BAB 5. PENUTUP

5.1. Kesimpulan

Berdasarkan hasil perancangan, implementasi, pengujian, dan analisis yang telah dilakukan pada sistem deteksi intrusi jaringan berbasis log Cowrie SSH menggunakan algoritma *Long Short-Term Memory*, diperoleh kesimpulan sebagai berikut:

1. Penelitian ini berhasil mengimplementasikan model *Network Intrusion Detection System* berbasis *Long Short-Term Memory* untuk mengklasifikasikan aktivitas pada log honeypot Cowrie SSH. Data log diproses melalui tahapan seleksi data Cowrie SSH, pembentukan *event sequence* berdasarkan *session*, pembentukan label *multiclass*, penghapusan fitur yang berpotensi menyebabkan *data leakage* dan *proxy label*, tokenisasi, *padding*, serta pembentukan input model. Kelas yang digunakan dalam penelitian ini terdiri dari *normal*, *scan_probe*, dan *brute_force*.
2. Meskipun hasil evaluasi menunjukkan nilai *accuracy* sebesar 99,30%, perlu ditegaskan bahwa label kelas *normal*, *scan_probe*, dan *brute_force* pada penelitian ini dibentuk melalui aturan heuristik berdasarkan pola *event* dalam *session*, bukan melalui proses verifikasi oleh pakar keamanan jaringan secara independen. Oleh karena itu, nilai *accuracy* yang tinggi tersebut sebaiknya dimaknai sebagai kemampuan model dalam mempelajari pola *sequence* sesuai aturan pelabelan yang digunakan, bukan sebagai representasi mutlak kemampuan model dalam mendeteksi serangan pada kondisi jaringan nyata yang jauh lebih beragam. Klaim performa model perlu disampaikan secara proporsional agar tidak menimbulkan kesan bahwa sistem telah siap diterapkan pada lingkungan produksi tanpa pengujian dan validasi lebih lanjut.
3. Model LSTM yang telah dikembangkan berhasil diintegrasikan ke dalam prototipe *dashboard* NIDS berbasis Flask, MongoDB, dan Docker. Hasil pengujian sistem menunjukkan bahwa *dashboard* mampu membaca log Cowrie SSH, membentuk token *session*, melakukan prediksi, menyimpan hasil, dan menampilkan klasifikasi aktivitas *normal*, *scan_probe*, dan *brute_force*. Sistem yang dibangun masih bersifat *detection-only*, sehingga sistem hanya melakukan deteksi dan pencatatan tanpa pemblokiran otomatis terhadap sumber serangan.

5.2. Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat digunakan untuk pengembangan penelitian selanjutnya, yaitu sebagai berikut:

1. Penelitian selanjutnya dapat menggunakan data log Cowrie SSH dengan jumlah dan variasi aktivitas yang lebih luas, seperti variasi alamat IP sumber, *username*, *password*, jumlah percobaan *login*, *command*, dan durasi *session*. Penggunaan variasi data yang lebih luas bertujuan agar model dapat mempelajari pola aktivitas yang lebih beragam.
2. Pengujian sistem dapat dilakukan pada lingkungan yang lebih mendekati kondisi nyata, misalnya menggunakan server atau VPS yang dapat menerima koneksi dari luar jaringan lokal. Pengujian pada lingkungan tersebut dapat membantu mengevaluasi kemampuan sistem dalam menangani aktivitas serangan yang lebih bervariasi.
3. *Dashboard* dapat dikembangkan dengan menambahkan fitur pendukung, seperti filter data, grafik tren serangan, ekspor laporan, sistem notifikasi, serta pengelolaan pengguna. Penambahan fitur tersebut dapat membantu proses pemantauan dan analisis hasil deteksi agar menjadi lebih informatif.
4. Penelitian selanjutnya dapat memperluas ruang lingkup sistem dengan membandingkan model LSTM terhadap algoritma lain, menambahkan layanan honeypot selain Cowrie SSH, atau mengembangkan mekanisme *active response* secara terbatas. Pengembangan tersebut perlu dilakukan dengan tetap memperhatikan risiko *false positive* agar tidak mengganggu aktivitas pengguna yang sah.
5. Penelitian selanjutnya disarankan untuk melakukan validasi label secara independen, misalnya dengan melibatkan pakar keamanan jaringan untuk memeriksa ulang sebagian sampel data secara manual, atau dengan membandingkan hasil pelabelan berdasarkan aturan heuristik terhadap basis pengetahuan ancaman (*threat intelligence*) yang telah teruji. Validasi independen ini penting dilakukan untuk memastikan bahwa label yang digunakan benar-benar merepresentasikan karakteristik serangan yang sesungguhnya, sehingga hasil evaluasi model dapat lebih dipertanggungjawabkan dan tidak menimbulkan kesan klaim berlebihan terhadap kemampuan sistem yang dikembangkan.
6. Penelitian selanjutnya disarankan untuk membandingkan performa model secara langsung antara dataset sebelum dan sesudah penyeimbangan (*imbalanced dataset*), sehingga besarnya kontribusi proses penyeimbangan data terhadap peningkatan performa model dapat terukur secara eksplisit. Selain itu, pengujian *Quality of Service* (QoS) juga disarankan untuk dilakukan, seperti pengukuran waktu tanggap (*latency*) prediksi, throughput sistem dalam menangani banyak log secara bersamaan, serta konsumsi sumber daya komputasi pada kondisi beban trafik yang tinggi. Pengujian QoS penting dilakukan mengingat evaluasi performa klasifikasi yang baik perlu diimbangi dengan kesiapan operasional sistem apabila hendak diterapkan pada lingkungan produksi yang menuntut waktu deteksi yang cepat dan stabil.