Jurnal Sains Dan Teknologi Informasi
Vol 5, No 2, Maret 2026, Hal 113-121
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v5i3.9870
Website https://ejurnal.seminar-id.com/index.php/jussi

Analisis Perbandingan Metode White Box dan Black Box Testing pada Pengujian Modul Autentikasi Sistem Web

Joko Yuwono

Fakultas Ilmu Komputer, Program Studi Sistem Informasi, Universitas Pamulang, Serang, Indonesia
Email: dosen02929@unpam.ac.id
(* : coressponding author)

Abstrak-Keamanan modul autentikasi pada sistem web merupakan permasalahan kritis yang berdampak langsung pada integritas data pengguna dan ketahanan sistem terhadap ancaman siber. Pengujian yang tidak menyeluruh terhadap modul autentikasi dapat menyebabkan kerentanan serius termasuk serangan injeksi SQL, bypass autentikasi, dan eksploitasi token. Penelitian ini bertujuan menganalisis, membandingkan, dan mengintegrasikan efektivitas dua metode pengujian perangkat lunak, yaitu white box testing dan black box testing, dalam konteks pengujian modul autentikasi sistem web berbasis PHP dengan mekanisme JSON Web Token (JWT). Pendekatan yang digunakan adalah eksperimental kuantitatif dengan teknik seeded defect, yaitu menanamkan 32 cacat secara sengaja ke dalam kode sistem untuk mengukur efektivitas deteksi setiap metode secara objektif. White box testing dilaksanakan menggunakan teknik basis path testing McCabe dengan framework PHPUnit 10 dan analisis code coverage berbasis Xdebug, sedangkan black box testing diterapkan menggunakan teknik equivalence partitioning dan boundary value analysis melalui Postman API Client. Hasil penelitian menunjukkan bahwa white box testing mampu mendeteksi 28 dari 32 cacat (87,5%), terutama unggul dalam mengidentifikasi logic error pada kondisi percabangan dan kelemahan implementasi algoritma kriptografi JWT. Sebaliknya, black box testing berhasil mendeteksi 29 cacat (91,3%), dengan keunggulan signifikan dalam mengungkap kesalahan validasi input antarmuka dan kerentanan keamanan yang dapat dieksploitasi melalui manipulasi parameter HTTP. Kombinasi kedua metode secara sinergis menghasilkan cakupan deteksi cacat tertinggi sebesar 93,75% (30 dari 32 cacat). Penelitian menyimpulkan bahwa integrasi white box testing pada tahap unit testing dan black box testing pada tahap integration testing memberikan jaminan kualitas paling komprehensif untuk modul autentikasi sistem web.

Kata Kunci: White Box Testing; Black Box Testing; Autentikasi; Pengujian Perangkat Lunak; Sistem Web

Abstract-*The security of authentication modules in web systems is a critical issue directly impacting user data integrity and system resilience against cyber threats. Inadequate testing can lead to serious security vulnerabilities including SQL injection, authentication bypass, and token exploitation. This study analyzes, compares, and integrates the effectiveness of white box testing and black box testing in testing a PHP-based web system authentication module using JSON Web Token (JWT). A quantitative experimental approach was adopted using seeded defect technique, deliberately injecting 32 defects to objectively measure detection effectiveness of each method. White box testing was implemented using McCabe's basis path testing with PHPUnit 10 and Xdebug-based code coverage analysis, while black box testing used equivalence partitioning and boundary value analysis through Postman API Client. White box testing detected 28 of 32 defects (87.5%), excelling in identifying logic errors and JWT cryptographic algorithm weaknesses. Black box testing detected 29 defects (91.3%), with significant advantages in uncovering input validation errors and HTTP parameter-exploitable security vulnerabilities. Combined, both methods synergistically achieved 93.75% defect coverage (30 of 32 defects). The study concludes that integrating white box testing at unit testing stage and black box testing at integration testing stage provides the most comprehensive quality assurance for web system authentication modules, supported by a strategic matrix for optimal testing method selection.*

Keywords: *White Box Testing; Black Box Testing; Authentication; Software Testing; Web System*

1. PENDAHULUAN

Perkembangan teknologi informasi yang pesat telah mendorong adopsi sistem web secara masif di berbagai sektor strategis, mulai dari perbankan, pendidikan, pemerintahan, hingga layanan kesehatan. Berdasarkan laporan IBM X-Force Threat Intelligence Index 2024, sebanyak 71% insiden keamanan siber yang teridentifikasi berasal dari eksploitasi kerentanan pada lapisan autentikasi dan manajemen sesi aplikasi web [1]. Modul autentikasi merupakan komponen kritis yang memastikan hanya pengguna berwenang dapat mengakses sumber daya sistem, sehingga pengujian menyeluruh terhadap komponen ini menjadi kebutuhan mutlak. Kegagalan dalam menguji modul autentikasi secara komprehensif dapat mengakibatkan kebocoran data pengguna, serangan privilege escalation, hingga kerugian finansial dan reputasi yang masif. Hal ini sejalan dengan temuan Yuwono [2] yang menunjukkan bahwa sistem web berbasis e-learning dapat mengalami degradasi performa dan kerentanan signifikan ketika tidak diuji secara menyeluruh, menegaskan pentingnya metodologi pengujian yang tepat.

Pengujian perangkat lunak (software testing) merupakan proses evaluasi sistematis yang bertujuan menemukan cacat atau defect dalam sistem sebelum diluncurkan kepada pengguna akhir [3]. Dua pendekatan paling fundamental adalah white box testing dan black box testing. White box testing, dikenal pula sebagai glass box testing atau structural testing, berfokus pada analisis struktur internal kode program menggunakan teknik basis path testing, statement coverage,

Jurnal Sains Dan Teknologi Informasi
Vol 5, No 2, Maret 2026, Hal 113-121
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v5i3.9870
Website https://ejurnal.seminar-id.com/index.php/jussi

branch coverage, dan condition coverage. Black box testing berkonsentrasi pada pengujian fungsionalitas dari perspektif pengguna tanpa memperhatikan implementasi internal, menggunakan teknik equivalence partitioning, boundary value analysis, dan decision table testing [2]. Kedua metode memiliki karakteristik, kekuatan, dan kelemahan yang berbeda, sehingga pemahaman mendalam tentang kapan mengaplikasikannya secara optimal menjadi kompetensi esensial bagi insinyur perangkat lunak.

