

# **BAB V**

## **PENUTUP**

### **5.1 Kesimpulan**

Dari hasil proses implementasi pengujian pada bab sebelumnya terlihat terjadi perbedaan antara *environment* docker compose, Kubernetes, dan penggunaan *Autoscaling*. Hal ini salah satunya di karenakan pod database mysql tidak mampu menangani koneksi yang berasal dari pod Aplikasi Moodle yang tinggi, dan berlanjut menjadikan Aplikasi Moodle memberikan *response* yaitu berupa HTTP Error 503 (Service Unavailable).

Mengingat aplikasi Moodle ini sendiri adalah aplikasi *stateful* yang menggunakan database untuk dapat bekerja. Dan fitur *Autoscaling* pada kubernetes sendiri lebih di peruntukkan pada aplikasi yang bersifat *stateless*.

### **5.2 Saran**

Permasalahan yang disebutkan pada kesimpulan mungkin dapat di selesaikan dengan beberapa metode implementasi database dengan skalabilitas tinggi, seperti database sharding, atau database operator, atau dalam Kubernetes menggunakan StatefulSet dan membuat replica layaknya Autoscaling.

Dalam proses pembangunan infrastruktur aplikasi ada banyak hal yang dapat di tentukan dalam jangka waktu tertentu, seperti apa aplikasi yang akan di jalankan, seperti apa data akan di kelola oleh server, dan bagaimana pengguna dapat berinteraksi dengan aplikasi tersebut. Dari data tersebut kita dapat memberikan

saran atau rekomendasi terhadap infrastruktur yang akan di bangun. Ada banyak sekali infrastruktur khususnya pada aplikasi web contohnya Elearning Moodle, mengingat aplikasi Moodle hanya memerlukan PHP dan Mysql untuk dapat menjalankan, maka infrastruktur tercepat dan termudah yang dapat di bangun adalah Web Server dengan Apache2 + PHP dan Mysql, dengan melupakan bagaimana pengguna berinteraksi dan berapa banyak pengguna yang mengakses aplikasi tersebut.

Mungkin dengan jumlah pengguna yang tidak banyak, infratraktur model ini cukup untuk digunakan di banyak kasus, akan tetapi dari sisi data yang tentu terus bertambah dan berkembang seiring waktu, menjadikan proses maintenance juga perlu rutin di lakukan. Ada beberapa hal yang dilakukan untuk menyiasati masalah tersebut, jika aplikasi digunakan dengan serentak maka solusi diantaranya adalah penjadwalan, pembuatan gelombang, atau buka tutup aplikasi. Semisal KRS dilakukan secara bertahap selama 4 hari, dimana hari pertama adalah jurusan A dan hari kedua untuk Jurusan B dan seterusnya. Selain itu juga mnggunakan solusi *Vertical Scaling* yaitu dengan menambah spesifikasi perangkat keras server yang digunakan. Mengingat infrastruktur ini masih tetap dipakai karena masih layak, hemat, dan mudah dalam proses pemasangannya.