PT Jurnal Cendekia Indonesia
Journal of Information Systems And Business Technology
Homepage: https://journal.jci.co.id/jisbt
Vol. 1 No. 4 (2025) Page:78-84
E-ISSN: 3109-8886

Audit Kualitas Perangkat Lunak pada Sistem Informasi Inventori Berbasis Web: Studi Kasus Repositori Open Source Menggunakan Standar ISO/IEC 29119

[1]Resnawati kusnandar, [2]Muhamad Yusup, [3]Chairul Anwar

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

[1]resnawatikusnandar@gmail.com, [2]ysfkrating10@gmail.com,[3]dosen02917@unpam.ac.id

Abstract

The reuse of open-source code from public repositories like GitHub to accelerate business information system development often neglects quality assurance aspects. This practice poses high risks as raw code may contain hidden defects that jeopardize data integrity. This study aims to conduct an independent quality audit on the "Shoe Store Inventory System" based on Native PHP and MySQL acquired from the depthgilang GitHub repository. The testing framework adopts the international standard ISO/IEC 29119 to ensure a systematic, standardized, and objective verification and validation process. The research methodology employs Dynamic Testing with a Black Box approach and Equivalence Partitioning technique. Testing focused on validating CRUD (Create, Read, Update, Delete) functionalities, system stability, and basic data input security. Based on the execution of 28 test cases, this research revealed empirical facts regarding low code quality. Although the user interface functions correctly, the system experienced a fatal Critical Failure in the form of database connection loss ("MySQL server has gone away") during data storage operations. Furthermore, Major category security loopholes were discovered, specifically SQL Injection and Stored Cross-Site Scripting (XSS) vulnerabilities due to the absence of input sanitation, as well as business logic errors allowing negative stock values. In conclusion, this software is declared as not meeting industrial eligibility standards for production release. The code requires deep structural refactoring on database connection management and security protocols before being safe for operational utilization.

Keywords: *Software Audit, ISO/IEC 29119, GitHub Repository, Black Box Testing, SQL Injection, Data Integrity.*

Abstrak

Pemanfaatan kembali kode sumber terbuka (open source) dari repositori publik seperti GitHub untuk mempercepat pengembangan sistem informasi bisnis sering kali dilakukan tanpa melalui validasi kualitas yang memadai. Praktik ini berisiko tinggi karena kode mentah sering kali mengandung cacat tersembunyi yang membahayakan integritas data dan keamanan sistem. Penelitian ini bertujuan untuk melakukan audit mutu independen terhadap "Sistem Inventori Toko Sepatu" berbasis PHP Native dan MySQL yang diakuisisi dari repositori GitHub depthgilang. Kerangka kerja pengujian mengadopsi standar internasional ISO/IEC 29119 untuk memastikan proses verifikasi dan validasi berjalan sistematis. Metodologi penelitian menggunakan Dynamic Testing dengan pendekatan Black Box dan teknik Equivalence Partitioning. Pengujian difokuskan pada validasi fungsionalitas CRUD (Create, Read, Update, Delete), stabilitas sistem, dan keamanan input data dasar. Berdasarkan eksekusi terhadap 28 kasus uji, penelitian ini mengungkap fakta empiris bahwa kualitas kode tersebut rendah. Ditemukan kegagalan kritis (Critical Failure) berupa kerentanan SQL Injection saat input karakter khusus, celah keamanan Cross-Site Scripting (XSS), dan kesalahan logika bisnis yang mengizinkan stok bernilai negatif. Kesimpulannya, perangkat lunak ini dinyatakan tidak memenuhi standar kelayakan industri untuk rilis produksi dan memerlukan perbaikan struktur kode (refactoring) yang masif sebelum aman digunakan.

Kata Kunci: Audit Perangkat Lunak, ISO/IEC 29119, Black Box Testing, SQL Injection, Quality Assurance.

