan menghasilkan citra 8-bit grayscale (256 tingkat keabuan). Perangkat ini mendukung USB 1.0, 1.1, dan 2.0 dengan waktu identifikasi kurang dari 1 detik, serta dilengkapi dengan enkripsi data 128-bit untuk menjaga keamanan. Ukuran fisiknya adalah 79 x 49 x

19 mm dan telah memenuhi standar industri *FCC Class B, CE*, serta *VCCI*. Sensor ini mampu membaca sidik jari hingga 360 posisi berbeda, termasuk pada kondisi jari kering, berminyak, atau sedikit rusak, serta dapat menolak bekas sidik jari sebelumnya pada kaca sensor. Dukungan *SDK* tersedia untuk berbagai bahasa pemrograman (*VB6, Delphi, VB.Net, VC++, Java, Linux*). Harga unit baru perangkat ini di pasaran sekitar Rp1.000.000,00, sehingga cukup terjangkau untuk kebutuhan penelitian maupun implementasi autentikasi biometrik.

### Test Execution

Pada tahap *Test Execution*, semua skenario pengujian yang telah dirancang dieksekusi pada lingkungan pengujian yang telah disiapkan. Pengujian dimulai dengan pengujian fungsionalitas positif dan negatif.

- 1) Pengujian fungsionalitas positif: Memastikan bahwa semua kolom dapat diisi, data dapat disimpan, dan verifikasi sidik jari berhasil.
- 2) Pengujian fungsionalitas negatif: Memastikan sistem memberikan peringatan yang sesuai ketika data yang tidak valid dimasukkan atau ketika verifikasi sidik jari gagal. Ditemukan bug minor di mana sistem dapat menyimpan data kosong ke dalam riwayat kunjungan, meskipun data tersebut dapat dihapus.

Untuk pengujian non-fungsional, hasilnya adalah sebagai berikut:

- a) *Performance Testing*: Menggunakan *WAPT*, didapatkan *Summary Report* dan *Performance Report* yang menunjukkan performa aplikasi. Waktu respons kurang dari 5 detik, namun tidak stabil, kemungkinan disebabkan oleh kelemahan pada konektivitas antara server aplikasi dan server database.



Gambar 14. Performance Testing Results Graph

Pada *Gambar 14*, terlihat bahwa pada 1 menit pertama, *response time* stabil, namun setelah melewati 1 menit, *response time* mulai tidak stabil hingga mencapai angka 30 detik. Setelah itu, pada 30 detik terakhir, *response time* kembali stabil seperti pada awal pengujian.

- b) *Reliability Testing*: Menggunakan *JMeter* dengan 200 *virtual users*, hasil yang diperoleh menunjukkan bahwa aplikasi mampu menangani beban tanpa error, dengan tingkat reliabilitas 100%.

**Rata - Rata Hasil Reliability Testing**

Label	Total Virtual User	Standar Deviasi	Throughput	Error Rate
HTTP Request	200 Virtual User	50 ms	7.9/sec	0,00%

Gambar 15. *Reliability Testing Summary Results Table*

Berdasarkan *Gambar 15*, aplikasi *Periksa.id* mampu menangani beban kerja dengan reliabilitas tinggi dan *response time* yang optimal untuk pengalaman pengguna yang baik, dengan tingkat *error rate* 0,00% dan *standard deviation* yang rendah.

- c) *Portability Testing*: Aplikasi berhasil diakses dan berfungsi dengan normal di browser *Microsoft Edge*, *Google Chrome*, dan *Mozilla Firefox*.
- d) *Security Testing*: Menggunakan *OWASP ZAP*, ditemukan 10 kerentanan keamanan yang tergolong *Minor*, seperti *Information Exposure* dan *Cross-Domain JavaScript Source File Inclusion*.



Gambar 16. *Security Testing Results Chart*

Pada *Gambar 16*, terlihat terdapat 2 kerentanan dengan tingkat keparahan *high*, diikuti oleh 10 kerentanan dengan tingkat keparahan *medium*, 9 kerentanan dengan tingkat keparahan *low*, serta 9 kerentanan dengan tingkat keparahan *informational*.

