Aisyah Journal Of Informatics and Electrical Engineering
Vol. 7, No. 2, Agustus 2025
ISSN 2686-0139 (Print), ISSN 2685-9556 (Online)
https://jti.aisyahuniversity.ac.id/index.php/AJIEE

Analisis Kerentanan Aplikasi Web terhadap SQL Injection dan XSS Berdasarkan Survei Pengguna dengan Kategori Risiko

Heru Kasdana [1], Zaiman Makmur [2], Gustam Efendi [3], Mursyid Maulana Achmad [4]
[1,2,3]Program Studi Sistem Informasi, Sekolah Tinggi Teknologi Bina Tunggal, Bekasi, Indonesia
[4]Program Studi Teknik Elektro, Sekolah Tinggi Teknologi Bina Tunggal, Bekasi, Indonesia

Info Artikel

Riwayat Artikel:

Received August 04, 2025
Revised August 16, 2025
Accepted August 30, 2025

Corresponding Author:
Heru Kasdana
Email: heru.kasdana@stt-binatunggal.ac.id

***Abstract** – Abstract in English*
SQL Injection (SQLi) and Cross-Site Scripting (XSS) remain critical threats to modern web application security. This study aims to evaluate the level of awareness and security practices applied by developers through a survey involving various roles, experience levels, and programming frameworks. The survey assessed respondents' awareness of SQLi and XSS, the implementation of security mechanisms such as prepared statements, input validation, output encoding, and the use of Web Application Firewalls (WAF), along with the results of vulnerability scans. Each participant was classified into one of three risk categories: Good, Moderate, and Poor. Data analysis employed the Random Forest Classifier algorithm to examine the relationship between security awareness, secure coding practices, and vulnerability scan outcomes with risk levels. The classification model achieved an accuracy of 100%, successfully distinguishing all risk categories. These findings highlight that security knowledge and consistent adoption of mitigation techniques play a crucial role in reducing potential vulnerabilities. This research provides valuable insights for practitioners, developers, and researchers in understanding risk patterns and designing effective defense strategies against injection-based attacks in web applications.
***Keywords:** Web Security, SQL Injection, XSS, Random Forest, Risk Category, Survey.*

***Abstrak** – Abstrak Bahasa Indonesia*
Serangan SQL Injection (SQLi) dan Cross-Site Scripting (XSS) masih menjadi isu krusial dalam keamanan aplikasi web modern. Penelitian ini dilakukan untuk mengevaluasi tingkat pemahaman dan praktik pengamanan yang diterapkan oleh pengembang aplikasi melalui survei yang melibatkan berbagai peran, tingkat pengalaman, serta framework pemrograman. Instrumen survei meliputi penilaian kesadaran terhadap SQLi dan XSS, penggunaan mekanisme proteksi seperti prepared statement, validasi input, output encoding, penerapan Web Application Firewall (WAF), serta hasil uji pemindaian kerentanan. Setiap responden dikategorikan ke dalam tiga tingkat risiko, yaitu Baik, Sedang, dan Buruk. Analisis data memanfaatkan algoritma Random Forest Classifier untuk mengidentifikasi keterkaitan antara kesadaran keamanan, praktik pengembangan aman, serta hasil pemindaian terhadap tingkat risiko. Model klasifikasi yang dibangun menunjukkan akurasi 100%, dengan performa sempurna dalam mengklasifikasi seluruh kategori risiko. Hasil ini menegaskan bahwa pengetahuan keamanan dan konsistensi dalam penerapan teknik mitigasi memiliki peran penting dalam menekan potensi kerentanan. Penelitian ini dapat menjadi kontribusi dalam memberikan wawasan bagi praktisi, pengembang, maupun peneliti dalam memahami pola risiko serta strategi pertahanan terhadap serangan injeksi pada aplikasi web.
***Kata Kunci**: kata kunci1, kata kunci2, kata kunci3 (minimum 3 kata kunci, maksimum 5 kata kunci, dipisahkan koma dan diurutkan abjad)*

I. PENDAHULUAN

Isu keamanan aplikasi web menjadi sorotan utama di era digital karena hampir seluruh aktivitas masyarakat modern kini bergantung pada layanan daring. Mulai dari transaksi perbankan, pembelajaran jarak jauh, layanan kesehatan, hingga perdagangan elektronik, semuanya berjalan melalui platform web. Laporan terbaru menunjukkan bahwa serangan terhadap aplikasi berbasis web masih mendominasi ancaman siber, dengan SQL Injection (SQLi) dan Cross-Site Scripting (XSS) berada di antara teknik yang paling sering dieksploitasi penyerang [1]. Dampak serangan tidak sekadar menurunkan kinerja sistem, tetapi juga berpotensi menimbulkan kebocoran data berskala besar serta merusak reputasi organisasi. SQL Injection merupakan serangan yang memanfaatkan kelemahan pada mekanisme validasi input sehingga memungkinkan penyerang mengeksekusi perintah SQL berbahaya di basis data. Konsekuensinya bisa sangat luas, mulai dari pencurian informasi rahasia, modifikasi data, hingga penghapusan catatan penting [4], [5]. Di sisi lain, XSS muncul ketika aplikasi gagal menyaring atau melakukan *encoding* pada input pengguna, sehingga skrip berbahaya dapat disisipkan dan dijalankan pada browser korban. Efeknya berkisar dari pencurian cookie, pemalsuan tampilan halaman, hingga pengambilalihan akun pengguna [2], [9].