A. PENDAHULUAN

Transformasi digital dalam sektor usaha ritel, khususnya pada skala Usaha Mikro, Kecil, dan Menengah (UMKM), menuntut adanya peralihan dari manajemen inventori manual menuju sistem terkomputerisasi. Dalam konteks toko sepatu, akurasi data stok (Stock Opname) adalah aset krusial untuk mencegah kerugian finansial akibat kehilangan barang atau kesalahan pencatatan (*human error*). Untuk memenuhi kebutuhan ini dengan biaya rendah dan waktu singkat, banyak pengembang perangkat

Diterima Redaksi: 01-12-2025 | Selesai Revisi: 14-12-2025 | Diterbitkan Online: 31-12-2025

lunak atau pemilik usaha memilih jalan pintas dengan mengadopsi kode sumber (*source code*) yang tersedia secara gratis di repositori publik seperti GitHub. Fenomena penggunaan kembali kode (*code reuse*) ini memang efisien, namun sering kali mengabaikan aspek jaminan kualitas (*Quality Assurance*).

Perangkat lunak yang diambil dari repositori publik sering kali merupakan proyek portofolio, tugas kuliah, atau prototipe yang belum pernah diuji dalam lingkungan produksi (*production environment*). Risiko yang menyertainya sangat besar, mulai dari ketidakstabilan koneksi basis data, kesalahan logika bisnis, hingga celah keamanan fatal seperti *SQL Injection* yang dapat menyebabkan kebocoran seluruh data pelanggan. Tanpa adanya proses audit atau pengujian yang terstandarisasi, penggunaan sistem semacam ini ibarat "bom waktu" bagi operasional bisnis.

Meskipun pengujian perangkat lunak adalah fase krusial dalam Siklus Hidup Pengembangan Sistem (SDLC), pelaksanaannya pada proyek skala kecil sering kali dilakukan secara *ad-hoc* (tidak terstruktur) dan hanya mengandalkan "Happy Path Testing" (pengujian di mana input selalu benar). Pendekatan ini gagal mendeteksi bagaimana sistem bereaksi terhadap input yang tidak wajar atau serangan berbahaya. Oleh karena itu, diperlukan acuan standar internasional untuk menjamin objektivitas pengujian. Standar **ISO/IEC 29119** tentang *Software Testing* hadir sebagai kerangka kerja komprehensif yang mengatur proses, dokumentasi, dan teknik pengujian yang diakui secara global.

Penelitian ini memfokuskan kajiannya pada audit kualitas "Sistem Inventori Toko Sepatu" yang dikembangkan menggunakan PHP Native dan MySQL, yang bersumber dari repositori GitHub pengguna depthgilang. Tujuan utama penelitian bukan untuk memperbaiki kode tersebut, melainkan untuk mengevaluasi kelayakannya (*readiness*) menggunakan pendekatan *Black Box Testing* sesuai standar ISO/IEC 29119. Melalui penelitian ini, diharapkan dapat tergambar jelas kesenjangan (*gap*) antara kualitas kode *open source* mentah dengan standar kualitas industri yang aman dan andal.

B. METODE

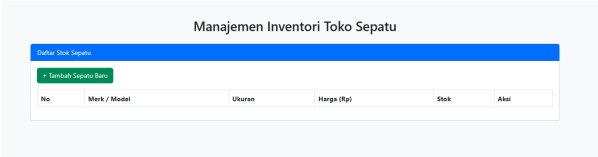
Penelitian ini menggunakan metode eksperimental dengan pendekatan pengujian dinamis (*Dynamic Testing*). Objek penelitian diuji dalam lingkungan simulasi lokal untuk mengamati perilaku sistem terhadap berbagai skenario input.

