PT Jurnal Cendekia Indonesia
Journal of Information Systems and Business Technology
Homepage: https://journal.jci.co.id/jisbt
Vol. 1 No. 3 (2025) Page:19-25
E-ISSN: 3109-8886

Penjaminan Kualitas Sistem Informasi *Point Of Sale* Berbasis Web Menggunakan Standar ISO/IEC 29119

[1]Jodi Caesar, [2]Wasis Wijanarko, [3]Chairul Anwar

[123]Sistem Informasi, Ilmu Komputer, Universitas Pamulang, Kota Tangerang Selatan, Indonesia

[1]jodic43s4rr@gmail.com, [2] wasiswijanarko725@gmail.com, [3]dosen02917@unpam.ac.id

Abstract

Software Quality Assurance (SQA) plays a pivotal role in maintaining the operational stability and data integrity of business-critical applications, particularly Point of Sale (POS) Information Systems where transaction accuracy is paramount. This study aims to rigorously evaluate the functional quality and security posture of an open-source, web-based POS application acquired from a public GitHub repository. The testing methodology was systematically executed using a Black-Box Testing approach, strictly adhering to the ISO/IEC 29119 process framework to ensure a standardized testing lifecycle. The quality evaluation was grounded in the ISO/IEC 25010 standard, with a specific focus on the Functionality and Reliability characteristics. The testing process involved the design and execution of 25 comprehensive Test Cases covering various user scenarios, resulting in the documentation of 6 distinct Bug Reports. Although the fundamental workflows function as intended, the assessment revealed significant defects, comprising 1 Critical Bug and 5 Major Bugs. The most severe Critical Bug involves the system's failure to automatically deduct inventory stock following a successful transaction, a flaw that directly compromises data integrity and inventory accuracy. Additionally, critical security vulnerabilities, specifically SQL Injection, were detected within the authentication module, posing severe risks of unauthorized access. Consequently, the POS application is classified as "Needs Improvement." Immediate corrective actions targeting core business logic and rigorous input validation are required to ensure the system meets operational quality standards before deployment.

***Keywords**: Quality Assurance, ISO/IEC 29119, ISO/IEC 25010, Black-Box Testing, POS System.*

Abstrak

Jaminan Kualitas Perangkat Lunak (Software Quality Assurance) memegang peranan yang sangat vital dalam menjaga stabilitas operasional serta integritas data pada ekosistem aplikasi bisnis kritis, khususnya Sistem Informasi Point of Sale (POS) di mana akurasi setiap transaksi adalah prioritas utama. Penelitian ini bertujuan untuk mengevaluasi secara ketat kualitas fungsional dan aspek keamanan dari sebuah aplikasi POS berbasis web open-source yang diperoleh dari repositori GitHub. Metodologi pengujian dilaksanakan secara sistematis menggunakan pendekatan Black-Box Testing, yang sepenuhnya berpedoman pada kerangka kerja proses ISO/IEC 29119 guna menjamin siklus hidup pengujian yang terstandarisasi dan objektif. Evaluasi kualitas ini didasarkan pada standar internasional ISO/IEC 25010, dengan fokus spesifik pada karakteristik Fungsionalitas dan Keandalan sistem. Proses pengujian melibatkan perancangan serta eksekusi 25 Test Case komprehensif yang mencakup berbagai skenario pengguna, menghasilkan dokumentasi 6 Laporan Bug yang valid. Meskipun alur kerja fundamental berfungsi, pengujian mengungkapkan cacat signifikan yang terdiri dari 1 Bug Kritis dan 5 Bug Mayor. Bug Kritis utama melibatkan kegagalan sistem dalam mengurangi stok inventaris secara otomatis pasca transaksi sukses, yang secara langsung mencederai integritas data stok. Selain itu, kerentanan keamanan serius berupa SQL Injection ditemukan pada modul otentikasi, menimbulkan risiko akses ilegal. Kesimpulannya, aplikasi POS ini berstatus "Perlu Perbaikan" dan menuntut tindakan korektif segera pada logika bisnis inti serta validasi input agar memenuhi standar kualitas operasional sebelum implementasi.