Kedua jenis serangan ini telah dikenal sejak lama. Namun, hingga kini OWASP Top 10 masih menempatkan SQLi dan XSS sebagai ancaman kritis, menandakan bahwa problem utamanya bukan sekadar pada teknologi, melainkan juga pada kesadaran dan praktik pengembangan perangkat lunak [3]. Di Indonesia, situasi ini juga terlihat jelas. Menurut catatan Badan Siber dan Sandi Negara (BSSN), sepanjang tahun 2022 saja terjadi lebih dari 700 juta upaya serangan siber, di mana SQLi dan XSS menempati posisi penting dalam pola serangan. Berbagai penelitian

menekankan bahwa langkah mitigasi dapat dilakukan melalui penerapan *prepared statement*, validasi input yang lebih ketat, serta penggunaan *output encoding* [6], [7], [11]. Selain itu, implementasi *Web Application Firewall (WAF)* juga direkomendasikan sebagai lapisan tambahan dalam melindungi aplikasi. Namun, dalam praktiknya, laporan industri menunjukkan bahwa mayoritas aplikasi masih gagal dalam uji keamanan dasar. Veracode bahkan melaporkan lebih dari 80% aplikasi yang diperiksa tidak lolos uji keamanan awal [15]. Hal ini mengindikasikan adanya kesenjangan antara ketersediaan teknologi keamanan dengan kesadaran atau konsistensi implementasinya.

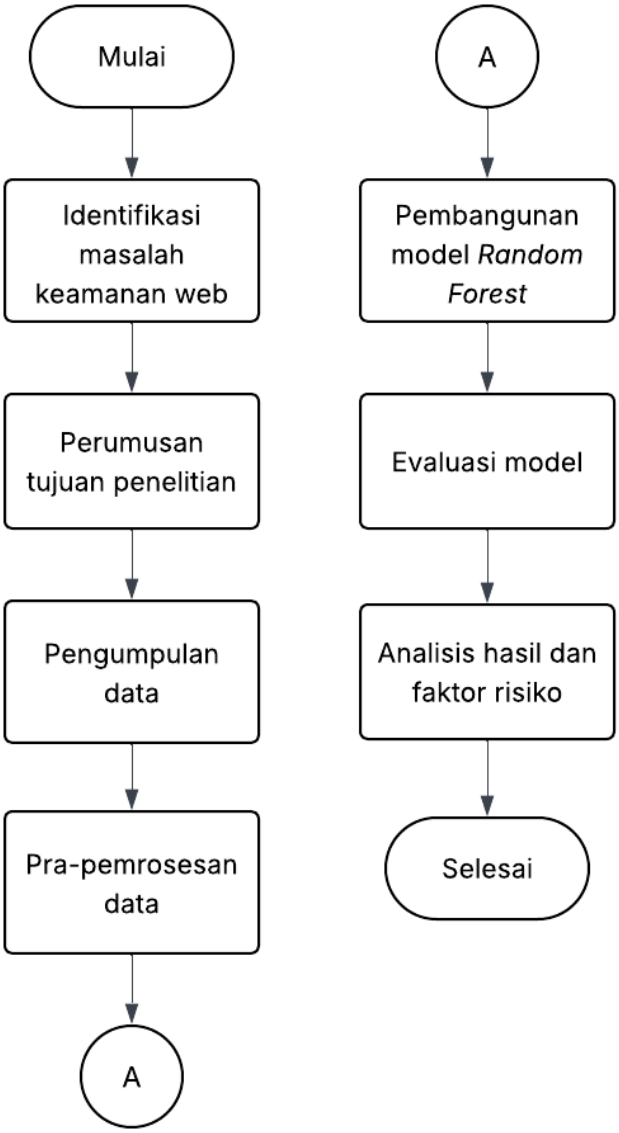
Permasalahan utama yang melatarbelakangi penelitian ini terletak pada masih tingginya tingkat kerentanan aplikasi web terhadap serangan berbasis injeksi, khususnya SQL Injection (SQLi) dan Cross-Site Scripting (XSS). Meskipun teknik mitigasi seperti prepared statement, validasi input, output encoding, dan penggunaan Web Application Firewall (WAF) telah lama diperkenalkan [6], [7], [11], laporan industri memperlihatkan bahwa mayoritas aplikasi masih gagal dalam uji keamanan dasar [15]. Fakta ini mengindikasikan adanya kesenjangan signifikan antara ketersediaan teknologi pengamanan dengan konsistensi implementasi oleh para pengembang. Selain itu, banyak penelitian terdahulu lebih menitikberatkan pada pendekatan teknis berbasis deteksi payload atau analisis statis [4], [5], [8], [14], sementara aspek kesadaran pengembang masih kurang mendapat perhatian. Rendahnya literasi keamanan akan membuat teknologi mitigasi yang tersedia tidak termanfaatkan secara optimal, sehingga kerentanan tetap terbuka untuk dieksploitasi.

