

Pengujian Kualitas Kode Website Pondok Pesantren Modern Menggunakan Standar ISO 5055 (Studi Kasus: Pondok Modern Nurul Hidayah)

*Website Code Quality Testing of Modern Islamic Boarding Schools Using ISO 5055 Standards
(Case Study: Nurul Hidayah Modern Islamic Boarding School)*

Fikrotun Najichah¹, Ryci Rahmatil Fiska²

^{1,2}*Teknik Informatika, Politeknik Negeri Bengkalis*

E-mail: ¹najichahfikrotun@gmail.com., ²rycirf@polbeng.ac.id

Received 31 May 2025; Revised 14 June 2025; Accepted 19 June 2025

Abstrak – Kualitas kode sumber pada sistem informasi berbasis web merupakan faktor krusial dalam menentukan keamanan, keandalan, efisiensi, dan kemudahan pemeliharaan perangkat lunak. Penelitian ini bertujuan untuk mengukur kualitas kode *website* pondok pesantren modern dengan menggunakan standar internasional ISO/IEC 5055, yang menekankan pada kelemahan struktural perangkat lunak. Studi ini menggunakan pendekatan *Software Testing Life Cycle* (STLC) dengan empat aspek evaluasi utama: *Security*, *Reliability*, *Performance Efficiency*, dan *Maintainability*. Tools yang digunakan meliputi SQLMap, PHPUnit, PHP CodeSniffer, dan PHP Metrics. Hasil pengujian menunjukkan bahwa sistem mampu menangkal sebagian besar serangan *SQL Injection* (skor *Security*: 75), seluruh unit pengujian berfungsi dengan baik meskipun terdapat peringatan *deprecated* (*Reliability*: 85), pelanggaran PSR-12 ditemukan pada penamaan dan visibilitas kode (*Performance Efficiency*: 60), serta tingkat kompleksitas kode yang tinggi memengaruhi *maintainability* (skor: 55). Dengan rata-rata skor keseluruhan sebesar 68,75, dapat disimpulkan bahwa kualitas kode berada pada kategori cukup, namun diperlukan *refactoring* dan penerapan standar penulisan kode yang lebih ketat. Hasil penelitian ini diharapkan dapat menjadi acuan awal dalam penerapan ISO 5055 pada sistem informasi pendidikan berskala kecil hingga menengah.

Kata kunci: ISO 5055, Keamanan, Keandalan, Efisiensi Kinerja, Pemeliharaan

Abstract - The quality of source code in web-based information systems is a crucial factor in determining the security, reliability, efficiency, and maintainability of software. This study aims to measure the quality of the code of a modern Islamic boarding school website using the international standard ISO/IEC 5055, which emphasizes structural weaknesses in software. This study uses the *Software Testing Life Cycle* (STLC) approach with four main evaluation aspects: *Security*, *Reliability*, *Performance Efficiency*, and *Maintainability*. The tools used include SQLMap, PHPUnit, PHP CodeSniffer, and PHP Metrics. The test results show that the system is able to ward off most *SQL Injection* attacks (*Security* score: 75), all test units function properly even though there are deprecated warnings (*Reliability*: 85), PSR-12 violations are found in code naming and visibility (*Performance Efficiency*: 60), and the high level of code complexity affects maintainability (score: 55). With an average overall score of 68.75, it can be concluded that the code quality is in the sufficient category, but refactoring and implementation of stricter code writing standards are needed. The results of this study are expected to be an initial reference in the implementation of ISO 5055 in small to medium-scale educational information systems.

Keywords: ISO 5055, Security, Reliability, Performance Efficiency, Maintainability

1. PENDAHULUAN

Dalam era digital saat ini, kualitas perangkat lunak memegang peranan penting dalam menentukan keandalan dan keamanan layanan berbasis web [1]. *Website* tidak hanya menjadi wajah utama suatu institusi, tetapi juga berfungsi sebagai media komunikasi, penyedia layanan informasi, dan sarana administrasi digital [2]. Oleh karena itu, penting bagi pengembang untuk memastikan kualitas kode perangkat lunak agar sistem dapat berfungsi optimal, aman, dan mudah dipelihara [3], [4].

Website Pondok Pesantren modern umumnya menyediakan informasi profil, kegiatan pendidikan, dan pendaftaran santri. Salah satu contohnya adalah *website* Pondok Modern Nurul Hidayah yang digunakan sebagai studi kasus dalam penelitian ini. Namun, sebagaimana banyak sistem informasi pendidikan lainnya, *website* ini belum memiliki dokumentasi pengujian kualitas kode yang sistematis dan objektif [5]. Kelemahan seperti struktur kode yang kompleks, potensi celah keamanan, dan ketidaksesuaian standar pengembangan perangkat lunak dapat menurunkan pengalaman pengguna dan menimbulkan risiko di sisi pemeliharaan jangka panjang [6].

Pemilihan *website* Pondok Modern Nurul Hidayah sebagai objek penelitian didasarkan pada kenyataan bahwa banyak institusi pendidikan kecil dan menengah di Indonesia menghadapi tantangan serupa, yaitu keterbatasan sumber daya teknis dan belum adanya penerapan standar pengujian kode yang sistematis. Meskipun terlihat sederhana, *website* ini mencerminkan struktur dan fungsionalitas umum yang juga dimiliki oleh sistem informasi pendidikan lainnya, seperti pendaftaran *online*, penyajian data santri, dan pengelolaan konten publik. Oleh karena itu, pengujian kualitas kode pada sistem ini dinilai relevan untuk memberikan gambaran nyata tentang bagaimana standar internasional seperti ISO 5055 dapat diimplementasikan pada konteks lokal yang sering kali terabaikan dalam penelitian akademik.

