

# BAB I PENDAHULUAN

## A. Latar Belakang

Perkembangan teknologi informasi telah mendorong transformasi digital pada pendidikan tinggi [1]. Perguruan tinggi membutuhkan sistem informasi untuk mengelola data akademik secara terstruktur, mulai dari data mahasiswa, kurikulum, mata kuliah, penilaian, hingga evaluasi capaian pembelajaran [2]. Dalam konteks tersebut, *Outcome Based Education* atau OBE digunakan karena menekankan ketercapaian luaran pembelajaran sebagai dasar perencanaan dan evaluasi pendidikan [3]. Pada tingkat program studi, luaran tersebut direpresentasikan dalam bentuk Capaian Pembelajaran Lulusan atau CPL yang digunakan untuk menilai ketercapaian kompetensi lulusan [4].

Sistem Penilaian Capaian Pembelajaran Lulusan berbasis OBE merupakan sistem informasi akademik yang membantu pengelolaan data CPL, CPMK, mata kuliah, nilai, dosen, mahasiswa, dan relasi akademik lainnya [5]. Dalam sistem seperti ini, *Dashboard* berperan menyajikan informasi akademik dalam bentuk ringkasan, grafik, tren, dan indikator *monitoring* [6]. Pada objek penelitian, *Dashboard* admin menampilkan data agregat seperti statistik CPL, jumlah mata kuliah aktif, jumlah mahasiswa, rata-rata CPL global, grafik CPL, tren semester, kelengkapan data, *alerts*, *insights*, analisis dosen, evaluasi mahasiswa terbatas, serta *Filter Dashboard*.

*Dashboard* admin tidak mengambil seluruh data mentah dari *Database* secara langsung, tetapi menampilkan hasil pengolahan *Backend* dalam bentuk ringkasan dan agregasi. Pemisahan antara data mentah *Database* dan data agregat *Dashboard* penting agar evaluasi performa tidak keliru, karena *Dashboard* berbasis data agregat memiliki karakteristik pemrosesan yang berbeda dengan halaman web sederhana [7]. Proses pemuatan *Dashboard* melibatkan pembacaan parameter *Filter*, akses data melalui Prisma ORM, *Query* ke MariaDB, perhitungan agregat, penyusunan *Response*, dan pengiriman data ke *Frontend*. Proses tersebut dapat memengaruhi *Response time*, *Throughput*, *Error rate*, dan kestabilan sistem ketika *Endpoint* diakses secara simultan. Selain itu, peningkatan beban pemrosesan juga dapat tercermin dari perubahan penggunaan sumber daya *Server*, seperti CPU dan memori, yang diamati sebagai indikator pendukung selama proses pengujian [8].

Sistem yang menjadi objek penelitian ini adalah Sistem Penilaian CPL berbasis OBE yang dikembangkan di lingkungan Universitas Nahdlatul Ulama Al Ghazali Cilacap. Sistem menggunakan React.js dan TypeScript pada *Frontend*, Node.js dengan Express.js pada *Backend*, Prisma ORM, serta MariaDB sebagai *Database*. Data yang digunakan berupa data *dummy* karena sistem belum digunakan secara

nyata dalam proses akademik. Data *dummy* digunakan sebagai konteks skala data sistem, bukan untuk menilai validitas akademik perhitungan CPL.

Permasalahan performa perlu dikaji karena *Dashboard* admin berpotensi diakses secara simultan atau diproses berulang ketika pengguna menggunakan *Filter* tertentu. Apabila *Endpoint Dashboard* membutuhkan waktu lama untuk menghasilkan *Response*, proses *monitoring* akademik dapat menjadi kurang efektif dan pengambilan keputusan berbasis data dapat terhambat [9]. Oleh karena itu, diperlukan evaluasi performa untuk mengetahui kemampuan *Endpoint Dashboard* Admin dalam menangani *Request* pengguna secara simultan. Evaluasi dilakukan melalui pengukuran metrik performa *Endpoint* dan didukung oleh pengamatan penggunaan CPU serta memori *Server* untuk membantu menginterpretasikan perubahan performa yang terjadi selama pengujian.

Evaluasi performa sistem berbasis web dilakukan melalui pengujian performa pada berbagai tingkat beban pengguna. Salah satu perangkat lunak yang banyak digunakan untuk tujuan tersebut adalah Apache JMeter [10]. Pengujian performa dilakukan untuk mengamati kemampuan sistem pada beban pengguna bertahap, sedangkan *Stress testing* digunakan untuk mengetahui perilaku sistem pada beban tinggi [11]. Dalam penelitian ini, Apache JMeter digunakan untuk menyimulasikan pengguna virtual dan mengirim *Request* HTTP/API ke *Endpoint Dashboard* admin [12].