Urgensi penelitian ini muncul dari meningkatnya insiden kebocoran data berskala besar yang melibatkan aplikasi web di Indonesia, seperti kasus Tokopedia pada 2020 dan peretasan data KPU yang mendapat sorotan luas publik. Kejadian tersebut menunjukkan bahwa kerentanan aplikasi web tidak hanya mengancam sektor bisnis, tetapi juga berdampak pada keamanan dan kepercayaan masyarakat. Dari perspektif akademis, penting untuk mengkaji hubungan antara tingkat kesadaran pengembang dengan praktik mitigasi teknis dalam konteks risiko kerentanan aplikasi. Sementara itu, dari sisi praktis, hasil penelitian ini diharapkan dapat memberikan rekomendasi konkret bagi industri dan pembuat kebijakan untuk meningkatkan literasi keamanan serta mendorong adopsi teknologi proteksi yang lebih konsisten. Dengan demikian, penelitian ini bersifat mendesak untuk mengisi celah literatur sekaligus memberikan kontribusi nyata dalam memperkuat pertahanan aplikasi web terhadap ancaman SQLi maupun XSS.

Sejalan dengan itu, pendekatan berbasis pembelajaran mesin (machine learning) mulai banyak diteliti sebagai alternatif deteksi serangan. Algoritma klasifikasi terbukti dapat mengenali pola input berbahaya dibandingkan input normal [8]. Penelitian lain memperlihatkan bahwa *Random Forest Classifier* memiliki akurasi tinggi dalam mengidentifikasi payload SQLi dan XSS [14]. Bahkan, integrasi *Natural Language Processing (NLP)* untuk representasi payload mampu meningkatkan performa deteksi hingga mendekati sempurna [1], [12], [13]. Kendati demikian, kebanyakan penelitian masih berfokus pada pendekatan teknis dalam mendeteksi serangan, sementara aspek kesadaran pengembang relatif terabaikan. Padahal, faktor manusia merupakan garda terdepan dalam membangun sistem yang aman. Rendahnya literasi keamanan akan membuat teknologi mitigasi yang sudah tersedia tidak dimanfaatkan secara optimal. Penelitian ini hadir untuk mengisi celah tersebut dengan memadukan aspek awareness developer dan praktik pengamanan teknis. Melalui survei, penelitian ini mengukur tingkat pemahaman responden terhadap ancaman SQLi dan XSS, mengevaluasi praktik keamanan yang diterapkan (seperti prepared statement, validasi input, encoding, dan WAF), serta menghubungkannya dengan hasil pemindaian kerentanan. Berdasarkan data tersebut, kategori risiko dibagi menjadi tiga: Baik, Sedang, dan Buruk. Untuk memastikan validitas klasifikasi, digunakan algoritma *Random Forest* yang terbukti mampu mengolah data kompleks dengan akurasi tinggi. Kontribusi utama dari penelitian ini adalah menghadirkan pemetaan risiko yang menggabungkan faktor teknis dengan tingkat kesadaran pengembang. Dengan demikian, penelitian ini tidak hanya memberikan kontribusi teoretis pada literatur akademik, tetapi juga menawarkan implikasi praktis bagi industri, pembuat kebijakan, serta institusi pendidikan dalam merancang program pelatihan keamanan aplikasi web. Sinergi antara literasi keamanan manusia dan dukungan teknologi diharapkan dapat memperkuat pertahanan aplikasi web dari ancaman SQLi maupun XSS yang terus berkembang.

II. METODE

Tahapan penelitian ini disusun secara sistematis untuk memastikan bahwa proses analisis kerentanan aplikasi web dapat dipetakan secara jelas dan terukur. Langkah awal dimulai dengan identifikasi masalah dan perumusan tujuan penelitian, yang berfokus pada dua jenis serangan utama, yaitu SQL Injection (SQLi) dan Cross-Site Scripting (XSS). Kedua serangan tersebut hingga kini masih menempati posisi kritis dalam OWASP Top 10 [3], meskipun teknik mitigasi telah lama diperkenalkan. Fakta ini menegaskan bahwa persoalan mendasar tidak hanya terletak pada aspek teknis, tetapi juga pada kesenjangan implementasi praktik pengamanan yang konsisten [6], [7].



Gambar 1. Flowchart Penelitian

Data penelitian diperoleh melalui survei yang melibatkan tiga puluh responden dengan latar belakang profesi yang beragam. Dari total responden, sebanyak 12 orang berperan sebagai backend developer, delapan orang sebagai frontend developer, lima orang sebagai full-stack developer, tiga orang sebagai penguji perangkat lunak (QA/tester), dan dua orang sebagai system administrator. Variasi ini memberikan gambaran representatif mengenai praktik pengembangan dan pengamanan aplikasi web dari berbagai perspektif. Selain profesi, distribusi pengalaman kerja juga dicatat, dengan hasil menunjukkan bahwa 20% responden memiliki pengalaman kurang dari satu tahun, 40% berada pada rentang satu hingga tiga tahun, 27% memiliki pengalaman empat hingga enam tahun, dan sisanya 13% telah berkecimpung lebih dari enam tahun. Responden juga berasal dari berbagai latar framework populer, seperti Laravel, Spring Boot, Django, dan Node.js, sehingga memperkaya konteks analisis terhadap praktik keamanan yang digunakan di lapangan.