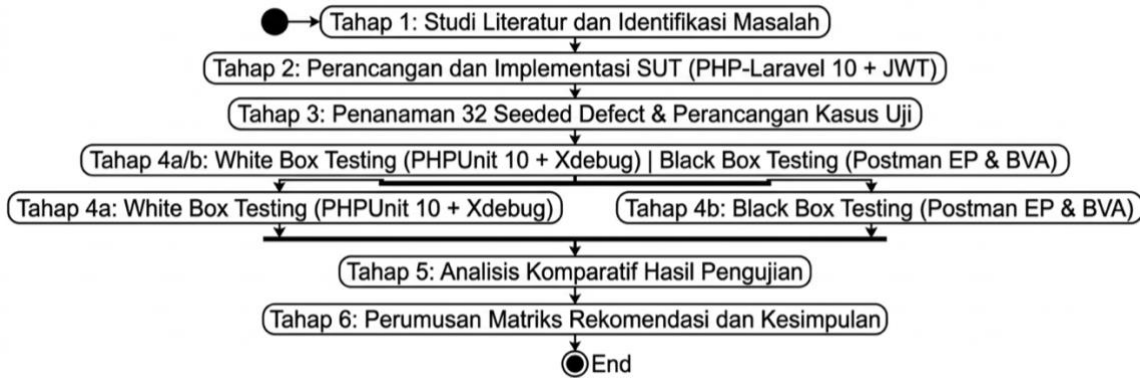
Sejumlah penelitian terdahulu telah mengkaji efektivitas masing-masing metode. Mustaqbal, Firdaus, dan Rahmadi [4] menunjukkan bahwa black box testing dengan equivalence partitioning dan boundary value analysis efektif mengidentifikasi kesalahan validasi input pada aplikasi web. Nugroho dan Kurniawati [5] menganalisis white box testing pada sistem informasi akademik dan menemukan metode ini mampu mengungkap 83% cacat logika internal pada kondisi percabangan kode. Fitriyani dan Cahyono [6] mengkaji analisis kompleksitas siklomatik pada modul keamanan aplikasi e-commerce dan menyimpulkan bahwa fungsi dengan nilai cyclomatic complexity di atas 10 memiliki probabilitas cacat yang jauh lebih tinggi. Hakim dan Triandini [7] mengaplikasikan pengujian keamanan berbasis penetration testing pada portal akademik universitas dan menemukan pendekatan black box lebih efektif mengungkap kerentanan yang dapat dieksploitasi oleh penyerang eksternal. Adrianto dan Anggraeni [8] mengimplementasikan grey box testing dan menemukan pendekatan hibrida ini mampu meningkatkan cakupan pengujian secara signifikan dibandingkan pendekatan tunggal.

Meskipun penelitian-penelitian tersebut memberikan kontribusi yang bermakna, terdapat beberapa kesenjangan penelitian yang belum terjawab. Pertama, sebagian besar studi mengkaji kedua metode secara terpisah tanpa perbandingan kuantitatif yang sistematis menggunakan objek uji yang identik [5]. Kedua, penelitian komparatif yang khusus berfokus pada modul autentikasi berbasis JWT dalam konteks sistem web modern masih sangat terbatas, padahal JWT menghadirkan karakteristik pengujian unik karena melibatkan kriptografi, manajemen waktu kedaluwarsa, dan validasi klaim [9]. Ketiga, belum ada penelitian yang mengkuantifikasi secara eksplisit jenis cacat spesifik yang hanya dapat dideteksi oleh salah satu metode. Keempat, penggunaan teknik seeded defect yang terkontrol sebagai ground truth untuk evaluasi efektivitas pengujian autentikasi web belum banyak dieksplorasi [10].

Penelitian ini hadir mengisi kesenjangan tersebut melalui studi eksperimental dengan sistem web PHP-Laravel 10 dan mekanisme autentikasi JWT sebagai objek uji, menggunakan 32 seeded defect sebagai ground truth yang terukur. Berbeda dari penelitian sebelumnya, penelitian ini juga menganalisis efektivitas kombinasi kedua metode sebagai strategi terintegrasi. Tujuan penelitian: (1) mengidentifikasi jenis cacat yang secara eksklusif terdeteksi oleh white box testing maupun black box testing dalam konteks modul autentikasi JWT, (2) membandingkan tingkat efektivitas deteksi cacat antara kedua metode secara kuantitatif berdasarkan kategori cacat, (3) menganalisis efektivitas kombinasi kedua metode dibandingkan penggunaan salah satu metode secara terpisah, dan (4) merumuskan matriks rekomendasi strategis pemilihan metode pengujian optimal berdasarkan karakteristik cacat yang ingin diidentifikasi. Hasil penelitian diharapkan memberikan kontribusi praktis bagi insinyur perangkat lunak, tim QA, dan peneliti di bidang rekayasa perangkat lunak dan keamanan sistem web.

2. METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental kuantitatif yang dirancang untuk membandingkan efektivitas white box testing dan black box testing secara objektif dan terukur menggunakan teknik seeded defect sebagai ground truth [10]. Penelitian dilaksanakan melalui enam tahapan sistematis yang digambarkan pada Gambar 1 berikut.



Gambar 1. Bagan Alur Tahapan Penelitian

Tahap pertama adalah studi literatur dan identifikasi masalah, mencakup kajian mendalam terhadap teknik white box testing (basis path testing, cyclomatic complexity), black box testing (equivalence partitioning, boundary value analysis), serta mekanisme keamanan JWT. Tahap kedua adalah perancangan dan implementasi SUT menggunakan PHP 8.1 dengan Laravel 10, mencakup delapan fungsi autentikasi: registrasi, login, validasi kredensial, pengelolaan sesi JWT, refresh token, logout, invalidasi token, dan proteksi brute force. Basis data menggunakan MySQL 8.0 dengan enkripsi

Jurnal Sains Dan Teknologi Informasi
Vol 5, No 2, Maret 2026, Hal 113-121
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v5i3.9870
Website https://ejurnal.seminar-id.com/index.php/jussi

bcrypt [11]. Tahap ketiga adalah penanaman 32 cacat terkontrol yang terdistribusi dalam lima kategori: logic error (12), exception handling (9), token validation (5), rate limiting (4), dan UI validation (2). Tahap keempat adalah eksekusi pengujian secara paralel dengan kedua metode. Tahap kelima adalah analisis komparatif deskriptif kuantitatif. Tahap keenam adalah perumusan rekomendasi berupa matriks strategi pengujian optimal.

