

BAB I

PENDAHULUAN

1.1.Latar Belakang Masalah

Dalam era modern ini, penggunaan *container* menjadi pilihan utama dalam pengelolaan dan distribusi aplikasi. *Container* seperti Docker dan Podman menawarkan sebuah lingkungan yang berbeda, terisolasi, dan fleksibel untuk menjalankan aplikasi tanpa mengkhawatirkan konflik dependensi atau masalah lingkungan[1]. Docker memiliki keunggulan berupa kontainer yang ringan dan sudah mencakup semua komponen yang dibutuhkan untuk menjalankan aplikasi, sehingga tidak bergantung pada perangkat lunak yang terpasang di *host*[2]. Podman sendiri memiliki keunggulan dimana para *developer* dapat mengelola *container* tanpa perlu menggunakan *daemon*[3]. Dengan kelebihan dari masing-masing *container* para *developer* dapat meningkatkan produktivitas dalam membuat *software* yang berkualitas.

Salah satu aspek penting dalam penerapan kontainerisasi adalah performa dari server tempat container dijalankan. Apache, Nginx, Cloudflare Server, LightTPD, Hiawatha, Cherokee, dan Apache Tomcat adalah beberapa contoh aplikasi *web server* yang umum digunakan di banyak belahan dunia[4]. Menurut data dari W3Techs hingga Januari 2025, Nginx menjadi *web server* urutan pertama yang paling banyak digunakan yaitu sebesar 34,0% disusul urutan kedua ada Apache yaitu sebesar 27,2% dan di urutan ketiga ada Cloudflare Server yaitu sebesar 23,0%[5][6].

Pada penelitian sebelumnya dalam menguji *web server* seperti penelitian yang berjudul “Analisis Perbandingan Performa Apache *Web Server* Dan Nginx Menggunakan Apache Jmeter”. Penelitian ini menyatakan bahwa Nginx lebih efisien dari segi *response time* dan lebih baik dari sisi *throughput*-nya[7]. Namun penelitian ini tidak menyebutkan secara detail jenis *container* yang digunakan. Lalu, terdapat penelitian yang membandingkan antara Docker dan Podman seperti penelitian yang berjudul “Analisis Perbandingan Kinerja Layanan *Container As A Service (CAAS)* Studi Kasus : Docker dan Podman”. Hasil penelitian ini menunjukkan bahwa Docker memiliki performa yang lebih

baik dibandingkan Podman ketika menjalankan 1-6 aplikasi. Setelah menerapkan tindakan penanggulangan terhadap serangan *container*, Podman menunjukkan keunggulan dalam performa dan penggunaan sumber daya dibandingkan Docker[1]. Namun penelitian ini tidak menyebutkan secara detail *web server* yang digunakan.

Berdasarkan latar belakang yang dijabarkan di atas, maka dilakukanlah penelitian yang nantinya menghasilkan *container* mana yang terbaik dengan menggunakan Apache sebagai *web server*-nya dan Docker dengan Podman sebagai *container*-nya. Penelitian ini menggunakan *tools* dari Apache Benchmark untuk mengukur *server response time* dan *CPU usage* yang bisa dilakukan oleh *web server*.

1.2.Rumusan Masalah

Rumusan masalah pada penelitian ini adalah apakah terdapat perbedaan signifikan dalam waktu respons server dan penggunaan CPU ketika *web server* Apache dijalankan dalam Docker dan Podman.

1.3.Tujuan Penelitian

Penelitian ini bertujuan untuk menganalisis dan membandingkan performa *web server* Apache yang dijalankan dalam lingkungan Docker dan Podman berdasarkan parameter waktu respons server (*server response time*) dan penggunaan CPU, guna menentukan teknologi *container* yang lebih optimal dalam pengelolaan *web server*.

1.4.Batasan Masalah

Untuk menghindari perluasan pembahasan yang berlebihan, berikut adalah penentuan batasan masalah dalam penelitian ini:

1. Menggunakan Apache Benchmark sebagai alat ukurnya.
2. Penelitian ini hanya menguji *server response time* dan *cpu usage*.
3. Tidak melakukan analisis terkait keamanan jaringan pada *web server* Apache.
4. Tidak menganalisis dampak pada sistem operasi.

1.5. Manfaat Penelitian

Penelitian ini diharapkan dapat menyajikan gambaran dan pengetahuan mengenai kinerja *web server* Apache saat dioperasikan dalam lingkungan Docker dan Podman, dengan memfokuskan pada parameter *server response time*, dan penggunaan CPU. Hasil penelitian ini diharapkan dapat menjadi kontribusi penting dalam literatur terkait penggunaan *web server* Apache dengan Docker dan Podman.

1.6. Sistematika Penulisan

Sistematika yang digunakan penulis dalam menyusun tugas akhir ini meliputi langkah-langkah berikut:

1. Studi Literatur

Menggali informasi dan pemahaman dari berbagai referensi, seperti buku, artikel, jurnal, dan sumber *online*, yang berkaitan dengan topik Virtualisasi, *Container*, *Web Server*.

2. Perancangan Sistem

Melakukan identifikasi terhadap permasalahan dan merancang arsitektur yang akan digunakan dalam pengembangan sistem.

3. Simulasi dan Implementasi

Menerapkan desain sistem yang telah dibuat untuk menjalankan simulasi dan implementasi.

4. Pengujian dan Analisis

Melakukan pengujian terhadap hasil simulasi dan implementasi untuk memperoleh data yang relevan sesuai dengan tujuan penelitian.

5. Penyusunan Laporan

Mendokumentasikan seluruh tahapan proses, mulai dari perancangan hingga analisis, dalam bentuk laporan tertulis.