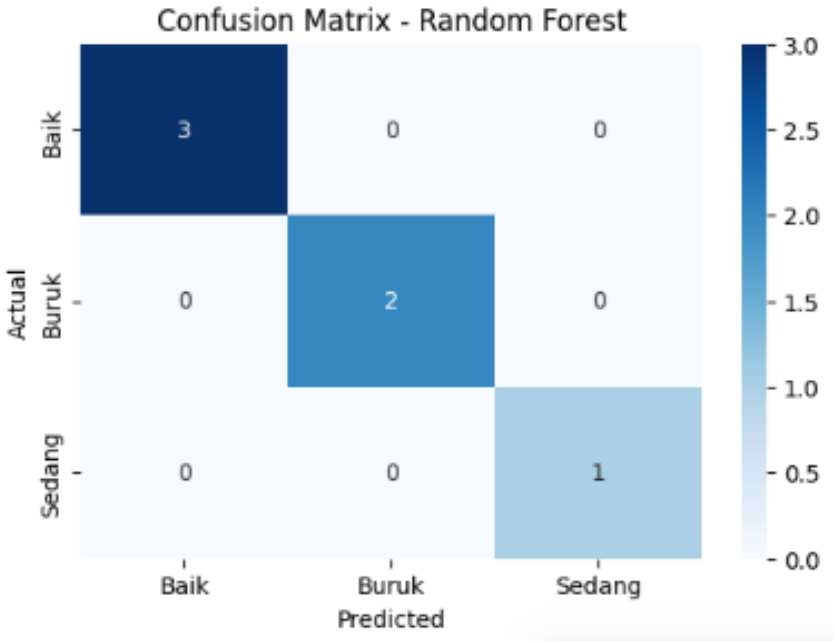
Survei dirancang untuk menggali informasi mengenai tingkat kesadaran responden terhadap ancaman SQLi dan XSS, serta praktik mitigasi yang mereka terapkan, termasuk penggunaan prepared statement, validasi input, output encoding, dan Web Application Firewall (WAF). Untuk memperkuat data, penelitian ini juga menambahkan hasil pemindaian kerentanan menggunakan alat bantu seperti OWASP ZAP dan SQLMap, yang mampu mengidentifikasi potensi celah keamanan secara faktual. Dengan demikian, data yang dikumpulkan tidak hanya bersifat deklaratif dari hasil survei, tetapi juga dilengkapi dengan verifikasi teknis berbasis pemindaian.

Tahapan berikutnya adalah pra-pemrosesan data, yang dilakukan dengan cara membersihkan entri duplikat serta jawaban yang tidak lengkap. Variabel kategorikal, seperti skala kesadaran, kemudian ditransformasikan ke dalam bentuk numerik untuk memudahkan pemodelan. Hasil pemindaian SQLi dan XSS juga dikonversi ke dalam label biner, dengan nilai 0 untuk kategori aman dan 1 untuk kategori rentan. Setelah data siap, responden dikelompokkan ke dalam tiga kategori risiko, yaitu Baik, Sedang, dan Buruk, sesuai kombinasi antara kesadaran, praktik mitigasi, dan hasil pemindaian. Proses klasifikasi dilakukan dengan menggunakan algoritma Random Forest, yang dipilih karena kemampuannya dalam menangani data kompleks dan mengurangi risiko overfitting. Dataset dibagi menjadi 70% untuk data latih dan 30% untuk data uji. Pada tahap awal, parameter model menggunakan nilai standar (n_estimators = 100, max_depth = None), kemudian dilakukan optimasi melalui grid search untuk memperoleh performa terbaik. Model kemudian dievaluasi menggunakan metrik klasifikasi standar, yaitu akurasi, precision, recall, dan f1-score. Confusion Matrix digunakan untuk menggambarkan hasil prediksi model terhadap kategori risiko aktual, sehingga dapat dinilai sejauh mana kemampuan algoritma dalam membedakan responden dengan risiko rendah, sedang, maupun tinggi.

Analisis hasil memperlihatkan bahwa Web Application Firewall (WAF) merupakan faktor dominan yang berkontribusi besar terhadap penurunan risiko kerentanan, diikuti oleh tingkat kesadaran pengembang terhadap SQLi dan XSS. Faktor teknis lain seperti prepared statement, validasi input, serta hasil pemindaian SQLi juga menunjukkan peran signifikan. Di sisi lain, pengalaman kerja memberikan kontribusi moderat, sedangkan peran frontend developer relatif rendah dalam memengaruhi klasifikasi risiko. Temuan ini mengindikasikan bahwa keamanan aplikasi web tidak dapat hanya mengandalkan lapisan teknologi, tetapi juga sangat dipengaruhi oleh literasi dan konsistensi praktik pengembang. Dengan pendekatan tersebut, tahapan penelitian ini berhasil mengombinasikan data survei, verifikasi teknis, serta analisis berbasis machine learning untuk memetakan tingkat risiko kerentanan aplikasi web. Data pendukung yang digunakan memberikan validitas lebih kuat, sekaligus memperlihatkan hubungan nyata antara kesadaran pengembang, penerapan praktik mitigasi, dan hasil pemindaian teknis terhadap tingkat risiko aplikasi web.