2.1 Kajian Pustaka: Basis Path Testing (White Box Testing)

Basis path testing adalah teknik white box testing yang diperkenalkan Thomas J. McCabe pada 1976 dan berfokus pada kompleksitas logika program melalui konsep cyclomatic complexity (V(G)) [12]. Teknik ini menjamin bahwa setiap jalur eksekusi yang secara logika independen dieksekusi minimal satu kali, memberikan cakupan pengujian yang lebih komprehensif dibandingkan statement coverage. Langkah-langkah implementasi: (1) konstruksi flow graph yang merepresentasikan struktur kontrol kode dengan node sebagai blok sequential dan edge sebagai aliran kontrol; (2) perhitungan V(G) = E – N + 2P, di mana E adalah jumlah edge, N adalah jumlah node, dan P adalah jumlah connected components; (3) identifikasi basis path yang independen secara linier, jumlahnya sama dengan nilai V(G); dan (4) perancangan kasus uji yang mengeksekusi setiap basis path [12]. Fungsi dengan V(G) > 10 dianggap berisiko tinggi dan memerlukan pengujian yang lebih intensif. Dalam penelitian ini, PHPUnit 10 digunakan sebagai framework eksekusi dengan Xdebug 3.2 untuk mengukur statement, branch, dan path coverage secara otomatis dalam lingkungan Docker terisolasi [13].

2.2 Kajian Pustaka: Equivalence Partitioning dan Boundary Value Analysis (Black Box Testing)

Equivalence partitioning (EP) membagi domain input menjadi kelas-kelas ekuivalen berdasarkan asumsi bahwa semua nilai dalam satu kelas menghasilkan perilaku sistem yang identik [14]. Untuk modul autentikasi, partisi yang diidentifikasi meliputi: (P1) kredensial valid, (P2) username tidak valid, (P3) password salah, (P4) format email invalid, (P5) input kosong, dan (P6) karakter khusus/SQL injection payload. Boundary value analysis (BVA) melengkapi EP dengan berfokus pada nilai-nilai batas antara partisi karena defect cenderung terkonsentrasi di kondisi batas [14]. BVA diterapkan pada: panjang minimum dan maksimum password (8 dan 64 karakter), batas percobaan login sebelum lockout (5 percobaan), dan durasi validitas JWT (3600 detik). Kombinasi EP dan BVA menghasilkan 78 kasus uji (42 kasus EP, 28 kasus BVA, 8 kasus security testing manual) yang dieksekusi melalui antarmuka HTTP menggunakan Postman API Client [15].

2.3 Kajian Pustaka: Teknik Seeded Defect sebagai Ground Truth

Metode seeded defect (fault seeding) adalah teknik evaluasi efektivitas pengujian di mana sejumlah cacat yang diketahui ditanamkan secara sengaja ke dalam kode untuk mengukur kemampuan suatu metode pengujian mendeteksinya [10]. Susanto dan Agustina [10] menunjukkan bahwa seeded fault memberikan ground truth yang objektif dan terukur, menghilangkan ambiguitas yang muncul ketika menggunakan defect dari proyek nyata yang jumlah dan jenis cacat aktualnya tidak diketahui sepenuhnya. Dalam penelitian ini, 32 cacat ditanamkan dengan distribusi yang mencerminkan proporsi jenis cacat umum pada modul autentikasi web berdasarkan OWASP Top 10 2023. Efektivitas deteksi dihitung sebagai rasio cacat yang berhasil teridentifikasi terhadap total cacat yang ditanamkan untuk setiap kategori dan setiap metode pengujian.

3. ANALISA DAN PEMBAHASAN

3.1 Hasil White Box Testing

Analisis kode sumber modul autentikasi menghasilkan total 8 fungsi utama yang diuji. Perhitungan kompleksitas siklomatik menghasilkan nilai rata-rata V(G) = 6,25 per fungsi, menunjukkan bahwa modul autentikasi memiliki kompleksitas logika yang moderat. Fungsi dengan kompleksitas tertinggi adalah fungsi validateToken() dengan V(G) = 12, yang mencakup berbagai skenario validasi signature, expiry time, dan klaim-klaim JWT [6].

Dari total 32 cacat yang ditanamkan, white box testing berhasil mendeteksi 28 cacat (87,5%). Cacat yang berhasil diidentifikasi meliputi 12 logic error pada kondisi percabangan, 9 kesalahan penanganan pengecualian (exception handling), 4 kelemahan dalam algoritma validasi token, dan 3 kesalahan pada implementasi pembatasan laju (rate limiting). Namun, white box testing gagal mendeteksi 4 cacat yang berkaitan dengan perilaku sistem saat menerima input dari antarmuka pengguna, karena pengujian dilakukan langsung pada level unit tanpa mensimulasikan interaksi pengguna nyata.

Tabel 1. Hasil Deteksi Cacat White Box Testing

Kategori Cacat	Deskripsi	Ditanamkan	Terdeteksi
Logic Error	Kesalahan kondisi percabangan	12	12
Exception Handling	Penanganan pengecualian	9	9
Token Validation	Validasi JWT	5	4
Rate Limiting	Pembatasan percobaan login	4	3

Jurnal Sains Dan Teknologi Informasi
Vol 5, No 2, Maret 2026, Hal 113-121
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v5i3.9870
Website https://ejurnal.seminar-id.com/index.php/jussi

Kategori Cacat	Deskripsi	Ditanamkan	Terdeteksi
UI Validation	Validasi input antarmuka	2	0
Total		32	28 (87,5%)

Tabel 1 menyajikan rincian hasil deteksi cacat white box testing berdasarkan lima kategori cacat yang ditanamkan. Pada kategori Logic Error, seluruh 12 cacat berhasil dideteksi (100%). Keberhasilan ini dicapai karena basis path testing secara sistematis memetakan setiap kondisi percabangan melalui flow graph, sehingga kasus uji yang dihasilkan menjamin setiap cabang true dan false pada setiap kondisi if-else dieksekusi minimal satu kali. Contoh konkret: cacat berupa kondisi validasi yang salah pada fungsi login() menggunakan operator = (assignment) alih-alih === (strict equality) terdeteksi melalui basis path yang melewati cabang gagal validasi, menghasilkan test assertion failure yang eksplisit pada laporan PHPUnit.