Selain itu, meskipun tampak sederhana, *website* Pondok Modern Nurul Hidayah memiliki fitur yang kompleks, termasuk sistem pendaftaran *online* berbasis formulir, autentikasi pengguna melalui *login*, interaksi dengan *database* MySQL, serta modul *backend* yang terintegrasi dengan *framework* *CodeIgniter*. Fitur-fitur ini mencerminkan kompleksitas struktural yang umum ditemukan pada sistem informasi pendidikan berskala kecil hingga menengah di Indonesia. Oleh karena itu, sistem ini merupakan objek yang representatif untuk pengujian menggunakan standar ISO 5055 yang berfokus pada kelemahan struktural kode, bukan hanya dari sisi ukuran atau kompleksitas organisasi, melainkan dari sudut pandang kualitas teknis kode dan *maintainability*-nya.

ISO 5055 merupakan standar internasional yang dikembangkan untuk menilai kualitas perangkat lunak berdasarkan empat karakteristik utama: *Security*, *Reliability*, *Performance Efficiency*, dan *Maintainability*. Standar ini berfokus pada deteksi kelemahan struktural dalam kode sumber perangkat lunak, dan berbeda dari ISO 25010 yang lebih menekankan pada karakteristik kualitas dari sisi pengguna akhir [7]. ISO 5055 memberikan pendekatan berbasis metrik dan automasi dalam mengevaluasi kualitas kode secara internal, sehingga sangat relevan digunakan dalam konteks pengujian perangkat lunak berbasis web [8]. Pendekatan serupa juga banyak digunakan dalam praktik evaluasi perangkat lunak, khususnya dalam pengujian berbasis otomatisasi yang memungkinkan identifikasi berbagai potensi kelemahan sistem. Melalui metode ini, hasil evaluasi dapat dianalisis lebih lanjut untuk mendukung upaya peningkatan kualitas dan keamanan perangkat lunak secara menyeluruh [9].

Beberapa penelitian sebelumnya telah menunjukkan pentingnya evaluasi kualitas kode menggunakan standar internasional. Penelitian *Measuring the Trustworthiness of Software with ISO/IEC 5055* oleh Bill Curtis membahas penggunaan ISO/IEC 5055:2021 untuk mengukur *trustworthiness* perangkat lunak secara otomatis dari kode sumber. Dengan menganalisis kelemahan struktural seperti *reliability* dan *security*, standar ini membantu mendeteksi risiko sejak tahap pengembangan. ISO 5055 menggunakan 138 kelemahan dari CWE dan dapat diterapkan dalam evaluasi kualitas, kontrak proyek, serta perbaikan berkelanjutan [7]. Penelitian oleh Rumabar dan Maria (2024) menggunakan ISO/IEC 25010 untuk mengevaluasi *ShopeePay* dan menunjukkan pentingnya *Security* dan *Performance Efficiency* dalam sistem pembayaran digital [10]. Penelitian pengujian aplikasi sistem monitoring perkuliahan menggunakan standar ISO 25010 menunjukkan bahwa aplikasi monitoring perkuliahan dengan kualitas *Maintainability* tinggi lebih mudah dikembangkan dan diperbaiki dalam jangka panjang [11]. Penelitian lain yang melakukan evaluasi kualitas sistem jurnal elektronik berbasis *open journal system* menggunakan ISO/IEC 25010 menggarisbawahi bahwa meskipun sistem jurnal OJS memiliki kinerja dan fungsionalitas yang baik, aspek keamanan masih menjadi titik lemah yang memerlukan perhatian khusus [12]. Penelitian-penelitian ini umumnya menggunakan ISO/IEC 25010, namun belum banyak yang mengadopsi ISO 5055 secara spesifik sebagai acuan evaluasi kualitas kode.

Berdasarkan tinjauan pustaka, masih sangat sedikit penelitian yang secara eksplisit menerapkan ISO 5055 pada sistem informasi pendidikan skala kecil-menengah. Sebagian besar studi sebelumnya lebih banyak menggunakan ISO 25010 yang berorientasi pada kualitas dari sudut pandang pengguna akhir. Hal ini menunjukkan adanya kesenjangan penelitian (*research gap*) terkait evaluasi kualitas dari aspek struktural kode sumber yang lebih teknis, yang dapat dijumpai melalui pendekatan ISO 5055.

Dengan demikian, penelitian ini berkontribusi dalam dua hal utama. Pertama, penelitian ini menerapkan standar ISO 5055 yang relatif baru dan belum banyak digunakan dalam pengujian sistem pendidikan berskala kecil-menengah di Indonesia, berbeda dengan penelitian terdahulu yang berfokus pada kualitas dari perspektif pengguna akhir. ISO 5055 memberikan pendekatan teknis yang lebih dalam melalui deteksi kelemahan struktural dalam kode sumber menggunakan metrik *Common Weakness Enumeration*, sehingga hasil evaluasinya menjadi lebih objektif, terukur, dan langsung ke akar masalah kualitas perangkat lunak dari sisi pengkodean internal. Kedua, penelitian ini menyusun kerangka pengujian praktis berbasis alat bantu otomatis, seperti SQL Map, PHPUnit, PHP *CodeSniffer*, dan PHP *Metrics* yang mudah diakses, *open-source*, serta dapat diterapkan di institusi dengan sumber daya terbatas. Pendekatan ini menekankan pada *white-box testing* berbasis kode *actual*, memberikan hasil yang lebih teknis dan *actionable* untuk *refactoring* kode dan peningkatan *maintainability*.