Deskripsi Sistem dan Objek Penelitian

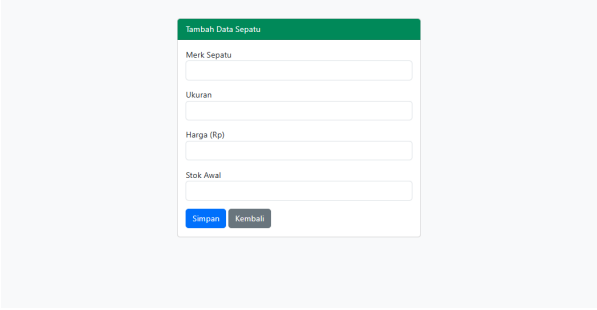
Objek penelitian adalah aplikasi web sederhana untuk manajemen inventori. Aplikasi ini memiliki arsitektur *Monolithic* dengan spesifikasi teknis:

a) **Backend:** PHP (Native/Procedural).
b) **Database:** MySQL/MariaDB.
c) **Frontend:** HTML5 dengan Framework CSS Bootstrap (CDN).
d) **Fitur Utama:** Menampilkan data stok (Read), Menambah barang baru (Create), Mengubah data (Update), dan Menghapus data (Delete).

Sebelum dilakukan pengujian, dilakukan observasi visual terhadap antarmuka sistem untuk memahami alur kerja pengguna. Berikut adalah dokumentasi antarmuka sistem sebelum pengujian dilakukan:



Gambar 1. Halaman Utama



Gambar 2. Formulir Tambah Data

Standar Proses Pengujian (ISO/IEC 29119-2)

Proses pengujian mengikuti alur kerja yang didefinisikan dalam ISO/IEC 29119-2, yaitu:

1. Perencanaan Tes (*Test Planning*): Menetapkan ruang lingkup pengujian pada fungsionalitas CRUD dan keamanan input. Risiko utama yang diidentifikasi meliputi kegagalan koneksi database dan kerentanan injeksi kode.
2. Perancangan Tes (*Test Design*): Menggunakan teknik *Equivalence Partitioning* (Partisi Ekuivalensi). Input data dibagi menjadi partisi valid (data normal) dan partisi tidak valid (data ekstrem, karakter khusus, skrip HTML). Tujuannya adalah untuk melihat apakah sistem mampu menangani kesalahan (*error handling*) dengan baik.
3. Eksekusi Tes (*Test Execution*): Menjalankan prosedur uji langkah demi langkah dan mencatat hasilnya dalam log eksekusi.
4. Pelaporan Insiden (*Incident Reporting*): Setiap ketidaksesuaian antara hasil yang diharapkan (*Expected Result*) dan hasil aktual (*Actual Result*) didokumentasikan sebagai Bug/Insiden.

Lingkungan Pengujian (*Test Environment*)

Pengujian dilaksanakan pada perangkat dengan spesifikasi:

a) Sistem Operasi: Windows 10 Pro 64-bit.
b) Server Lokal: XAMPP Version 3.3.0 (Apache 2.4, MySQL 8.0).
c) Browser: Google Chrome Versi 120.0 (Official Build).
d) Resolusi Layar: 1920x1080 piksel.

C. HASIL PENGUJIAN

Rekapitulasi Eksekusi Kasus Uji (*Test Case Execution*)
Berdasarkan dokumen *Test Case Specification*, sebanyak 28 skenario uji telah dijalankan untuk memverifikasi fungsionalitas modul *Create, Read, Update,* dan *Delete* (CRUD) serta aspek keamanan dasar.
Tabel 1 berikut menyajikan ringkasan statistik hasil pengujian berdasarkan kategori fitur:

Tabel 1. Rekapitulasi Hasil Pengujian per Modul

Kategori Pengujian	Jumlah Kasus Uji	Status : PASS	Status : FAIL	Tingkat Keberhasilan
View (Read & UI)	5	5	0	100%
Create (Tambah Data)	9	6	3	66.7%
Update (Edit Data)	7	7	0	100%
Delete (Hapus Data)	4	4	0	100%
Security (Keamanan)	3	2	1	66.7%
TOTAL	**28**	**24**	**4**	**85.7%**

