BAB I
PENDAHULUAN

1.1. Latar Belakang

Di era modern, perkembangan teknologi informasi memiliki peran penting dalam mendukung berbagai aktivitas, seperti mengelola, memproses, memperoleh, menyusun, dan menyimpan data dengan berbagai metode untuk menghasilkan informasi yang berkualitas dan akurat (Supartini Reni, 2023). Namun, seiring dengan pesatnya perkembangan teknologi informasi, kemudahan akses dan pengolahan informasi yang ditawarkannya bagi masyarakat juga membawa potensi terjadinya serangan siber. Data statistik dari Badan Siber dan Sandi Negara (BSSN) menunjukkan bahwa pada tahun 2022 terjadi 370,02 juta serangan siber di Indonesia. Salah satu serangan siber yang paling umum adalah *code injection* (Kurt Baker, 2024). Serangan *code injection*, seperti *SQL Injection* dan *Cross-Site Scripting (XSS)*, memungkinkan *hacker* mengeksploitasi celah keamanan pada aplikasi web untuk melakukan aktivitas berbahaya, seperti memodifikasi data, mengganggu operasional bisnis, hingga merusak data (Alnabulsi et al., 2024).

SQL Injection adalah kerentanan yang terjadi ketika *hacker* memiliki kemampuan untuk memanipulasi *Structured Query Language (SQL)* (Sari et al., 2023). Kerentanan ini memungkinkan pengguna atau program aplikasi untuk berinteraksi dengan *database* dengan cara memasukkan data baru, menghapus data lama, dan mengubah data yang sudah disimpan (Alghawazi et al., 2023). Dampaknya *hacker* dapat memperoleh akses tidak sah ke data sensitif, yang mengakibatkan kebocoran informasi atau bahkan merusak sistem database. Serangan SQL Injection sangat berbahaya karena sering digunakan untuk mencari data penting, seperti informasi pribadi, kredensial pengguna, atau data sensitif lainnya, yang berpotensi disalahgunakan untuk aktivitas berbahaya. Di sisi lain, *Open Web Appliclation Security Project (OWASP)* menyebutkan bahwa serangan SQL Injection menduduki perangkat ke-3 sebagai resiko keamanan aplikasi website paling kritis pada tahun 2021, dengan 274.000 insiden yang dilaporkan (Stiawan et al., 2023).

Cross-Site Scripting (XSS) adalah sebuah bentuk eksploitasi keamanan di mana *hacker* menyisipkan kode berbahaya (biasanya berupa Javascript) ke dalam halaman web di sisi klien (Chandra et al., 2024). Cross-Site Scripting (XSS) dapat mengakibatkan data sensitif diubah dan diekspos. Salah satu metode yang paling umum adalah mengakses sesi atau mencuri *cookie* untuk mengumpulkan informasi rahasia (Wibowo, 2021). Hal ini dapat menyebabkan berbagai masalah, termasuk pencurian identitas, akses tidak sah ke akun pengguna, dan kerugian finansial. Cross-Site Scripting secara konsisten muncul dalam daftar 10 resiko keamanan aplikasi web teratas OWASP, yang mencakup 37,2% dari insiden yang laporkan dan masih menjadi ancaman yang berisiko tinggi (Stiawan et al., 2023).

Deteksi dini terhadap serangan *code injection* merupakan kunci dalam memitigasi potensi kerugian yang ditimbulkan (Arum Sari, 2024). Namun, hingga kini pendeteksian *code injection* masih sering dilakukan secara manual, misalnya dengan menganalisis log paket jaringan pada router atau *access point* (Dzulnufrie Hafriadi & Ardiansyah, 2024). Pendekatan ini memiliki keterbatasan signifikan, terutama dalam hal skalabilitas, karena tidak memungkinkan untuk memeriksa setiap konten paket jaringan secara menyeluruh akibat volume data yang sangat besar (Crespo-Martínez et al., 2023). Di sisi lain, upaya pencegahan masih banyak mengandalkan metode *penetration testing* yang bersifat manual, di mana serangan disimulasikan untuk mengidentifikasi dan mengevaluasi titik kerentanan dalam aplikasi web (Bastian et al., 2020). Pendekatan ini tidak hanya membutuhkan waktu yang lama, tetapi juga sumber daya manusia yang terampil untuk meninjau dan menganalisis hasil secara menyeluruh. Mengingat volume trafik internet yang sangat masif, melakukan analisis manual terhadap setiap request yang masuk menjadi tidak realistis. Oleh karena itu, dibutuhkan sistem yang mampu mengotomatiskan proses klasifikasi, dengan membedakan apakah suatu request bersifat ‘berbahaya’ atau ‘aman’ secara real-time. Klasifikasi otomatis semacam ini menjadi garis pertahanan awal yang krusial untuk memfilter ancaman sebelum menimbulkan kerusakan yang lebih besar. Untuk menjawab tantangan tersebut, pendekatan berbasis *machine learning* menjadi salah satu solusi yang menjanjikan. *Machine learning* merupakan cabang dari kecerdasan buatan (AI) yang memungkinkan sistem komputer mempelajari pola dari data yang ada dan membuat

