

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengujian perangkat lunak (*software testing*) merupakan langkah kritis dalam siklus pengembangan perangkat lunak. Pada tahap ini dipastikan bahwa perangkat lunak yang dikembangkan berfungsi sesuai harapan, memenuhi persyaratan yang telah ditetapkan, dan tidak memiliki masalah yang dapat mengganggu kinerja atau keamanan sistem (Romindo et al., 2023). Salah satu aspek dalam pengujian adalah performa perangkat lunak. Pengujian performa memastikan bahwa perangkat lunak dapat beroperasi dengan efisien dan sesuai dengan ekspektasi pengguna pada waktu *run-time*. Pengujian ini dilakukan dengan mengukur parameter sistem yang menjadi kunci, seperti utilisasi sumber daya perangkat lunak, waktu respon, dan tingkat kesalahan (Wibisono & Baskoro, 2002). Dengan dilakukannya pengukuran dan pengujian performa maka dapat mengidentifikasi dan mengatasi potensi *bottleneck*, memperbaiki masalah kinerja, dan memastikan bahwa perangkat lunak berfungsi dengan baik yang akhirnya akan membangun kepercayaan pengguna (Wijaya et al., 2023). Namun ketika sebuah perangkat lunak menjadi lebih rumit dan harus dikembangkan lagi, dan dalam proses pengembangannya dibutuhkan waktu yang lebih cepat maka untuk meningkatkan efisiensi pengujian biasanya menggunakan *automation testing* (Yutia & Satrinia, 2021).

Pengujian otomatis dan pengujian manual berbeda dalam hal cara pelaksanaannya. Pengujian otomatis dilakukan dengan menyusun skrip pengujian,

sedangkan pengujian manual dilakukan secara manual oleh manusia. Pengujian otomatis memiliki beberapa kelebihan dibandingkan pengujian manual, yaitu dapat dijalankan berulang kali dengan biaya yang lebih rendah dan kecepatan yang lebih cepat. Oleh karena itu, pengujian otomatis menjadi sangat penting dalam proses pengembangan perangkat lunak (Wang, 2018).

Salah satu pengujian yang dilakukan untuk mengukur performa adalah *load testing* yang mengukur respon sistem dengan berbagai kondisi beban. Dalam konteks ini, pengujian sangat berguna untuk mengevaluasi bagaimana perangkat lunak berperilaku terhadap sejumlah pengguna yang melakukan akses secara bersamaan (Permatasari et al., 2020). *Load testing* akan menciptakan simulasi akses yang bersamaan ke aplikasi web atau situs web, yang berarti lebih efisien daripada mengharuskan banyak pengguna sebenarnya untuk mengakses secara bersamaan. Dengan simulasi tersebut, dapat diidentifikasi potensi masalah dan mengevaluasi performa aplikasi sebelum peluncuran secara luas (Permatasari et al., 2019).

Application Programming Interface (API) merupakan salah satu faktor yang sangat berpengaruh terhadap performa perangkat lunak dan aplikasi. API memainkan peran kunci dengan menyediakan kerangka kerja untuk berinteraksi dengan komponen perangkat lunak lainnya yang memungkinkan pertukaran data dan fungsionalitas yang esensial. Sebagai contoh, aplikasi berbasis web sering kali sangat tergantung pada API untuk berkomunikasi dengan server, layanan, atau perangkat lainnya (Agustika et al., 2021).

SiUKT merupakan salah satu sistem informasi yang berfungsi untuk melayani mahasiswa di lingkungan Universitas Siliwangi yang berkaitan dengan pengajuan

penyesuaian Uang Kuliah Tunggal (UKT). Sistem informasi ini dikembangkan oleh Unit Penunjang Akademis Teknologi Informasi dan Komunikasi (UPA TIK) Universitas Siliwangi. Tujuan dikembangkannya SiUKT adalah untuk memberikan keringanan biaya kuliah bagi mahasiswa yang kurang mampu secara ekonomi. Menurut *Web Developer* SiUKT dalam wawancara memberitahukan bahwa batas *user* pada *website* SiUKT adalah 300 *user* per detik serta *website* pernah diuji menggunakan *Blackbox* untuk menguji fungsi API pada *website*. Terdapat lebih dari 10 *endpoint*, dimana *endpoint-endpoint* tersebut dapat dikategorikan menjadi dua, yaitu *endpoint* publik dan *endpoint* privat. Pada penelitian ini akan menggunakan metode *Load Testing*, metode ini akan membantu mengidentifikasi bagaimana sistem dapat menangani beban kerja yang lebih tinggi dan memastikan performa sistem tetap optimal bahkan saat terjadi peningkatan penggunaan.