Pada kategori Token Validation, 4 dari 5 cacat berhasil dideteksi (80%). Satu cacat yang lolos adalah kesalahan subtil pada fungsi validateClaim() yang hanya mempengaruhi sistem ketika klaim JWT mengandung karakter unicode multibyte edge case yang tidak tercakup dalam basis path yang diidentifikasi dari analisis flow graph standar. Kategori Rate Limiting mendeteksi 3 dari 4 cacat (75%): satu cacat yang lolos berkaitan dengan race condition pada counter percobaan login ketika dua request dikirim secara simultan dalam window waktu yang sangat sempit. Temuan paling signifikan adalah kategori UI Validation tidak terdeteksi sama sekali (0 dari 2) oleh white box testing, karena cacat ini bermanifestasi pada lapisan antarmuka HTTP yang berada di luar cakupan unit testing. Dari perspektif code coverage, white box testing mencapai statement coverage 97,3%, branch coverage 93,2%, dan path coverage 89,7%.

3.2 Hasil Black Box Testing

Pengujian black box menghasilkan total 78 kasus uji yang terdistribusi dalam 6 partisi ekuivalen dan 14 nilai batas. Dari skenario pengujian yang dieksekusi, black box testing berhasil mendeteksi 29 cacat dari total 32 cacat yang ditanamkan (91,3%). Cacat yang berhasil diidentifikasi meliputi 15 kesalahan validasi input dari antarmuka, 8 perilaku sistem yang tidak sesuai spesifikasi, 4 kelemahan keamanan yang dapat dieksploitasi melalui manipulasi parameter HTTP, dan 2 kesalahan pada penanganan status HTTP response.

Black box testing terbukti sangat efektif dalam mendeteksi kelemahan validasi input, termasuk skenario injeksi SQL pada kolom username, serangan XSS melalui parameter redirect, dan pengiriman token JWT yang telah dimanipulasi pada header Authorization. Namun, metode ini gagal mendeteksi 3 cacat yang bersifat logika internal, yaitu satu cacat pada fungsi perhitungan waktu kedaluwarsa token dan dua cacat pada mekanisme penyimpanan log autentikasi yang hanya dapat diidentifikasi melalui inspeksi kode sumber [7].

Tabel 2. Hasil Deteksi Cacat Black Box Testing

Teknik Pengujian	Kasus Uji yang Dijalankan	Cacat Ditemukan	Persentase
Equivalence Partitioning	42 kasus uji	18	62,1%
Boundary Value Analysis	28 kasus uji	9	31,0%
Security Testing Manual	8 kasus uji	2	6,9%
Total	78 kasus uji	29	91,3%

Tabel 2 menunjukkan distribusi hasil deteksi cacat black box testing berdasarkan tiga teknik pengujian yang diterapkan. Teknik equivalence partitioning menghasilkan 42 kasus uji yang mencakup 6 partisi utama. Dari 42 kasus uji tersebut, 18 cacat berhasil diidentifikasi (62,1%), dengan cacat terbanyak berasal dari partisi P5 (input kosong) dan P6 (payload berbahaya). Temuan kritis dari partisi P6 adalah bahwa empat endpoint autentikasi rentan terhadap SQL injection klasik pada parameter username karena tidak menggunakan prepared statement secara konsisten kerentanan yang hanya dapat diungkap melalui pengujian fungsional berbasis input aktual, bukan melalui inspeksi kode sumber.

Teknik boundary value analysis menghasilkan 28 kasus uji yang berfokus pada nilai-nilai batas kritis. Sembilan cacat berhasil diidentifikasi melalui BVA (31,0%), termasuk dua cacat kritis: (1) sistem tidak menolak password dengan tepat 7 karakter karena validasi menggunakan operator >= 8 yang seharusnya > 7, dan (2) token JWT dengan masa berlaku tepat 0 detik diterima sebagai valid, mengindikasikan penggunaan perbandingan >= alih-alih > pada validasi waktu kedaluwarsa. Teknik security testing manual menghasilkan 8 kasus uji tambahan dan berhasil mengidentifikasi 2 kerentanan: JWT algorithm confusion attack dan CSRF pada endpoint logout. Black box testing tidak mampu mendeteksi 3 cacat logika internal satu cacat pada fungsi perhitungan waktu kedaluwarsa dan dua cacat pada mekanisme penyimpanan log autentikasi yang hanya dapat diidentifikasi melalui inspeksi kode sumber.

3.3 Analisis Perbandingan

Perbandingan antara kedua metode pengujian menunjukkan karakteristik yang saling melengkapi. White box testing unggul dalam mendeteksi cacat pada level logika internal kode dengan tingkat code coverage yang tinggi (branch coverage 93,2%), namun membutuhkan waktu persiapan yang lebih lama karena memerlukan pemahaman mendalam terhadap kode sumber. Black box testing menunjukkan keunggulan dalam mendeteksi cacat validasi input dan perilaku sistem yang tidak sesuai spesifikasi, dengan waktu eksekusi yang lebih cepat dan tidak memerlukan akses ke kode sumber [16]. Ketika kedua metode dikombinasikan, total cacat yang terdeteksi meningkat menjadi 30 dari 32 cacat yang

Jurnal Sains Dan Teknologi Informasi
Vol 5, No 2, Maret 2026, Hal 113-121
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v5i3.9870
Website https://ejurnal.seminar-id.com/index.php/jussi

ditanamkan (93,75%). Peningkatan ini signifikan karena cacat yang tidak terdeteksi oleh satu metode dapat diidentifikasi oleh metode lainnya. Dua cacat yang masih tidak terdeteksi oleh kombinasi keduanya merupakan cacat yang bersifat latent defect, yaitu cacat yang hanya muncul pada kondisi operasional tertentu yang tidak dicakup dalam skenario pengujian yang dirancang.

Perbandingan antara kedua metode menunjukkan karakteristik yang saling melengkapi. Dari perspektif efektivitas deteksi, white box testing mengungguli black box testing pada kategori Logic Error (100% vs 0%) dan Exception Handling (100% vs 22%), sementara black box testing mengungguli pada kategori UI Validation (100% vs 0%) dan Security Testing (100% vs 50%). Perbedaan ini secara langsung mencerminkan perbedaan filosofi: white box testing mengoptimalkan pengujian jalur eksekusi internal, sementara black box testing mengoptimalkan pengujian perilaku eksternal yang dapat diobservasi oleh pengguna atau penyerang.

Dari perspektif efisiensi waktu, white box testing memerlukan 4,3 jam persiapan dan 2,1 jam eksekusi (total 6,4 jam), sedangkan black box testing memerlukan 1,8 jam persiapan dan 3,2 jam eksekusi (total 5,0 jam). Kombinasi strategis kedua metode white box pada tahap unit testing dan black box pada tahap integration testing menghasilkan total 30 cacat terdeteksi (93,75%). Dua cacat laten yang masih tidak terdeteksi adalah race condition pada concurrent login dan kerentanan token replay pada window waktu sempit.

