

BAB I

PENDAHULUAN

1.1. Latar Belakang

REST API (*Representational State Transfer Application Programming Interface*) adalah salah satu arsitektur dalam pengembangan perangkat lunak untuk menghubungkan sistem antara klien dengan server yang menggunakan *protocol* HTTP untuk melakukan proses pengiriman dan penerimaan data dengan format JSON atau XML. REST API bertujuan untuk mendukung integrasi antara aplikasi web dan aplikasi *mobile* [1]. Keunggulan dari REST API yaitu integrasi antara *backend service* dan *front end* dapat dilakukan dengan mudah dan lebih fleksibel. Dengan menggunakan URL dan HTTP protocol REST API dapat memetakan data dalam database ke endpoint yang sesuai [2]. Melalui proses ini server dan klien dapat bekerja secara independen, dimana jika terdapat perubahan pada sisi server ataupun klien sisi yang lainnya tidak ikut terpengaruh selama kontrak API dijalankan.

Performa REST API harus diperhatikan dan dioptimalkan agar memberikan pengalaman pengguna yang baik. Untuk memenuhi performa REST API yang baik harus memenuhi beberapa kriteria seperti respon data yang cepat dan akurat agar tidak menjadi hambatan bagi pengguna. Tak hanya itu, REST API harus dapat menangani banyak *request* dari klien tanpa mengonsumsi sumber daya yang tinggi, sehingga biaya penggunaan server dapat ditekan oleh pengelola aplikasi [3].

PT. Bimbingan Belajar Himalaya merupakan salah satu perusahaan pada bidang Pendidikan dengan menyediakan layanan bimbingan belajar pada berbagai jenjang. Mulai dari sekolah dasar hingga sekolah menengah atas bahkan sampai dengan persiapan masuk ke perguruan tinggi negeri. Salah satu layanan yang ada pada bimbingan belajar tersebut yaitu aplikasi Jadi Juara. Aplikasi Jadi Juara merupakan aplikasi yang memiliki layanan materi pembelajaran dan latihan soal-soal secara *online*. Aplikasi Jadi Juara saat ini tersedia pada google playstore.

Saat ini aplikasi Jadi Juara menggunakan arsitektur REST API untuk komunikasi antara *backend service* dan *front end* pada aplikasi Jadi Juara. REST

API ini menggunakan framework Laravel untuk pengembangan *backend service* dengan menggunakan format JSON sebagai respon datanya. REST API aplikasi Jadi Juara saat ini dapat berjalan dengan baik pada keadaan normal, namun terdapat permasalahan apabila terjadi lonjakan pengguna yang tinggi seperti saat mengadakan ujian bersama. Akibat dari lonjakan pengguna tersebut beban yang ditanggung pada server lebih tinggi daripada biasanya, sehingga menyebabkan penurunan performa waktu respon REST API pada *backend service* dalam mengolah data, bahkan server untuk menjalankan *backend service* pernah mengalami gangguan layanan (*downtime*). Hal ini memberikan pengalaman pengguna yang kurang baik bagi pengguna aplikasi Jadi Juara. Menurut [4] 53% pengguna aplikasi *mobile* akan meninggalkan situs jika membutuhkan waktu 3 detik untuk memuat halaman. Maka dari itu REST API pada aplikasi Jadi Juara perlu dioptimalkan agar dapat melayani apabila terjadi lonjakan pengguna.

Pada penelitian ini REST API aplikasi Jadi Juara dioptimalkan menggunakan metode *in-memory cache* dan *queue*. *Queue* merupakan metode untuk mengatur dan memproses tugas yang ada pada *backend service* secara asinkron dengan cara menempatkan tugas-tugas dalam antrian agar diproses di latar belakang secara asinkron [5]. Dengan menggunakan *queue* tugas-tugas yang berat pada endpoint REST API aplikasi Jadi Juara seperti pengiriman email atau tugas-tugas berat lainnya dapat dijalankan tanpa mengganggu pengalaman pengguna. Sedangkan *in-memory cache* adalah proses menyimpan data yang dibutuhkan oleh aplikasi ke memori [6]. *In-memory cache* digunakan untuk mengurangi *query* ke database secara terus menerus. Hal ini karena *query* ke database membutuhkan *latency* yang lebih tinggi dan *bandwidth* yang lebih rendah [7]. Pada penelitian yang dilakukan oleh [8] membuktikan bahwa metode *queue* dan *in-memory cache* dapat meningkatkan performa REST API. Pada penelitian tersebut penggunaan *queue* dan *in-memory cache* dapat menurunkan penggunaan CPU sebesar 1,4%. Penelitian lain juga membuktikan bahwa metode *queue* dapat menurunkan waktu respon dari 15 detik menjadi mendekati 0 milidetik, sedangkan metode *in-memory cache* dapat menurunkan waktu respon dari 15 milidetik menjadi 10 milidetik [5].

Selain itu, pada penelitian ini dilakukan juga penggantian web server dari Nginx dengan PHP-FPM ke web server FrankenPHP dengan konfigurasi *worker mode*. Menurut penelitian [9] FrankenPHP dengan konfigurasi *worker mode* memiliki performa waktu respon yang lebih singkat pada *concurrent* tinggi daripada web server yang lainnya, yang mana pada web server FrankenPHP saat menangani 1000 *concurrent* dapat menghasilkan waktu respon sekitar 180 milidetik.

