BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam era digital, sistem informasi yang dibangun menggunakan teknologi internet sangat rentan terhadap serangan serangan *Distributed Denial of Service* (*DDoS*). Serangan *DDoS* adalah salah satu ancaman terhadap keamanan sistem informasi yang paling umum dan paling sulit diatasi[1]. Dalam serangan *DDoS*, serangan yang dilakukan oleh banyak sumber daya (*botnet*) yang terkoordinasi untuk menghancurkan sistem informasi dengan cara mengalirkan lalu lintas trafik yang besar dan tidak sah ke sistem. Serangan ini membanjiri target dengan lalu lintas data yang sangat besar dari berbagai sumber, sehingga mengakibatkan kelebihan beban (*overload*) pada server dan mengganggu akses bagi pengguna yang sah[2]. Dalam konteks aplikasi web yang dibangun menggunakan *Flask API*, serangan *DDoS* dapat menyebabkan penurunan kinerja yang parah, bahkan berpotensi melumpuhkan aplikasi sepenuhnya[3].

Dalam konteks aplikasi web yang dibangun dengan menggunakan *Flask API*, ancaman dari serangan *DDoS* menjadi semakin relevan. *Flask*, sebagai salah satu *framework web* yang populer untuk pembuatan *aplikasi web*, menawarkan kemudahan dan fleksibilitas dalam pengembangan[4]. Namun, kesederhanaan dan kemudahan akses ini juga membuat *Flask API* rentan terhadap serangan *DDoS*. Ketika aplikasi web yang dibangun dengan *Flask API* menghadapi serangan *DDoS*, dampaknya dapat sangat merugikan, mulai dari penurunan kinerja yang signifikan hingga kemungkinan terjadinya kerusakan sistem yang total[5]. Namun, kerentanan terhadap serangan *DDoS* memerlukan langkah-langkah mitigasi yang efektif. Salah satu pendekatan yang menjanjikan adalah penggunaan *traffic limiter*[6], [7].

Kerentanan *Flask API* terhadap serangan *DDoS* menekankan kebutuhan mendesak akan solusi mitigasi yang efektif. Salah satu pendekatan yang menjanjikan untuk mengatasi masalah ini adalah penggunaan traffic limiter. *Traffic limiter* merupakan alat yang dapat digunakan untuk mengatur dan

digunakan untuk menghambat serangan *DDoS* dengan cara mengurangi jumlah lalu lintas trafik yang masuk ke sistem[2], [9]. Dengan menggunakan *Traffic Limiter*, sistem dapat lebih stabil dan lebih aman dari serangan-serangan *DDoS*[10]. Dengan mengatur batas ini, *traffic limiter* dapat mencegah lonjakan lalu lintas yang mendadak dan melindungi aplikasi dari serangan *DDoS*. Namun, penerapan traffic limiter pada *Flask API* memerlukan pertimbangan yang cermat. Penting untuk memastikan bahwa penerapan traffic limiter tidak mengganggu pengguna yang sah dan tetap menjaga kinerja aplikasi secara keseluruhan. Ini berarti bahwa batas yang diterapkan harus seimbang—cukup ketat untuk mencegah serangan, tetapi tidak terlalu ketat sehingga mengganggu akses normal pengguna[11], [12]. Penelitian ini bertujuan untuk mengeksplorasi implementasi *traffic limiter* dalam konteks *Flask API*, menguji efektivitasnya dalam menangkal serangan *DDoS*, serta menganalisis dampaknya terhadap kinerja aplikasi web.

Berdasarkan permasalahan yang telah diuraikan, penelitian ini memiliki tujuan utama untuk melakukan remediasi terhadap serangan *DDoS* pada *Flask API* melalui penggunaan *traffic limiter*. Penelitian ini akan menyelidiki bagaimana traffic limiter diimplementasikan dalam lingkungan *Flask API*, mengukur efektivitasnya dalam menghadapi serangan *DDoS*, dan menganalisis dampaknya terhadap kinerja sistem secara keseluruhan. Melalui penelitian ini, diharapkan dapat ditemukan solusi tentang cara yang efektif untuk meningkatkan ketahanan *Flask API* terhadap serangan *DDoS*. Dengan mengeksplorasi dan menguji penggunaan *traffic limiter*, penelitian ini bertujuan untuk memberikan kontribusi yang berarti dalam strategi mitigasi serangan siber, sehingga membantu pengembang dan penyedia layanan dalam menjaga keamanan dan performa aplikasi *web*.