Selain itu, penelitian ini secara eksplisit mengaitkan hasil pengujian dengan nilai kuantitatif (skor 0-100) berdasarkan keberhasilan pengujian dan metrik struktural seperti *complexity* > 20, yang jarang ditemukan dalam studi sebelumnya. Penilaian kuantitatif ini membuat hasil evaluasi lebih transparan, replikatif, dan mudah di bandingkan antara proyek yang satu dengan lainnya, serta membuka ruang untuk pembuatan *baseline* kualitas kode di sektor pendidikan lokal.

Melalui pendekatan ini, diharapkan hasil pengujian dapat memberikan dampak nyata dalam peningkatan kualitas perangkat lunak berbasis web di lingkungan pendidikan, serta mendorong penerapan standar internasional yang lebih luas di sektor pengembangan sistem informasi lokal.

2. METODE PENELITIAN

ISO 5055 berbeda dengan ISO/IEC 25010 yang lebih fokus pada karakteristik eksternal seperti *usability* dan *functionality*. ISO 5055 menilai kualitas dari dalam kode sumber secara struktural melalui metrik berbasis kelemahan umum (*Common Weakness Enumeration/CWE*), seperti tingkat kompleksitas, kesalahan autentikasi, atau struktur program yang tidak efisien. Karena fokusnya pada analisis teknis dan *maintainability*, ISO 5055 lebih sesuai digunakan dalam konteks audit kode sistem pendidikan kecil seperti *website* pondok pesantren yang biasanya digunakan tanpa pedoman kualitas formal.

Website pondok modern nurul hidayah merupakan portal berbasis web yang menyediakan informasi pendaftaran santri, profil lembaga, berita kegiatan, dan manajemen data santri. Sistem ini dibangun menggunakan *framework CodeIgniter* dan basis data MySQL. Dengan fungsionalitas seperti login pengguna, validasi input form, dan manajemen konten. Sistem ini mewakili struktur umum sistem informasi pondok pesantren berskala kecil hingga menengah di Indonesia.

Secara umum, *website* pondok pesantren modern memiliki karakteristik yang khas, yaitu menyajikan fitur-fitur seperti formulir pendaftaran santri online, jadwal kegiatan keagamaan, pengumuman hasil seleksi, dan informasi kurikulum. Fitur-fitur ini memerlukan pengolahan input dari pengguna, interaksi dengan basis data, dan kontrol autentikasi, sehingga memiliki tantangan teknis yang serupa dalam aspek keamanan, keandalan, dan kemudahan pemeliharaan kode. Hal ini menjadikan *website* pondok pesantren sebagai objek penelitian yang unik dan penting untuk diteliti menggunakan pendekatan berbasis ISO 5055.

Metode penelitian yang digunakan pada penelitian ini adalah *Software Testing Life Cycle* (STLC), Berikut tahapannya:



Gambar 1 *Software Testing Life Cycle* (STLC)

Penelitian ini menggunakan metode *Software Testing Life Cycle* (STLC). Penggunaan metode *Software Testing Life Cycle* (STLC) untuk pengujian kualitas website Pondok Modern Nurul Hidayah dengan standar ISO 5055 yang memiliki banyak keuntungan. Metode ini memberikan kerangka kerja yang teratur, memastikan setiap tahap pengujian dilakukan dengan baik, dari perencanaan hingga pelaporan. Ini membantu menemukan masalah lebih awal, sehingga dapat diperbaiki lebih cepat dan dengan biaya yang lebih rendah. STLC memastikan semua aspek kualitas terpenuhi. Secara keseluruhan, STLC membuat proses pengembangan lebih efisien dan terstruktur.

1. *Requirement Analysis*: analisis kebutuhan dilakukan untuk menentukan jenis dan Teknik pengujian berdasarkan empat karakteristik kualitas perangkat lunak dalam standar ISO 5055:
 - a) *Security*, diuji menggunakan *sqlmap* untuk mendeteksi kerentanan keamanan seperti *SQL Injection*.
 - b) *Reliability*, diuji menggunakan *PHPUnit* untuk menilai stabilitas dan efisiensi *backend*.
 - c) *Performance Efficiency*, diuji menggunakan *PHP CodeSniffer* untuk mengidentifikasi kesalahan standar kode.
 - d) *Maintainability*, diuji menggunakan *PHP Metrics* untuk menilai kompleksitas, *bug*, dan kemudahan pemeliharaan kode.
2. *Test Planning*: langkah ini meliputi penentuan strategi, pemilihan alat pengujian (*sqlmap*, *PHPUnit*, *PHP CodeSniffer*, *PHP Metrics*), penyusunan jadwal, alokasi sumber daya, dan dokumentasi rencana pengujian.
3. *Test Case Development*: *test case* dirancang dan didokumentasikan untuk masing-masing karakteristik. Contoh pengujian:
 - a) *Security*, pengujian *SQL Injection*, validasi input, proteksi *CSRF*. Berikut *test case*-nya

Tabel 1 *Test Case Security*

No	Test Case	Hasil yang diharapkan	Ya	Tidak
1	<i>SQL Injection</i> pada <i>Login Form</i>	Sistem menolak <i>login</i> dan memberikan pesan <i>error</i> yang aman	✓	
2	<i>Validasi Input Login</i>	Sistem menolak input tidak valid dan memberikan pesan <i>error</i> aman	✓	
3	Pengujian <i>Brute Force</i>	Sistem memblokir permintaan setelah beberapa upaya gagal	✓	
4	Validasi Proteksi <i>CSRF</i>	Server menolak permintaan tanpa token <i>CSRF</i>	✓	
5	Mekanisme <i>Logging</i> dan Pesan <i>Error</i>	Pesan error tidak mengungkapkan informasi teknis	✓	
6	Audit Akses Halaman yang Dilindungi	Pengguna diarahkan ke halaman login jika belum terautentikasi	✓	
7	Keamanan <i>Directories</i>	Server menolak akses langsung atau menampilkan error ' <i>403 Forbidden</i> '	✓	

