

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Kemajuan teknologi yang pesat membuat aktivitas mencari informasi lebih mudah dilakukan dan memungkinkan berbagai jenis pekerjaan diselesaikan dengan cepat. Salah satunya adalah penggunaan teknologi web. Saat ini, banyak institusi dan organisasi bergantung pada aplikasi berbasis web yang dirancang untuk mempermudah pekerjaan mereka dan mempercepat penyebaran informasi. *E-learning* atau pembelajaran daring berbasis web menjadi banyak digunakan oleh perguruan tinggi. Salah satu aplikasi pembelajaran daring berbasis web yang sering digunakan adalah *Learning Management System Moodle*. Kemenristekdikti dalam hal ini juga mengikuti perkembangan pembelajaran online dengan mengembangkan *open online courses* seperti Sistem Pembelajaran Daring (SPADA) [1].

Moodle sebagai aplikasi berbasis web sangat bergantung pada infrastruktur web server. Sebagai salah satu komponen kunci yang dibutuhkan oleh aplikasi web, web server diharuskan dapat memberikan performa yang optimal dalam menangani banyak permintaan secara bersamaan. Masalah yang disebabkan oleh performa web server yang tidak optimal karena ketidakmampuan menangani permintaan dalam jumlah besar adalah dapat mengganggu atau menghentikan aplikasi web yang sedang digunakan. Di dalam aplikasi web terdapat dua bentuk scripting, yaitu *client-side scripting* dan *server-side scripting* [2]. *Client-side scripts* adalah program yang diproses dalam browser klien. Jenis skrip ini adalah program kecil yang diunduh dan dijalankan oleh browser. *Server-side scripting* diproses oleh web server ketika pengguna meminta informasi. Jenis skrip ini dapat dijalankan sebelum halaman web dimuat. Seperti yang dijelaskan di atas dua bentuk scripting tersebut diproses dengan cara berbeda oleh web server dalam menjalankan aplikasi web.

Selain itu, web server juga membutuhkan server agar dapat dijalankan. Web server dapat dijalankan pada server *baremetal* atau Docker yang memiliki kelebihan dan kekurangan masing-masing. Maka dari itu *performance testing* dilakukan untuk mengukur dan melakukan komparasi performa Moodle pada server

*baremetal* dan Docker. *Performance testing* digunakan untuk mengetahui efisiensi performa komponen atau sistem yang memainkan peran dalam menentukan tingkat kualitas yang dapat diterima untuk pengguna [3]. Alat pengujian yang dapat digunakan untuk melakukan *performance testing* antara lain adalah Apache JMeter. Apache JMeter digunakan untuk mengukur dan menganalisis performa server *baremetal* dan Docker. Hasil, analisis, dan temuan dari pengujian ini dapat membantu pengembang dalam memahami performa Moodle di server *baremetal* dan Docker serta merancang dan mengembangkan sistem yang sesuai dengan kebutuhan.

## 1.2. Rumusan Masalah

Berdasarkan uraian latar belakang masalah di atas maka dapat diidentifikasi permasalahan yang terjadi sebagai berikut:

1. Bagaimana performa Moodle pada *baremetal* dan Docker?
2. Bagaimana perbandingan performa pada *baremetal* dan Docker?
3. Bagaimana komponen penunjang seperti web server Apache dan PHP-FPM mempengaruhi performa Moodle pada *baremetal* dan Docker?

## 1.3. Tujuan dan Manfaat

1. Mengetahui performa Moodle pada *baremetal* dan Docker.
2. Mengetahui perbandingan performa Moodle pada *baremetal* dan Docker.
3. Mampu menganalisa pengaruh Web Server Apache dan PHP-FPM pada performa Moodle pada *baremetal* dan Docker.

## 1.4. Batasan Masalah

1. Implementasi kontainer menggunakan Docker.
2. Implementasi web server menggunakan Apache2.
3. Implementasi *database* menggunakan MariaDB.
4. Implementasi Moodle dengan konfigurasi *default*.
5. Parameter yang diamati *throughput*, *Response Time*, *error*, dan *Resource Utilization* (prosesor dan memori).
6. Perubahan konfigurasi hanya dilakukan pada Apache dan PHP-FPM.

## 1.5. Metode Penelitian

Dalam pembuatan tugas akhir ini dilakukan metodologi sebagai berikut:

### 1. Studi Literatur

Pengumpulan referensi yang berkaitan dengan *performance testing* seperti menemukan parameter yang dapat digunakan, menentukan *test case* dan *test scenario*. Sumber referensi dapat berupa artikel, buku, jurnal, atau penelitian yang telah dilakukan sebelumnya.

### 2. Analisis Sistem

Proses analisis yang dilakukan meliputi analisis struktur website Moodle. Selain itu dilakukan juga analisis terhadap struktur kebutuhan perangkat keras, perangkat lunak serta data hasil *performance Testing*.

### 3. Perancangan Sistem

Pada perancangan sistem meliputi gambaran umum sistem, diagram alur data, diagram alur sistem, perancangan *performance Testing*.

### 4. Implementasi Sistem

Tahap implementasi sistem dilakukan dengan mengimplementasikan rancangan yang sudah dibuat untuk kemudian digunakan menguji kemampuan server *baremetal* dan Docker dalam menangani *request* dari pengguna.

### 5. Pengujian Sistem

Setelah mendapatkan hasil dari implementasi sistem, penulis melakukan pengujian dan perbaikan sistem secara berkala untuk mendapatkan hasil yang maksimal.

### 6. Penulisan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan buku tugas akhir sesuai dengan rancangan yang telah dilakukan dan sesuai dengan kaidah penulisan sehingga menjadi tugas akhir yang utuh.