BAB VI
PENUTUP

Kesimpulan dan hasil yang telah diselesaikan dalam penelitian ini, dengan referensi serta teori yang telah disampaikan pada bab-bab sebelumnya akan disampaikan dalam bab ini. Saran serta rekomendasi pada penelitian ini juga akan dipaparkan untuk pengembangan kedepannya.

6.1 Kesimpulan

Dengan menggabungkan teknologi Security Information and Event Management (SIEM) berbasis Wazuh, model klasifikasi URL berbasis BERT, dan Squid Proxy Server untuk mekanisme pemblokiran, penelitian ini berhasil merancang dan menerapkan sistem deteksi dan pencegahan akses terhadap URL berbahaya. Sebagai hasil dari pengujian yang dilakukan, dapat disimpulkan bahwa:

1. Dengan menggunakan dataset URL publik, model klasifikasi URL berbasis BERT berhasil membedakan URL benign dan *malicious* dengan akurasi tinggi sebesar 98,65%, dengan nilai precision 97,72%, recall 98,42%, dan F1-score 98,07%. Ini menunjukkan bahwa model memiliki kemampuan untuk melakukan deteksi URL berbahaya secara konsisten dan andal.
2. Flask API berfungsi dengan baik sebagai antarmuka antara model dan Squid Proxy Server. Ketika seseorang mencoba mengakses URL tertentu, API akan mengirimkan URL tersebut ke model, dan hasil klasifikasi ("*malicious*" atau "benign") menentukan keputusan pemblokiran.
3. Pengguna akan diarahkan ke halaman blokir khusus yang ditampilkan melalui web server Flask dengan URL berbahaya, sementara dengan URL aman, akses akan diteruskan sebagaimana mestinya.
4. Semua modul sistem, termasuk tokenisasi URL, klasifikasi, filtering proxy, pengiriman log, dan visualisasi dashboard, bekerja sesuai harapan dengan pengujian white box dan black box.

6.2 Saran

Beberapa rekomendasi untuk pengembangan dan perbaikan sistem di masa depan meliputi:

1. Penanganan HTTPS Secara Menyeluruh: Squid Proxy saat ini tidak dapat melakukan inspeksi HTTPS secara transparan. Untuk menginspeksi konten HTTPS lebih dalam, dapat menggunakan SSL Bump atau proxy chaining dengan layer filtering lainnya.
2. Model Tambahan: Model BERT dapat disempurnakan untuk meningkatkan kinerja klasifikasi dengan menggunakan teknik fine-tuning pada kumpulan data yang lebih besar.
3. Pengujian Lebih Luas di Jaringan Produksi: Disarankan untuk menguji sistem pada skala jaringan yang lebih besar dan kompleks (dengan banyak klien) untuk memastikan skalabilitas dan performa sistem dalam kondisi nyata.
4. Auto-Update Threat Intelligence: Implementasikan sistem untuk memperbarui model atau blacklist URL secara otomatis dari sumber threat intelligence seperti PhishTank, Google Safe Browsing, atau URLHaus.