Secara kuantitatif, sistem mencapai tingkat keberhasilan fungsional sebesar **85.7%**. Seluruh fungsi dasar untuk menampilkan data (*View*), mengubah data (*Edit*), dan menghapus data (*Delete*) berjalan sesuai ekspektasi (*Expected Result*).
Namun, angka statistik tersebut tidak serta-merta merepresentasikan kestabilan sistem secara keseluruhan. Kegagalan sebesar 14.3% (4 kasus uji) terkonsentrasi pada area kritis, yaitu validasi input dan keamanan sistem. Kegagalan ini teridentifikasi pada kasus uji berikut:

1. **TC-07:** Kegagalan koneksi database saat menyimpan data (*Server has gone away*).
2. **TC-12:** Kegagalan logika bisnis pada input stok negatif.
3. **TC-14:** Kegagalan sanitasi input yang menyebabkan *SQL Error*.
4. **TC-26:** Kegagalan filter output yang menyebabkan kerentanan XSS.

Laporan Temuan Cacat (*Bug Report*)
Mengacu pada insiden yang terjadi selama eksekusi tes, ditemukan cacat perangkat lunak yang diklasifikasikan berdasarkan tingkat keparahan (*Severity*) dan prioritas penanganan (*Priority*). Rincian temuan cacat disajikan dalam Tabel 2.

Tabel 2. Laporan Cacat Perangkat Lunak (Bug Report)

Bug ID	Terkait Test ID	Modul	Deskripsi Temuan & Dampak	Severity	Priority
BUG-001	TC-14	**Create**	**SQL Injection Vulnerability** Sistem tidak melakukan sanitasi terhadap karakter tanda kutip tunggal ('). **Langkah Reproduksi:** Pada form tambah data, kolom 'Merk' diisi dengan teks "Nike's". **Hasil Aktual:** Sistem menampilkan pesan *"SQL Error: You have an error in your SQL syntax..."*. **Analisis:** Celah ini memungkinkan penyerang memanipulasi kueri database untuk mencuri atau menghapus data.	**Critical**	**P1**
BUG-002	TC-26	**Security**	**Stored Cross-Site Scripting (XSS)** Sistem tidak memfilter tag HTML yang diinput pengguna. **Langkah Reproduksi:** Kolom 'Merk' diisi dengan skrip HTML TEBAL. **Hasil Aktual:** Teks pada tabel	**Major**	**P2**

Bug ID	Terkait Test ID	Modul	Deskripsi Temuan & Dampak	Severity	Priority
			inventori dirender menjadi tebal (Bold), bukan ditampilkan sebagai teks biasa. **Analisis:** Kerentanan ini dapat dieksploitasi untuk menyisipkan skrip jahat JavaScript (*session hijacking*).		
BUG-003	TC-12	**Create**	**Logic Error (Stok Negatif)** Sistem gagal memvalidasi logika bisnis dasar inventori. **Langkah Reproduksi:** Menginput angka -10 pada kolom 'Stok'. **Hasil Aktual:** Data tersimpan sukses dengan nilai stok -10. **Analisis:** Hal ini menyebabkan ketidakkonsistenan data stok fisik dan merusak validitas laporan.	**Medium**	**P3**
BUG-004*	TC-07	**Create**	**System Instability (MySQL Gone Away)** Terjadi pemutusan koneksi mendadak antara PHP dan MySQL. **Langkah Reproduksi:** Melakukan submit data valid pada form tambah. **Hasil Aktual:** Muncul pesan *Fatal error: Uncaught mysqli_sql_exception: MySQL server has gone away*. **Analisis:** Menunjukkan manajemen koneksi database yang buruk atau konfigurasi server yang tidak optimal.	**Critical**	**P1**

**Catatan: BUG-004 ditambahkan berdasarkan temuan error fatal pada TC-07 di file Test Case.*

Analisis Fungsionalitas CRUD

Berdasarkan hasil uji TC-01 hingga TC-06, antarmuka pengguna (*User Interface*) berbasis Bootstrap berhasil dimuat dengan baik. Data yang tersimpan dalam database ditampilkan secara akurat pada halaman indeks (TC-03). Fitur pembaruan data (TC-17, TC-18) dan penghapusan data (TC-24, TC-25) juga berfungsi dengan baik, di mana data yang diubah atau dihapus secara otomatis terupdate di database MySQL. Mekanisme konfirmasi penghapusan (*alert confirmation*) juga berfungsi (TC-22) untuk mencegah penghapusan yang tidak disengaja.