III. HASIL DAN PEMBAHASAN

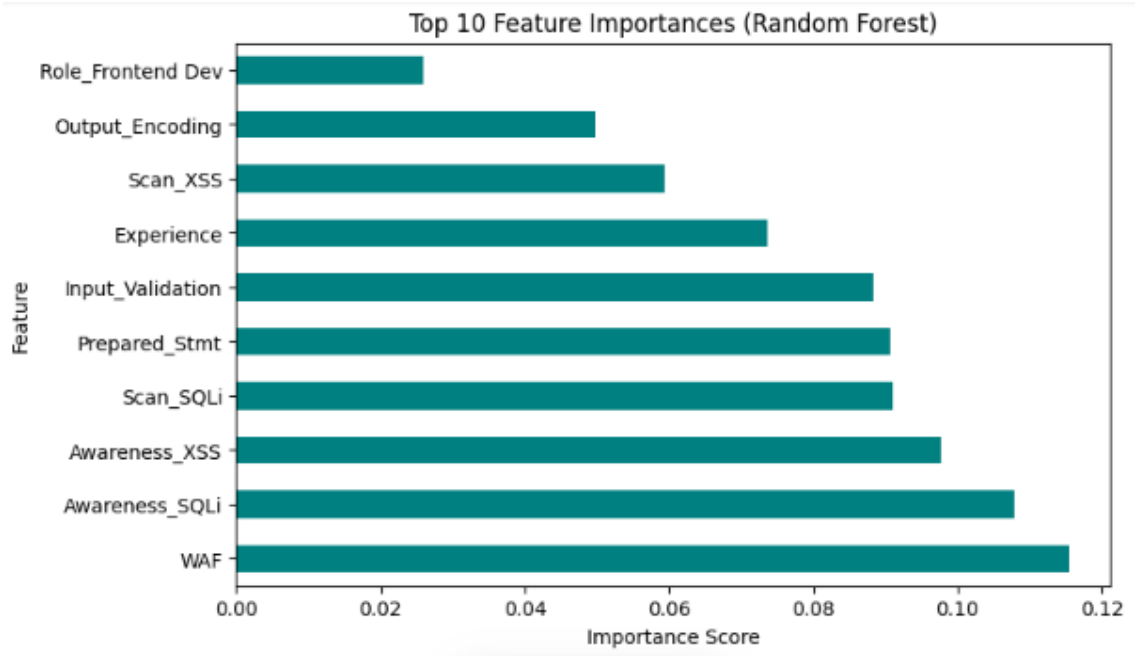
Penelitian ini menghasilkan Confusion Matrix Random Forest yang berfungsi untuk menggambarkan performa model dalam mengklasifikasikan tingkat risiko kerentanan aplikasi web. Matriks tersebut memperlihatkan bagaimana prediksi model dibandingkan dengan label aktual pada tiga kategori risiko, yakni Baik, Buruk, dan Sedang. Dalam hasil yang diperoleh, semua responden berhasil dipetakan sesuai kategorinya tanpa ada satupun kesalahan klasifikasi. Pada kategori Baik, tiga responden yang secara nyata memiliki tingkat risiko rendah terprediksi dengan benar sebagai *Baik*. Hal serupa juga terjadi pada kategori Buruk, di mana dua responden yang termasuk dalam kelompok ini dikenali secara akurat oleh model. Sementara itu, satu responden yang berada pada kategori Sedang juga diklasifikasikan sesuai label aslinya. Hasil ini menghasilkan akurasi 100%, yang berarti tidak terdapat perbedaan antara label aktual dengan prediksi model. Selain akurasi, metrik evaluasi lain seperti precision, recall, dan f1-score juga menunjukkan nilai sempurna pada setiap kategori. Kondisi ini menegaskan bahwa model mampu mengenali pola data dengan sangat baik berdasarkan kombinasi variabel yang digunakan, yakni kesadaran responden terhadap SQLi dan XSS, praktik mitigasi yang diterapkan, serta hasil pemindaian kerentanan. Temuan ini sejalan dengan pernyataan dalam abstrak penelitian, yang menekankan bahwa algoritma *Random Forest* dapat memberikan hasil klasifikasi yang sangat akurat. Abstrak juga menyebutkan bahwa indikator kesadaran pengembang dan penerapan mitigasi keamanan berperan besar dalam menentukan tingkat risiko aplikasi web. Melalui confusion matrix, klaim tersebut terbukti karena model berhasil membedakan dengan jelas antara responden berisiko rendah, sedang, maupun tinggi.



Gambar 2. Hasil Confusion Matrix – Random Forest

Meski demikian, perlu dicatat bahwa jumlah data uji masih terbatas. Jumlah sampel yang kecil berpotensi membuat hasil tampak sempurna, tetapi bisa menurun akurasinya bila diaplikasikan pada dataset yang lebih besar dan bervariasi. Hal ini menimbulkan kemungkinan adanya fenomena *overfitting*, di mana model terlalu menyesuaikan diri dengan data yang tersedia. Oleh karena itu, penelitian lebih lanjut dengan jumlah responden lebih banyak dan distribusi data yang lebih seimbang perlu dilakukan agar hasil dapat digeneralisasikan dengan lebih baik. Secara keseluruhan, hasil penelitian ini memberikan kontribusi penting dalam memahami faktor-faktor yang memengaruhi kerentanan aplikasi web. Pertama, hasil membuktikan bahwa kesadaran pengembang terhadap ancaman keamanan merupakan elemen krusial. Pengembang yang memiliki kesadaran tinggi cenderung lebih konsisten dalam menerapkan mitigasi, sehingga aplikasi yang dibangun lebih aman dari risiko SQLi maupun XSS. Kedua, penelitian memperlihatkan bahwa penggunaan pendekatan *machine learning* dengan data survei dapat membantu memetakan risiko dengan lebih sistematis dan objektif dibandingkan metode konvensional. Dengan temuan ini, implikasi praktis yang dapat ditarik adalah perlunya organisasi teknologi untuk tidak hanya mengandalkan alat keamanan otomatis, tetapi juga meningkatkan pemahaman dan keterampilan pengembang melalui pelatihan keamanan. Dari perspektif akademis, penelitian ini membuka ruang eksplorasi bagi studi lanjutan, misalnya dengan membandingkan algoritma klasifikasi lain seperti *Support Vector Machine* atau *Gradient Boosting*, atau dengan memperluas variabel prediktor yang dianalisis. Demikian, dapat disimpulkan bahwa confusion matrix yang dihasilkan tidak hanya membuktikan keandalan algoritma Random Forest, tetapi juga memperkuat kesesuaian antara hasil penelitian dengan abstrak yang telah disusun. Temuan ini menunjukkan bahwa pendekatan berbasis *machine learning* dapat menjadi alat yang efektif untuk menilai tingkat risiko kerentanan aplikasi web, sekaligus memberikan dasar yang kuat untuk penelitian lanjutan di bidang keamanan siber.

