BAB V
PENUTUP

5.1 Kesimpulan

Kesimpulannya, *Traffic Limiter* merupakan komponen krusial dalam strategi remediasi *DDoS* bagi aplikasi *Flask API*. Kemampuannya dalam membatasi *trafik* berlebih, fleksibilitas penyesuaian, dan dampak positif pada kinerja aplikasi menjadikan pendekatan ini solusi yang efektif dan layak implementasi. Setelah tahap implementasi dan pengujian, *execution time* dari proses *API* ada pada angka tertinggi 1000ms dan rata rata pada 420ms. Utilitas cpu sendiri berada pada *load* 80% tertinggi dan berada stabil dengan rata rata 41-43% Namun, keberhasilan dalam menghadapi ancaman *DDoS* yang terus berkembang memerlukan kombinasi *Traffic Limiter* dengan solusi keamanan cerdas lainnya untuk membangun pertahanan yang komprehensif dan adaptif. Penggunaan *traffic limiter* merupakan pendekatan yang efektif dalam mitigasi serangan *DDoS* pada aplikasi berbasis *Flask API*. Implementasi teknik ini dapat meningkatkan keamanan dan ketersediaan layanan dengan cara yang relatif sederhana dan tidak memerlukan sumber daya yang besar. Namun, penelitian ini juga mencatat bahwa teknik ini lebih efektif ketika dikombinasikan dengan strategi keamanan lainnya, seperti deteksi dini serangan dan penggunaan *firewall* yang lebih canggih. Penelitian lanjutan direkomendasikan untuk mengeksplorasi kombinasi teknik mitigasi yang lebih kompleks guna menghadapi serangan *DDoS* yang semakin canggih.

5.2 Saran

1. Integrasi *Machine Learning* untuk Deteksi Proaktif:

   Mengembangkan model prediksi serangan *DDoS* yang menganalisis historis trafik, karakteristik permintaan, dan sinyal ancaman *real-time* untuk memprediksi dan mencegah serangan sebelum terjadi *overload*
2. Analisis *Behavioristic* dan *Adaptif*:

   Menyelidiki teknik behavioural biometrics untuk aplikasi *Flask API*. Pahami pola penggunaan normal pengguna dan aplikasi untuk membangun profil trafik yang lebih akurat, membedakan trafik legit dari ancaman dengan lebih baik
3. Evaluasi terhadap Varian *DDoS* yang Canggih:

Menggunakan metode *DDoS reflection/amplification* yang memanfaatkan protokol seperti *DNS* dan *NTP*. Mengembangkan solusi kolaboratif dengan *DNS* server dan jaringan untuk mengurangi dampak serangan tersebut.

4. Skalabilitas dan Efisiensi dalam Klasterisasi

   Menggunakan metode *DDoS* reflection/amplification yang memanfaatkan protokol seperti *DNS* dan *NTP*. Mengembangkan solusi kolaboratif dengan *DNS* server dan jaringan untuk mengurangi dampak serangan tersebut.

5. Skalabilitas dan Efisiensi dalam Klasterisasi

   Mengintegrasikan *Traffic Limiter* dalam arsitektur *microservices* dan *cloud-native* dengan fokus pada skalabilitas dan efisiensi pemrosesan trafik dalam lingkungan yang kompleks. Mengevaluasi pendekatan distributed *rate limiting* menggunakan teknologi seperti *Redis* atau *Kafka* untuk menangani beban tinggi dan distribusi trafik secara efisien di *cluster server*