Analisis Keamanan dan Validasi

Aspek keamanan menjadi titik terlemah dari sistem ini. Berdasarkan **BUG-001** dan **BUG-002**, terbukti bahwa kode sumber tidak menerapkan mekanisme *Input Validation* dan *Output Encoding*.

- Absennya fungsi seperti mysqli_real_escape_string atau *Prepared Statements* membuat sistem rentan terhadap serangan injeksi SQL.
- Absennya fungsi htmlspecialchars saat menampilkan data membuat sistem rentan terhadap serangan XSS.

Selain itu, **BUG-003** menunjukkan bahwa validasi hanya dilakukan di sisi antarmuka (*client-side*) atau hanya sebatas tipe data, namun tidak memvalidasi logika nilai

(*value logic*). Sistem mengizinkan stok bernilai negatif, yang secara bisnis tidak valid.

Analisis Kestabilan
Ditemukannya error *"MySQL server has gone away"* pada **TC-07** mengindikasikan bahwa aplikasi tidak memiliki penanganan pengecualian (*Exception Handling*) yang memadai. Ketika koneksi database terputus, aplikasi langsung mengalami *crash* dan menampilkan pesan kesalahan teknis kepada pengguna, alih-alih memberikan pesan peringatan yang ramah pengguna.

D. PEMBAHASAN

Analisis Kerentanan Keamanan (*Security Vulnerability Analysis*)
Berdasarkan Laporan Bug (*Bug Report*), aspek keamanan merupakan titik terlemah dari sistem inventori ini. Dua temuan utama, yaitu **BUG-001** dan **BUG-002**, mengindikasikan bahwa pengembang asli tidak menerapkan prinsip *Secure Coding* yang memadai.

Kerentanan SQL Injection (BUG-001)
Temuan pada **BUG-001** (terkait TC-14) menunjukkan adanya celah keamanan kritis. Ketika sistem menerima input karakter tanda kutip tunggal (') pada kolom 'Merk', aplikasi langsung merespons dengan pesan kesalahan sintaks database (*SQL Syntax Error*).
Secara teknis, hal ini membuktikan bahwa aplikasi menyusun kueri SQL menggunakan metode penggabungan string (*string concatenation*) secara langsung dari input pengguna, tanpa melalui mekanisme sanitasi atau *parameterized queries*. Dalam terminologi keamanan siber, ini adalah celah **SQL Injection**.

- **Akar Masalah:** Tidak adanya penggunaan fungsi mysqli_real_escape_string() atau *Prepared Statements* pada kode PHP sisi server (*backend*).
- **Implikasi Risiko:** Penyerang dapat memanipulasi input untuk mengeksekusi perintah database yang tidak sah. Skenario terburuk meliputi pencurian seluruh data inventori (*Data Exfiltration*) atau penghapusan tabel secara permanen (*Data Loss*) menggunakan perintah DROP TABLE. Mengacu pada standar ISO/IEC 29119, cacat ini dikategorikan sebagai *Critical Severity* yang mewajibkan penundaan rilis sistem.

Kerentanan Stored Cross-Site Scripting (BUG-002)
Temuan **BUG-002** (terkait TC-26) mengungkap kegagalan sistem dalam menangani input berbasis HTML. Saat penguji memasukkan tag HTML TEBAL, sistem merender teks tersebut menjadi tebal di tabel inventori. Ini mengonfirmasi keberadaan celah **Stored XSS (Cross-Site Scripting)**.

- **Akar Masalah:** Sistem mempercayai input pengguna sepenuhnya dan menampilkannya kembali (*echo*) ke browser tanpa melalui proses *Output Encoding* atau filter karakter khusus (seperti mengubah < menjadi <).
- **Implikasi Risiko:** Meskipun dalam pengujian hanya menggunakan tag HTML sederhana, celah ini memungkinkan penyisipan skrip JavaScript berbahaya. Jika dimanfaatkan oleh pihak tidak bertanggung jawab, skrip tersebut dapat berjalan otomatis di browser administrator untuk mencuri *session cookies* (pembajakan sesi) atau mengalihkan pengguna ke situs *phishing*.