8	Content Security Policy (CSP)	Browser memblokir eksekusi skrip eksternal yang tidak tepercaya	✓	
---	-------------------------------	---	---	--

b) *Reliability*, waktu respons model/API, penanganan null pointer. Berikut test case-nya:

Tabel 2 Test Case Reliability

No	Test Case	Hasil yang diharapkan	Ya	Tidak
1	Uji waktu respon fungsi model	Waktu <i>respons</i> < 1 detik	✓	
2	Uji waktu eksekusi <i>query database</i>	Waktu eksekusi <i>query</i> < 500ms	✓	
3	Uji waktu respon API	API merespon dengan cepat dan valid	✓	
4	Uji penanganan Null Pointer Exception	Tidak terjadi <i>Null Pointer Exception</i>	✓	
5	Uji efisiensi pemrosesan data (dataset besar)	Pemrosesan data dalam waktu < 2 detik	✓	

c) *Performance Efficiency*, struktur penamaan dan visibilitas kode. Berikut test case-nya:

Tabel 3 Test Case Performance Efficiency

No	Test Case	Hasil yang diharapkan	Ya	Tidak
1	Memeriksa deklarasi simbol baru (kelas, fungsi, dll.)	Tidak ada efek samping logika atau kesalahan lainnya	✓	
2	Memeriksa pemisahan <i>header block</i> dengan satu baris kosong	Setiap <i>blok header</i> dipisahkan dengan baris kosong tunggal	✓	
3	Memeriksa format penamaan kelas Agenda_m	Nama kelas sesuai format <i>PascalCase</i>	✓	

4	Memeriksa properti \$table dengan awalan <i>underscore</i>	Properti \$table diawali dengan <i>underscore</i>	✓	
5	Memeriksa deklarasi visibilitas untuk <i>method</i> data	<i>Method</i> data dideklarasikan dengan visibilitas yang benar	✓	

d) *Maintainability*, ketaatan terhadap standar format kode. Berikut *test case*-nya:

Tabel 4 *Test Case Maintainability*

No	<i>Test Case</i>	Hasil yang diharapkan	Ya	Tidak
1	File Harus Mendeklarasikan Simbol Baru (Kelas, Fungsi, Konstanta)	Hanya deklarasi simbol baru yang ada, tanpa logika dengan efek samping lain.	✓	
2	<i>Blok Header</i> Dipisahkan dengan Satu Baris Kosong	Setiap <i>blok header</i> dipisahkan dengan satu baris kosong.	✓	
3	Karakter Akhir Baris yang Tidak Valid	Karakter akhir baris harus \n, tanpa \r\n.	✓	
4	Nama Kelas Tidak Sesuai Format <i>PascalCase</i>	Nama kelas mengikuti format <i>PascalCase</i> (misal: EkstrakurikulerM).	✓	
5	Properti Tidak Dideklarasikan dengan Visibilitas yang Tepat	Properti harus dideklarasikan dengan visibilitas yang tepat dan tidak menggunakan <i>underscore</i> untuk mendefinisikan visibilitas.	✓	
6	Metode Tidak Dideklarasikan dengan Visibilitas yang Tepat	Metode harus dideklarasikan dengan visibilitas yang tepat.	✓	

7	File Harus Diakhiri dengan Satu Baris Baru	File PHP harus diakhiri dengan satu baris baru (\n).	✓	
---	--	--	---	--

4. *Enviroinment*: spesifikasi pengujian meliputi perangkat keras (Intel i5, RAM 8 GB) dan perangkat lunak (*Windows 10*, *VScode*, *sqlmap*, *PHPUnit*, *PHP CodeSniffer*, *PHP Metrics*).
5. *Test Excecution*: pengujian dilakukan dengan menjalankan *test case*:
 - a) *Security*: halaman *login* diuji terhadap *SQL Injection* dan *bypass login*, hasil menunjukkan sistem cukup aman dengan pemblokiran akses ilegal.
 - b) *Reliability*: model *Agenda_m.php* diuji dengan *PHPUnit* dan hasilnya menunjukkan stabilitas tanpa *error*.
 - c) *Performance Efficiency*: hasil dari *PHP CodeSniffer* menunjukkan beberapa pelanggaran pada format penulisan kode.
 - d) *Maintainability*: hasil dari *PHP Metrics* menunjukan kompleksitas kode yang cukup tinggi (nilai *complexity*: 23.34), menandakan perlunya *refactoring*.
6. *Test Cycle Closure*: tahapan akhir ini berupa penyusunan laporan hasil pengujian, evaluasi proses, dan simpulan terhadap kualitas sistem berdasarkan ISO 5055. Hasil pengujian digunakan sebagai dasar perbaikan kode dan peningkatan kualitas *website*.

Metode pengujian yang diusulkan dalam penelitian ini tidak hanya menyelaraskan dengan standar ISO 5055, tetapi juga dirancang agar mudah diadopsi oleh institusi kecil dengan keterbatasan teknis. Pendekatan ini menggabungkan alat uji *open-source* dan skenario pengujian berbasis kasus nyata, sehingga dapat berfungsi sebagai model awal bagi penelitian serupa di lingkungan pendidikan.

