

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Pesatnya pertumbuhan sumber informasi dan keterbatasan waktu, metodologi pembelajaran telah berubah. Pada saat ini mahasiswa dapat memperoleh pengetahuan melalui sistem *e-learning*, yang menawarkan fleksibilitas dalam mengakses materi kapan saja dan di mana saja. *E-learning* mendukung proses pembelajaran mahasiswa dengan fitur-fitur seperti manajemen konten, perlindungan data, manajemen pembelajaran dan pengiriman materi, evaluasi, serta kontrol akses (Patil dan Chandankhede, 2022). Meskipun demikian, sistem *E-Learning* membutuhkan pengelolaan pembelajaran yang lebih efisien dan efektif. Dengan jumlah pengguna yang mengakses layanan *e-learning*, kinerja *website* menjadi faktor kunci yang harus diperhatikan. *Website* yang lambat atau tidak responsif dapat menimbulkan pengalaman buruk bagi pengguna dan mempengaruhi efektivitas dalam proses pembelajaran. Oleh karena itu, diperlukan sistem *server* yang andal dan skalabel untuk menangani beban permintaan yang tinggi, memastikan *website* tetap dapat berfungsi secara optimal (Riskiono dan Pasha, 2020).

Kinerja *website* dapat diukur menggunakan beberapa metrik utama, seperti *throughput*, *response time*, dan *CPU usage*, untuk mengetahui kemampuan sistem dalam menangani permintaan pengguna dan mempertahankan performa layanan pada berbagai tingkat beban. Pengukuran terhadap metrik-metrik tersebut penting untuk mengevaluasi efektivitas suatu sistem dalam melayani pengguna secara optimal, terutama pada kondisi trafik yang tinggi (Sidik dan Safar, 2023). Dalam konteks penelitian ini, *website* yang menjadi objek kajian adalah *website* e-learning JTI Slearn milik Jurusan Teknologi Informasi Politeknik Negeri Jember yang digunakan sebagai media pembelajaran daring oleh dosen dan mahasiswa. Tingginya intensitas penggunaan *website* tersebut, terutama pada periode tertentu seperti ujian, menyebabkan peningkatan beban sistem yang berdampak pada performa layanan.

Berdasarkan kondisi tersebut, *website* e-learning JTI Slearn perlu dikaji lebih lanjut terkait kemampuan sistem dalam menangani peningkatan beban pengguna. Berdasarkan hasil pemantauan penggunaan sumber daya *server*, *website* JTI Slearn sempat mengalami peningkatan beban sistem yang cukup tinggi pada saat pelaksanaan Ujian Tengah Semester. Berdasarkan hasil pemantauan penggunaan sumber daya server yang dilakukan selama lima hari, penggunaan CPU tercatat mencapai 80%–93% pada beberapa periode pengamatan. Berdasarkan hasil pemantauan penggunaan sumber daya server yang dilakukan selama lima hari pada rentang waktu 07.00–15.00 WIB setiap hari, penggunaan CPU tercatat mencapai 80%–93% pada beberapa periode pengamatan. Beban tertinggi terjadi pada Hari ke-3, yaitu sekitar 09.20–10.00 WIB dan 11.50–14.20 WIB, Hari ke-4 sekitar 10.30–11.50 WIB, serta Hari ke-5 sekitar 07.20–07.40 WIB, 08.40–10.00 WIB, dan 11.50–13.50 WIB. Tingginya utilisasi CPU tersebut menunjukkan bahwa server mengalami tekanan beban kerja yang signifikan sehingga berpotensi menyebabkan penurunan kualitas layanan dan gangguan akses pada *website* JTI Slearn. Hal ini terjadi karena sistem masih berjalan pada satu *instance* saja, sehingga ketika terjadi satu kesalahan atau beban berlebih, seluruh layanan ikut terdampak dan menyebabkan *website* tidak dapat diakses oleh pengguna. Lonjakan penggunaan sumber daya *server* ini menyebabkan penurunan performa layanan, seperti lambatnya waktu respon dan bahkan kegagalan akses bagi sebagian pengguna. Pengujian performa mandiri terhadap *website* JTI SLearn menunjukkan bahwa sistem masih mampu menangani hingga 150 *virtual users* (VUs) dengan penggunaan CPU mencapai 88%, *memory* aplikasi sebesar 1,3–1,4 GiB, *network traffic* maksimum sebesar 72,2 Mb/s, dan *Disk IOPs* maksimum sebesar 7,95 io/s. Namun, saat beban meningkat menjadi 160 VUs, terjadi peningkatan penggunaan CPU hingga 93% dengan *network traffic* maksimum mencapai 80,0 Mb/s dan *Disk IOPs* maksimum sebesar 8,05 io/s yang menunjukkan bahwa sistem mulai mengalami *overload* akibat keterbatasan *resource* pada *single instance*. Penurunan performa pada 160 VUs mengindikasikan adanya batas beban maksimal yang telah terlampaui, sehingga diperlukan optimasi sistem agar *website* tetap stabil dan responsif dalam kondisi trafik tinggi.