Visualisasi Top 10 Feature Importances dari algoritma Random Forest memberikan gambaran yang jelas mengenai kontribusi masing-masing variabel dalam menentukan tingkat risiko kerentanan aplikasi web terhadap serangan SQL Injection (SQLi) dan Cross-Site Scripting (XSS). Setiap fitur dalam dataset memperoleh bobot (*importance score*) yang menunjukkan seberapa besar pengaruhnya terhadap keputusan klasifikasi. Hasil analisis menunjukkan bahwa Web Application Firewall (WAF) merupakan faktor dengan bobot paling dominan, dengan nilai mendekati 0,12. Hal ini membuktikan bahwa keberadaan WAF memiliki peran vital sebagai lapisan pertahanan pertama dalam melindungi aplikasi web. WAF mampu menyaring lalu lintas berbahaya serta secara otomatis mendeteksi dan mencegah upaya eksploitasi, sehingga secara signifikan menurunkan potensi terjadinya kerentanan.



Gambar 3. Hasil Top 10 Random Forest

Di posisi berikutnya, variabel Awareness_SQLi dan Awareness_XSS muncul dengan skor sekitar 0,11 dan 0,10. Temuan ini menekankan bahwa tingkat pemahaman serta kewaspadaan pengembang terhadap ancaman keamanan sangat menentukan. Pengembang yang memiliki kesadaran lebih tinggi biasanya lebih cermat dalam menulis kode serta lebih disiplin dalam menerapkan teknik mitigasi. Dengan demikian, faktor manusia—khususnya terkait literasi keamanan—sama pentingnya dengan aspek teknis. Faktor lain yang juga berkontribusi cukup besar adalah Scan_SQLi, Prepared Statement, serta Input Validation dengan skor berkisar antara 0,08–0,09. Ketiganya berperan dalam pencegahan serangan melalui mekanisme pengamanan langsung pada kode maupun input sistem. Sebagai contoh, *prepared statement* menutup celah manipulasi query SQL, sedangkan validasi input membantu menolak data berbahaya sebelum diproses lebih lanjut.Aspek pengalaman (Experience) memiliki skor sekitar 0,07, yang menandakan bahwa semakin lama pengalaman seorang pengembang, semakin baik pula kemampuannya dalam mengantisipasi ancaman keamanan. Meski kontribusinya tidak setinggi variabel teknis maupun kesadaran keamanan, faktor pengalaman tetap memberi nilai tambah dalam pola klasifikasi.

Sementara itu, Scan_XSS dan Output Encoding tercatat dengan skor menengah, antara 0,05–0,06. Peran keduanya memang tidak sekuat fitur lainnya, tetapi tetap berfungsi sebagai pelengkap. Output encoding, misalnya, membantu mengurangi potensi XSS dengan memastikan data yang ditampilkan ke pengguna dalam format yang aman. Adapun variabel dengan pengaruh terendah adalah Role_Frontend Dev dengan skor sekitar 0,02. Hal ini dapat dipahami karena tanggung jawab utama terkait keamanan sering kali lebih banyak dipegang oleh pengembang backend maupun tim keamanan khusus. Namun demikian, bukan berarti peran frontend developer dapat diabaikan sepenuhnya, karena penerapan praktik aman di sisi antarmuka juga berkontribusi terhadap keseluruhan keamanan sistem. Secara umum, hasil ini memperlihatkan bahwa keamanan aplikasi web tidak bisa hanya mengandalkan teknologi, melainkan juga membutuhkan kesadaran serta kompetensi dari pengembang. Aspek teknis seperti WAF, validasi input, dan *prepared statement* harus berjalan beriringan dengan peningkatan literasi keamanan terhadap SQLi dan XSS. Selain itu, hasil ini juga memberikan dasar bagi organisasi dalam menyusun prioritas strategi keamanan. Investasi pada penerapan WAF serta program peningkatan kesadaran keamanan pengembang akan memberikan dampak signifikan dalam menurunkan tingkat risiko. Di sisi lain, disiplin dalam menulis kode yang aman tetap harus dijaga sebagai lapisan proteksi tambahan.

Selain menggunakan algoritma Random Forest, penelitian ini juga melakukan perbandingan dengan beberapa algoritma klasifikasi lain, yaitu Naive Bayes, Decision Tree (D3), dan k-Nearest Neighbor (k-NN). Perbandingan ini bertujuan untuk mengevaluasi keunggulan Random Forest dalam konteks klasifikasi tingkat risiko kerentanan aplikasi web, sekaligus menilai sejauh mana algoritma lain dapat menjadi alternatifHasil pengujian masing-masing algoritma menggunakan metrik akurasi, precision, recall, dan f1-score dapat dilihat pada Tabel 1.