ISO 5055 mendasarkan penelitiannya pada daftar kelemahan struktural yang telah diklasifikasikan oleh CWE (*Common Weakness Enumeration*). Kelemahan tersebut mencakup berbagai jenis kesalahan umum dalam pemrograman seperti *SQL Injection*, *Null Pointer*, dan tingginya kompleksitas logika. Beberapa alat pengujian seperti *PHP Metrics* dan *PHP CodeSniffer* memberikan hasil dalam bentuk angka atau jumlah pelanggaran, yang dapat digunakan untuk menilai kualitas kode. Sebagai contoh, nilai *Cyclomatic Complexity* sebesar 23.34 menandakan bahwa tingkat kompleksitas logika dalam satu fungsi atau modul cukup tinggi.

3. HASIL DAN PEMBAHASAN

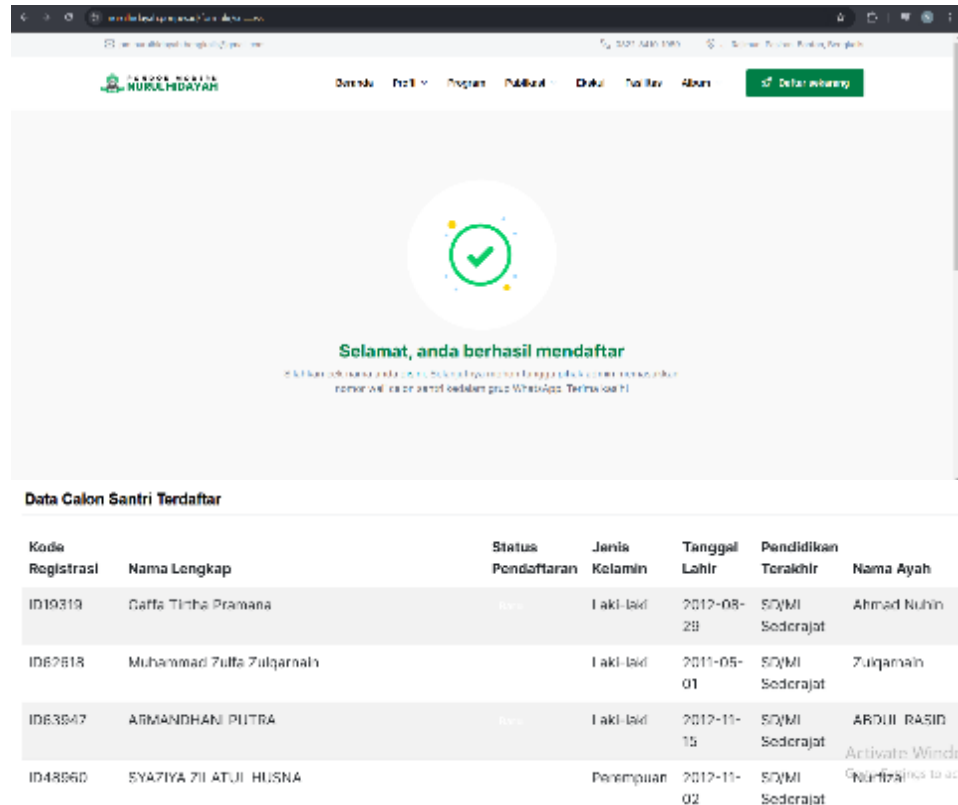
Penelitian ini melakukan pengujian terhadap kualitas kode *website* Pondok Modern Nurul Hidayah berdasarkan empat karakteristik utama ISO 5055:

3.1 *Test Execution*

1. Hasil pengujian *Security*

Berdasarkan ISO/IEC 5055, aspek *security* menilai kelemahan struktural seperti kontrol akses yang tidak memadai (CWE–285) dan kerentanan injeksi seperti *SQL Injection*. Kelemahan autentikasi pada halaman pendaftaran mencerminkan kurangnya kontrol akses yang sesuai. Sementara itu, sistem yang berhasil menolak serangan injeksi menunjukkan penerapan mitigasi struktural yang sesuai dengan kriteria ISO 5055.

- a) Kesalahan otentikasi pada akses halaman pengisian formulir pendaftaran

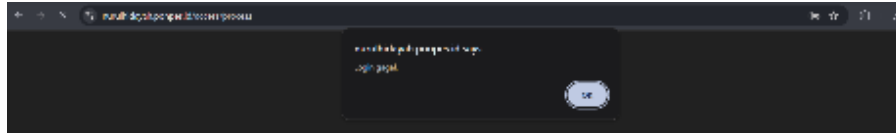


Gambar 2 Pengisian Formulir Pendaftaran

Ditemukan kelemahan pada halaman pendaftaran (/formulir/register) yang dapat diakses tanpa autentikasi. Hal ini menunjukkan tidak adanya kontrol akses pada *resource* yang semestinya dilindungi. Selain itu, pengujian menggunakan *sqlmap* memicu respon 403 *Forbidden*, yang menunjukkan adanya *firewall* atau proteksi server terhadap serangan otomatis. Secara keseluruhan, sistem memiliki perlindungan cukup baik, namun perlu perbaikan pada akses kontrol halaman publik.

b) Bypass Login Form

Gambar 3 Form Login



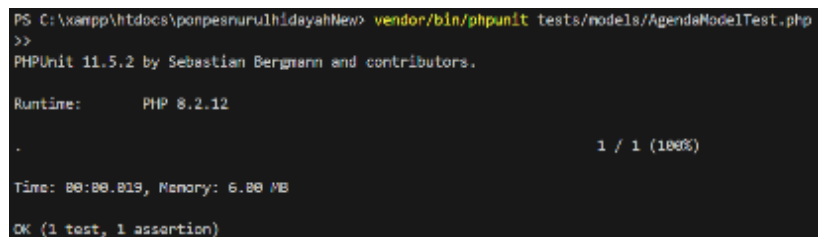
Gambar 4 Login Gagal

Hasil pengujian keamanan menunjukkan bahwa sebagian besar serangan berbasis SQL Injection berhasil difilter oleh sistem. Uji input manipulatif seperti : ' or '1'=1 | admin' or '1'=1'-- | admin") or ("1"="1"-- | ' or '1'=1) gagal melewati otentikasi, menandakan adanya filter input yang efektif, kemungkinan besar difasilitasi oleh *framework CodeIgniter*.