keputusan berdasarkan pembelajaran tersebut (Hasan et al., 2022). Dengan memanfaatkan *machine learning*, sistem deteksi dapat terus memperbarui pengetahuannya dari data serangan yang telah teridentifikasi, sehingga akurasinya meningkat seiring waktu dan mampu menghadapi berbagai jenis *code injection* yang semakin kompleks.

Terdapat berbagai teknik klasifikasi yang baik dalam literatur, termasuk artificial neural networks, k-nearest- neighbors classifier, decision trees, Bayesian classifier dan *Support Vector Machine* (SVM) algorithm (Muawanah et al., 2023). Di antara teknik klasifikasi ini, algoritma *machine learning* yang akan digunakan untuk mendeteksi serangan code injection adalah *Support Vector Machine* (SVM). Support Vector Machine (SVM) secara luas dikenal sebagai salah satu teknik yang paling sering digunakan untuk mengekstraksi fitur dari dataset (Rivera-Romero et al., 2024). Selain itu, SVM unggul dalam akurasi dibandingkan algoritma lain. Berdasarkan hasil pengujian yang telah dilakukan pada lima algoritma klasifikasi, yaitu: (a) Naive Bayes, (b) Logistic Regression, (c) Gradient Boosting, (d) Support Vector Machine, dan (e) K-Nearest Neighbor, SVM mampu menghasilkan tingkat akurasi tertinggi dibandingkan algoritma lainnya (Triloka et al., 2022). *Support Vector Machine* bekerja dengan memanfaatkan ruang hipotesis yang terdiri dari fungsi linear dua arah dalam ruang fitur berdimensi tinggi (Nurkholis et al., 2022). Dalam kasus klasifikasi multi-kelas, SVM dapat diperluas untuk menangani lebih dari dua kelas dengan menerapkan pendekatan *one-vs-all* dan *one-vs-one* (Sunitha & Raju, 2021). Pendekatan multi-kelas digunakan dalam penelitian ini karena dalam praktik dunia nyata, seorang analis keamanan perlu mengetahui jenis serangan secara spesifik untuk dapat mengambil tindakan yang tepat dan cepat. Model klasifikasi biner yang hanya membedakan antara *serangan* dan *bukan serangan* kurang memberikan informasi yang dapat langsung ditindaklanjuti. Sebaliknya, model multi-kelas memungkinkan sistem secara langsung mengidentifikasi tipe serangan, seperti *SQL Injection* atau *Cross-Site Scripting (XSS)*, sehingga dapat meningkatkan ketepatan respons keamanan. Penerapan klasifikasi multi-kelas juga memberikan sejumlah manfaat penting, antara lain peningkatan akurasi deteksi, efisiensi dalam pengambilan keputusan, serta memungkinkan integrasi dengan sistem keamanan yang lebih canggih—semua ini

menjadi kunci dalam membangun sistem pertahanan yang adaptif dan responsif terhadap berbagai jenis ancaman siber (Razali et al., 2025).

Penelitian ini memilih untuk fokus pada dua jenis serangan code injection, yaitu *SQL Injection* dan *Cross-Site Scripting* (XSS), dengan beberapa pertimbangan yang mendukung keputusan ini. *SQL Injectio*n maupun XSS memiliki pola serangan yang mirip, yaitu sama-sama melibatkan penyisipan kode berbahaya ke dalam aplikasi web untuk mengeksploitasi celah keamanan. Kemiripan metode serangan ini menjadikan keduanya relevan untuk dipelajari secara bersamaan dalam konteks pengembangan model deteksi berbasis machine learning. Dengan memahami pola umum yang mendasari kedua jenis serangan ini, model yang dikembangkan dapat lebih fokus pada deteksi karakteristik serangan code injection secara keseluruhan.