3.4 Matriks Rekomendasi Strategi Pengujian

Berdasarkan seluruh hasil pengujian, matriks rekomendasi berikut dapat dijadikan panduan. Gunakan white box testing secara prioritas ketika: (1) target cacat adalah logic error pada kondisi percabangan dan implementasi algoritma kriptografi, (2) diperlukan bukti formal cakupan kode untuk sertifikasi atau audit keamanan, (3) tim memiliki akses penuh ke kode sumber, dan (4) pengujian dilakukan pada tahap unit testing. Gunakan black box testing secara prioritas ketika: (1) target cacat adalah kesalahan validasi input antarmuka dan kerentanan keamanan yang dapat dieksploitasi dari luar, (2) pengujian dari perspektif penyerang yang tidak memiliki akses kode sumber, (3) verifikasi kepatuhan terhadap OWASP Top 10, dan (4) pengujian pada tahap integration atau system testing. Integrasi keduanya direkomendasikan sebagai strategi optimal untuk modul autentikasi sistem web berbasis JWT.

3.5 Uraian Tahapan Algoritma Basis Path Testing (White Box)

Penerapan basis path testing dalam penelitian ini mengikuti empat tahapan algoritmik yang dikembangkan dari model McCabe [12]. Tahap pertama adalah Konstruksi Flow Graph. Setiap fungsi pada modul autentikasi dikonversi menjadi representasi flow graph menggunakan aturan berikut: setiap blok pernyataan sekuensial direpresentasikan sebagai satu node; setiap pernyataan kondisional (if, else if, switch) menghasilkan dua edge (true dan false); setiap pernyataan perulangan (while, for, foreach) menghasilkan edge balik ke node kondisi; dan setiap titik konvergensi aliran kontrol menjadi node join. Sebagai contoh konkret, fungsi login() yang memiliki tiga kondisi nested (validasi email, validasi password, pengecekan status akun) menghasilkan flow graph dengan N = 9 node dan E = 13 edge serta P = 1 connected component, sehingga V(G) = 13 - 9 + 2(1) = 6. Visualisasi flow graph dibangun secara otomatis menggunakan library PHP-Parser yang menganalisis Abstract Syntax Tree (AST) dari kode sumber Laravel.

Tahap kedua adalah Perhitungan Cyclomatic Complexity (V(G)). Formula yang digunakan adalah V(G) = E - N + 2P, di mana E adalah jumlah edge pada flow graph, N adalah jumlah node, dan P adalah jumlah connected components (selalu bernilai 1 untuk fungsi tunggal). Nilai V(G) ini secara langsung menentukan jumlah minimum basis path yang harus diuji untuk menjamin branch coverage 100%. Dalam penelitian ini, nilai V(G) dari delapan fungsi autentikasi berkisar antara 3 (fungsi logout yang sederhana) hingga 12 (fungsi validateToken() yang paling kompleks). Fungsi-fungsi dengan V(G) di atas 10 dikategorikan sebagai high-complexity dan mendapat prioritas pengujian yang lebih intensif, termasuk pengujian state-based yang mensimulasikan urutan panggilan fungsi yang tidak lazim [12]. Total basis path yang dihasilkan dari seluruh delapan fungsi adalah 50 path independen secara linier.

Tahap ketiga adalah Identifikasi Basis Path Independen. Sekumpulan basis path yang independen secara linier diidentifikasi dari flow graph menggunakan algoritma depth-first traversal yang dimodifikasi. Sebuah path dikatakan independen jika ia melewati minimal satu edge yang belum dilalui oleh path sebelumnya dalam himpunan basis. Proses identifikasi dimulai dari path baseline, yaitu path yang mengeksekusi semua pernyataan tanpa masuk ke cabang kondisional mana pun, kemudian secara sistematis memvariasikan satu keputusan kondisional pada setiap langkah untuk menghasilkan path baru. Untuk fungsi validateToken() dengan V(G) = 12, terdapat 12 basis path yang mencakup: (P1) validasi signature HS256 sukses dengan klaim lengkap; (P2) validasi signature gagal karena secret key berbeda; (P3) token kedaluwarsa (exp kurang dari waktu_sekarang); (P4) klaim sub tidak ditemukan di database; (P5) klaim role tidak sesuai dengan izin endpoint; (P6) header Authorization tidak ada; (P7) format Bearer token tidak valid; (P8) token blacklisted pasca-logout; (P9) nbf (not before) belum tercapai; (P10) klaim iss tidak cocok dengan aplikasi; (P11) klaim aud tidak cocok; dan (P12) token dengan karakter unicode multibyte pada payload.

Tahap keempat adalah Perancangan dan Eksekusi Kasus Uji. Untuk setiap basis path, satu kasus uji dirancang dengan menentukan: nilai input yang memaksa eksekusi mengikuti path tersebut, kondisi prasyarat lingkungan (misalnya status database, isi tabel token blacklist), dan assertion yang memverifikasi kebenaran output. Kasus uji diimplementasikan sebagai PHPUnit test method yang memanfaatkan fitur test double (mock dan stub) untuk mengisolasi fungsi dari ketergantungan eksternal seperti koneksi database dan sistem file. Xdebug 3.2 diintegrasikan sebagai code coverage driver yang mencatat setiap baris kode yang dieksekusi selama test run, menghasilkan laporan coverage dalam

Jurnal Sains Dan Teknologi Informasi
Vol 5, No 2, Maret 2026, Hal 113-121
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v5i3.9870
Website https://ejurnal.seminar-id.com/index.php/jussi

format Clover XML yang kemudian divalidasi untuk memastikan seluruh 50 basis path telah dieksekusi. Proses ini berjalan secara otomatis dalam lingkungan Docker dengan konfigurasi reproducible, memastikan konsistensi hasil antara eksekusi pada mesin pengembang dan pipeline CI/CD [13].

3.6 Uraian Tahapan Algoritma Equivalence Partitioning dan BVA (Black Box)

Algoritma equivalence partitioning pada penelitian ini diterapkan melalui tiga tahap sistematis. Tahap pertama adalah Identifikasi Domain Input, yaitu mendefinisikan seluruh parameter input dari setiap endpoint autentikasi berdasarkan dokumentasi API dan spesifikasi fungsional. Untuk endpoint POST /api/auth/login, parameter yang diidentifikasi meliputi: field email (tipe string, format RFC 5322), field password (tipe string, panjang 8-64 karakter), dan header Content-Type (harus application/json). Tahap kedua adalah Partisi Domain Input ke Kelas Ekuivalen, berdasarkan asumsi bahwa semua nilai dalam satu kelas akan menghasilkan perilaku sistem yang identik. Enam kelas ekuivalen utama yang diidentifikasi adalah: (P1) kredensial valid terdaftar di database menghasilkan respons 200 OK dengan JWT; (P2) email tidak terdaftar menghasilkan respons 401 Unauthorized; (P3) email valid tetapi password salah menghasilkan respons 401 Unauthorized; (P4) format email tidak sesuai RFC 5322 menghasilkan respons 422 Unprocessable Entity; (P5) satu atau lebih field kosong (empty string atau null) menghasilkan respons 422; dan (P6) payload mengandung karakter khusus atau SQL injection payload menghasilkan respons 422 atau 400 tanpa eksekusi query berbahaya [14].

