BAB V

PENUTUP

5.1 KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan mengenai deteksi serangan *malware* jenis *Trojan Horse* menggunakan metode *Convolutional Neural Network (CNN)*, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Sistem deteksi berbasis CNN mampu mengidentifikasi *malware* jenis *Trojan Horse* dengan akurasi yang cukup tinggi. Model CNN yang diterapkan mampu mengekstraksi fitur dari data secara otomatis tanpa perlu proses ekstraksi manual, menjadikan sistem lebih efisien dalam klasifikasi *malware*.
2. Penggunaan *EarlyStopping* dalam proses pelatihan meningkatkan efisiensi dan mencegah *overfitting*. *EarlyStopping* secara adaptif menghentikan pelatihan saat model tidak mengalami peningkatan performa pada data validasi, yang berdampak positif pada generalisasi model.
3. Pembagian data latih dan uji (split 80:20 dan 70:30) memengaruhi hasil evaluasi. Percobaan menunjukkan bahwa proporsi data yang lebih besar untuk pelatihan dapat memberikan akurasi yang sedikit lebih tinggi, namun perbedaan tidak signifikan ketika model dilatih dengan optimal dan menggunakan validasi yang baik.

4. *Dataset* yang mengandung beragam varian *ransomware* (seperti Ako, Shade, Pysa, Maze, dan Conti) memberikan tantangan tersendiri dalam membedakan Trojan dari jenis *malware* lain. Namun, dengan preprocessing dan labeling yang tepat, sistem mampu mendeteksi *Trojan Horse* dengan presisi yang baik.

5.2 SARAN

Untuk pengembangan lebih lanjut, berikut adalah beberapa saran yang dapat dipertimbangkan:

1. Perluasan *dataset* dan diversifikasi jenis *Trojan Horse*. Semakin banyak varian dan sampel *malware* yang digunakan, semakin baik model dalam mengenali pola serangan yang beragam.
2. Eksplorasi metode lain selain CNN. Metode seperti LSTM, *Transformer*, atau kombinasi CNN-RNN dapat diuji untuk melihat perbandingan performa terhadap jenis data berbeda (terutama log atau urutan API *call*).
3. Penggunaan data dinamis. Selain data statis seperti metadata *file* atau *bytecode*, penggunaan data dinamis (seperti aktivitas jaringan atau API *behavior*) dapat meningkatkan akurasi pendeteksian *Trojan Horse*.
4. Integrasi ke dalam sistem keamanan nyata. Sistem deteksi ini sebaiknya diuji pada lingkungan nyata seperti *endpoint protection system* atau *gateway firewall* untuk mengetahui performa secara *real-time*.
5. Evaluasi dengan metrik lebih luas. Selain akurasi, metrik seperti *precision*, *recall*, F1-*score*, dan *confusion matrix* perlu dikaji lebih mendalam untuk memastikan bahwa sistem tidak hanya akurat, tapi juga andal dalam kondisi sebenarnya.