Selain itu, penelitian ini memilih dua jenis serangan karena cakupan yang lebih spesifik ini memungkinkan analisis yang lebih mendalam tanpa kehilangan fokus. Jika hanya meneliti satu jenis serangan, hasil penelitian mungkin terlalu sempit untuk memberikan gambaran menyeluruh tentang tantangan deteksi code injection. Di sisi lain, meneliti lebih dari dua jenis serangan, seperti menambahkan Command Injection atau serangan lainnya, berpotensi memperluas lingkup penelitian hingga melebihi batas waktu dan sumber daya yang tersedia, seperti jumlah data yang harus dianalisis, kompleksitas algoritma yang harus dikembangkan, serta waktu yang diperlukan untuk menguji dan mengevaluasi model deteksi. Oleh karena itu, penelitian ini berupaya mencapai keseimbangan antara kedalaman analisis dan cakupan yang relevan, yaitu dengan memilih jenis serangan yang cukup representatif untuk menunjukkan tantangan deteksi code injection, namun tidak terlalu luas sehingga mengurangi fokus dan efisiensi dalam pengolahan data serta pengembangan model.

1.2. Rumusan Masalah

Serangan code injection, terutama SQL Injection dan Cross-Site Scripting (XSS), merupakan ancaman signifikan terhadap keamanan aplikasi web. Serangan ini dapat menyebabkan kebocoran data sensitif, kerugian finansial, dan berbagai masalah lainnya bagi pengguna dan organisasi. Pendeteksian dini terhadap

serangan-serangan ini sangat penting untuk memitigasi dampak yang ditimbulkan. Namun, metode deteksi manual yang masih banyak digunakan saat ini tidak efisien karena jumlah data yang besar, yang membuat pemeriksaan menyeluruh setiap paket jaringan sulit dilakukan. Secara mendalam, penelitian ini menjawab pertanyaan :

1. Bagaimana membangun model klasifikasi multi-class berbasis SVM untuk mendeteksi dan membedakan serangan *SQL Injection*, XSS, dan trafik normal yang akurat?
2. Bagaimana hasil pengujian model SVM dibandingkan algoritma lain seperti *Random Forest*, *Logistic Regression* dan *K-Nearest Neighbors* dalam mengklasifikasikan jenis serangan code injection seperti *SQL Injection* dan XSS?
3. Bagaimana tingkat efektivitas model *Support Vector Machine* (SVM) ketika diintegrasikan ke dalam sistem deteksi *real-time* untuk mengidentifikasi serangan *SQL Injection* dan XSS?

1.3. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah diuraikan, tujuan dari penelitian ini adalah sebagai berikut:

1. Membangun dan mengoptimalkan sebuah model klasifikasi multi-kelas berbasis *Support Vector Machine* (SVM) yang mampu mendeteksi serta membedakan antara serangan *SQL Injection*, *XSS*, dan trafik *Normal* dengan akurasi tinggi.

2. Membandingkan performa model *Support Vector Machine* (SVM) dengan algoritma pembanding lainnya, yaitu *Random Forest*, *Logistic Regression*, dan *K-Nearest Neighbors*, untuk menentukan model yang paling efektif dalam mengklasifikasikan serangan.

3. Mengevaluasi tingkat efektivitas model *Support Vector Machine* (SVM) yang telah dioptimalkan ketika diimplementasikan ke dalam sebuah sistem deteksi *real-time* pada lingkungan simulasi.

1.4. Batasan dan Asumsi Penelitian

Dalam penelitian ini, terdapat beberapa batasan yang ditetapkan untuk menjaga fokus penelitian dan memastikan hasil yang diharapkan dapat dicapai secara efektif. Adapun batasan-batasan tersebut adalah sebagai berikut:

1. Hanya berfokus pada deteksi serangan SQL Injection dan Cross-Site Scripting, serta trafik normal.
2. Dataset diperoleh melalui simulasi serangan menggunakan aplikasi web OWASP Juice Shop.
3. Penelitian dilakukan dalam lingkup simulasi dan tidak diuji pada aplikasi web di dunia nyata.
4. Simulasi serangan dilakukan dalam lingkungan jaringan lokal, di mana *host attacker* dan *host victim* berada dalam satu jaringan yang sama.