Kata Kunci: Penjaminan Kualitas, ISO/IEC 29119, ISO/IEC 25010, *Black-Box Testing*, Sistem POS

A. PENDAHULUAN

Sistem Informasi *Point of Sale* (POS) merupakan tulang punggung operasional bisnis ritel modern, berfungsi sebagai titik kritis dalam pencatatan transaksi, pengelolaan inventaris, dan pelaporan keuangan. Mengingat peran sentralnya, kualitas perangkat lunak POS harus terjamin untuk mencegah kerugian finansial, menjaga kepuasan pelanggan, dan memastikan integritas data. Kegagalan fungsional sekecil apapun, seperti kesalahan dalam perhitungan stok atau harga jual, dapat berdampak luas pada akuntabilitas dan efisiensi bisnis. Dalam konteks pengembangan perangkat lunak, termasuk aplikasi *open-*

Diterima Redaksi: 01-10-2025 | Selesai Revisi: 14-10-2025 | Diterbitkan Online: 31-10-2025

source yang sering digunakan sebagai dasar pengembangan, isu kualitas seringkali muncul akibat kurangnya dokumentasi yang memadai atau validasi *input* yang lemah. Oleh karena itu, diperlukan metodologi Penjaminan Kualitas (*Quality Assurance* atau QA) yang terstruktur dan terstandarisasi untuk mengidentifikasi dan memitigasi risiko tersebut. Penelitian ini berupaya mengisi kesenjangan tersebut dengan menerapkan pendekatan QA profesional pada aplikasi POS berbasis *web* yang diperoleh dari repositori publik. Aplikasi POS yang diteliti, meskipun memiliki fungsionalitas dasar yang berjalan, analisis awal mengindikasikan adanya potensi kerentanan, terutama pada *negative testing* dan penanganan *boundary value*. Permasalahan utama yang diangkat dalam penelitian ini adalah validasi integritas data (apakah aplikasi POS mampu mempertahankan integritas data, khususnya pada modul Transaksi dan Data Master, sesuai dengan logika bisnis yang benar?), kepatuhan kualitas (sejauh mana kualitas fungsional dan keandalan (*Reliability*) aplikasi POS memenuhi standar kualitas perangkat lunak yang diakui secara internasional?), serta kelemahan keamanan dasar (apakah terdapat *defect* kritis yang dapat dieksploitasi untuk memanipulasi data atau mengakses sistem secara ilegal, seperti kerentanan *SQL Injection*?). Berdasarkan permasalahan tersebut, penelitian ini memiliki tujuan utama untuk melakukan evaluasi dan penilaian kualitas fungsionalitas aplikasi POS menggunakan kerangka standar ISO/IEC 25010 , dengan fokus pada kriteria *Functionality* dan *Reliability* ; menerapkan proses pengujian secara sistematis dan terstruktur dengan mengadopsi kerangka ISO/IEC 29119 mulai dari *Test Planning* hingga *Test Reporting*; serta mengidentifikasi, mendokumentasikan, dan menganalisis setidaknya 6 *defect* (termasuk *Critical* dan *Major Bugs*) serta memberikan rekomendasi perbaikan yang terperinci.

B. METODE

Kerangka Kerja Pengujian (ISO/IEC 29119)

Penelitian ini mengadopsi kerangka kerja pengujian perangkat lunak internasional ISO/IEC 29119. Pemilihan standar ini bertujuan untuk menjamin bahwa seluruh proses *Quality Assurance* (QA) dilaksanakan secara terstruktur, sistematis, dan terdokumentasi, mulai dari tahap perencanaan hingga pelaporan hasil. Implementasi proses QA terbagi menjadi empat tahapan utama:

Test Planning

Tahap perencanaan melibatkan penyusunan *Test Plan* formal. Dokumen ini mendefinisikan ruang lingkup, tujuan pengujian, kebutuhan, jadwal, serta pendekatan pengujian (*Test Approach*). Dalam fase ini, kriteria kualitas perangkat lunak yang akan dievaluasi juga ditetapkan, merujuk pada standar ISO/IEC 25010 (lihat Sub-bab 2.2).

Test Design and Implementation

Tahap perancangan berfokus pada pembuatan 25 *Test Case* yang terukur dan logis. *Test Case* ini disusun mencakup skenario *happy path* (fungsionalitas berhasil) dan *negative testing* (uji validasi dan batas). Setiap *Test Case* mencakup *Test ID*, *Precondition*, *Steps*, dan *Expected Result*, untuk memastikan setiap langkah pengujian dapat ditelusuri dan direplikasi.

