BAB V
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil implementasi, pengujian, dan analisis yang telah dilakukan pada penelitian ini, dapat ditarik kesimpulan sebagai berikut:

1. Sistem deteksi intrusi berbasis framework T-Pot yang terintegrasi dengan SIEM dan notifikasi Telegram berhasil dibangun dan diverifikasi. Sistem ini mengintegrasikan Cowrie, Dionaea, dan Suricata sebagai komponen deteksi, Filebeat dan Logstash sebagai pipeline pengumpulan serta pemrosesan log, Elasticsearch sebagai media penyimpanan dan indexing, Kibana sebagai dashboard monitoring, ElastAlert sebagai engine notifikasi berbasis rule, serta Telegram sebagai media penerimaan alert real-time. Selama pengujian serangan organik selama 16 hari efektif, sistem berhasil menangkap 6.332.283 event dari 28.224 alamat IP unik yang tersebar di 163 negara, serta mengirimkan 14.229 notifikasi ke Telegram secara otomatis. Integrasi T-Pot dengan SIEM juga terbukti meningkatkan kualitas informasi keamanan melalui proses korelasi dan enrichment event, seperti penambahan konteks GeoIP, klasifikasi alert Suricata berbasis ruleset Emerging Threats, serta enrichment hash malware melalui script malware_vt_checker.py. Dengan demikian, event mentah dari honeypot dapat diolah menjadi informasi keamanan yang lebih bermakna bagi administrator.
2. Pengukuran kecepatan notifikasi end-to-end (MTTD) pada pengujian terkontrol menghasilkan nilai mean gabungan sebesar 24,902 detik dari 150 iterasi. Empat dari lima skenario memenuhi target MTTD ≤ 30 detik, yaitu Malware Upload Tanpa VirusTotal (15,067 detik), Brute Force (18,786 detik), Port Scanning (22,769 detik), dan DDoS / SYN Flood (28,703 detik). Skenario Malware Upload dengan VirusTotal menghasilkan MTTD 39,185 detik akibat *overhead enrichment* API eksternal, namun secara komparatif terbukti bahwa pipeline inti sistem tanpa *enrichment* eksternal mampu beroperasi jauh di bawah target. Analisis bottleneck menunjukkan bahwa segmen T2→T3 (ElastAlert polling dan Telegram API) mendominasi latency pada empat skenario pertama dengan kontribusi 92–98%, sedangkan pada skenario Malware dengan VirusTotal, bottleneck bergeser ke segmen T1→T2 akibat delay *enrichment* selama rata-rata

24,185 detik. Perbandingan antara kedua skenario malware mengungkapkan *trade-off* antara kecepatan deteksi dan kedalaman analisis ancaman — Skenario 5 (tanpa VirusTotal) unggul dalam kecepatan dengan MTTD 15,067 detik namun hanya menghasilkan informasi dasar berupa hash MD5, sedangkan Skenario 4 (dengan VirusTotal) meskipun lebih lambat menghasilkan informasi *enrichment* yang jauh lebih kaya mencakup klasifikasi dari 70+ antivirus engine, identifikasi CVE terkait (cve-2017-0147/EternalBlue), dan reputasi file dari komunitas VirusTotal.

3. Sistem menghasilkan konsistensi notifikasi yang sempurna dengan success rate 100% dari seluruh 150 iterasi pengujian terkontrol. Tidak terdapat satu pun event serangan yang gagal menghasilkan notifikasi Telegram dari kelima skenario yang diuji, membuktikan bahwa pipeline deteksi berjalan reliabel tanpa kehilangan data. Pencapaian *zero event loss* pada skenario yang melibatkan dependensi API eksternal mengonfirmasi bahwa mekanisme rate limiting yang diterapkan pada script `malware_vt_checker.py` berhasil menjaga stabilitas sistem.

5.2. Saran

Berdasarkan temuan dan keterbatasan yang diidentifikasi selama penelitian, beberapa saran diajukan untuk pengembangan dan implementasi sistem serupa di masa mendatang

1. Optimasi konfigurasi ElastAlert perlu dilakukan untuk mengurangi latency pada proses pengiriman notifikasi. Berdasarkan hasil analisis, segmen T2–T3 menjadi salah satu kontributor terbesar terhadap total *end-to-end latency* pada beberapa skenario pengujian. Oleh karena itu, penelitian selanjutnya dapat menguji variasi konfigurasi seperti interval polling, buffer time, realert, dan mekanisme throttling untuk mengetahui pengaruhnya terhadap kecepatan notifikasi dan beban query ke Elasticsearch.
2. Proses enrichment malware menggunakan VirusTotal perlu dioptimalkan karena menjadi salah satu penyebab meningkatnya latency pada skenario malware upload dengan enrichment. Penelitian selanjutnya dapat menerapkan mekanisme caching hash lokal, penyimpanan hasil query sebelumnya, atau penggunaan threat intelligence lokal untuk mengurangi ketergantungan terhadap API eksternal. Dengan demikian, sistem tetap dapat memperoleh informasi tambahan

terkait malware tanpa menghasilkan delay yang terlalu besar pada proses notifikasi.

3. Mekanisme pencatatan timestamp T2 dan T3 sebaiknya dibuat otomatis untuk meningkatkan akurasi dan konsistensi pengukuran latency. Pada penelitian ini, beberapa proses pencatatan masih bergantung pada ekstraksi log secara manual dari beberapa sumber. Penelitian selanjutnya dapat mengembangkan script otomatis untuk mengambil timestamp dari seluruh komponen pipeline, mulai dari waktu serangan dijalankan, waktu log terindeks di Elasticsearch, hingga waktu notifikasi diterima di Telegram. Otomatisasi ini dapat mengurangi risiko kesalahan pencatatan manual dan mempermudah replikasi pengujian.
4. Periode dan cakupan pengujian serangan organik dapat diperluas agar data yang diperoleh lebih representatif. Penelitian ini menggunakan satu server honeypot dengan periode pengamatan tertentu, sehingga pola serangan yang diperoleh sangat bergantung pada eksposur jaringan dan alamat IP yang digunakan. Penelitian selanjutnya dapat memperpanjang durasi observasi atau menggunakan beberapa sensor honeypot pada lokasi jaringan yang berbeda untuk membandingkan variasi pola serangan, distribusi sumber serangan, dan jenis event yang berhasil ditangkap.
5. Skenario pengujian terkontrol dapat diperluas dengan tetap mempertahankan fokus pada evaluasi deteksi dan notifikasi. Penelitian selanjutnya dapat menambahkan variasi serangan yang masih relevan dengan komponen T-Pot, seperti variasi brute force dengan wordlist berbeda, intensitas port scanning yang berbeda, variasi traffic flooding, serta beberapa jenis file uji pada skenario malware upload. Penambahan variasi ini bertujuan untuk menguji stabilitas sistem dalam kondisi yang lebih beragam tanpa mengubah fokus utama penelitian dari deteksi intrusi dan notifikasi *real-time*.