BAB 1. PENDAHULUAN

1.1 Latar Belakang

Pesatnya pertumbuhan dan penggunaan internet di berbagai sektor telah mendorong transformasi digital di Indonesia (Gunawan dkk., 2020). Di era teknologi informasi saat ini, kebutuhan akan ketersediaan tinggi (*high availability*) dan keandalan layanan digital menjadi sangat kritis (Šimon dkk., 2023). Untuk menjawab tantangan tersebut, sistem aplikasi modern semakin mengadopsi arsitektur berbasis kontainer, di mana Kubernetes telah muncul sebagai platform orkestrasi terkemuka yang memungkinkan organisasi mengelola dan menyebarkan aplikasi berskala besar secara efisien (Aqasizade dkk., 2024). Keandalan sistem berbasis web ini sangat penting untuk memastikan kelancaran operasional, terutama saat menghadapi periode dengan aktivitas atau beban trafik yang tinggi (Annisa dkk., 2026). Di sisi lain, peningkatan penggunaan layanan digital berskala besar ini juga diiringi dengan meningkatnya ancaman keamanan siber yang mengintai infrastruktur tersebut.

Berdasarkan laporan APJII (Asosiasi Penyelenggara Jasa Internet Indonesia) tahun 2024, jumlah pengguna internet di Indonesia mencapai 221 juta jiwa (Asosiasi Penyelenggara Jasa Internet Indonesia, 2026). Angka tersebut mencakup 79,5% dari total populasi yang dilaporkan oleh BPS (Badan Pusat Statistik) yaitu sebanyak 281,6 juta jiwa (Badan Pusat Statistik, 2026). Data ini menggambarkan skala besar penetrasi internet di Indonesia, sekaligus memperlihatkan potensi risiko keamanan siber yang ada di Indonesia.

Ancaman keamanan siber di Indonesia tercermin dari laporan BSSN (Badan Siber dan Sandi Negara) tahun 2024, yang mencatat sebanyak 330,5 juta anomali lalu lintas jaringan. Salah satu ancaman utama yang diprediksi terus meningkat pada tahun 2025 adalah serangan DDoS *(Distributed Denial of Service)*. Ancaman ini menjadi perhatian karena dapat mengganggu operasional layanan digital secara signifikan (Badan Siber dan Sandi Negara, 2025).

DDoS, khususnya serangan HTTP(S) Flood, merupakan salah satu ancaman yang berdampak signifikan untuk mengganggu ketersediaan layanan website dengan membanjiri server target dengan permintaan HTTP(S) seperti *GET* dan *POST* secara berlebihan (Sarmah dkk., 2025). Serangan HTTP(S) Flood, memanfaatkan permintaan yang tampak sah atau menyerupai trafik pengguna asli dan kompleks atau berat untuk diproses dengan tujuan menghabiskan sumber daya server, sehingga menyebabkan penurunan performa atau kelumpuhan layanan (Fadhilah dkk., 2023). Dalam banyak kasus, sistem tidak mengalami kegagalan total, tetapi mengalami degradasi layanan yang berdampak pada menurunnya kualitas layanan. Untuk itu dibutuhkan solusi dalam mengatasi masalah penurunan performa atau kelumpuhan pada layanan server.

Untuk mengatasi permasalahan tersebut, berbagai pendekatan telah dikembangkan. Dari sisi keamanan, solusi seperti ModSecurity dapat diterapkan. ModSecurity adalah WAF *(Web Application Firewall)* yang merupakan alat keamanan berbasis aturan yang dapat dikonfigurasi untuk mendeteksi dan mencegah serangan siber, termasuk DDoS. Dengan aturan yang telah ditetapkan, ModSecurity mampu mengidentifikasi pola serangan dan memblokirnya (Zain dkk., 2023). Fadhilah dkk. (2023) menunjukkan bahwa serangan DDoS jenis SlowHTTP dapat dimitigasi secara efektif menggunakan ModSecurity dan ModAntiLoris, yang dimana sangat relevan untuk strategi perlindungan terhadap serangan HTTP(S) Flood. Selain itu, Indrajid, dkk. (2023) menganalisis dampak serangan SYN Flood, menegaskan kerentanan infrastruktur web terhadap variasi serangan DDoS.