1.2 Rumusan Masalah

Berdasar latar belakang masalah pada 1.1, rumusan masalah yang hendak diselesaikan pada penelitian ini adalah menerapkan limitasi pada penggunaan *API (Flask API)* guna mengurangi kemungkinan terjadinya *DDoS* karena permintaan berlebih dan tidak sah yang tidak dapat ditangani oleh *gateway API*. Penelitian ini dibatasi pada penggunaan *static limiter* dengan batas permintaan per periode waktu yang telah ditentukan (misalnya, 2 permintaan per hari). Fokus penelitian akan

mengevaluasi kemampuan *static limiter* dalam menangkal serangan *DDoS* dengan melakukan limitasi pada pengguna maupun *gateway* dari Flask API dengan tujuan meredam *traffic* yang masuk dalam bentuk *request* dari pengguna yang melewati batas *threshold* yang ditentukan pada konfigurasi *limiter*. Dari latar belakang masalah tersebut, penulis pun merumuskan pertanyaan penelitian:

1. Bagaimana efektivitas penerapan static limiter dalam mengurangi dampak serangan DDoS pada Flask API?
2. Bagaimana static limiter yang diterapkan pada Flask API dapat mengurangi dampak serangan DDoS?

1.3 Batasan Masalah

Agar penelitian ini lebih terarah dan sesuai dengan rumusan masalah yang telah dipaparkan sebelumnya, peneliti membuat batasan masalah. Batasan masalah yang ditetapkan dalam penelitian ini adalah sebagai berikut:

1. **Jenis Serangan *DDoS***: Penelitian ini hanya akan fokus pada mitigasi serangan *DDoS* volumetrik sederhana dengan menggunakan *static limiter*. Serangan *DDoS* jenis lain, seperti serangan protokol atau serangan aplikasi, tidak akan dipertimbangkan dalam penelitian ini.
2. **Algoritma *Traffic Limiter***: Penelitian ini akan menggunakan algoritma *static limiter* yang sederhana, yaitu membatasi jumlah permintaan per periode waktu tertentu. Algoritma *traffic limiter* yang lebih kompleks, seperti *token bucket* atau *leaky bucket*, tidak akan dibahas dalam penelitian ini.
3. **Lingkungan Pengujian**: Evaluasi efektivitas static limiter akan dilakukan dalam lingkungan simulasi serangan *DDoS*. Pengujian dalam lingkungan produksi tidak akan dilakukan untuk menghindari risiko terhadap aplikasi yang berjalan.
4. **Fokus Aplikasi**: Penelitian ini akan fokus pada implementasi dan evaluasi *static limiter* pada *Flask API*. Meskipun prinsip-prinsipnya dapat diterapkan pada framework lain, penelitian ini tidak akan

membahas secara spesifik implementasi pada framework selain *Flask*.

5. **Perancangan Infrastruktur**: Perancangan Infrastruktur dibuat seminimal mungkin untuk melihat efektivitas dan efisiensi pada penggunaan sumber daya.
6. **Pengujian Infrastruktur**: Pengujian Infrastruktur tidak berlangsung lama, namun singkat dan dilakukan repetitif beberapa kali hingga limit pada traffic terlampaui, hal ini dilakukan karena keterbatasan infrastruktur dan sumber daya.

Dengan batasan masalah yang jelas seperti di atas, penelitian ini akan lebih terarah dan memberikan kontribusi yang lebih spesifik terhadap pemahaman tentang mitigasi serangan *DDoS* pada *Flask API* menggunakan *static limiter*.

1.4 Tujuan Penelitian

Tujuan utama yang ingin dicapai dalam penelitian ini adalah untuk merancang dan mengembangkan sebuah skema atau sistem yang sederhana namun efektif dalam mitigasi serangan *Distributed Denial of Service* (*DDoS*). Sistem ini akan menggunakan *traffic limiter* yang dirancang khusus untuk diterapkan pada *Flask API*. Dengan adanya *traffic limiter* ini, diharapkan sumber daya yang tersedia pada *API* dapat dikelola dengan lebih baik, serta dapat membatasi jumlah permintaan yang diterima dalam setiap periode waktu yang telah ditentukan sebelumnya. Melalui pendekatan ini, penulis berupaya untuk meningkatkan ketahanan *API* terhadap serangan *DDoS* dan memastikan bahwa layanan tetap dapat diakses oleh pengguna yang sah.