Salah satu teknik yang umum digunakan untuk meningkatkan kinerja *website* adalah *load balancing*, yaitu metode untuk membagi lalu lintas secara merata ke beberapa *Instance* agar tidak terjadi beban yang berlebihan pada satu titik. Dengan penerapan *load balancing*, sistem dapat menjaga kestabilan layanan, mempercepat waktu respons, dan memastikan layanan tetap optimal saat terjadi lonjakan akses pengguna secara tiba-tiba (Riskiono dan Darwis, 2020). Menurut Adila dan Hadiwandura (2024) dalam penelitiannya menunjukkan bahwa penerapan *load balancing* sangat penting dalam menjaga performa dan keandalan *website*, yang secara langsung berkontribusi terhadap peningkatan kinerja *website*. Hasil penelitian mereka mengungkap bahwa penggunaan *load balancing* lebih unggul daripada *server* tunggal. Pada pengujian dengan 100 *request*, penggunaan *load balancing* mencatat nilai *throughput* sebesar 1,6 Kb/s dan *response time* sebesar 11,79 milidetik, lebih baik dibandingkan *server* tunggal yang hanya mencapai *throughput* 1,4 Kb/s dengan *response time* 13,87 milidetik (Adila dan Hadiwandura, 2024).

Peningkatan kinerja sistem juga dapat dicapai melalui mekanisme *auto scaling*. *Auto scaling* adalah fitur penting dalam orkestrasi kontainer Kubernetes yang memungkinkan sistem menyesuaikan jumlah sumber daya secara otomatis sesuai dengan perubahan beban kerja. Peningkatan kinerja sistem juga dapat dicapai melalui mekanisme *auto scaling*. Salah satu metode yang paling efektif adalah *Horizontal Pod Autoscaler* (HPA), sebuah fitur dalam Kubernetes yang memungkinkan sistem secara otomatis menyesuaikan jumlah replika *pod* berdasarkan pemantauan metrik seperti penggunaan CPU. Ketika beban meningkat, HPA akan menambah jumlah *pod* untuk menangani permintaan, dan menguranginya kembali saat beban menurun, sehingga memastikan efisiensi sumber daya. Menurut (Gurohman dkk., 2024) menunjukkan bahwa penerapan HPA dan Redis *Cluster* pada *platform e-learning Moodle* meningkatkan performa sistem hingga 4,3% dibandingkan pendekatan monolitik, serta mampu menjaga ketersediaan layanan saat trafik tinggi. Hal serupa juga ditemukan dalam penelitian oleh (Penggali dkk., 2024), yang menyatakan bahwa layanan CMS dengan konfigurasi HPA mampu menurunkan tingkat kegagalan permintaan secara

signifikan dari 100% menjadi hanya 14% saat diuji dengan 200 pengguna simultan. Berdasarkan hasil-hasil tersebut, dapat disimpulkan bahwa integrasi *load balancing* dan *auto scaling* melalui HPA dalam lingkungan Kubernetes sangat efektif untuk meningkatkan elastisitas dan stabilitas layanan, terutama pada sistem *e-learning* yang menghadapi dinamika trafik pengguna yang tinggi.

