

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Pada era pandemi ini banyak instansi Pendidikan beralih dari pembelajaran *offline* ke pembelajaran online menggunakan platform *e-learning*, akan tetapi banyak permasalahan yang muncul pada praktiknya, salah satunya adalah permasalahan pada lambatnya server *e-learning*, ketika trafik naik pada aplikasi, otomatis beban request server juga naik. Jika server tidak mampu untuk mengatasi beban request ketika trafik naik, maka pengguna aplikasi akan terganggu. gangguan yang terjadi diantaranya dikarenakan kondisi server tidak mampu untuk menerima banyaknya *request* yang masuk dan sampai ada beberapa kasus server mati karena server *overload*. Jika hal ini terjadi tentu akan mengganggu proses belajar mahasiswa maupun siswa, sehingga proses belajar akan kurang maksimal. Pada saat ini permasalahan ini terjadi pada server *e-learning* yang ada di jurusan Teknologi Informasi Politeknik Negeri Jember, sehingga banyak mahasiswa yang mengeluh karena lambatnya server *e-learning*, setelah dianalisis ternyata hal tersebut terjadi karena server tidak mampu menangani trafik yang tinggi.

Untuk mengatasi permasalahan yang terjadi pada server diperlukan pembagian beban kerja atau disebut *Load Balancing*. *Load Balancing* digunakan untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, sehingga yang menerima beban pada saat terjadi proses bisnis pada aplikasi tidak hanya satu server, ada dua atau lebih server yang lain, sehingga pada saat trafik naik server tidak akan menerima beban yang banyak karena pembagian beban oleh *load balancer*. Load balancing masih kurang efektif, karena load balancing pada tingkat server terlalu besar *resource* yang digunakan sehingga akan memakan banyak *resource*. Solusi lainnya adalah menggunakan unit yang lebih kecil dari server *baremetal* atau *virtual machine* (VM) yaitu *container*. Dengan menggunakan *container resource* yang digunakan lebih kecil dari pada menggunakan server *baremetal* atau VM. pada *container* sisi *load balancing* tidak ada permasalahan akan tetapi dengan *resource* kecil, untuk *load balancing* pastinya akan sulit untuk mengelola banyak *container*, untuk *scaling up* Ketika trafik naik

juga akan sulit dikarenakan sysadmin perlu menambahkan *container* ke konfigurasi load balancing. Pada server yang ada pada jurusan TI Politeknik Negeri Jember menggunakan arsitektur *Virtual Machine*(VM). VM menggunakan *resources* yang besar dan membutuhkan banyak usaha dan waktu Ketika melakukan *scaling up* ketika trafik naik.

Maka dari itu kami memiliki solusi untuk mengatasi permasalahan yang terjadi, yaitu dengan cara menggunakan Kubernetes untuk orkestrasi *container*. Kubernetes dapat mengatasi permasalahan yang dialami beberapa platform. Dengan Kubernetes *load balancing* bisa menggunakan Service Kubernetes dan masih ada beberapa opsi yang dapat digunakan untuk load balancing pada kubernetes, dan untuk mengelola banyak *container* kita lebih efektif menggunakan Kubernetes. Pada Kubernetes untuk mengatasi permasalahan Ketika trafik naik ataupun trafik turun dapat menggunakan autoscaling atau HPA (Horizontal Pod Autoscaling) fitur dari Kubernetes. Ketika trafik naik maka akan otomatis *scaling up* pada *container* atau pada Kubernetes disebut dengan Pod. Ketika trafik turun, Maka pod yang tadi ditambahkan Ketika *scaling up* akan dikurangi dengan sesuai parameter yang ditentukan. Untuk proses *scaling up* ataupun *scaling down* tidak membutuhkan banyak usaha dan waktu, dikarenakan menggunakan *container*. Pada proposal ini penulis menggunakan Docker sebagai *container engine* yang di integrasi kan dengan Kubernetes.

1.2 Rumusan Masalah

Berdasarkan dari uraian masalah yang dijelaskan pada latar belakang di atas, maka penulis mendapatkan pokok permasalahan sebagai berikut:

1. Bagaimana mewujudkan *load balancing* pada server *e-learning* dengan kubernetes dan docker untuk mengatasi lambatnya *e-learning* ?
2. Bagaimana menerapkan *autoscaling* bersamaan dengan *load balancing* pada server *e-learning* kubernetes dan docker?
3. Bagaimana pengujian hasil penerapan metode *load balancing* dan *autoscaling* pada server *e-learning* dengan kubernetes dan docker ?

1.3 Tujuan

Berdasarkan rumusan masalah, tujuan dari penelitian ini adalah:

1. Menyelesaikan permasalahan lambatnya server e-learning dengan mengimplementasikan *load balancing* menggunakan Kubernetes dan Docker.
2. Mengimplementasikan *Autoscaling* bersamaan dengan *load balancing* server e-learning dengan Kubernetes dan Docker.
3. Melakukan pengujian terhadap metode *load balancing* dan *Autoscaling E-learning* server yang digunakan.

1.4 Manfaat

Adapun manfaat yang diharapkan dari pembuatan laporan ini adalah sebagai berikut:

1. Teratasi permasalahan lambatnya server *e-learning*.
2. Proses pembelajaran tidak terganggu sehingga mahasiswa atau siswa dapat belajar dengan maksimal dengan menggunakan *e-learning*.
3. Menemukan metode paling efektif untuk *Load balancing* dan *Autoscaling E-learning* Server.