1.5 Manfaat Penelitian

Penelitian ini bertujuan untuk menerapkan limitasi pada penggunaan *API (Flask API)* guna mengurangi kemungkinan terjadinya serangan *DDoS* akibat permintaan berlebih dan permintaan tidak sah yang tidak dapat ditangani oleh gateway *API*. Fokus penelitian ini terbatas pada penggunaan *static limiter* dengan menetapkan batas jumlah permintaan per periode waktu tertentu (misalnya, 2

permintaan per hari). Dengan demikian, penelitian ini akan mengevaluasi efektivitas *static limiter* dalam menangkal serangan *DDoS* dan memberikan wawasan mengenai implementasi teknik pembatasan yang dapat meningkatkan keamanan *API*.

1.6 Sistematika Penulisan

Sistematik penulisan berisikan garis besar atau gambaran secara umum penelitian ini sehingga mempermudah pemahaman alur isi :

1. BAB I PENDAHULUAN

   Bab pendahuluan menyajikan uraian mengenai latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan laporan penelitian. Uraian dalam bab ini bertujuan untuk menjelaskan permasalahan yang dikaji, meliputi bagaimana permasalahan tersebut muncul, lingkup spesifik yang terkait, usulan solusi terhadap permasalahan yang ada, serta struktur penyajian laporan penelitian ini.

2. BAB II TINJAUAN PUSTAKA DAN DASAR TEORI

   Bab tinjauan pustaka dan dasar teori menyajikan pemaparan sistematis mengenai referensi dari penelitian-penelitian lain yang relevan dengan pokok-pokok atau poin-poin penting yang dibahas dalam penelitian ini, serta penjelasan mengenai teori yang terkait dengan poin-poin tersebut. Uraian dalam bab ini bertujuan untuk memberikan gambaran umum tentang poin-poin penting yang dikaji, serta melakukan perbandingan dan menjelaskan perbedaan yang ada antara penelitian ini dan penelitian-penelitian lain yang relevan.

3. BAB III METODE PENELITIAN

   Bab metode penelitian menyajikan pemaparan mengenai alat dan bahan yang digunakan, tahapan yang dilalui, serta rancangan sistem atau solusi terhadap permasalahan yang dibahas dalam penelitian ini. Uraian pada bab ini bertujuan untuk menjelaskan secara rinci spesifikasi alat dan bahan, alur tahapan dalam penerapan solusi, serta desain rancangan solusi,

sehingga dapat memberikan gambaran yang berguna sebagai acuan untuk penggunaan kembali (reuse) dan pertimbangan dalam pengembangan lebih lanjut.

4. BAB IV HASIL DAN PEMBAHASAN

Bab hasil dan pembahasan memuat pemaparan terkait hasil-hasil yang didapatkan dalam implementasi dan pengujian dari rancangan sistem atau solusi pada penelitian penelitian ini. Pemaparan pada bab ini bertujuan untuk menyampaikan dan menjelaskan terkait apa saja yang dicapai serta hasil yang didapatkan dari implementasi dan pengujian sistem atau solusi yang diusulkan, sehingga dapat menjadi bahan atau materi pembelajaran serta menjadi acuan dalam pertimbangan untuk pengembangan lebih lanjut atau untuk alternatif penyelesaian dari permasalahan yang serupa.

5. BAB V PENUTUP

Bab ini secara komprehensif menyajikan dan menganalisis hasil implementasi dan pengujian rancangan sistem/solusi, mengungkapkan capaian dan hasil konkrit yang dicapai. Dengan demikian, pemaparan ini memberikan wawasan mendalam atas efektivitas dan kemanjuran solusi, serta menjadi referensi berharga untuk pembelajaran, pertimbangan pengembangan lanjutan, dan alternatif penyelesaian permasalahan serupa, melalui interpretasi yang tajam dan implikasi praktis yang jelas.