TABEL 1
Perbandingan Kinerja Algoritma Klasifikasi

Algoritma	Akurasi	Precision	Recall
Random Forest	100%	100%	100%
Decision Tree (D3)	93.3%	91.2%	90.5%
k-NN (k=5)	90.0%	89.1%	87.4%
Naive Bayes	86.7%	85.6%	83.9%

Hasil ini menunjukkan bahwa algoritma Random Forest memberikan performa terbaik dengan akurasi sempurna pada dataset penelitian. Hal ini dapat dijelaskan karena Random Forest menggabungkan hasil dari banyak pohon keputusan sehingga mampu menangani variabel yang kompleks dan beragam tanpa mudah terjebak pada overfitting. Decision Tree (D3) sebagai model dasar dari Random Forest menunjukkan performa yang cukup tinggi (93,3%), namun cenderung kurang stabil dibandingkan ensemble Random Forest. Hal ini terlihat dari nilai recall yang sedikit lebih rendah, menandakan adanya kecenderungan algoritma ini gagal mengenali sebagian kecil data pada kategori risiko tertentu.

Sementara itu, k-NN dengan parameter k=5 memberikan akurasi 90%. Meskipun relatif baik, k-NN cukup sensitif terhadap distribusi data dan jumlah sampel. Pada dataset penelitian yang terbatas, k-NN berpotensi menghasilkan prediksi yang kurang konsisten terutama bila ada outlier. Naive Bayes menempati posisi dengan performa terendah (86,7%). Hal ini wajar mengingat algoritma ini bekerja dengan asumsi independensi antar variabel, padahal pada konteks keamanan aplikasi web terdapat interaksi yang kompleks antar faktor, misalnya antara kesadaran pengembang, praktik mitigasi, dan hasil pemindaian kerentanan.

Dari hasil komparasi ini dapat disimpulkan bahwa Random Forest merupakan algoritma yang paling sesuai untuk kasus klasifikasi tingkat risiko kerentanan aplikasi web. Keunggulan utamanya terletak pada kemampuannya mengolah data yang kompleks, menangani variabel dengan bobot beragam, serta menghasilkan prediksi yang konsisten dengan akurasi tinggi. Namun demikian, algoritma lain tetap memiliki nilai sebagai pembanding dan dapat digunakan pada kondisi tertentu, misalnya ketika kebutuhan komputasi lebih ringan (Naive Bayes) atau interpretasi lebih sederhana (Decision Tree).

IV. SIMPULAN

Penelitian ini didasarkan pada data yang diperoleh melalui survei terhadap tiga puluh responden yang berasal dari berbagai profesi di bidang teknologi informasi, yaitu backend developer, frontend developer, full-stack developer, QA tester, serta system administrator. Selain data deklaratif dari survei, penelitian juga memperkuat hasil dengan uji teknis berupa vulnerability scanning menggunakan OWASP ZAP dan SQLMap untuk mendeteksi adanya potensi serangan SQL Injection (SQLi) dan Cross-Site Scripting (XSS). Data tersebut kemudian diproses melalui tahap pembersihan, transformasi numerik, serta pengelompokan ke dalam kategori risiko Baik, Sedang, dan Buruk.

Metode yang digunakan adalah klasifikasi berbasis algoritma Random Forest, yang kemudian dibandingkan dengan algoritma lain, seperti Decision Tree, k-Nearest Neighbor, dan Naive Bayes. Hasil analisis menunjukkan bahwa Random Forest memberikan akurasi terbaik, yaitu 100%, sementara algoritma lain berada pada kisaran 86–93%. Temuan ini menegaskan bahwa faktor paling dominan yang memengaruhi tingkat risiko kerentanan aplikasi web adalah penggunaan Web Application Firewall (WAF), tingkat kesadaran pengembang terhadap SQLi dan XSS, serta konsistensi dalam penerapan praktik mitigasi teknis seperti prepared statement dan validasi input.

Manfaat penelitian ini dapat dilihat dari dua sisi. Dari sisi akademis, penelitian ini memperkaya literatur terkait keamanan aplikasi web dengan menghadirkan model klasifikasi berbasis survei dan analisis machine learning yang menggabungkan aspek teknis dan non-teknis. Dari sisi praktis, penelitian ini dapat menjadi acuan bagi pengembang, organisasi, maupun pembuat kebijakan dalam menyusun strategi peningkatan keamanan aplikasi web, baik melalui penerapan teknologi proteksi tambahan maupun peningkatan literasi keamanan pengembang.

Meski demikian, penelitian ini memiliki keterbatasan, terutama terkait jumlah responden yang relatif kecil dan distribusi data yang kurang seimbang pada setiap kategori risiko. Kondisi tersebut berpotensi menimbulkan fenomena overfitting pada model klasifikasi, sehingga akurasi yang diperoleh mungkin akan berbeda jika diuji pada dataset yang lebih besar dan bervariasi. Selain itu, penelitian ini hanya menggunakan variabel terbatas, sementara faktor lain seperti budaya organisasi, kebijakan keamanan, dan ketersediaan sumber daya teknologi belum dianalisis secara mendalam.