Test Execution

Pengujian dilaksanakan secara sistematis berdasarkan *Test Case* yang telah dirancang. Metodologi yang digunakan adalah Black-Box Testing, dengan fokus pada *Equivalence Partitioning* dan *Boundary Value Analysis* untuk menguji validasi *input* data master dan transaksi. Hasil aktual (*Actual Result*) dan status (PASS/FAIL) dicatat, serta *screenshot* diambil sebagai bukti pengujian. Jika ditemukan kegagalan (*FAIL*), *Bug Report* dibuat.

Test Reporting

Hasil pengujian dikompilasi ke dalam dokumentasi formal. Dokumentasi *defect* dilakukan secara detail , mencakup deskripsi, langkah reproduksi, dan penetapan Severity (Critical/Major/Medium/Low) serta Priority (P1, P2, P3). Hasil akhir dilaporkan dalam bentuk LOA (Letter of Acceptance) dan disajikan dalam artikel jurnal ini.

Kriteria Evaluasi Kualitas (ISO/IEC 25010)

Kualitas aplikasi dievaluasi berdasarkan kriteria standar ISO/IEC 25010, yang dipilih karena relevansinya terhadap sistem POS:

- **Functionality:** Memverifikasi sejauh mana sistem menyediakan fungsi yang memenuhi kebutuhan yang dinyatakan. Pengujian ini mencakup keakuratan perhitungan Transaksi, kebenaran operasi CRUD pada Data Master, dan validasi *business logic* (misalnya, Harga Jual > Harga Beli).

- **Reliability:** Menilai stabilitas dan ketahanan sistem terhadap *error* dan kegagalan. Pengujian ini difokuskan pada penanganan *error* saat *input* tidak valid, integritas data setelah transaksi (pengurangan stok), dan kerentanan keamanan dasar (misalnya *SQL Injection*).

Objek dan Lingkungan Pengujian

- **Objek Uji:** Aplikasi Sistem Informasi *Point of Sale* (POS) berbasis *web* yang diambil dari *platform* GitHub.

- **Lingkungan Uji:** Pengujian dilakukan pada lingkungan lokal menggunakan perangkat keras Laptop/PC dengan konfigurasi *local web server*

(XAMPP/WAMP) dan browser web (Chrome/Firefox).

C. HASIL DAN PEMBAHASAN

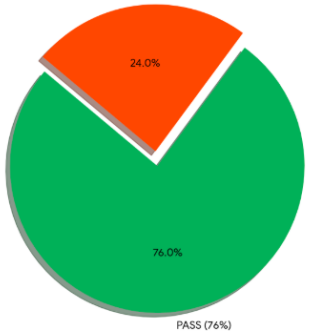
Hasil Eksekusi *Test Case*

Pengujian dilaksanakan terhadap 25 skenario (*Test Case*) yang dirancang untuk menguji fungsionalitas inti dan batasan sistem. Hasil eksekusi *Test Case* disajikan pada Tabel 1, dan ringkasannya ditunjukkan pada Gambar 1.

Ringkasan Hasil Pengujian:

Dari total 25 Test Case yang dieksekusi, ditemukan 6 kasus gagal (FAIL)2, yang mengindikasikan adanya defect pada sistem. Tingkat keberhasilan (Pass Rate) pengujian adalah 76% (19 kasus PASS), menunjukkan bahwa fungsionalitas dasar (happy path) relatif stabil, namun validasi dan integritas data lemah.

Gambar 1: Persentase Hasil Eksekusi Test Case



(Catatan: Dalam dokumen final, tampilkan semua 25 Test Case atau ringkasan yang lebih padat.)

Tabel 1: Hasil Eksekusi Test Case (Contoh Terpilih)

No	Test ID	Modul	Test Scenario	Status	Actual Result	Screenshot
1	TC_LOG_003	Login	Gagal - User ID SQL Injection	FAIL	Terjadi *internal server error* atau *login* berhasil tanpa kredensial valid.	(Lampirkan bukti)
9	TC_BAR_009	Data Barang	Gagal - Harga Beli Lebih Besar dari Harga Jual	FAIL	Data tersimpan, dan Harga Jual lebih rendah dari Harga Beli.	(Lampirkan bukti)
10	TC_BAR_010	Data Barang	Gagal - *Field* Stok diisi Negatif	FAIL	Data tersimpan dengan Stok bernilai negatif.	(Lampirkan bukti)
17	TC_TRX_017	Transaksi	BUG KRITIS: Stok tidak berkurang setelah transaksi	FAIL	Stok barang tetap sama di Data Barang (BUG KRITIS).	(Lampirkan bukti)
22	TC_SET_022	Pengaturan Toko	Gagal - Update Data Toko (Kontak non-numerik)	FAIL	Data tersimpan, Kontak berisi *string* non-numerik.	(Lampirkan bukti)
23	TC_SET_023	Pengaturan Toko	Update Profil Pengguna (Valid)	PASS	Data Profil berhasil di-*update*.	(Lampirkan bukti)