1.5. Manfaat Penelitian

Berdasarkan tujuan yang ingin dicapai, penelitian ini diharapkan memberikan manfaat, baik secara langsung maupun tidak langsung, dalam bidang pendidikan. Manfaat dari penelitian ini antara lain :

1. Manfaat Akademis
    - Menambah literatur dalam bidang keamanan siber, khususnya terkait penggunaan machine learning untuk mendeteksi serangan SQL Injection (SQLi) dan Cross-Site Scripting (XSS).
    - Memberikan referensi baru terkait implementasi algoritma Support Vector Machine (SVM) dalam klasifikasi multi-kelas pada keamanan web.

2. Manfaat Praktis
    - Menyediakan model deteksi serangan otomatis yang dapat digunakan sebagai bagian dari sistem pertahanan aplikasi web untuk mengurangi risiko eksploitasi oleh penyerang.

- Membantu pengembang aplikasi web dalam mengenali pola-pola serangan SQLi dan XSS, sehingga dapat meningkatkan keamanan aplikasi mereka.

1.6. Sistematika Penulisan

Sistematika penulisan Proposal Tugas Akhir ini disusun untuk memudahkan pembaca dalam memahami alur pemikiran dan hasil penelitian yang dilakukan. Berikut adalah uraian masing-masing bab yang terdapat dalam Proposal Tugas Akhir ini :

a. BAB I PENDAHULUAN

Bab ini berisi latar belakang penelitian yang menjelaskan pentingnya deteksi dini serangan SQL Injection dan Cross-Site Scripting (XSS) dalam keamanan web. Selain itu, dirumuskan juga permasalahan penelitian, tujuan penelitian, batasan penelitian, manfaat penelitian serta sistematika penulisan. Pada bagian ini, disajikan juga alasan penggunaan algoritma Support Vector Machine (SVM) dalam mendeteksi serangan code injection.

b. BAB II LANDASAN TEORI

Bab ini memuat teori-teori pendukung yang relevan dengan penelitian. Pembahasan meliputi penjelasan tentang serangan SQL Injection dan XSS , penjelasan konsep Machine Learning dengan fokus utama pada algoritma Support Vector Machine (SVM), termasuk cara kerja dan metode multi-class classification seperti One-vs-All dan One-vs-One, penjelasan tentang K-fold cross validation dan Confusion Matrix. Selain itu terdapat penelitian terdahulu yang mendukung penggunaan SVM untuk deteksi serangan.

c. BAB III METOLOGI PENELITIAN

Bab ini berisi sistematika penyelesaian masalah yang menjelaskan menjelaskan langkah-langkah yang akan dilakukan dalam proses penelitian, mulai dari pengumpulan data, preprocessing, pengujian model sampai pengujian sistem.

d. BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Bab ini menguraikan secara rinci proses pelaksanaan pengumpulan data primer melalui simulasi serangan pada aplikasi OWASP Juice Shop. Bagian ini juga merinci tahapan pengolahan data yang mencakup *pre-processing*, transformasi fitur menggunakan Word2Vec dan metode lainnya, hingga proses pelatihan model *machine learning* beserta hasil optimasi *hyperparameter* menggunakan GridSearchCV.

e. BAB V ANALISIS DAN PEMBAHASAN

Bab ini berfokus pada analisis dan pembahasan hasil penelitian. Bagian ini menyajikan evaluasi mendalam terhadap performa model SVM menggunakan *confusion matrix* dan metrik lainnya. Bab ini juga memuat analisis komparatif dengan model pembanding seperti *Random Forest*, *Logistic Regression*, dan *K-Nearest Neighbors*. Selain itu, dilaporkan dan dibahas juga hasil pengujian sistem deteksi secara *real-time*.

f. BAB VI KESIMPULAN DAN SARAN

Bab ini merupakan bab penutup yang berisi rangkuman kesimpulan dari keseluruhan hasil penelitian dan analisis yang telah dilakukan