Berdasarkan penelitian yang dilakukan untuk mengukur performa pada aplikasi Smart Activity Tracking dengan metode *Load Testing* pada setiap *endpoint* API, pengujian berfokus pada *response time*, *throughput*, dan *error rate*. Hasil penelitian menunjukkan bahwa performa API menurun dengan peningkatan beban *request*, tetapi semua *endpoint* mampu menangani *request* dengan baik dan *error rate* 0.0% (Setiawan, Adnyana, et al., 2022).

Terdapat penelitian lain yang dilakukan, pengujian API pada aplikasi DiTenun dengan menggunakan Jmeter untuk mengukur performa. Performa diukur dengan *performance test* dan *load testing* dengan sejumlah *virtual user* secara bersamaan. Meskipun performa sebagian besar API menurun dengan peningkatan jumlah pengguna, namun tidak ada kesalahan yang ditemukan. Hasil *baseline web service*

menunjukkan *response time* dapat digunakan sebagai acuan. Hasil penelitian ini dapat membantu pengelola DiTenun meningkatkan performa aplikasi (Barus et al., 2021).

Hasil dari kedua penelitian tersebut memberikan wawasan mengenai performa aplikasi berbasis API. Dapat dipahami bahwa performa dapat berubah seiring dengan peningkatan beban, sehingga pengelola aplikasi dapat mengambil langkah-langkah untuk menjaga kualitas layanan. Selain itu, pengukuran *baseline web service* memberikan panduan yang berguna untuk pemantauan dan perbaikan performa dimasa depan. Maka dari itu, penelitian ini dimaksudkan untuk melakukan pengujian API *website* dengan tujuan perbaikan performa aplikasi SiUKT. Pengujian ini dilakukan untuk mengevaluasi bagaimana aplikasi menangani beban pengguna yang semakin meningkat, respon aplikasi ketika terjadinya perubahan beban, dan mengidentifikasi potensi masalah performa. Pengujian dilakukan menggunakan Jmeter untuk mengukur *response time*, *throughput*, dan *error rate*. Data dari hasil pengujian akan menjadi landasan penting dalam menentukan adanya perbaikan atau tidak yang harus dilakukan pada *website* SiUKT.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka rumusan masalah pada penelitian ini adalah sebagai berikut:

- a. Bagaimana performa API pada SiUKT berdasarkan hasil pengujian menggunakan metode *load test*?
- b. Bagaimana hasil analisis pengujian yang dilakukan menggunakan metode *load test* dan apa rekomendasi yang dapat diberikan kepada bagian UPA TIK untuk meningkatkan performa API SiUKT?

1.3 Tujuan Penelitian

Tujuan penelitian ini adalah sebagai berikut:

- a. Mengukur dan mengevaluasi performa *baseline response time* serta menganalisis performa perilaku *throughput* API pada SiUKT menggunakan metode *load test*.
- b. Melakukan analisis hasil pengujian menggunakan metode *load test* pada API SiUKT dan memberikan rekomendasi kepada bagian UPA TIK untuk meningkatkan performa SiUKT.

1.4 Batasan Masalah

Beberapa batasan masalah dari penelitian ini adalah sebagai berikut:

- a. Objek pengujian pada penelitian ini adalah SiUKT Universitas Siliwangi.
- b. Pengujian yang akan dilakukan sebanyak 10 *end point API*.
- c. Pengujian performa dilakukan menggunakan metode *load test*.
- d. Alat yang digunakan untuk melakukan pengujian adalah Jmeter .
- e. Pengujian dilakukan meliputi pengujian *API*, *throughput* dan *baseline response*.

1.5 Manfaat Penelitian

Manfaat penelitian ini adalah sebagai berikut:

- a. Memberikan pemahaman mengenai performa awal API SiUKT dengan mengukur *baseline response time* dan perilaku *throughput* . Hal tersebut dapat menetapkan standar awal kinerja sistem, sehingga dapat mengevaluasi perbaikan masa depan.
- b. Rekomendasi berdasarkan hasil analisis dapat membantu bagian UPA TIK dalam mengoptimalkan infrastruktur atau kode program untuk meningkatkan performa SiUKT, sehingga dapat mengurangi kemungkinan terjadinya downtime atau masalah performa yang dapat mengganggu ketika sistem diakses oleh *user*.