BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan aplikasi web yang semakin pesat menjadikan aspek keamanan sebagai perhatian utama dalam pengembangannya. Aplikasi web yang tidak dilengkapi dengan mekanisme keamanan yang memadai berpotensi menjadi target serangan siber. Salah satu jenis serangan yang masih sering ditemukan adalah *Cross-Site Scripting* (*XSS*). Serangan ini memungkinkan penyerang menyisipkan skrip berbahaya ke dalam halaman web yang kemudian dieksekusi pada sisi pengguna. Dampak dari serangan *XSS* dapat berupa pencurian data, pengambilalihan sesi pengguna, hingga penyalahgunaan informasi sensitif.

Tingginya risiko yang ditimbulkan oleh serangan *XSS* tercermin pada berbagai laporan dan hasil penelitian. Dalam laporan OWASP Top 10 tahun 2021, *Cross-Site Scripting* (*XSS*) diklasifikasikan sebagai bagian dari kategori Injection, yang menempati urutan ketiga kerentanan aplikasi web paling umum secara global Temuan tersebut mengindikasikan bahwa XSS merupakan ancaman yang signifikan secara global. Selain itu dari penelitian (Azizah dkk., 2026) menunjukan bahwa dalam web instunsi Universitas di temukan seperti reflected XSS dan Stord XSS Kondisi ini menunjukkan bahwa permasalahan *XSS* masih relevan untuk dikaji.

Permasalahan keamanan aplikasi web juga terlihat dalam konteks nasional berdasarkan laporan resmi pemerintah. Laporan Information Technology Security Assessment (ITSA) yang diterbitkan oleh BSSN menunjukkan bahwa aplikasi berbasis web masih ditemukan memiliki berbagai kerentanan keamanan.Kerentanan tersebut berpotensi menimbulkan dampak seperti kebocoran data dan gangguan layanan. Hal ini menunjukkan bahwa tingkat keamanan aplikasi web masih perlu ditingkatkan. Oleh karena itu, isu keamanan aplikasi web menjadi perhatian penting.

Keberagaman pola dan karakteristik serangan *XSS* menyebabkan proses deteksi menjadi semakin menantang. Variasi payload dan teknik penyamaran dapat menurunkan performa pendekatan deteksi konvensional yang umumnya berbasis

aturan statis. Pendekatan tersebut cenderung efektif pada pola serangan sederhana, namun kurang adaptif terhadap variasi serangan yang kompleks. Akibatnya, tingkat akurasi deteksi dapat menurun pada kondisi tertentu. Hal ini menunjukkan adanya keterbatasan pendekatan deteksi konvensional.

Berdasarkan kondisi tersebut, pendekatan berbasis *Machine Learning* mulai banyak dikaji dalam penelitian deteksi serangan *XSS*. Pendekatan ini memungkinkan model mempelajari karakteristik serangan secara otomatis dari data. Evaluasi performa model menjadi aspek penting untuk menilai efektivitas pendekatan tersebut dalam mendeteksi berbagai variasi serangan *XSS*. Oleh karena itu, penelitian ini difokuskan pada analisis performa model *Machine Learning* dalam mendeteksi serangan *XSS*, tanpa membahas aspek implementasi sistem keamanan secara langsung.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah dipaparkan, dapat diidentifikasi beberapa permasalahan mengenai keamanan web dan deteksi serangan *XSS* sebagai berikut:

1. Meningkatnya kompleksitas serangan *XSS* pada aplikasi web

   Serangan *XSS* pada aplikasi web terus berkembang baik dari sisi jumlah maupun kompleksitas payload. Teknik serangan yang semakin bervariasi menimbulkan tantangan dalam proses deteksi, khususnya pada payload yang telah mengalami penyamaran.
2. Keterbatasan metode deteksi berbasis aturan

   Pendekatan deteksi *XSS* berbasis signature dan rule-based memiliki keterbatasan dalam mengenali variasi payload *XSS* yang kompleks. Metode ini cenderung kurang adaptif terhadap teknik serangan baru yang tidak sesuai dengan pola yang telah didefinisikan sebelumnya.
3. Keterbatasan model *Machine Learning* Tunggal

   Pendekatan *Machine Learning* yang menggunakan satu jenis model memiliki keterbatasan masing-masing. Model seperti *Random Forest* efisien dalam pemrosesan, namun kurang optimal dalam memahami konteks teks, sedangkan model berbasis *transformer* seperti *BERT* memiliki

kemampuan representasi konteks yang lebih baik namun membutuhkan sumber daya komputasi yang lebih besar.