2. Hasil pengujian *Reliability*

Dalam ISO 5055, *reliability* merujuk pada kemampuan sistem untuk berfungsi secara benar dan stabil dalam berbagai kondisi operasional. Uji unit yang berhasil tanpa error menunjukkan tidak adanya kelemahan struktural terkait kesalahan fatal (CWE-248) maupun *exception runtime* yang tidak ditangani. Peringatan *deprecation* menandakan adanya potensi risiko masa depan jika tidak diperbarui, namun saat ini belum menimbulkan gangguan fungsional [7][10].

Pengujian *reliability* menggunakan PHPUnit menunjukkan bahwa semua fungsi dalam model AgendaModelTest.php, berhasil diuji dengan status lulus. Tidak ditemukan *error* fatal dalam proses pengujian unit. Namun, terdapat peringatan *deprecation* yang menunjukkan penggunaan kode atau fungsi yang tidak lagi direkomendasikan pada versi PHP terbaru. Hal ini perlu menjadi perhatian untuk menjaga kompatibilitas jangka panjang.



Gambar 5 Hasil Pengujian

Pengujian performa fungsi model, waktu eksekusi *query*, dan API *response* semuanya berada dalam batas wajar (< 2 detik). Ini menandakan bahwa *website* cukup handal dan stabil dalam berbagai skenario pengujian. Meskipun penelitian ini berfokus pada ISO 5055, hasil ini mendukung prinsip yang dikemukakan dalam ISO 25010 seperti yang diteliti oleh Rumabar dan Maria [10], di mana aspek *reliability* dianggap penting untuk menjaga performa jangka panjang sistem digital. Penelitian ini menunjukkan bahwa ISO 5055 juga mampu mengidentifikasi aspek keandalan melalui pendekatan berbasis pengujian unit dan stabilitas fungsi.

3. Hasil pengujian *Performance Efficiency*

Aspek *performance efficiency* dalam ISO 5055 menilai efisiensi struktural kode, termasuk penulisan, konsistensi visibilitas, dan standar pemrograman. Pelanggaran PSR-12 seperti penamaan tidak sesuai dan visibilitas metode yang tidak dideklarasikan termasuk dalam kategori CWE-398 (*Code Quality Issue*) dan menunjukkan rendahnya efisiensi struktural kode. Hal ini dapat memperlambat pemrosesan dan menyulitkan optimasi performa dalam jangka Panjang [7][12].

```
PS C:\xampp\htdocs\parpen\cuhiday\wms\..> vendor/bin/phpcs --standard=PSR12 application\models\Agenda_n.php

FILE: C:\xampp\htdocs\parpen\cuhiday\wms\application\models\Agenda_n.php
-----
FOUND 8 ERRORS AND 2 WARNINGS AFFECTING 6 LINES
-----
 1 | WARNING | [ ] A file should declare new symbols (classes, functions,
   |          | constants, etc.) and cause no other side effects, or it
   |          | should execute logic with side effects, but should not do
   |          | both. The first symbol is defined on line 4 and the first
   |          | side effect is on line 3.
 2 | ERROR   | [x] Header blocks must be separated by a single blank line
 3 | ERROR   | [x] End of line character is invalid; expected "\n" but found
   |          | "\r\n"
 4 | ERROR   | [ ] Each class must be in a namespace of at least one level (a
   |          | top level vendor name)
 4 | ERROR   | [ ] Class name "Agenda_n" is not in PascalCase format
 6 | WARNING | [ ] Property name "$ table" should not be prefixed with an
   |          | underscore to indicate visibility
50 | ERROR   | [ ] Visibility must be declared on method "data"
55 | ERROR   | [ ] Method name "Agenda_n::jurnal_data()" is not in camel caps
   |          | format
55 | ERROR   | [ ] Visibility must be declared on method "jurnal_data"
61 | ERROR   | [x] Expected 1 newline at end of file; 0 found
-----
PHPCS CAN FIX THE 5 MARKED SHEFF VIOLATIONS AUTOMATICALLY
-----

Time: 5.38 secs, Memory: 8MB
```

Gambar 6 Hasil Pengujian pada File agenda m.php

Dalam melakukan pengujian kode ditemukan beberapa pelanggaran, sebagai berikut:

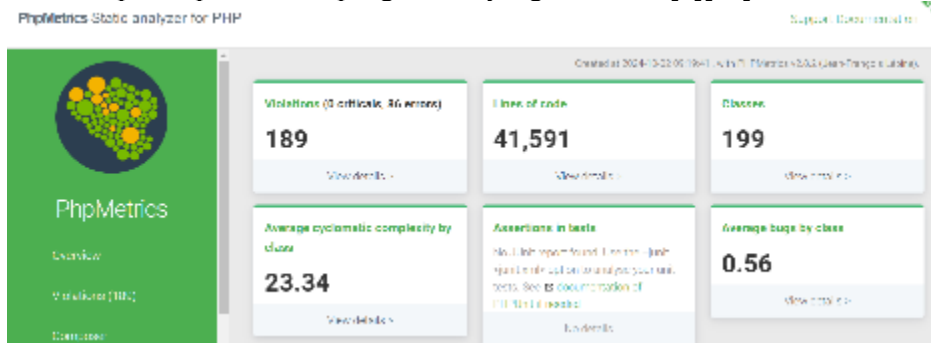
Tabel 5 Temuan Pelanggaran Format Kode (PSR-12)