Pengujian dilakukan pada dua lingkungan, yaitu *server* lokal dan *Virtual Private Server* (VPS) milik Universitas Nahdlatul Ulama Al Ghazali (UNUGHA) Cilacap yang diakses melalui jaringan internet menggunakan domain obe.unugha.ac.id dan selanjutnya disebut VPS UNUGHA, menggunakan *endpoint*, data, skenario jumlah pengguna, dan parameter pengukuran yang sama. Pengujian pada kedua lingkungan diperlukan untuk mengetahui perubahan performa sistem pada konfigurasi *server* yang berbeda serta membandingkan kemampuan masing-masing lingkungan dalam menangani beban yang sama. Perbandingan tersebut menggambarkan performa sistem pada dua lingkungan aktual dan tidak dimaksudkan sebagai pengujian perangkat keras yang dikendalikan secara eksperimental.

Pengujian difokuskan pada performa proses *request-response endpoint* yang menghasilkan data agregat dari *backend*, bukan pada proses *rendering* React, tampilan *frontend*, atau keseluruhan fitur sistem. Parameter utama yang digunakan meliputi *response time*, *throughput*, dan *error rate*. Pengujian juga didukung dengan pemantauan penggunaan CPU dan memori *server* menggunakan Glances melalui *Secure Shell* atau SSH untuk membantu menginterpretasikan perubahan performa pada setiap skenario.

Penelitian sebelumnya telah membahas implementasi OBE, *dashboard* akademik, dan pengujian performa aplikasi web. Namun, kajian yang secara khusus mengevaluasi performa *endpoint Dashboard* Admin Sistem CPL berbasis OBE sebagai penghasil data agregat pada dua lingkungan *server* masih terbatas. Penelitian mengenai OBE umumnya berfokus pada implementasi kurikulum, sistem pemantauan capaian pembelajaran, atau penyajian *dashboard*. Sementara itu, penelitian *performance testing* umumnya menguji aplikasi web secara keseluruhan tanpa menyoroti karakteristik *dashboard* akademik berbasis data agregat [13].

Berdasarkan keterbatasan tersebut, kontribusi penelitian ini terletak pada evaluasi performa *endpoint API Dashboard* Admin Sistem CPL berbasis OBE pada *server* lokal dan VPS UNUGHA. Evaluasi dilakukan melalui pengukuran *response time*, *throughput*, dan *error rate*, serta didukung pemantauan penggunaan CPU dan memori. Hasil pengujian pada masing-masing *server* dianalisis dan dibandingkan pada skenario beban yang sama untuk mengetahui titik awal penurunan performa dan kestabilan sistem.

Berdasarkan hasil analisis parameter performa, penelitian selanjutnya mengidentifikasi dugaan *bottleneck* yang berkaitan dengan pemrosesan *endpoint*, proses Node.js, dan kapasitas lingkungan *server*. Identifikasi tersebut menjadi dasar dalam menyusun rekomendasi perbaikan teknis berupa optimalisasi akses data, penyesuaian pemanfaatan prosesor berdasarkan lingkungan *server*, dan pengembangan mekanisme *caching*. Rekomendasi tersebut tidak diimplementasikan maupun diuji ulang dalam penelitian ini.

Dengan demikian, penelitian ini penting dilakukan untuk memberikan gambaran mengenai kemampuan *Dashboard* Admin Sistem CPL berbasis OBE dalam menangani permintaan pengguna secara simultan pada dua lingkungan *server*. Hasil penelitian diharapkan dapat menunjukkan pola perubahan performa, perbedaan kemampuan *server* lokal dan VPS UNUGHA, titik awal penurunan performa, dugaan *bottleneck*, serta rekomendasi teknis untuk pengembangan sistem selanjutnya.

## **B. Rumusan Masalah**

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

1. Bagaimana performa *endpoint Dashboard* Admin Sistem CPL berbasis OBE pada *server* lokal dan VPS UNUGHA berdasarkan parameter *response time*, *throughput*, *error rate*, penggunaan CPU, penggunaan memori, dan kestabilan sistem pada berbagai tingkat beban pengguna?

2. Pada skenario jumlah pengguna berapa *endpoint Dashboard* Admin mulai menunjukkan penurunan performa pada *server* lokal dan VPS UNUGHA, serta bagaimana perbandingan performa kedua lingkungan tersebut?
3. Faktor teknis apa yang diduga berkontribusi terhadap munculnya *bottleneck* berdasarkan hasil analisis parameter performa, serta rekomendasi perbaikan apa yang dapat disusun untuk pengembangan sistem selanjutnya?