Melengkapi solusi keamanan seperti ModSecurity, diperlukan pendekatan tambahan dari sisi performa. Pemanfaatan teknologi modern seperti Kubernetes dengan fitur Nginx Caching dapat meningkatkan efisiensi dan ketahanan infrastruktur web terhadap lonjakan lalu lintas jaringan (Gautham S dkk., 2021). Dengan caching, Nginx dapat menyimpan konten statis atau respons yang dihasilkan sementara, sehingga meminimalkan pemrosesan berulang terhadap permintaan HTTP(S) yang berat, seperti yang sering dimanfaatkan dalam serangan HTTP(S) Flood.

Efektivitas penerapan ModSecurity sebagai mekanisme filtering serta Nginx Caching sebagai mekanisme optimasi dalam meningkatkan ketersediaan layanan pada sistem berbasis Kubernetes menjadi aspek penting yang perlu dianalisis. Penelitian ini berfokus pada pendekatan berlapis yang menggabungkan ModSecurity dan Nginx Caching dalam Kubernetes untuk menciptakan strategi yang kuat dalam mitigasi serangan HTTP(S) Flood. ModSecurity berperan dalam mendeteksi dan memblokir permintaan berbahaya melalui aturan keamanan, sementara Nginx Caching mengurangi beban server dengan menyimpan respons yang sering diakses, menjaga ketersediaan layanan bagi pengguna sah.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana kinerja layanan Content Management System pada klaster Kubernetes tanpa mekanisme mitigasi dalam menghadapi serangan HTTP(S) Flood?
2. Bagaimana pengaruh penerapan Nginx Caching terhadap performa layanan Content Management System pada klaster Kubernetes dalam kondisi serangan HTTP(S) Flood?
3. Bagaimana pengaruh penerapan ModSecurity terhadap ketersediaan layanan (*availability*) Content Management System pada sistem berbasis Kubernetes?
4. Bagaimana efektivitas kombinasi ModSecurity dan Nginx Caching dalam meningkatkan ketersediaan layanan Content Management System dan performa sistem pada klaster Kubernetes?

1.3 Tujuan

Tujuan dari penelitian ini adalah:

1. Menganalisis kinerja layanan Content Management System pada klaster Kubernetes tanpa mekanisme mitigasi terhadap serangan HTTP(S) Flood.
2. Menganalisis pengaruh Nginx Caching terhadap performa layanan Content Management System pada klaster Kubernetes.

3. Menganalisis pengaruh ModSecurity terhadap ketersediaan layanan (*availability*) Content Management System pada klaster Kubernetes.
4. Menganalisis efektivitas kombinasi ModSecurity dan Nginx Caching dalam meningkatkan ketersediaan layanan Content Management System dan performa system pada klaster Kubernetes.

1.4 Manfaat

Penelitian ini diharapkan memberikan manfaat sebagai berikut:

1. Memberikan referensi empiris mengenai efektivitas mekanisme mitigasi HTTP(S) Flood pada lingkungan Kubernetes.
2. Menjadi dasar pertimbangan dalam memilih pendekatan keamanan (filtering) dan optimasi performa (caching) dalam menjaga ketersediaan layanan.
3. Memberikan gambaran mengenai trade-off antara keamanan dan performa dalam sistem berbasis container.

1.5 Batasan Masalah

Untuk menjaga ruang lingkup penelitian tetap terfokus dan dapat diselesaikan sesuai waktu dan sumber daya yang tersedia, penelitian ini dibatasi pada hal-hal berikut:

1. Penelitian difokuskan pada mitigasi serangan HTTP(S) Flood dan tidak mencakup jenis serangan DDoS lainnya.
2. Pengujian dilakukan menggunakan simulasi serangan HTTP(S) Flood dengan Apache JMeter.
3. Implementasi menggunakan ModSecurity dengan aturan standar (OWASP Core Rule Set) dan Nginx Caching berbasis reverse proxy.
4. Sistem diuji dalam lingkungan Kubernetes dengan konfigurasi yang telah ditentukan.
5. Parameter pengujian meliputi latency, throughput, dan request per second dalam kondisi beban terkontrol.