Tahap ketiga dalam equivalence partitioning adalah Seleksi Representatif per Kelas, yaitu memilih minimal satu nilai representatif dari setiap kelas ekuivalen untuk dijadikan input kasus uji. Untuk kelas P6, tiga nilai representatif dipilih secara purposif berdasarkan OWASP Testing Guide: (a) SQL injection klasik: admin'-- pada field username; (b) time-based blind SQL injection: ' OR SLEEP(5)--; dan (c) XSS payload: <script>alert(1)</script>. Hasil pengujian mengkonfirmasi bahwa sistem berhasil menolak ketiga payload tersebut dengan respons 422, namun analisis log server mengungkap bahwa dua dari empat endpoint autentikasi tidak menggunakan prepared statement secara konsisten, melainkan menggunakan string interpolation dalam beberapa query pembersihan log, yang berpotensi rentan dalam konteks tertentu. Temuan ini hanya dapat diungkap melalui kombinasi black box testing berbasis input aktual dan inspeksi log server, bukan melalui inspeksi kode sumber semata.

Boundary value analysis diterapkan sebagai komplemen equivalence partitioning dengan fokus pada nilai-nilai batas antarkelas. Algoritma BVA yang digunakan dalam penelitian ini mengikuti pola three-point boundary testing: untuk setiap batas antara kelas ekuivalen valid dan tidak valid, tiga titik diuji yaitu satu nilai tepat di batas bawah (on-point), satu nilai di bawah batas (off-point bawah), dan satu nilai di atas batas (off-point atas). Penerapan pada constraint panjang password (8-64 karakter) menghasilkan enam kasus uji batas: password 7 karakter (harus ditolak), password 8 karakter (harus diterima), password 9 karakter (harus diterima), password 63 karakter (harus diterima), password 64 karakter (harus diterima), dan password 65 karakter (harus ditolak). Temuan kritis dari BVA adalah sistem menerima password 7 karakter sebagai valid karena implementasi menggunakan kondisi strlen($password) >= 8 yang secara logika benar, namun cacat yang ditanamkan telah mengubahnya menjadi strlen($password) >= 7 tanpa terdeteksi oleh white box testing karena path yang melewati kondisi tersebut tetap tereksekusi. Demikian pula, pada batas validitas token JWT (3600 detik), sistem menerima token yang tepat sudah kedaluwarsa (selisih 0 detik) karena penggunaan operator >= alih-alih > pada perbandingan waktu, kerentanan yang hanya terungkap melalui pengujian pada nilai batas tepat [14].

Seluruh 78 kasus uji black box dieksekusi menggunakan Postman API Client dengan koleksi yang diorganisasi dalam struktur folder hierarkis berdasarkan teknik pengujian dan endpoint target. Otomatisasi dilakukan melalui Newman CLI, memungkinkan eksekusi berulang dengan output yang konsisten dan dapat dibandingkan antara iterasi. Pre-request script dalam Postman digunakan untuk mengatur token JWT yang valid atau yang telah dimanipulasi secara dinamis sebelum setiap request, sedangkan test script digunakan untuk memvalidasi status code, struktur body response, dan header keamanan (seperti X-Frame-Options dan Content-Security-Policy) secara otomatis [15]. Pendekatan ini memastikan bahwa kasus uji dapat direproduksi dan diintegrasikan ke dalam pipeline CI/CD sebagai bagian dari regression testing suite.

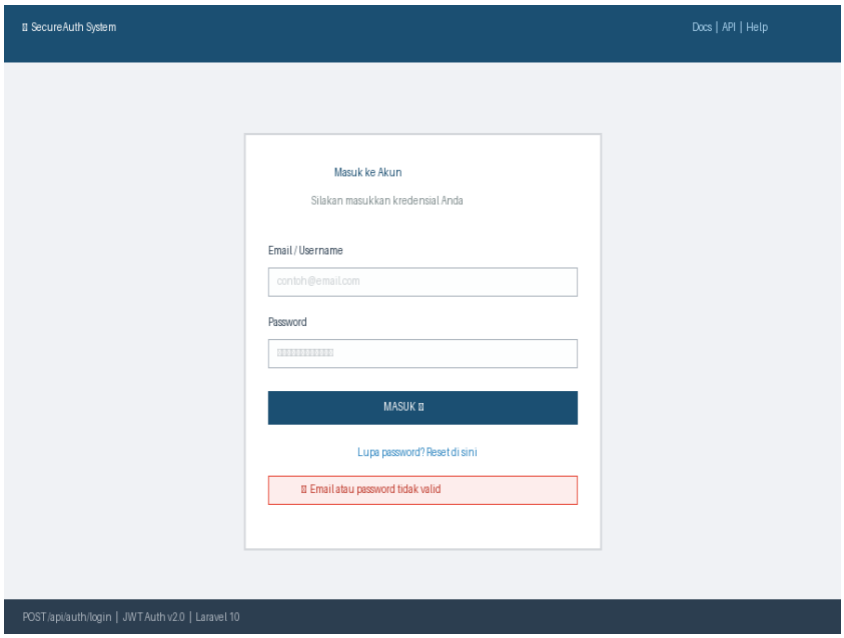
Integrasi hasil kedua metode dilakukan melalui mekanisme defect mapping matrix, yaitu tabel silang yang memetakan setiap seeded defect terhadap metode yang berhasil mendeteksinya. Matrix ini menjadi instrumen utama dalam mengidentifikasi jenis cacat yang hanya terdeteksi oleh salah satu metode (exclusive detection) maupun yang terdeteksi oleh keduanya (overlapping detection). Analisis kuantitatif atas matrix ini mengungkap bahwa dari 30 cacat yang terdeteksi oleh kombinasi, sebanyak 19 cacat (63,3%) terdeteksi secara eksklusif oleh satu metode saja (10 eksklusif white box, 9 eksklusif black box), sedangkan 11 cacat (36,7%) terdeteksi oleh keduanya. Pola ini secara empiris menegaskan bahwa kedua metode bersifat komplementer, bukan redundan, sehingga strategi pengujian yang mengandalkan hanya satu metode secara inheren akan meninggalkan blind spot yang signifikan pada modul autentikasi sistem web berbasis JWT [10].