Penelitian ini dilakukan pada sistem *e-learning* JTI Slearn yang digunakan oleh Jurusan Teknologi Informasi, Politeknik Negeri Jember. Pemilihan tempat ini didasarkan pada adanya permasalahan nyata berupa peningkatan beban pengguna yang menyebabkan penurunan performa sistem. Melalui penelitian ini, ditujukan untuk memperoleh solusi teknis yang aplikatif dalam meningkatkan kinerja sistem, khususnya melalui penerapan teknik *load balancing*. Dengan adanya peningkatan performa, sistem JTI Slearn mampu memberikan layanan yang optimal bagi seluruh penggunanya, baik dosen maupun mahasiswa, tanpa mengalami gangguan meskipun berada pada kondisi beban tinggi.

1.2 Rumusan Masalah

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

- a. Bagaimana teknik *load balancing* dan *auto scaling* yang diimplementasikan pada website JTI Slearn dapat meningkatkan kinerja dalam menghadapi peningkatan beban akses pengguna secara bersamaan?
- b. Bagaimana perbandingan kinerja antara sistem tanpa atau dengan *load balancing* dan *auto scaling* pada website JTI Slearn?

1.3 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

- a. Menerapkan dan mengevaluasi teknik *load balancing* dan *auto scaling* untuk meningkatkan kinerja website JTI Slearn.
- b. Membandingkan kinerja website JTI Slearn sebelum dan sesudah menggunakan *load balancing* dan *auto scaling* berdasarkan metrik *throughput*, *response time*, dan *error rate*.

1.4 Manfaat

Penelitian ini diharapkan memberikan manfaat sebagai berikut:

- a. Membantu JTI Slearn dalam mengatasi permasalahan penurunan performa saat terjadi peningkatan beban pengguna, sehingga sistem tetap dapat diakses dengan cepat dan stabil.
- b. Meningkatkan performansi dan skalabilitas sistem JTI Slearn melalui penerapan *load balancing* dan *auto scaling*, sehingga layanan tetap optimal dalam berbagai kondisi penggunaan.
- c. Memberikan panduan teknis implementasi infrastruktur *server* yang lebih efisien untuk JTI Slearn, khususnya dalam pemanfaatan teknologi Kubernetes.
- d. Mendukung peningkatan kualitas layanan pembelajaran daring di lingkungan Jurusan Teknologi Informasi, karena dosen dan mahasiswa dapat mengakses materi tanpa terganggu kendala teknis.
- e. Menjadi dasar pengembangan lebih lanjut bagi tim pengelola sistem JTI Slearn dalam merancang arsitektur yang tangguh dan berorientasi pada performa tinggi.

1.5 Batasan Masalah

Agar penelitian lebih fokus dan terarah, batasan masalah dalam penelitian ini adalah sebagai berikut:

- a. Penelitian hanya dilakukan pada *website e-learning* JTI Slearn di Jurusan Teknologi Informasi, Politeknik Negeri Jember.
- b. Penerapan *load balancing* dan *auto scaling* dilakukan menggunakan teknologi Kubernetes, dengan *tools* dan konfigurasi yang telah disesuaikan.
- c. Pengujian kinerja sistem dilakukan menggunakan Grafana K6 (*Load Testing Tool*) dan difokuskan pada metrik *throughput*, *response time*, dan *error rate*, serta didukung oleh pemantauan penggunaan CPU, *memory*, *network*, dan *Disk IOPs*.
- d. Penelitian tidak membahas aspek keamanan, UI/UX, maupun kualitas materi *e-learning*, tetapi hanya berfokus pada kinerja sistem *server*.