## Test Closure

Pada tahap *Test Closure*, semua pengujian yang telah dilakukan disimpulkan dan didokumentasikan. Ringkasan hasil eksekusi pengujian menunjukkan bahwa semua *test case* berhasil dieksekusi tanpa ada yang gagal.

## Pembahasan

Hasil pengujian kualitas aplikasi *Periksa.id* menggunakan metode *Software Testing Life Cycle (STLC)* berdasarkan temuan yang telah disajikan di bab sebelumnya. Analisis ini bertujuan untuk menentukan apakah aplikasi *Periksa.id* telah memenuhi karakteristik sebuah aplikasi web yang berkualitas. Dari tahap awal pengujian, yaitu *Requirement Analysis*, peneliti berhasil mengidentifikasi fungsi utama aplikasi, khususnya pada halaman input *Catatan Perkembangan Pasien Terintegrasi (CPPT)*. Temuan ini sejalan dengan penelitian yang dilakukan oleh Handayani *et al.* (2020), yang menekankan pentingnya pemahaman kebutuhan sistem untuk merancang pengujian yang komprehensif. Informasi yang diperoleh mencakup alur proses bisnis dan *Use Case Diagram* yang menggambarkan interaksi antara dokter dan pasien. Pada tahap ini, peneliti juga melakukan observasi terhadap arsitektur sistem dan *Standard Operating Procedure (SOP)* rumah sakit, yang sangat membantu dalam menentukan target pengujian yang spesifik.

Sebagaimana dicatat oleh Widiyanto (2021), pemahaman terhadap infrastruktur dan prosedur rumah sakit sangat berperan dalam merancang pengujian yang akurat dan relevan. *Requirement Traceability Matrix (RTM)* yang dibuat berfungsi sebagai alat penting untuk memastikan bahwa setiap kebutuhan yang dianalisis akan diuji secara menyeluruh, sesuai dengan pendekatan yang diusulkan oleh Tohirin *et al.* (2019), yang menggunakan matriks untuk menghindari pengujian yang tidak terstruktur. Pada tahap *Test Planning*, peneliti berhasil menyusun rencana pengujian yang mencakup jenis pengujian, komponen, pendamping pengujian, dan jadwal pelaksanaan yang rinci. Rencana ini menjadi panduan yang jelas untuk seluruh proses pengujian, memastikan bahwa setiap aspek yang penting telah terencana dengan baik, mirip dengan pendekatan yang digunakan oleh Sukums *et al.* (2015) yang mengutamakan perencanaan pengujian yang



terstruktur dan sistematis. Pada tahap *Test Case Design*, skenario pengujian yang dirancang berdasarkan RTM dan hasil diskusi dengan staf IT terbukti efektif. Dengan adanya 9 skenario pengujian, yang meliputi pengujian fungsional dan non-fungsional, peneliti dapat melakukan pengujian secara terstruktur dan sistematis, yang sejalan dengan praktik terbaik yang disarankan oleh Chukwu *et al.* (2022) dalam melakukan pengujian berbasis kasus. Setiap *test case* memiliki prasyarat dan langkah-langkah yang jelas, yang membantu meminimalkan kesalahan selama eksekusi, serta memudahkan pelacakan jika terjadi kegagalan (*defect*). Tahap *Test Environment Setup* juga berjalan lancar, di mana semua perangkat keras, perangkat lunak, dan *tools* pengujian dipersiapkan dengan baik, seperti yang dijelaskan dalam penelitian oleh Ebnehoseini *et al.* (2019) mengenai pentingnya pengaturan lingkungan pengujian yang tepat. Secara keseluruhan, tahap *Test Execution* menunjukkan bahwa aplikasi Periksa.id telah berfungsi dengan baik pada aspek fungsional, portabilitas, dan reliabilitas. Sistem mampu menangani input yang salah dengan memberikan respons yang tepat, serta berhasil menjalankan verifikasi dan penyimpanan data sesuai spesifikasi yang ditentukan.

