BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil implementasi, pengujian, dan analisis yang telah dilakukan pada sistem keamanan IoT menggunakan kriptografi AES-128 dan *Intrusion Detection System* (IDS) berbasis LSTM, dapat ditarik beberapa kesimpulan sebagai jawaban atas rumusan masalah penelitian:

1. Implementasi AES-128

   Algoritma enkripsi AES-128 berhasil diimplementasikan pada simulasi perangkat IoT untuk mengamankan transmisi data. Pengujian fungsionalitas secara terisolasi pada lingkungan arsitektur 32-bit membuktikan bahwa mekanisme ini secara efektif menjamin aspek kerahasiaan (*confidentiality*) dengan mengubah data sensor menjadi *ciphertext* heksadesimal. Proses enkripsi berjalan sangat efisien dengan latensi yang sangat rendah (sekitar 1 ms pada lingkungan terbatas, dan 0,13–0,22 ms pada lingkungan komputasi 64-bit), sehingga terbukti tidak menjadi faktor penghambat (*bottleneck*) pada aliran data perangkat *edge*.

2. Pembangunan Model LSTM

   Model *Deep Learning* berbasis LSTM berhasil dibangun untuk mendeteksi anomali jaringan IoT menggunakan arsitektur satu *layer* LSTM (64 unit), *Dropout* (0,2), dan *Dense layer* (aktivasi Sigmoid). Penggunaan *dataset* CICIoT2023 yang dikombinasikan dengan teknik SMOTE yang diterapkan *secara eksklusif* hanya pada data latih terbukti berhasil mengatasi ketidakseimbangan kelas tanpa memicu kebocoran data (*data leakage*). Model mampu mengenali pola serangan siber dengan sangat baik sekaligus mempertahankan kepekaan terhadap trafik normal, dibuktikan dengan nilai akurasi keseluruhan mencapai 99%, nilai *Recall* kelas Normal sebesar 0,99, dan *F1-Score* kelas Normal sebesar 0,83 pada pengujian menggunakan 440.000 data uji murni.

3. Akurasi dan Optimasi TFLite

   Proses optimasi model menggunakan TensorFlow Lite dengan teknik *dynamic range quantization* terbukti mampu mengompresi model tanpa mengorbankan akurasi prediksi. Model teroptimasi ini mempertahankan performa luar biasa dengan tingkat *False Alarm* yang sangat rendah (hanya 64 dari 10.316 data Normal yang salah diblokir). Pengujian latensi komparatif komprehensif (5 kali pengujian dengan 1.000 sampel acak) menunjukkan bahwa model TFLite mencatat rata-rata waktu inferensi 0,848 ms, jauh melampaui model standar (.h5) yang membutuhkan 113,85 ms. Hal ini menghasilkan faktor percepatan (*speedup*) sebesar 134,3 kali lipat, menjadikan model sangat siap dan layak ditanamkan pada perangkat dengan komputasi terbatas.

4. Dampak Integrasi terhadap Latensi

   Simulasi integrasi sistem keamanan berlapis (*Defense in Depth*) secara *end-to-end* yang menggabungkan proses enkripsi AES-128 dan inferensi deteksi anomali (IDS TFLite) di sisi pengirim (*edge processing*) menunjukkan kinerja yang sangat memuaskan. Jika diakumulasikan, total latensi komputasi keamanan yang ditambahkan sebelum paket dikirimkan hanya berada di kisaran ~1,0 ms. Angka ini jauh di bawah ambang batas toleransi aplikasi IoT *real-time* (umumnya < 100 ms). Hal ini mengonfirmasi bahwa integrasi keamanan data dan kecerdasan buatan dapat berjalan beriringan dengan performa maksimal.

Secara keseluruhan, keempat rumusan masalah telah terjawab. Enkripsi AES-128 dapat diimplementasikan secara mulus, model LSTM mampu mendeteksi anomali dengan akurasi tinggi setelah dioptimasi dengan TFLite, dan kombinasi keduanya memberikan perlindungan jaringan berlapis dengan penambahan latensi yang sangat minimal.

5.2 Saran

Berdasarkan temuan dan batasan yang diidentifikasi selama penelitian, terdapat beberapa rekomendasi krusial untuk pengembangan sistem ke depannya. Pertama, dari sisi arsitektur *Deep Learning*, penelitian ini menggunakan timesteps = 1 sehingga model secara komputasi beroperasi layaknya *Dense Neural Network*. Penelitian selanjutnya sangat disarankan untuk menggunakan teknik *sliding window* dengan timesteps ≥ 10 agar model LSTM dapat memanfaatkan memori sekuensialnya secara optimal dalam mendeteksi pola serangan berkelanjutan yang kompleks, seperti serangan DDoS multi-vektor. Selain itu, eksplorasi optimasi tingkat lanjut seperti *full integer quantization* atau pemangkasan bobot jaringan (*pruning*) pada TensorFlow Lite dapat dilakukan untuk menargetkan implementasi *Machine Learning* pada perangkat mikrokontroler yang jauh lebih kecil (*TinyML*), seperti Arduino atau ESP32.

Kedua, dari sisi implementasi sistem, pengujian komparasi arsitektur dalam penelitian ini masih menggunakan simulasi pada Virtual Machine (32-bit) dan *Instance Cloud* (64-bit). Oleh karena itu, langkah paling krusial berikutnya adalah melakukan implementasi langsung pada wujud fisik perangkat keras 64-bit secara utuh, seperti pada perangkat Raspberry Pi 4 atau Pi 5. Pengujian pada perangkat keras asli ini bertujuan untuk memvalidasi integrasi model secara *real-time* sekaligus mengukur metrik lingkungan fisik secara nyata, seperti fluktuasi suhu prosesor perangkat, konsumsi daya energi, dan stabilitas *runtime* saat sistem keamanan beroperasi penuh.

Ketiga, dari aspek perlindungan kriptografi, standar keamanan data dapat ditingkatkan lebih jauh dengan mengimplementasikan algoritma pertukaran kunci dinamis, seperti *Elliptic-Curve Diffie-Hellman* (ECDH). Lebih lanjut, penggunaan mode operasi enkripsi yang lebih mutakhir seperti AES-GCM (*Galois/Counter Mode*) juga sangat direkomendasikan. Mode ini tidak hanya menjamin kerahasiaan transmisi, tetapi secara *default* sudah mencakup autentikasi data bawaan untuk mencegah manipulasi atau alterasi *ciphertext* oleh pihak ketiga selama paket berada di dalam jaringan.