### C. Batasan Masalah

Agar penelitian dapat dilakukan secara lebih terarah dan sesuai dengan ruang lingkup skripsi, maka penelitian ini memiliki beberapa batasan sebagai berikut:

1. Penelitian difokuskan pada evaluasi performa tiga *endpoint API Dashboard* Admin Sistem CPL berbasis OBE, yaitu `/api/dashboard/stats`, `/api/dashboard/dosen`, dan `/api/dashboard/students`.
2. Pengujian menggunakan data *dummy* yang disusun untuk merepresentasikan struktur, relasi, volume, dan karakteristik data pada Sistem CPL.
3. Pengujian dilakukan menggunakan Apache JMeter dengan skenario beban sebesar 1, 10, 20, 50, 100, 200, dan 500 pengguna virtual.
4. Parameter utama yang dianalisis meliputi *response time*, *throughput*, dan *error rate*, sedangkan penggunaan CPU, penggunaan memori, dan kestabilan sistem digunakan sebagai parameter pendukung.
5. Pengujian dilakukan pada server lokal dan VPS UNUGHA menggunakan *endpoint*, data, parameter, serta skenario beban yang sama. Server lokal diakses melalui jaringan lokal, sedangkan VPS UNUGHA diakses oleh laptop penguji melalui jaringan internet menggunakan domain `obe.unugha.ac.id`. Nilai *response time* pada VPS UNUGHA merupakan waktu respons *end-to-end* dan penelitian tidak memisahkan latensi jaringan dari waktu pemrosesan server.
6. Penelitian difokuskan pada proses *request-response endpoint backend* dan tidak mencakup proses *rendering frontend*, pengujian antarmuka pengguna, *code profiling*, penelusuran kueri secara *real-time*, implementasi rekomendasi perbaikan, maupun pengujian ulang setelah optimasi diterapkan.

### D. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dijelaskan, maka tujuan penelitian ini adalah sebagai berikut:

1. Menganalisis performa *endpoint Dashboard* Admin Sistem CPL berbasis OBE pada *server* lokal dan VPS UNUGHA berdasarkan parameter *response*

*time, throughput, error rate*, penggunaan CPU, penggunaan memori, dan kestabilan sistem pada berbagai tingkat beban pengguna.

2. Menentukan skenario jumlah pengguna yang menyebabkan *endpoint Dashboard Admin* mulai mengalami penurunan performa pada masing-masing *server* serta membandingkan performa *server* lokal dan VPS UNUGHA pada skenario yang sama.
3. Mengidentifikasi dugaan *bottleneck* berdasarkan hasil analisis parameter performa dan menyusun rekomendasi teknis untuk meningkatkan performa sistem pada pengembangan selanjutnya.

## **E. Manfaat Penelitian**

### **1. Manfaat Teoritis**

Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan ilmu pengetahuan, khususnya pada bidang rekayasa perangkat lunak dan pengujian performa sistem. Manfaat teoretis penelitian ini adalah sebagai berikut:

- a. Menambah referensi mengenai evaluasi performa aplikasi web berbasis *Outcome-Based Education*.
- b. Memberikan contoh penerapan *load testing* dan *stress testing* pada *endpoint API* sistem akademik berbasis web.
- c. Memperkuat pemahaman mengenai hubungan antara peningkatan beban pengguna, perubahan parameter performa, dan penggunaan sumber daya *server*.
- d. Memberikan gambaran mengenai perbandingan perilaku sistem yang sama ketika dijalankan pada dua lingkungan *server* dengan kapasitas yang berbeda.
- e. Menjadi acuan bagi penelitian selanjutnya yang membahas identifikasi *bottleneck* dan optimasi performa *dashboard* akademik berbasis data agregat.

### **2. Manfaat Praktis**

Penelitian ini diharapkan memberikan manfaat praktis bagi pihak yang berkaitan dengan pengembangan dan penggunaan Sistem CPL. Manfaat tersebut adalah sebagai berikut:

- a. Bagi peneliti, penelitian ini dapat meningkatkan pemahaman mengenai pengujian, analisis, dan perbandingan performa sistem menggunakan Apache JMeter.
- b. Bagi pengembang sistem, penelitian ini memberikan informasi mengenai pola penurunan performa, *endpoint* yang memiliki beban

pemrosesan lebih tinggi, dan bagian sistem yang diduga mengalami *bottleneck*.

- c. Bagi program studi atau instansi, penelitian ini memberikan gambaran awal mengenai kemampuan *Dashboard* Admin ketika dijalankan pada *server* lokal dan VPS UNUGHA sebelum sistem digunakan secara lebih luas.
- d. Bagi pengguna sistem, hasil penelitian dapat mendukung peningkatan kualitas layanan *Dashboard* Admin pada pengembangan berikutnya.
- e. Bagi pengembangan sistem selanjutnya, rekomendasi teknis yang dihasilkan dapat digunakan sebagai arahan awal dalam melakukan *query refactoring*, penggunaan indeks basis data, migrasi menuju *external caching*, penyesuaian jumlah proses aplikasi, penerapan PM2 *cluster mode*, dan peningkatan kapasitas *server* sesuai kebutuhan.