File	Pelanggaran	Rekomendasi
agenda_m.php	Nama kelas tidak <i>PascalCase</i>	Gunakan AgendaM
agenda_m.php	Metode tanpa visibilitas	Tambahkan pulic
agenda_m.php	Tidak ada baris kosong di akhir file	Tambahkan \n
agenda_m.php	Karakter akhir \r\n	Ubah ke \n

Penemuan pelanggaran terhadap standar PSR-12 dalam penelitian ini sejalan dengan penelitian Murdiani dan Umar [12] yang mengidentifikasi struktur kode sebagai faktor penting dalam performa sistem jurnal OJS. Hal ini menunjukkan bahwa efisiensi kode tidak hanya penting pada sistem besar, tetapi juga berdampak pada sistem pendidikan berskala kecil.

4. Hasil pengujian *Maintainability*

Maintainability dalam ISO 5055 dikaitkan dengan kompleksitas logika dan struktur kode yang mendukung pemeliharaan jangka panjang. Nilai *Cyclomatic Complexity* yang tinggi (23.34) menunjukkan kelemahan dalam modularitas (CWE-710: *Excessive Code Complexity*). Banyaknya pelanggaran standar format kode juga memperburuk *maintainability* sistem. Hal ini mengindikasikan perlunya *refactoring*, dokumentasi yang lebih baik, dan penerapan standar pengkodean yang konsisten [7][11].



Gambar 7 Pengujian PHP *Metrics*

Pengujian menggunakan PHP *Metrics* menunjukkan bahwa nilai *Cyclomatic Complexity* kode mencapai 23,34, yang dikategorikan tinggi. Nilai ini mengindikasikan bahwa bagian-bagian tertentu dalam kode memiliki logika yang rumit, sehingga berpotensi menyulitkan pemahaman dan pemeliharaan oleh pengembang lain. Selain itu, ditemukan sejumlah *code violations* dan potensi *bug* pada beberapa file, yang menunjukkan perlunya proses *refactoring*, serta peningkatan dokumentasi, kompleksitas yang tinggi dapat memperlambat *debugging*, meningkatkan resiko kesalahan selama pengembangan, serta menyulitkan integrasi atau migrasi fitur baru.

Temuan ini sejalan dengan Lamada et al. [11] yang menunjukkan bahwa *maintainability* tinggi memudahkan proses pengembangan dan pemeliharaan jangka panjang. Meskipun mereka menggunakan ISO 25010, studi ini membuktikan bahwa ISO 5055 dapat memberikan deteksi struktural yang lebih terukur melalui metrik seperti *Cyclomatic Complexity*.

Tabel 6 Rekapitulasi Hasil Pengujian Kode Website Berdasarkan Aspek ISO 5055

Aspek ISO 5055	Alat Uji	Hasil Utama	Penilaian
<i>Security</i>	SQL Map	Sistem berhasil menolak sebagian besar SQLi, celah autentikasi ditemukan	Cukup, perlu perbaikan
<i>Reliability</i>	PHPUnit	Semua unit test lulus, ada warning deprecated	Baik, perlu update kode
<i>Performance Efficiency</i>	PHP CodeSniffer	Banyak pelanggaran PSR-12, format penamaan dan visibilitas tidak konsisten	Kurang, perlu <i>refactor</i>
<i>Maintainability</i>	PHP Metrics	Nilai <i>complexity</i> = 23.34, banyak kode <i>violation</i>	Rendah, perlu <i>refactor</i>

3.3 Test Cycle Closure

a) *Security*

Sistem telah mampu menangkal upaya SQL *Injection* dan *bypass login* berkat filterisasi input yang baik serta proteksi dari *Web Application Firewall* (WAF). Namun, terdapat kelemahan pada halaman pendaftaran yang dapat diakses tanpa autentikasi, berpotensi menimbulkan risiko kebocoran data.

b) *Reliability*

Seluruh model berhasil diuji menggunakan PHPUnit dengan hasil lulus 100%, namun ditemukan sejumlah peringatan *deprecation* yang menunjukkan penggunaan fungsi usang. Waktu pengujian tergolong cepat dan efisien, meskipun cakupan pengujian masih dapat ditingkatkan.

c) *Performance Efficiency*

Pengujian dengan PHP *CodeSniffer* menunjukkan bahwa sebagian besar file tidak mematuhi standar penulisan PSR-12, seperti ketidaksesuaian penamaan, absennya *namespace*, dan kurangnya deklarasi visibilitas metode, yang dapat mempersulit pemeliharaan.

d) *Maintainability*

Hasil dari PHP *Metrics* mengindikasikan kompleksitas kode yang cukup tinggi, banyaknya *dependencies* antar kelas, serta jumlah pelanggaran standar penulisan yang signifikan. Meskipun struktur komentar tergolong lengkap, hal ini belum cukup untuk mengimbangi kompleksitas logika kode. Oleh karena itu, perlu dilakukan *refactoring* dan penerapan

standar pengkodean yang lebih ketat untuk meningkatkan *maintainability* secara keseluruhan.