Analisis Kesalahan Logika Bisnis (*Business Logic Error Analysis*)

Selain aspek keamanan, pengujian juga menemukan kelemahan pada validasi aturan bisnis. **BUG-003** (terkait TC-12) menunjukkan bahwa sistem mengizinkan penyimpanan data stok dengan nilai negatif (contoh: -10).

- **Analisis Teknis:** Sistem kemungkinan hanya memvalidasi tipe data input (harus *Integer*), tetapi tidak memvalidasi batasan nilai (*Range Validation*). Dalam skrip PHP, kondisi if ($stok < 0) tidak diimplementasikan sebelum data disimpan.
- **Dampak Operasional:** Dalam konteks manajemen gudang nyata, stok fisik tidak mungkin bernilai negatif. Data stok negatif akan mengacaukan perhitungan aset toko dan laporan audit stok (*stock opname*). Hal ini menurunkan tingkat kepercayaan pengguna terhadap akurasi sistem.

Evaluasi Kualitas Kode Open Source
Hasil audit ini memberikan gambaran empiris mengenai risiko penggunaan kode *open source* dari repositori publik yang tidak terverifikasi. Meskipun secara antarmuka (*User Interface*) sistem terlihat profesional menggunakan *framework* Bootstrap, struktur kode di baliknya (*backend*) sangat rapuh.
Penggunaan kode mentah seperti ini tanpa proses *Quality Assurance* (QA) sangat berbahaya bagi UMKM. Sistem yang diuji terbukti lebih mengutamakan "fitur yang berjalan" (*functional correctness*) dibandingkan "sistem yang aman dan andal" (*secure and robust system*). Temuan-temuan ini menegaskan pentingnya peran pengujian independen berbasis standar seperti ISO/IEC 29119 sebelum sebuah perangkat lunak diadopsi ke lingkungan produksi.

Status Kelayakan Sistem (*System Readiness*)
Berdasarkan kriteria penerimaan (*Acceptance Criteria*) yang ditetapkan dalam *Test Plan*, sebuah sistem dinyatakan layak rilis jika tidak terdapat *bug* dengan status *Critical* atau *Major* yang masih terbuka (*open*).
Mengingat keberadaan **BUG-001 (Critical)** dan **BUG-002 (Major)**, maka sistem inventori ini dinyatakan **TIDAK LAYAK (REJECTED)** untuk diimplementasikan. Diperlukan perbaikan kode (*refactoring*) yang signifikan, terutama pada modul koneksi database dan validasi input, sebelum sistem dapat diuji ulang (*re-testing*).

E. KESIMPULAN

Kesimpulan

Penelitian ini bertujuan untuk melakukan audit kualitas independen terhadap perangkat lunak "Sistem Inventori Toko Sepatu" berbasis web yang bersumber dari repositori publik GitHub, menggunakan standar pengujian ISO/IEC 29119. Berdasarkan serangkaian pengujian *Black Box* yang telah dilaksanakan terhadap 28 skenario uji, dapat ditarik kesimpulan sebagai berikut:

1. **Fungsionalitas Dasar:** Secara umum, fitur utama manajemen data CRUD (*Create, Read, Update, Delete*) berfungsi sesuai spesifikasi dasar. Sistem mampu menampilkan data, memperbarui informasi stok, dan menghapus data dengan benar pada kondisi penggunaan normal (*happy path*). Tingkat keberhasilan fungsional tercatat sebesar 85.7%.
2. **Keamanan Sistem:** Sistem memiliki tingkat keamanan yang **SANGAT RENDAH** dan tergolong **KRITIS**. Ditemukannya celah keamanan *SQL Injection* (BUG-001) dan *Stored Cross-Site Scripting* (BUG-002) membuktikan bahwa kode sumber tidak menerapkan mekanisme sanitasi input dan enkripsi output. Hal ini membuat sistem sangat rentan terhadap serangan peretasan yang dapat menyebabkan kebocoran atau kerusakan data permanen.
3. **Integritas Logika Bisnis:** Sistem gagal memvalidasi logika bisnis yang fundamental. Sistem mengizinkan input stok bernilai negatif (BUG-003), yang secara operasional tidak valid dan dapat merusak akurasi pelaporan inventori.
4. **Kelayakan Rilis:** Berdasarkan kriteria penerimaan (*Acceptance Criteria*) yang mensyaratkan tidak adanya *bug* dengan status *Critical* atau *Major*, maka perangkat lunak ini dinyatakan **TIDAK LAYAK (REJECTED)** untuk digunakan secara langsung di lingkungan produksi (*production environment*) tanpa perbaikan menyeluruh.

Saran

Berdasarkan temuan cacat perangkat lunak di atas, disarankan langkah-langkah perbaikan dan mitigasi sebagai berikut bagi pengembang yang ingin menggunakan atau melanjutkan pengembangan sistem ini:

1. **Perbaikan Keamanan (Security Hardening):**
   a) Wajib mengganti metode kueri database konvensional dengan **Prepared Statements** (menggunakan PDO atau MySQLi) untuk menutup celah *SQL Injection* secara total.
   b) Menerapkan fungsi *escaping* seperti htmlspecialchars() pada setiap data yang ditampilkan ke browser untuk mencegah serangan *XSS*.
2. **Validasi Sisi Server (Server-Side Validation):** Menambahkan logika validasi pada skrip PHP untuk memastikan nilai numerik (seperti Harga dan Stok) tidak boleh kurang dari nol (*non-negative*). Validasi tidak boleh hanya mengandalkan atribut HTML di sisi *client*.
3. **Manajemen Koneksi Database:** Mengimplementasikan blok penanganan kesalahan (*try-catch block*) pada modul koneksi database untuk menangani kegagalan koneksi secara elegan, sehingga pengguna tidak dihadapkan pada pesan kesalahan sistem (*fatal error*) yang membingungkan.
4. **Penerapan Standar QA:** Disarankan untuk selalu menerapkan prosedur pengujian standar (seperti ISO/IEC 29119) sebelum merilis aplikasi, terutama jika kode bersumber dari pihak ketiga, guna meminimalisir risiko kegagalan sistem di kemudian hari.

F. DAFTAR PUSTAKA

Anwar, C., & Riyanto, J. (2019). Perancangan Sistem Informasi Human Resources Development Pada PT. Semacom Integrated. International Journal of Education, Science, Technology, and Engineering (IJESTE), 2(1), 19-38. https://doi.org/10.36079/lamintang.ijeste-0201.16

Anwar, C. ., Sumerli A, C. H. ., Hady, S. ., Rahayu, N. ., & Kraugusteeliana, K. . (2023). The Application of Mobile Security Framework (MOBSF) and Mobile Application Security Testing Guide to Ensure the Security in Mobile Commerce Applications. Jurnal Sistim Informasi Dan Teknologi, 5(2), 97–102. https://doi.org/10.37034/jsisfotek.v5i2.231

Anwar, C. (2024). Rekomendasi Teknis Untuk Pengolahan Data Berbasis Web. Jurnal Informatika Utama, 2(1), 50–54. https://doi.org/10.55903/jitu.v2i1.166

Anwar, C., Jagat, L. S., Yanti, I., Anjarsari, E., & Sholihah, N. A. (2023). Pengembangan media pembelajaran berbasis teknologi untuk meningkatkan kemampuan anak. Caruban: Jurnal Ilmiah Ilmu Pendidikan Dasar, 6(2), 154-163.

Anwar, C. (2022). Application of Academic Information System With Extreme Programming Method (Case Study: Jakarta International Polytechnic).