4. Tingginya kesalahan klasifikasi dalam deteksi *XSS*

   Model deteksi *XSS* yang ada masih berpotensi menghasilkan kesalahan klasifikasi berupa *False Positive* dan false negative. Kondisi ini menunjukkan perlunya evaluasi yang lebih mendalam terhadap kemampuan model dalam membedakan payload *XSS* dan data normal.

5. Kesulitan mendeteksi payload *XSS* yang telah disamarkan

   Payload *XSS* modern sering menggunakan teknik penyamaran seperti encoding, manipulasi karakter, dan penyisipan komentar untuk menghindari deteksi. Teknik ini menyulitkan model konvensional dalam mengenali pola serangan secara akurat.

6. Perlunya analisis *trade-off* antara akurasi dan efisiensi model

   Sebagian besar penelitian masih berfokus pada peningkatan akurasi deteksi, tanpa mempertimbangkan efisiensi komputasi. Padahal, analisis trade-off antara akurasi dan waktu inferensi model menjadi aspek penting dalam evaluasi performa model Machine Learning.

1.3 Rumusan Masalah

1. Bagaimana mengembangkan model *hybrid* deteksi *XSS* yang mengintegrasikan *Random Forest* sebagai metode dasar dengan *BERT* sebagai metode lanjutan?
2. Bagaimana tingkat akurasi model *hybrid* yang dikembangkan dalam mendeteksi payload *XSS*?
3. Bagaimana kinerja dan Confident score model *hybrid* dalam mendeteksi serangan *XSS* dengan teknik penyamaran?

1.4 Tujuan

1. Bagaimana mengembangkan model *hybrid* deteksi *XSS* yang mengintegrasikan *Random Forest* sebagai metode dasar dengan *BERT* sebagai metode lanjutan?
2. Bagaimana tingkat akurasi model *hybrid* yang dikembangkan dalam mendeteksi payload *XSS*?

3. .Bagaimana kinerja dan confined score model *hybrid* dalam mendeteksi serangan *XSS* dengan teknik penyamaran?

1.5 Batasan Masalah

1. Penelitian ini hanya berfokus pada klasifikasi payload *Cross-Site Scripting* (*XSS*) dan tidak mencakup jenis kerentanan web lainnya seperti SQL Injection atau CSRF.
2. Dataset yang digunakan terbatas pada payload *XSS* yang tersedia secara publik dari Kaggle dan sumber payload tervalidasi lainnya, tanpa melibatkan data dari lingkungan produksi atau serangan aktual.
3. Evaluasi model dilakukan dalam lingkungan terkontrol menggunakan dataset statis, bukan pada implementasi produksi atau sistem keamanan web aktif.
4. Model yang digunakan merupakan *hybrid* model berbasis *Machine Learning* yang mengombinasikan *Random Forest* dan *BERT* dengan skenario penggunaan terbatas (*use case-based*), tanpa melakukan optimasi lanjutan terhadap arsitektur model.
5. Klasifikasi dibatasi pada dua kelas, yaitu payload berbahaya (*XSS*) dan payload normal, tanpa kategorisasi detail jenis serangan *XSS*.
6. Penelitian menitikberatkan pada analisis performa model machine learning, bukan pada implementasi sistem keamanan web secara menyeluruh.
7. Hasil penelitian berupa metrik kinerja model seperti akurasi, presisi, dan *recall* berdasarkan analisis dataset, tanpa melakukan eksekusi payload atau pengujian pada aplikasi web nyata.
8. Teknik penyamaran yang diuji terbatas pada variasi encoding dan obfuscation yang tersedia dalam dataset yang digunakan, dan tidak mencakup serangan adaptif atau teknik penghindaran tingkat lanjut.
9. Implementasi penelitian menggunakan sumber daya akademik terbatas seperti *Google colab* dan komputer lokal, tanpa menggunakan infrastruktur komputasi khusus.

1.6 Manfaat

1. Manfaat Teoritis

Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan pendekatan *hybrid Machine Learning* untuk deteksi serangan *XSS* serta menjadi referensi integrasi antara model *Machine Learning* tradisional dan model berbasis *transformer*.

2. Manfaat Praktis

   Penelitian ini dapat memberikan gambaran performa model *hybrid* dalam mendeteksi serangan *XSS* dan menjadi acuan awal bagi penelitian selanjutnya di bidang keamanan aplikasi web.

3. Manfaat Metodologis

   Penelitian ini menyediakan skenario evaluasi dan pembandingan performa model deteksi *XSS* yang dapat digunakan sebagai rujukan metodologi pada penelitian serupa, khususnya dalam pengujian model berbasis *Machine Learning* dan *transformer* menggunakan dataset statis.