Pemilihan *tools* untuk pengembangan REST API memiliki peran penting karena dapat mempengaruhi kinerja server termasuk waktu respon, penggunaan CPU dan penggunaan memori [10]. Maka dari itu Redis dipilih sebagai *tools* untuk menerapkan metode *queue* dan *in-memory cache*. Redis dipilih karena memiliki performa yang luar biasa dengan *latency* yang konsisten rendah [11].

Untuk menghitung kinerja dari REST API diperlukan pengujian *load testing*. *Load testing* adalah pengujian kinerja dengan mengukur respon sistem di berbagai kondisi beban yang membantu menentukan bagaimana perangkat lunak berperilaku saat diakses oleh pengguna secara bersamaan [12]. Pengukuran kinerja *load testing* dapat dilihat pada parameter *request per second* dan *time per request* [13], [14]. Tak hanya itu, penggunaan sumber daya seperti penggunaan CPU dan memori akan dievaluasi dalam beberapa scenario pengujian untuk melihat karakteristik performa dari REST API [15].

Berdasarkan permasalahan dan solusi yang telah diuraikan, penelitian ini bertujuan untuk mengoptimalkan performa REST API aplikasi Jadi Juara dengan mengimplementasikan metode *queue*, *in-memory cache* dan mengganti penggunaan web server dari Nginx ke FrankenPHP serta metode *queue* dan *in-memory cache* dengan FrankenPHP. Melalui optimasi ini, diharapkan aplikasi Jadi Juara mampu memberikan pengalaman pengguna yang lebih baik, terutama dalam menghadapi lonjakan pengguna secara bersamaan. Hasil penelitian ini juga diharapkan dapat memberikan kontribusi dalam pengembangan aplikasi berbasis REST API PHP yang efisien, serta menjadi referensi bagi pengembang aplikasi lainnya dalam menangani permasalahan serupa.

1.2.Rumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan terdapat beberapa rumusan masalah sebagai berikut.

1. Bagaimana cara melakukan peningkatan performa REST API pada aplikasi Jadi Juara?
2. Bagaimana performa REST API saat melayani lonjakan intensitas pengguna setelah dilakukan optimasi?
3. Apa rekomendasi terbaik untuk peningkatan performa REST API agar tidak terjadi *server down* pada REST API aplikasi Jadi Juara?

1.3.Tujuan Penelitian

Pada penelitian ini memiliki tujuan yang akan dicapai, berikut tujuan dari penelitian ini:

1. Mengoptimalkan performa REST API pada aplikasi Jadi Juara agar mampu menangani lonjakan jumlah pengguna secara bersamaan dengan respon yang cepat dan stabil.
2. Mengimplementasikan *queue* dan *in-memory caching* menggunakan Redis untuk meningkatkan efisiensi pengolahan data dan waktu respons REST API.
3. Mengimplementasikan FrankenPHP sebagai web server untuk meningkatkan performa REST API.
4. Menganalisis dan membuktikan keunggulan *queue* dan *in-memory caching* dengan Redis untuk meningkatkan performa REST API.
5. Menganalisis dampak penggunaan FrankenPHP untuk meningkatkan performa REST API.

1.4. Manfaat Penelitian

Pada penelitian ini diharapkan memberikan kontribusi dalam berbagai aspek. Berikut merupakan manfaat dari penelitian ini:

1. Penelitian ini bermanfaat untuk mengukur performa REST API pada framework Laravel.
2. Penelitian ini dapat digunakan sebagai rujukan untuk meningkatkan kualitas dan efisiensi REST API Laravel.

3. Penelitian ini dapat digunakan sebagai referensi penggunaan *Queue* dan *in-memory cache* dengan Redis untuk meningkatkan performa REST API.
4. Penelitian ini dapat digunakan sebagai rujukan untuk menentukan web server yang akan digunakan pada aplikasi berbasis framework Laravel.

1.5. Batasan Masalah

Untuk menjaga penelitian ini tetap terarah dan tidak meluas dari topik yang dituju, beberapa batasan masalah ditetapkan. Berikut Batasan masalah yang diterapkan pada penelitian ini:

1. Penelitian ini hanya akan fokus pada optimasi performa REST API pada aplikasi Jadi Juara yang menggunakan framework Laravel sebagai *backend*.
2. Penelitian ini terbatas pada implementasi *queue* dan *in-memory caching* menggunakan Redis untuk meningkatkan efisiensi pengolahan data dan respons REST API.
3. Penelitian ini hanya menggunakan web server FrankenPHP sebagai optimasi web server.
4. Penelitian ini tidak akan membahas aspek pengembangan *frontend* aplikasi atau fitur lain selain REST API dalam aplikasi Jadi Juara.
5. Penelitian ini hanya akan menguji dan menganalisis kinerja REST API dari sisi server, dan tidak akan melibatkan pengujian pada perangkat pengguna atau faktor eksternal yang dapat mempengaruhi kinerja aplikasi.
6. Penelitian ini akan mengukur kinerja REST API dengan menggunakan metode *load testing* untuk menganalisis respon sistem dalam berbagai kondisi beban pengguna secara bersamaan.
7. FrankenPHP yang digunakan pada penelitian ini hanya menggunakan *worker mode*.
8. Parameter load testing hanya *request per second*, *time per request*, penggunaan CPU dan penggunaan Memori.
9. Karena keterbatasan provider email, maka endpoint yang menggunakan pengiriman email hanya akan dibatasi dengan maksimal 10 concurrent

Halaman ini sengaja dikosongkan