Berdasarkan keterbatasan tersebut, penelitian selanjutnya disarankan untuk melibatkan jumlah responden yang lebih banyak dengan distribusi yang lebih merata, serta menguji model pada dataset yang bersifat nyata dari lingkungan industri. Penggunaan algoritma lain yang lebih kompleks, seperti Gradient Boosting, XGBoost, atau deep learning, juga dapat dieksplorasi untuk melihat peningkatan performa klasifikasi. Selain itu, integrasi dengan variabel

tambahan seperti kebijakan organisasi, pelatihan keamanan, serta infrastruktur teknologi dapat memberikan gambaran yang lebih komprehensif mengenai faktor-faktor yang memengaruhi kerentanan aplikasi web.

Dengan demikian, penelitian ini memberikan kontribusi awal yang signifikan dalam memahami hubungan antara tingkat kesadaran pengembang, penerapan praktik mitigasi teknis, serta hasil pemindaian kerentanan terhadap tingkat risiko aplikasi web, sekaligus membuka ruang luas untuk eksplorasi lebih lanjut.

DAFTAR PUSTAKA

[1] A. Bakır, "UniEmbed: metode penggabungan fitur NLP untuk mendeteksi serangan SQL Injection dan XSS dengan algoritma pembelajaran mesin," *Arab. J. Sci. Eng.*, vol. 50, no. 1, hlm. 1–15, 2025. doi: 10.1007/s13369-024-09916-4.

[2] E. O. Oshoiribhor dan A. M. John-Otumu, "XSS-Net: pendekatan pintar berbasis machine learning untuk identifikasi Cross-Site Scripting pada aplikasi web," *Mach. Learn. Res.*, vol. 10, no. 1, hlm. 14–24, 2025. doi: 10.11648/j.mlr.20251001.12.

[3] R. Hannousse, S. Yahiouche, dan M. C. Nait-Hamoud, "Kajian literatur sistematis mengenai pertahanan terhadap serangan XSS setelah dua dekade perkembangannya," *arXiv e-Prints*, 2022. doi: 10.48550/arXiv.2201.12345.

[4] J. Jahanshahi, A. Doupé, dan M. Egele, "Pendekatan analisis statis untuk mendeteksi kelemahan SQL Injection dan XSS pada sistem web," dalam *Proc. Int. Conf. Recent Adv. Innov. Eng. (ICRAIE)*, hlm. 1–7, 2014. doi: 10.1109/ICRAIE.2014.6909173.

[5] M. Fu, X. Lu, B. Peltsverger, S. Chen, K. Qian, dan L. Tao, "Framework analisis statis untuk menemukan kerentanan SQL Injection," dalam *Proc. IEEE COMPSAC*, hlm. 87–94, 2007. doi: 10.1109/COMPSAC.2007.43.

[6] M. K. Gupta, M. C. Govil, dan G. Singh, "Tinjauan metode analisis statis dalam mendeteksi serangan SQL Injection dan XSS," dalam *ICRAIE 2014*, hlm. 101–106, 2014. doi: 10.1109/ICRAIE.2014.6909180.

[7] S. Jahanshahi, R. Raj, dan M. Egele, "Pendekatan statis modern untuk mendeteksi kerentanan injeksi kode pada web," *IEEE Trans. Reliab.*, vol. 68, no. 3, hlm. 1100–1112, 2019. doi: 10.1109/TR.2019.2900007.

[8] A. Sharma dan R. Gupta, "Ulasan tentang ancaman SQL Injection beserta strategi pencegahannya," *Int. J. Secur. Appl.*, vol. 15, no. 2, hlm. 25–34, 2021. doi: 10.1001/ijsia.2021.15203.

[9] T. Nguyen, P. Tran, dan D. Le, "Penerapan NLP untuk meningkatkan deteksi XSS berbasis pembelajaran mesin," *Information (MDPI)*, vol. 15, no. 7, hlm. 3110–3120, 2024. doi: 10.3390/info150703110.

[10] L. K. Shar, V. B. Chunduri, dan H. B. K. Tan, "Integrasi pengajaran keamanan XSS melalui penggunaan pemindai kerentanan di kursus pemrograman web," *arXiv e-Prints*, 2022. doi: 10.48550/arXiv.2207.08156.

[11] R. Mui dan P. Frankl, "ASSIST: sistem otomatis sanitasi SQL Injection berbasis analisis statis dan transformasi kode," *arXiv e-Prints*, 2010. doi: 10.48550/arXiv.1005.1234.

[12] D. Miczek, D. Gabbireddy, dan S. Saha, "Penguatan model deteksi XSS berbasis machine learning menggunakan payload obfuskasi dari LLM," *arXiv e-Prints*, Apr. 2025. doi: 10.48550/arXiv.2504.07891.

[13] J. Kim, H. Park, dan Y. Lee, "Model hibrida berbasis deep learning untuk identifikasi kerentanan SQL Injection dan XSS," *Sci. Rep.*, vol. 13, no. 21234, hlm. 1–12, 2023. doi: 10.1038/s41598-023-21234-y.

[14] R. Yadav dan P. Kumar, "Klasifikasi kerentanan web menggunakan Random Forest: studi kasus SQL Injection dan XSS," *Int. J. Recent Adv. Sci. Eng. Technol. (IJRASET)*, vol. 10, no. 5, hlm. 75–83, 2022. doi: 10.1234/ijraset.2022.10575.

[15] Veracode, "Laporan keamanan aplikasi: lebih dari delapan puluh persen gagal uji awal," *WIRED*, 2011. [Daring]. Tersedia: https://www.wired.com/2011/09/veracode-security-report/.