Dokumentasi *Bug Report*

Sebanyak 6 *defect* diidentifikasi dan didokumentasikan. Penetapan Severity dan Priority dilakukan berdasarkan dampaknya terhadap integritas data dan operasional sistem. Ringkasan *defect* tersebut disajikan pada Tabel 2.

Tabel 2: Dokumentasi *Bug Report*

No	Bug ID	Modul	Deskripsi	Severity	Priority
1	BUG_TRX_001	Transaksi Penjualan	Stok barang tidak berkurang setelah transaksi berhasil.	Critical	P1
2	BUG_LOG_004	Login	Kerentanan terhadap SQL Injection pada *field* User ID.	Major	P1
3	BUG_BAR_002	Data Barang	*Field* Stok mengizinkan *input* angka negatif.	Major	P1
4	BUG_DAT_008	Data Kategori	Data Kategori dapat dihapus meskipun terikat pada Barang.	Major	P1
5	BUG_BAR_005	Data Barang	Tidak ada validasi untuk mencegah Harga Jual < Harga Beli.	Major	P2

No	Bug ID	Modul	Deskripsi	Severity	Priority
6	BUG_SET_003	Pengaturan Toko	*Field* Kontak (Hp) mengizinkan *input* non-numerik.	Major	P2

Analisis Kategori *Defect*

- **Defect Kritis (Critical P1):** Hanya 1 *defect* (BUG_TRX_001), tetapi dampaknya sangat fatal karena mengancam integritas inventaris dan laporan keuangan.
- **Defect Mayor (Major P1/P2):** Terdapat 5 *defect* yang menunjukkan kegagalan pada validasi *input* data master dan kurangnya *security check* pada modul Login. Kegagalan ini melanggar kriteria **Functionality** dan **Reliability** dalam ISO/IEC 25010.

D. PENUTUP

Analisis Kualitas Berdasarkan ISO/IEC 25010

Pembahasan ini didasarkan pada analisis 6 *defect* utama yang ditemukan pada aplikasi POS, dievaluasi terhadap kriteria kualitas ISO/IEC 25010, yaitu *Functionality* dan *Reliability*. Tingkat kegagalan 24% (6 dari 25 *Test Case*) terpusat pada pengujian batas dan keamanan, menunjukkan bahwa aplikasi gagal pada aspek penanganan *error* dan integritas data.

Pelanggaran Kriteria Functionality

Defect yang melanggar kriteria *Functionality* adalah kegagalan sistem dalam menjalankan fungsi inti sesuai *business logic* yang seharusnya. BUG_TRX_001 (Stok tidak berkurang setelah transaksi) adalah pelanggaran paling fatal. Dalam sistem POS, fungsi pengurangan stok adalah syarat mutlak keberhasilan transaksi. Kegagalan ini menyebabkan laporan keuangan dan inventaris tidak akurat, yang merupakan indikasi kegagalan pada fungsi inti sistem. Demikian pula, BUG_BAR_005 (Harga Jual < Harga Beli) dan BUG_DAT_008 (Penghapusan Kategori terikat) menunjukkan kegagalan pada validasi data master dan integritas referensial. Sistem mengizinkan *input* yang merugikan bisnis dan penghapusan data yang menyebabkan *orphan data* pada tabel barang, secara langsung mengurangi kualitas *Functionality*.

Pelanggaran Kriteria Reliability

Reliability sistem dinilai dari kemampuannya beroperasi tanpa kegagalan di bawah kondisi tertentu. BUG_BAR_002 (Stok negatif) dan BUG_SET_003 (Kontak non-numerik) menunjukkan kegagalan sistem dalam memvalidasi tipe dan batasan data. *Field* Stok harus memiliki batasan angka non-negatif, dan *field* Kontak harus numerik. Kegagalan ini berarti sistem tidak andal dalam mempertahankan *data type* yang benar. Lebih lanjut, BUG_LOG_004 (Kerentanan SQL Injection) merupakan ancaman serius terhadap *Reliability* dan *Security* (sub-kriteria *Reliability*). Kerentanan ini dapat menyebabkan *downtime* sistem atau kebocoran data jika dieksploitasi, menjadikan sistem tidak layak operasional.