Pengujian *portability* menunjukkan bahwa aplikasi dapat diakses dan digunakan dengan baik di berbagai browser, yang menunjukkan fleksibilitasnya, sesuai dengan temuan Widiyanto (2021) mengenai pentingnya kemampuan aplikasi untuk beradaptasi dengan berbagai platform. Pengujian reliabilitas juga menunjukkan hasil yang sangat memuaskan, di mana sistem mampu menangani 200 pengguna secara bersamaan tanpa adanya error, membuktikan kestabilan dan ketahanannya, yang konsisten dengan hasil penelitian oleh Tohirin *et al.* (2019). Meskipun demikian, ada beberapa kelemahan yang ditemukan, terutama pada aspek performa dan keamanan. Pada pengujian performa, waktu respons sistem masih belum optimal, yang dapat menghambat efisiensi kerja, sebagaimana ditemukan oleh Yoshiura *et al.* (2017) dalam penelitian mereka yang menunjukkan bahwa waktu respons aplikasi menjadi masalah utama dalam pengujian performa. Pada pengujian keamanan, terdeteksi beberapa kerentanan yang perlu diperbaiki untuk melindungi data pasien, yang sejalan dengan temuan oleh Sayyadi Tooranloo *et al.* (2021), yang mengaitkan masalah keamanan dalam

sistem manajemen rumah sakit dengan pengujian yang tidak memadai. Hal ini mengindikasikan bahwa meskipun secara fungsional aplikasi sudah baik, ada area-area tertentu yang memerlukan perbaikan mendalam untuk mencapai kualitas yang lebih tinggi. Secara khusus, hasil dari *Test Closure* menggarisbawahi bahwa ada dua skenario yang dapat digolongkan berdasarkan tingkat keparahannya: satu dengan tingkat keparahan *Major* pada keamanan (*Security Testing*) dan satu dengan tingkat keparahan *Minor* pada fungsionalitas (*Negative Testing*). Skenario pertama dengan *test case ID TC09* memiliki tingkat keparahan *Major* karena terdeteksi 11 kerentanan yang berpotensi menyebabkan kebocoran data dan pemalsuan data, yang tidak memenuhi karakteristik *CLA Triad* (Kerahasiaan, Integritas, Ketersediaan). Kerentanan ini bisa dikategorikan sebagai *Critical* dan harus menjadi prioritas perbaikan dengan melakukan analisis lebih lanjut melalui *Penetration Testing* untuk membuktikan eksploitasi celah keamanan tersebut, sebagaimana yang direkomendasikan oleh Odekunle *et al.* (2017) untuk mengidentifikasi dan menangani kerentanan sistem. Skenario kedua dengan *test case ID TC03* memiliki tingkat keparahan *Minor* karena sistem dapat menyimpan data kosong di *history* kunjungan.

Meskipun data ini dapat dihapus, adanya bug ini menunjukkan kekurangan dalam validasi input, yang konsisten dengan temuan oleh Afrizal *et al.* (2019) mengenai pentingnya pengujian yang mendalam terhadap validasi input. Berdasarkan temuan kelemahan pada aplikasi, kelemahan ini tidak berdampak langsung pada kegiatan operasional rumah sakit. Namun, kelemahan ini menjadi celah bagi peretas yang berniat untuk melakukan tindak kriminal dengan memanfaatkan kerentanan yang ada, sejalan dengan temuan yang dicatat oleh Chukwu *et al.* (2022) mengenai kerentanannya yang berpotensi dieksploitasi oleh pihak yang tidak bertanggung jawab.

