BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Pada tahun 2021, Badan Siber dan Sandi Negara (BSSN) menerima 332 aduan serangan siber yang didominasi pada kasus SQL Injection sebanyak 79 aduan melebihi serangan lainnya diikuti serangan ransomware, *Cross-Site Scripting* (XSS), konten negatif, information disclosure, dan phising[1],[2]. Selain itu, tercatat ada beberapa serangan lainnya hingga Juni 2024 ini yang diawali dengan bocornya data unik calon pemilih pada pemilu 2024, hingga mencapai 204 juta data yang diperjualbelikan di situs BreachForums dengan 500 ribu sampel dari situs Komisi Pemilihan Umum (KPU)[3] dan puncaknya pada akhir Juni 2024 diduga data milik Kementerian Komunikasi dan Informatika telah bocor dan dibandrol dengan harga sekitar Rp. 1,9 miliar yang berisikan data pribadi meliputi NIK dan detail rekening perbankan yang hingga saat ini, belum ada langkah tegas dari Kominfo terkait permasalahan ini[4]. Maraknya kasus pembobolan data penduduk Indonesia, validasi input yang ketat, Salah satu penelitian yang dilakukan oleh Xuan et al. (2015) menggunakan algoritma Random Forest untuk mendeteksi serangan SQL Injection dengan tingkat akurasi mencapai 98.6%. Mereka menggunakan dataset yang terdiri dari 40.000 *query* SQL, di mana 20.000 di antaranya merupakan *query* normal dan 20.000 lainnya mengandung serangan SQL Injection. Hasil penelitian mereka menunjukkan bahwa model Random Forest mampu mengenali pola-pola yang mencurigakan dalam *query* SQL dengan sangat baik.

Dari permasalahan SQL Injection dan penelitian terkait, peneliti menggunakan model Random Forest yang didesain untuk mengklasifikasi *query* SQL dari dataset yang digunakan yang berasal dari Kaggle dengan pendekatan *Bag of Words* dan encode unigram pada *preprocessing* dengan encode unigram beserta *Feature extraction* menggunakan *Count vectorizer* lalu, performa dari beberapa metriks akan dievaluasi menggunakan Confussion Matrix sekaligus memodifikasi

beberapa *hyperparameter* (*hyperparameter tuning*) seperti jumlah pohon keputusan (n_estimators), kedalaman maksimum pohon (max_depth), dan jumlah fitur yang dipertimbangkan saat mencari split terbaik (max_features) disesuaikan dengan dataset yang digunakan dalam uji coba kemudian, masing-masing percobaan akan dibandingkan dan diakhir percobaan akan dilakukan evaluasi lanjutan menggunakan metode *Cross validation* untuk meminimalisir terjadinya *overfitting* pada model yang digunakan. Beberapa pendekatan tersebut dikombinasi untuk mengembangkan model Random Forest agar dapat memperoleh informasi yang relevan dari data teks dan mengklasifikasikan *query* SQL dengan akurasi yang lebih tinggi serta optimal sehingga, beberapa *query* yang mencurigakan dapat terdeteksi dengan lebih cepat sebagai alternatif pencegahan dini.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah di atas, dapat dirumuskan sebuah permasalahan tentang:

- Bagaimana cara pembuatan model machine learning Random Forest untuk mendeteksi serangan SQL Injection dengan melakukan *hyperparameter tuning*, menggunakan metode *Bag of Words*, dan preprocessing dengan encode unigram beserta *Feature extraction* menggunakan *Count vectorizer* dan evaluasi dengan confusion matrix beserta *Cross validation*?
- Apakah kombinasi pendekatan ini dapat meningkatkan tingkat deteksi serangan SQL Injection secara signifikan?
- Bagaimana pengaruh dari penggunaan *hyperparameter tuning* pada Random Forest terhadap kinerja deteksi serangan ini?

1.3 Batasan Masalah

Untuk mempersempit pembahasan pada skripsi ini, maka dibuat batasan-batasan sebagai berikut:

a. Sistem deteksi dirancang menggunakan bahasa pemrograman Python (ipynb) di platform Google Colaboratory.

b. Dataset bersumber dari Kaggle.com yang terdiri dari 33761 data berbentuk *query* acak.
c. Penelitian mencakup analisis data (EDA), perancangan model, dan visualisasi performa model, termasuk implementasi Random Forest dengan *hyperparameter* dari masing-masing percobaan model sebagai pembanding.
d. Beberapa model tersebut dilakukan pendekatan menggunakan *hyperparameter tuning* dan *feature extraction* dengan *Bag of Words* serta, *function Count vectorizer*.
e. Dari beberapa uji coba model akan diambil hasil *accuracy*, *precision*, f1-score, dan *recall* untuk dilakukan perbandingan model terbaik dilihat sebatas dari seberapa tinggi *accuracy* yang dihasilkan.
f. Visualisasi hasil menggunakan confusion matrix.
g. Penelitian ini dilakukan hanya sampai tahap pembuatan model machine learning, adapun pengujian beberapa *hyperparameter* yang ditentukan pada model Random Forest untuk meningkatkan performanya hanya untuk pertimbangan penelitian.

1.4 Tujuan Penelitian

Tujuan yang ingin diraih dalam pembuatan skripsi ini adalah membangun sebuah sistem yang lebih baik dengan pemanfaatan model machine learning Random Forest sebagai deteksi dini serangan SQL Injection dan meningkatkan matriks performa yang berlaku pada Model Random Forest.

1.5 Sistematika Penulisan

Bab I Pendahuluan, berisi: latar belakang, rumusan masalah dan hipotesis, batasan masalah, tujuan penelitian, dan sistematika penulisan.

Bab II Landasan Teori, berisi: hasil penelitian sejenis yang sudah pernah dilakukan sebelumnya, teori penunjang, dan referensi berupa buku, jurnal, dan laporan skripsi/tesis.

Bab III Metodologi Penelitian, berisi: penjelasan mengenai metode penelitian yang digunakan untuk memahami dan mengekplorasi obyek penelitian, hasil observasi / pengumpulan data, masalah yang terdapat pada obyek, dan gambaran umum proyek atau obyek penelitian, hingga Rencana Alur Penelitian.

Bab IV Pembahasan, berisi: rancangan proyek, implementasi *coding* dan desain, serta evaluasi rancangan. Selanjutnya alur pengerjaan proyek, metode testing, hingga hasil akhir penelitian dan pembahasan analisis hasil akhir penelitian, termasuk pembahasan hasil-hasil uji coba (testing).

Data hasil akhir pengujian dapat berupa grafik, tabel, data analisis, dan lain-lain, dengan pembahasan.

Bab V Penutup, berisi kesimpulan dari hasil akhir penilaian proyek, dan saran.