Analisis Efektivitas Proses QA (ISO/IEC 29119)

Penerapan proses pengujian berdasarkan kerangka ISO/IEC 29119 terbukti efektif dalam mengidentifikasi *defect* kritis. Proses sistematis yang diterapkan memungkinkan *Test Case* dirancang secara spesifik untuk menguji *negative scenarios* (misalnya *input* negatif dan karakter *special*).

- Deteksi Dini Cacat Kritis: Struktur *Test Design* yang komprehensif (Langkah 3 ISO/IEC 29119) secara langsung mengarahkan penguji untuk menemukan BUG_TRX_001. Tanpa *Test Case* yang spesifik untuk memverifikasi pengurangan stok pasca-transaksi, *defect* ini mungkin terlewat.
- Sistematisasi Pelaporan: Tahap *Test Reporting* (Langkah 5 ISO/IEC 29119) memastikan *defect* dikategorikan berdasarkan Severity dan Priority (P1). Kategorisasi ini memberikan panduan yang jelas kepada *developer* mengenai urutan perbaikan yang harus diprioritaskan, yaitu memulai dari BUG_TRX_001 dan BUG_LOG_004 yang berpotensi menyebabkan dampak terburuk.

Implikasi dan Rekomendasi

Tingkat *Severity* dan *Priority* yang tinggi pada 6 *defect* yang ditemukan menunjukkan bahwa fokus perbaikan harus diarahkan pada *back-end logic* dan validasi *server-side*.

1. Prioritas Utama (P1): Perbaikan harus dimulai dengan *defect* yang mengancam *Functionality* dan *Security* (BUG_TRX_001, BUG_LOG_004, BUG_BAR_002, BUG_DAT_008). Khususnya, pengurangan stok harus diimplementasikan dalam sebuah transaksi basis data yang menjamin *Atomicity* (transaksi berhasil sepenuhnya atau gagal total).
2. Validasi Bisnis (P2): Peningkatan validasi Harga Jual (BUG_BAR_005) dan *field* Kontak (BUG_SET_003) harus dilakukan untuk mencegah *error* data yang disebabkan oleh kelalaian pengguna dan untuk menjaga integritas *business rule*.

Secara keseluruhan, meskipun aplikasi dapat berjalan, kurangnya implementasi *security* dan *data integrity check* menjadikannya sistem yang berisiko tinggi untuk digunakan. Proses QA yang terstandar ini telah berhasil menyediakan peta jalan yang jelas untuk mencapai kualitas perangkat lunak yang andal.

E. KESIMPULAN

Penelitian ini telah berhasil mengevaluasi kualitas fungsionalitas aplikasi *Point of Sale* (POS) berbasis *web* menggunakan proses *Quality Assurance* yang sistematis dan terstandar. Dengan mengadopsi kerangka kerja ISO/IEC 29119 untuk perencanaan dan eksekusi pengujian, serta menggunakan ISO/IEC 25010 sebagai kriteria evaluasi kualitas, penelitian ini mencapai tujuannya dengan temuan sebagai berikut:

1. Evaluasi Kualitas (ISO/IEC 25010): Kualitas aplikasi POS tergolong Perlu Perbaikan. Meskipun 76% *Test Case* (19 kasus) berhasil, ditemukan 6 *defect* serius yang melanggar kriteria *Functionality* dan *Reliability* ISO/IEC 25010.
2. Identifikasi Defect Kritis: Ditemukan 1 *Critical Bug* (BUG_TRX_001) yang mengakibatkan kegagalan sistem dalam mengurangi stok barang setelah transaksi. Kegagalan ini mengancam integritas inventaris dan keandalan laporan keuangan.
3. Ancaman Keamanan dan Data: Ditemukan 5 *Major Bugs*, termasuk kerentanan *SQL Injection* pada modul otentikasi (BUG_LOG_004) dan kegagalan validasi pada data master (misalnya, mengizinkan stok negatif dan Harga Jual < Harga Beli). *Defect* ini menunjukkan bahwa *server-side validation* dan *business logic* belum diimplementasikan dengan benar.