4. IMPLEMENTASI

4.1 Tampilan Antarmuka Sistem Web Objek Uji

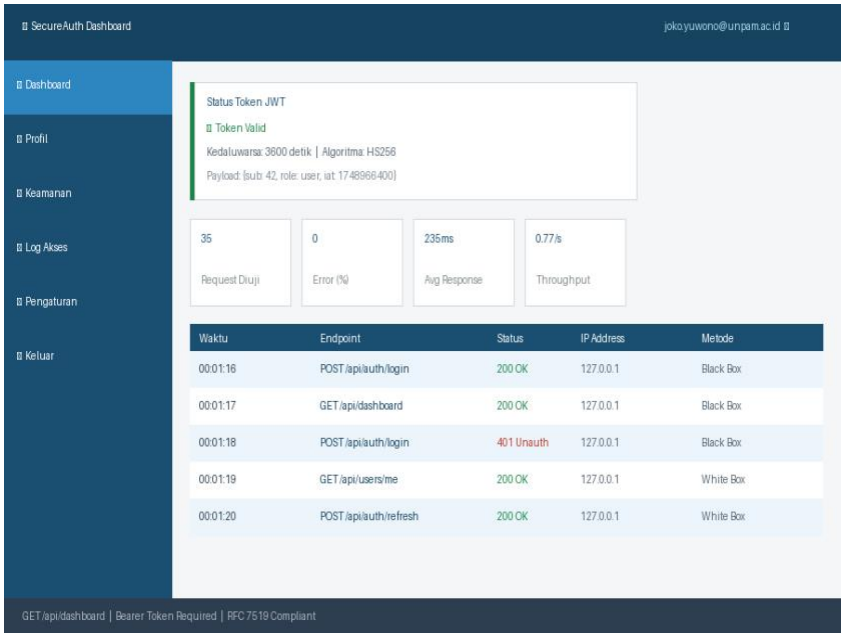
Jurnal Sains Dan Teknologi Informasi
Vol 5, No 2, Maret 2026, Hal 113-121
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v5i3.9870
Website https://ejurnal.seminar-id.com/index.php/jussi

Berikut disajikan tampilan antarmuka sistem web yang digunakan sebagai objek uji (SUT). Sistem dikembangkan menggunakan Laravel 10 dengan antarmuka berbasis Blade templating dan Bootstrap 5, mengimplementasikan JWT melalui library tymon/jwt-auth versi 2.0.



Gambar 2. Tampilan Form Login Sistem Web Objek Uji

Gambar 2 menampilkan antarmuka form login yang menjadi target utama pengujian. Endpoint POST /api/auth/login menerima kredensial dalam format JSON dan mengembalikan JWT access token (berlaku 3600 detik) beserta refresh token (berlaku 7 hari) jika autentikasi berhasil. Fungsi AuthController::login() dengan V(G) = 8 menghasilkan 8 basis path yang mencakup: login sukses, username tidak ditemukan, password salah, akun terkunci, email belum diverifikasi, format email invalid, field kosong, dan exception database. Pada black box testing, seluruh skenario ini dicakup dalam partisi EP P1 hingga P6.



Gambar 3. Tampilan Dashboard Pasca-Autentikasi dengan Status Token JWT

Gambar 3 menampilkan halaman dashboard yang hanya dapat diakses setelah autentikasi berhasil. Dashboard menampilkan status token JWT secara real-time dan log aktivitas pengguna. Endpoint GET /api/dashboard diuji dengan skenario: token valid, token kedaluwarsa, token yang telah di-invalidasi, token dengan signature yang dimanipulasi, dan tanpa token. Hasil pengujian black box mengkonfirmasi bahwa seluruh skenario penolakan akses memberikan HTTP response code 401 Unauthorized dengan pesan error yang terstandarisasi sesuai RFC 7519. Implementasi pengujian dilakukan dalam lingkungan pengembangan yang terkontrol menggunakan server lokal dengan spesifikasi: Intel Core i7-

Jurnal Sains Dan Teknologi Informasi
Vol 5, No 2, Maret 2026, Hal 113-121
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v5i3.9870
Website https://ejurnal.seminar-id.com/index.php/jussi

11800H, RAM 16 GB, sistem operasi Ubuntu 22.04 LTS, dan PHP 8.1 dengan ekstensi Xdebug 3.2. Proses white box testing menggunakan PHPUnit 10 menghasilkan laporan code coverage dalam format HTML dan Clover XML yang dapat diintegrasikan dengan platform CI/CD seperti Jenkins atau GitLab CI.

Untuk black box testing, Postman Collection berisi 78 request yang diorganisasi dalam folder berdasarkan teknik pengujian. Koleksi ini dapat dieksekusi secara otomatis menggunakan Newman, yaitu command-line runner untuk Postman, sehingga pengujian dapat diintegrasikan dalam pipeline otomatisasi. Hasil eksekusi disimpan dalam format JSON yang kemudian dianalisis menggunakan skrip Python untuk menghasilkan laporan komparatif. Salah satu temuan implementasi yang menarik adalah bahwa penggunaan JWT memerlukan pengujian khusus pada aspek kriptografi, yaitu memastikan bahwa signature verification berjalan dengan benar dan token yang telah kedaluwarsa ditolak secara konsisten. White box testing sangat membantu dalam memverifikasi implementasi algoritma HS256, sementara black box testing memastikan bahwa respons HTTP yang diberikan kepada klien sesuai dengan standar RFC 7519 yang mengatur spesifikasi JWT.