Tabel 7 Kesimpulan Hasil dari Pengujian

Aspek ISO 5055	Hasil Pengujian	Skor(0-100)	Keterangan
<i>Security</i>	Menangkal <i>SQL Injection</i> dan <i>bypass login</i> dengan baik, namun halaman pendaftaran terbuka tanpa autentikasi.	75/100	Keamanan dasar sudah baik, namun ada potensi kebocoran data yang perlu ditangani.
<i>Reliability</i>	Pengujian PHPUnit berhasil 100%, namun ada peringatan deprecation.	85/100	Fungsionalitas baik, tetapi ada penggunaan fungsi usang yang perlu diperbarui.
<i>Performance Efficiency</i>	Banyak pelanggaran standar PSR-1 seperti penamaan dan visibilitas metode.	60/100	Penulisan kode tidak efisien dan menyulitkan pemeliharaan
<i>Maintainability</i>	Kompleksitas kode tinggi, banyak <i>dependencies</i> , dan pelanggaran standar, komentar cukup lengkap namun tidak menutupi kompleksitas.	55/100	Perlu <i>refactoring</i> besar dan penerapan standar yang lebih ketat

4. KESIMPULAN DAN SARAN

Berdasarkan pengujian kualitas kode website Pondok Modern Nurul Hidayah menggunakan standar ISO 5055, diperoleh skor rata-rata 68,75 dari empat karakteristik utama. Aspek keamanan (*Security*) menunjukkan perlindungan cukup baik terhadap serangan *SQL Injection*, meskipun ditemukan celah autentikasi pada halaman pendaftaran (skor: 75). Aspek Keandalan (*Reliability*) meraih skor tertinggi yaitu 85, dengan seluruh unit berhasil diuji tanpa error, namun masih terdapat fungsi usang yang menimbulkan peringatan *deprecated*. Efisiensi kinerja (*Performance Efficiency*) hanya mendapat skor 60 karena banyak pelanggaran standar penulisan kode PSR-12. Maintainability menjadi aspek terendah dengan skor 55, disebabkan oleh tingginya kompleksitas kode dan banyaknya pelanggaran struktur kode.

Secara keseluruhan, kualitas kode berada dalam kategori cukup, namun perlu peningkatan signifikan khususnya pada aspek *maintainability* dan efisiensi kode untuk memastikan keberlanjutan dan skalabilitas sistem.

DAFTAR PUSTAKA

- [1] C. Kartiko, "Evaluasi Kualitas Aplikasi Web Pemantau Menggunakan Model Pengujian Perangkat Lunak ISO/IEC 9126," 2019.
- [2] B. Devi, A. Iqbal, S. Fadli, and W. Murniati, "Analisa Website Desa Darmaji Menggunakan ISO/IEC 25010 (Studi Kasus : Website Desa Darmaji)," *INNOVATIVE: Journal Of Social Science Research*, 2024.
- [3] R. A. Wijaya and K. Karmilasari, "Pengukuran Kualitas Website Pengurus Cabang NU Depok Menggunakan Software Metric," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 10, no. 3, pp. 438–443, Dec. 2021, doi: 10.32736/sisfokom.v10i3.1267.
- [4] A. Ghaita, R. Pertiwi, M. K. Sabariah, and M. Adrian, "Penjaminan mutu perangkat lunak pada project Aplikasi Manajemen Informasi Fakultas Industri Kreatif(MI-FIK) di

- Universitas Telkom Berdasarkan ISO 25010,” *Jurnal Penelitian Informatika*, vol. 2, pp. 32–39, 2024, doi: 10.25124/logic.v2i1.7420.
- [5] A. Ijudin and A. Saifudin, “Pengujian Black Box pada Aplikasi Berita Online dengan Menggunakan Metode Boundary Value Analysis,” vol. 5, no. 1, 2020, [Online]. Available: <http://openjournal.unpam.ac.id/index.php/informatika>
- [6] N. Ratnadhita, Y. Sudianto, and A. Kusumawati, “ISO/IEC 25010 : Analisis Kualitas Sistem E-learning sebagai Media Pembelajaran Online,” *Journal of Information System, Graphics, Hospitality and Technology*, vol. 5, no. 1, pp. 8–20, Mar. 2023, doi: 10.37823/insight.v5i1.302.
- [7] B. Curtis, “Measuring the Trustworthiness of Software with ISO/IEC 5055,” Jul. 2022. [Online]. Available: <https://www.iiconsortium.org/foundational-publications.htm>
- [8] Washington Technology Solutions (WaTech), “SOFTWARE QUALITY BEST PRACTICES GUIDELINE,” Jun. 2023.
- [9] K. Nisa, A. Putra, R. A. Siregar, and M. Dedi Irawan, “Analisis Website Tapanuli Tengah Menggunakan Metode Open Web Application Security Project Zap (Owasp Zap),” vol. 3, no. 4, pp. 308–316, 2022, doi: 10.47065/bit.v3i1.
- [10] B. I. Rumabar and E. Maria, “Evaluasi Kualitas Shopeepay Menggunakan ISO/IEC 25010,” *Jurnal Sistem Informasi Bisnis*, vol. 14, no. 1, pp. 54–61, Jan. 2024, doi: 10.21456/vol14iss1pp54-61.
- [11] M. S. Lamada, A. Sa’ban Miru, and R. Amalia, “Pengujian Aplikasi Sistem Monitoring Perkuliahan Menggunakan Standar ISO 25010,” vol. 3, no. 3, p. 1, 2020.
- [12] D. Murdiani and R. Umar, “EVALUASI KUALITAS SISTEM JURNAL ELEKTRONIK BERBASIS OPEN JOURNAL SYSTEM (OJS) MENGGUNAKAN ISO/IEC 25010,” *BACA: JURNAL DOKUMENTASI DAN INFORMASI*, vol. 41, no. 1, p. 75, May 2020, doi: 10.14203/j.baca.v41i1.588.