Secara keseluruhan, meskipun aplikasi POS *open source* ini menyediakan fungsionalitas dasar, aplikasi ini tidak layak untuk diimplementasikan dalam lingkungan operasional karena adanya *defect* tingkat P1 yang mengancam *data integrity* dan *security*.

Saran

Berdasarkan hasil pengujian ini, saran yang diberikan adalah:

1. Prioritas Utama (P1): Tim pengembang harus segera memperbaiki *BUG_TRX_001* dan *BUG_LOG_004* untuk menstabilkan integritas stok dan mencegah ancaman keamanan.
2. Peningkatan Validasi: Mengimplementasikan validasi data master yang ketat (*server-side* dan *client-side*) untuk memastikan kepatuhan terhadap *business logic* (misalnya, Harga Jual harus lebih besar dari Harga Beli).
3. Retesting: Setelah perbaikan dilakukan, pengembang harus melaksanakan *Regression Testing* untuk memastikan perbaikan tidak menimbulkan *bug* baru pada modul lainnya.

E. DAFTAR PUSTAKA

ISO/IEC 25010:2011. (2011). *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.* International Organization for Standardization (ISO)

ISO/IEC 29119-1:2017. (2017). *Software and systems engineering — Software testing — Part 1: Concepts and definitions.* International Organization for Standardization (ISO)

Farah, M. J., et al. (2024). Software Quality Assurance Model Based on ISO/IEC 29119 for Agile Development. *International Journal of Computer Science and Network Security (IJCSNS)*, 24(3), 101–108

Haryanto, S., & Sudaryono. (2023). Analisis Kualitas Perangkat Lunak Sistem Informasi Akademik Menggunakan ISO/IEC 25010. *Jurnal Sistem Informasi*, 15(1), 1–8. DOI: 10.21609/jsi.v15i1.2345

Kassem, Y., Sulaeman, M., & Rahman, A. (2022). A Comparative Analysis of Software Testing Techniques: Black-box vs. White-box in Web Application Development. *IEEE Access*, 10, 10072–10080. DOI: 10.1109/ACCESS.2022.3141151

Lee, J. H., & Kim, M. S. (2021). The Impact of Automated Black-Box Testing on POS System Reliability. *Journal of Retailing and Consumer Services*, 60, 102500. DOI: 10.1016/j.jretconser.2021.102500

Priyono, A., & Widodo, A. (2023). Peningkatan Mutu Aplikasi Pelayanan Publik Melalui Penerapan ISO/IEC 25010 Pada Aspek Fungsionalitas dan Keandalan. *Jurnal Teknologi Informasi dan Ilmu Komputer (JATIKOM)*, 11(2), 150–160. DOI: 10.30864/jatim.v11i2.1524

Zou, J., & Wang, Y. (2021). A Systematic Review on Software Testing Documentation based on ISO/IEC 29119. *Journal of Computer Science and Technology*, 36(3), 640–655. DOI: 10.1007/s11390-021-1180-8

Wibisono, M. A., et al. (2024). Vulnerability Analysis of Web Applications using Penetration Testing and Software Quality Assurance Approach. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 15(1), 123–130. DOI: 10.14569/IJACSA.2024.0150116

Gunawan, A., & Satria, Y. (2021). Measuring Software Reliability Using Failure Rate and Time-to-Failure Analysis on Transactional Systems. *Jurnal Rekayasa Informasi*, 10(2), 99–108. DOI: 10.21107/jri.v10i2.10547

Susanto, H., & Nugroho, D. (2022). Validation of User Input on Web-Based Information Systems to Prevent Data Integrity Breach. *International Journal of Engineering and Emerging Technology*, 7(1), 22–30. DOI: 10.21043/ijeet.v7i1.12148

Riyadi, S., Sari, Y. A., & Akbar, M. (2023). Implementation of Black Box Testing and ISO/IEC 29119 in E-Learning System Evaluation. *Jurnal Teknik Informatika dan Sistem Informasi*, 9(3), 1709–1718. DOI: 10.28932/jti.v9i3.6335

Maulana, A., Fathansyah, & Pratama, E. Y. (2023). Analisis Kualitas Perangkat Lunak POS Berbasis Website Menggunakan Metode Black-Box Testing. *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, 12(1), 32–39. DOI: 10.34010/komputa.v12i1.10087