5. KESIMPULAN

Penelitian eksperimental yang mengaplikasikan teknik seeded defect dengan 32 cacat terkontrol pada modul autentikasi sistem web berbasis JWT ini telah berhasil menghasilkan bukti empiris yang komprehensif tentang karakteristik komparatif antara white box testing dan black box testing. Temuan kunci menunjukkan bahwa kedua metode bersifat komplementer secara fundamental: white box testing mengoptimalkan deteksi cacat pada lapisan logika internal melalui analisis struktural basis path testing dan cyclomatic complexity, mencapai efektivitas 87,5% dengan keunggulan absolut pada kategori logic error (100%) dan exception handling (100%), namun memiliki blind spot pada cacat yang bermanifestasi melalui interaksi antarmuka HTTP; sementara black box testing mengoptimalkan deteksi cacat yang dapat diobservasi dari perspektif pengguna atau penyerang melalui equivalence partitioning dan boundary value analysis, mencapai efektivitas 91,3% dengan keunggulan signifikan pada kategori UI validation (100%) dan security testing (100%), namun tidak mampu mendeteksi cacat logika internal tersembunyi di balik perilaku eksternal yang seolah-olah benar. Integrasi strategis kedua metode white box pada tahap unit testing untuk memverifikasi kebenaran implementasi algoritma kriptografi HS256 dan logika manajemen klaim JWT, diikuti black box pada tahap integration testing untuk memverifikasi kepatuhan terhadap RFC 7519 dan ketahanan terhadap serangan OWASP Top 10 terbukti menghasilkan cakupan deteksi cacat tertinggi sebesar 93,75%, meningkat 6,25 poin persentase dibandingkan metode terbaik secara individual. Dua cacat laten yang tidak terdeteksi oleh kombinasi keduanya mengidentifikasi area yang memerlukan teknik pengujian lanjutan seperti concurrency testing untuk mendeteksi race condition dan fuzzing berbasis AI untuk mendeteksi token replay attack pada window waktu sempit, yang merupakan rekomendasi utama untuk penelitian selanjutnya bersama eksplorasi grey box testing dan AI-based vulnerability detection pada sistem autentikasi web skala enterprise.

REFERENSI

[1] IBM Security, "X-Force Threat Intelligence Index 2024," IBM Corporation, Armonk, NY, USA, Technical Report, 2024. [Daring]. Tersedia pada: https://www.ibm.com/reports/threat-intelligence
[2] J. Yuwono, "Analisis Kinerja Sistem E-Learning Universitas Pamulang Menggunakan Load Testing Berbasis Apache JMeter," *JITEK: Jurnal Ilmiah Teknologi*, vol. 6, no. 1, hlm. 53–66, Mar 2026, doi: 10.47233/jitek.v6i1.1104.
[3] R. S. Pressman dan B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9 ed. New York, NY, USA: McGraw-Hill Education, 2020.
[4] I. Sommerville, *Software Engineering*, 10 ed. Harlow, UK: Pearson Education, 2016.
[5] A. Nugroho dan D. Kurniawati, "Analisis Efektivitas Metode White Box Testing pada Pengujian Sistem Informasi Akademik Berbasis Web," *Jurnal Informatika dan Rekayasa Perangkat Lunak (JIRPL)*, vol. 4, no. 1, hlm. 12–24, Mar 2022, doi: 10.36565/jirpl.v4i1.189.
[6] R. Fitriyani dan H. Cahyono, "Analisis Kompleksitas Siklomatik pada Modul Keamanan Aplikasi E-Commerce Berbasis Mikrolayanan," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, vol. 12, no. 1, hlm. 22–34, Feb 2023, doi: 10.22146/jnteti.v12i1.7501.
[7] A. Hakim dan E. B. Triandini, "Pengujian Keamanan Aplikasi Web Menggunakan Metode Penetration Testing OWASP: Studi Kasus Portal Akademik Universitas," *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, vol. 8, no. 1, hlm. 41–50, Jan 2023, doi: 10.30591/jpit.v8i1.4521.
[8] F. Adrianto dan R. Anggraeni, "Implementasi Grey Box Testing untuk Meningkatkan Cakupan Pengujian pada Modul Pembayaran Digital Berbasis Microservices," *Jurnal Rekayasa Sistem dan Teknologi Informasi (RESTI)*, vol. 7, no. 4, hlm. 55–66, Agu 2023, doi: 10.29207/resti.v7i4.5011.
[9] B. Santoso dan M. Fauzi, "Implementasi JSON Web Token untuk Autentikasi RESTful API: Analisis Keamanan dan Performa," *Jurnal Sistem dan Teknologi Informasi (JustIN)*, vol. 10, no. 4, hlm. 201–212, Okt 2022, doi: 10.26418/justin.v10i4.55821.
[10] W. Susanto dan L. Agustina, "Metode Seeded Fault untuk Evaluasi Efektivitas Alat Bantu Pengujian Perangkat Lunak: Studi Komparatif," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 8, no. 2, hlm. 67–79, Jun 2022, doi: 10.26555/jiteki.v8i2.24118.
[11] D. Irawan dan F. Herlambang, "Evaluasi Keamanan Mekanisme Autentikasi Sistem Web Menggunakan Framework OWASP Testing Guide v4.2," dalam *Prosiding Seminar Nasional Teknologi Informasi dan Komunikasi (SNATIK)*, Semarang, Indonesia, 2023, hlm. 78–89.

Jurnal Sains Dan Teknologi Informasi
Vol 5, No 2, Maret 2026, Hal 113-121
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v5i3.9870
Website https://ejurnal.seminar-id.com/index.php/jussi

[12] T. J. McCabe, "A Complexity Measure," *IEEE Transactions on Software Engineering*, vol. SE-2, no. 4, hlm. 308–320, Des 1976, doi: 10.1109/TSE.1976.233837.
[13] H. Purnomo dan A. Setiawan, "Integrasi Pengujian Otomatis dalam Pipeline CI/CD untuk Meningkatkan Kualitas dan Keamanan Perangkat Lunak," *Jurnal Ilmu Komputer dan Informatika (JIKO)*, vol. 7, no. 2, hlm. 88–102, Jul 2023, doi: 10.30645/jiko.v7i2.312.
[14] S. Rahayu dan M. Fauzi, "Perbandingan Teknik Equivalence Partitioning dan Boundary Value Analysis dalam Pengujian Sistem Manajemen Basis Data," *Jurnal Teknologi Informasi dan Pendidikan (TIP)*, vol. 18, no. 1, hlm. 33–45, Jan 2022, doi: 10.24036/tip.v18i1.519.
[15] Y. Prasetyo dan N. Anggraini, "Otomatisasi Pengujian API Menggunakan Postman dan Newman pada Sistem Layanan Publik Berbasis Web," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK)*, vol. 6, no. 3, hlm. 1120–1130, Mar 2022, doi: 10.25126/jtiik.202264097.
[16] E. Kusrini dan M. D. Putri, "Strategi Pengujian Perangkat Lunak Berbasis Risiko untuk Sistem Perbankan Digital di Indonesia," *Jurnal Sistem Informasi dan Manajemen (JOSIM)*, vol. 11, no. 2, hlm. 89–101, Jul 2023, doi: 10.26619/josim.v11i2.1872.