#### 4. Kesimpulan dan Saran

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa penerapan metode *Software Testing Life Cycle* (STLC) dalam pengujian kualitas aplikasi Periksa.id di Rumah Sakit Soeharto Heerdjan telah berjalan dengan efektif dan menghasilkan temuan yang komprehensif. Tahap

*Requirement Analysis* berhasil mengidentifikasi alur proses bisnis dan *Use Case Diagram* dari aplikasi, serta observasi terhadap arsitektur sistem dan *Standard Operating Procedure (SOP)* rumah sakit. Pada tahap *Test Planning*, peneliti berhasil menyusun rencana pengujian yang terstruktur, mencakup jenis pengujian fungsional dan non-fungsional, yang memberikan panduan jelas untuk seluruh proses pengujian. Tahap *Test Case Design* menghasilkan 9 skenario pengujian yang rinci dan terukur, yang mencakup pengujian positif, negatif, dan non-fungsional, memungkinkan pengujian dilakukan secara sistematis. Pada tahap *Test Environment Setup*, lingkungan pengujian yang memadai telah disiapkan, termasuk perangkat keras, *tools* pengujian, dan akun khusus untuk memastikan pengujian berjalan lancar. Tahap *Test Execution* menunjukkan bahwa aplikasi berfungsi dengan baik pada aspek fungsional, portabilitas, dan reliabilitas. Pengujian ini menunjukkan bahwa aplikasi mampu menangani berbagai jenis input, dapat diakses dengan baik di berbagai browser, serta mampu menangani beban pengguna tanpa terjadi kesalahan, dengan tingkat reliabilitas 100%. Namun, pada pengujian performa, ditemukan bahwa waktu respons masih tidak konsisten, yang perlu diperbaiki untuk meningkatkan efisiensi sistem.

Tahap *Test Closure* mengidentifikasi adanya bug minor pada fungsionalitas, khususnya terkait input data kosong, serta bug *major* pada aspek keamanan (*security*). Bug minor pada fungsionalitas ini menunjukkan adanya kekurangan dalam validasi input, yang perlu diperbaiki pada sistem backend aplikasi agar data kosong tidak dapat tersimpan. Sedangkan bug *major* yang ditemukan pada aspek keamanan memerlukan pengujian *penetration testing* oleh tim profesional dengan standar internasional untuk menemukan dan mengatasi kerentanannya, agar aplikasi dapat mencapai tingkat keamanan yang lebih baik. Secara keseluruhan, meskipun aplikasi Periksa.id telah menunjukkan performa yang baik dalam pengujian fungsional dan reliabilitas, terdapat beberapa area yang masih perlu perbaikan, terutama dalam hal keamanan dan performa. Oleh karena itu, disarankan agar pengujian lebih lanjut dilakukan untuk memperbaiki kerentanan yang ada dan meningkatkan kualitas sistem secara menyeluruh.

## 5. Daftar Pustaka

- Afrizal, S. H., Handayani, P. W., Hidayanto, A. N., Eryando, T., Budiharsana, M., & Martha, E. (2019). Barriers and challenges to primary health care information system (PHCIS) adoption from health management perspective: A qualitative study. *Informatics in Medicine Unlocked*, 17, 100198. <https://doi.org/10.1016/j.imu.2019.100198>.
- Aisyah, D. N., Setiawan, A. H., Lokopessy, A. F., Faradiba, N., Setiaji, S., Manikam, L., & Kozlakidis, Z. (2024). The information and communication technology maturity assessment at primary health care services across 9 provinces in Indonesia: Evaluation study. *JMIR Medical Informatics*, 12(1), e55959. <https://doi.org/10.2196/55959>.
- Amin, M. (2018). ICT for rural area development in Indonesia: A literature review. *Jurnal Ilmiah Teknologi dan Informasi ASIA*, 12(2), 32–42. <https://doi.org/10.30818/jitu.1.2.1881>.
- Atashi, A., Khajouei, R., Azizi, A., & Dadashi, A. (2016). User interface problems of a nationwide inpatient information system: A heuristic evaluation. *Applied Clinical Informatics*, 7(1), 89–100. <https://doi.org/10.4338/ACI-2015-07-RA-0086>.
- Barreto, J. O. (2017). A web-based information system for a regional public mental healthcare service network in Brazil. *International Journal of Mental Health Systems*, 11(1), 1. <https://doi.org/10.1186/s13033-016-0117-z>.
- Chukwu, E., Garg, L., Foday, E., Konomanyi, A., Wright, R., & Smart, F. (2022). Electricity, computing hardware, and internet infrastructures in health facilities in Sierra Leone: Field mapping study. *JMIR Medical Informatics*, 10(2), e30040. <https://doi.org/10.2196/30040>.
- Ebnehoseini, Z., Tabesh, H., Deldar, K., Mostafavi, S. M., & Tara, M. (2019). Determining the hospital information system (HIS) success rate:

- Development of a new instrument and case study. *Open Access Macedonian Journal of Medical Sciences*, 7(9), 1407–1414. <https://doi.org/10.3889/oamjms.2019.294>.
- Handayani, P. W., Meiganto, A., & Hidayanto, A. N. (2020). Critical success factors for hospital information system implementation in Indonesia. *Health Policy and Technology*, 9(3), 311–319. <https://doi.org/10.1016/j.hlpt.2020.07.002>.
- Handayani, P. W., Yazid, S., Bressan, S., & Sampe, A. F. (2020). Information and communication technology recommendations for the further development of a robust national electronic health strategy for epidemics and pandemics. *Jurnal Sistem Informasi*, 16(2), 31–42. <https://doi.org/10.21609/jsi.v16i2.979>.
- Karamagi, H. C., Muneene, D., Droti, B., Apolot, R. R., Baguma, M., Banura, A., ... & Okeibunor, J. C. (2022). eHealth or e-chaos: The use of digital health interventions for health systems strengthening in sub-Saharan Africa over the last 10 years: A scoping review. *Journal of Global Health*, 12, 04090. <https://doi.org/10.7189/jogh.12.04090>.
- Odekunle, F. F., Odekunle, R. O., & Shankar, S. (2017). Why sub-Saharan Africa lags in electronic health record adoption and possible strategies to increase its adoption in this region. *International Journal of Health Sciences*, 11(4), 59–64. <https://pubmed.ncbi.nlm.nih.gov/29085270>.
- Pitaloka, A. A., & Nugroho, A. P. (2021). Digital transformation in Indonesia health care services: Social, ethical and legal issues. *Journal of Science, Technology and Innovation Policy*, 6(1), 51–66. <https://doi.org/10.14203/STIPM.2021.301>.
- Sayyadi Tooranloo, H., Sepideh, S., & Arezoo Sadat, A. (2021). Evaluation of failure causes in employing hospital information systems. *Journal of System Management*, 6(3), 31–76. <https://doi.org/10.30495/JSM.2021.678894>.
- Sukums, F., Mensah, N., Mpembeni, R., Massawe, S., Duysburgh, E., Williams, A., ... & Kaltschmidt, J. (2015). Promising adoption of an electronic clinical decision support system for antenatal and intrapartum care in rural primary healthcare facilities in sub-Saharan Africa: The QUALMAT experience. *International Journal of Medical Informatics*, 84(9), 647–657. <https://doi.org/10.1016/j.ijmedinf.2015.05.002>.
- Tohirin, T., Mauludyansah, W. A., Setyawan, S. E., & Widiyanto, S. R. (2019). Analisis kualitas dan penerapan Software Quality Assurance pada situs web e-Clinic menggunakan model ISO/IEC 9126. *MULTINETICS*, 5(2), 107–113. <https://doi.org/10.32722/multinetics.v5i2.2761>.
- Widiyanto, A. (2021). The challenges of hospital information system implementation: A case study of a public hospital in Indonesia. *Jurnal Farmasi Sains dan Komunitas*, 18(1), 7–13. <https://doi.org/10.24071/jpsc.003010>.
- Yoshiura, V. T., de Azevedo-Marques, J. M., Rzewuska, M., Ferreira, L. A., Pereira, A. M., Rehem, T. C., ... & Barreto, J. O. (2017). A web-based information system for a regional public mental healthcare service network in Brazil. *International Journal of Mental Health Systems*, 11(1), 1. <https://doi.org/10.1186/s13033-016-0117-z>.