

BAB IV KESIMPULAN

4.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan mengenai implementasi arsitektur *microservice* pada *API (Application Programming Interface)* EVIRA mobile. Hasil Performa *API* EVIRA mobile menggunakan pengujian *Load Testing API* dapat lolos pengujian dan menangani semua permintaan dengan baik dan tanpa adanya error, hal ini dibuktikan dengan hasil pengujian *API* dengan *Error Rate* sebesar 0%. Hasil Pengujian menunjukkan bahwa *response time* pada *Endpoint* Login mengalami peningkatan seiring bertambahnya *virtual users*. Namun pada *Endpoint* lain mengalami ketidakstabilan dalam menangani permintaan pengguna.

Pengujian menunjukkan bahwa waktu *response time* selalu naik seiring bertambahnya jumlah *virtual users* pada *endpoint* Login dengan 50 *virtual users* mendapatkan hasil persentil 95% selama 2,5 detik, dengan 100 *virtual users* mendapatkan hasil waktu respon persentil 95% selama 5,5 detik dan dengan 150 *virtual users* mendapatkan hasil respon persentil 95% selama 21,7 detik. Namun beberapa *endpoint* memiliki *response time* yang tidak menentu seiring bertambahnya *virtual users*. Pada *Endpoint* grafik_bentuk_pt dengan 50 *virtual users* mendapatkan hasil persentil 95% selama 0,5 detik, dengan 100 *virtual users* mendapatkan hasil waktu respon persentil 95% selama 2,6 detik dan dengan 150 *virtual users* mendapatkan hasil respon persentil 95% selama 0,9 detik. Dengan menggunakan *virtual users* 50 menuju ke 100 *endpoint* grafik_bentuk_pt mengalami kenaikan waktu respon, namun mengalami penurunan waktu respon dari *virtual users* 100 ke 150.

4.2 Saran

Dalam menyempurnakan penelitian tentang pengembangan aplikasi berbasis arsitektur *microservice*, adapun saran dari penulis untuk menyempurnakan penelitian pengembangan arsitektur *microservice* untuk di penelitian selanjutnya. Berikut saran dari penulis untuk pengembangan dan penelitian aplikasi berbasis arsitektur *microservice* kedepannya.

a. Pengembangan Layanan Aplikasi

Pengembangan aplikasi *API Microservice EVIRA mobile* menggunakan teknologi *node js* dan *flask*. Beberapa layanan pada *API EVIRA* masih mengalami kegagalan saat melakukan pengujian performa waktu response, untuk mengatasi hal tersebut pengembangan layanan yang membutuhkan proses lama atau berat dapat menggunakan bahasa pemrograman atau *framework* yang memiliki performa tinggi seperti *golang*.

b. Deployment Aplikasi

Dalam proses penyebaran aplikasi atau deployment *API EVIRA mobile* menggunakan teknologi *kontainerisasi* dengan *docker*. Namun *docker* masih belum mampu dalam melakukan otomatisasi dan orkestrasi *kontainer*. Dalam mendukung performa *microservice* sistem orkestrasi *kontainer* menggunakan platform *kubernetes* dan *docker swarm* menjadi solusi dengan sistem otomatisasi *deployment* dan replikasi *kontainer* berdasarkan beban dari *kontainer* yang memaksimalkan potensi dari arsitektur *microservice*.