Anwar, C., Kom, S., Kom, M., Santiari, C. N. P. L., & Sitorus, Z. (2023). Buku Referensi Sistem Informasi Berbasis Kearifan Lokal.

Samsumar, L. D., Nasiroh, S., Farizy, S., Anwar, C., Mursyidin, I. H., Rosdiyanto, R., ... & Prastyo, D. (2025). Keamanan Sistem Informasi: Perlindungan Data dan Privasi di Era Digital

Indra, S., Anwar, C., Kom, S., Asparizal, S., Kom, M., Nur, R. A., ... & Hafrida, L. Komputer dan Masyarakat. CV Rey Media Grafika.

Wijayanti, R. R., S ST, M. M. S. I., Anwar, C., Kom, S., Indra, S., Kom, M., ... & Kom, M. (2023). Arsitektur dan Organisasi Komputer. CV Rey Media Grafika.

Handayani, T., Silalahi, L. M., Nugroho, S. S. P., Anwar, C., Mursyidin, I. H., Sumantri, A., ... & Yulianti, B. (2025). Pengantar Sistem Informasi: Konsep, Teknologi, dan Implementasi.

Anwar, C., & Harits, A. (2025). Perancangan Sistem Kuisioner Penilaian Kapabilitas Framework COBIT 2019. Jurnal Informatika Utama, 3(1), 42-51.

Anwar, C., & Sunardi, D. (2024). Pelatihan Pengembangan Ide Bisnis Inovatif Berbasis Teknologi Informasi Dan Komunikasi (TIK) Untuk Siswa/Siswi Dan Masyarakat Umum Di SMK Nusantara Bojonggede. JIPM: Jurnal Inovasi Pengabdian Masyarakat, 2(2), 53-57.

Samsumar, L. D., Firdaus, M., Windyasari, V. S., Rachendu, S., Anwar, C., Haq, F. A. S. N., ... & Kusumaningrum, A. (2025). Sistem Informasi Manajemen: Strategi, Desain, dan Penerapan.

Handijono, A., Anwar, C., & Harits, A. (2025). Pemanfaatan Penggunaan Sosial Media Dengan Bijak Dalam Teknologi Informasi Di Era Digital Di SMK Media Informatika. Attamkiim: Jurnal Pengabdian Masyarakat, 2(1), 58-64.

Anwar, C., Handijono, A., & Harits, A. (2025). Pemanfaatan Penggunaan Sosial Media Dengan Bijak Dalam Teknologi Informasi Di Era Digital Di SMK Media Informatika. Journal of Community Service Synergy, 1(1), 71-77

Aisyah, S., Anwar, C., Satmoko, N. D., & Nuryanto, U. W. (2023). Role of Product Quality and Store Atmosphere on Purchase Decision of Clothing Product Vintage Vibes. JEMSI (Jurnal Ekonomi, Manajemen, Dan Akuntansi), 9(1), 172-178.

Farizy, S., Trisnawan, A. B., Silalahi, L. M., Yuliadi, B., Anwar, C., Alamsyah, D., ... & Sitorus, B. B. (2025). Buku Ajar Jaringan Komputer: Dari Teori Dasar Hingga Jaringan Nirkabel

TRISNAWAN, A. B., HASANUDIN, M., HANDAYANI, T., ANWAR, C., ZAENUDDIN, I., WAYAHDI, M. R., ... & MARTADINATA, A. T. (2025). Buku Ajar Rekayasa Perangkat Lunak: Prinsip, Praktik, dan Teknologi Modern.

Anwar, C., Ramadhani, G., Aditiya, M. Z., & Sari, P. A. (2025). Pemanfaatan Cloud Computing untuk Solusi Disaster Recovery dan Kontinuitas Bisnis Sistem Informasi Utama (Studi Kasus: Universitas Pamulang). Journal of Information Systems and Business Technology, 1(1), 161-166.

Anwar, C. Prediction Of Academic Achievement Of Pamulang University Students Using Artificial Neural Networks.

Repositori GitHub "depthgilang". (2025). *Source Code inventori sepatu*. Diakses dari https://github.com/depthgilang.