

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan dari proses pembuatan sistem *freelance* berbasis web dengan menerapkan arsitektur mikroservis dan orkestrasi kubernetes menggunakan paradigma domain-driven design dengan studi kasus dari M-Knows Consulting sebagai berikut:

1. Proses perancangan arsitektur mikroservis pada proyek ini berhasil dibuat menggunakan paradigma DDD dengan bantuan prinsip bounded context yang memecah model ERD menjadi beberapa domain bisnis. Pemisahan ini diperlukan analisis terhadap tiap tabel di dalam ERD yang dibagi berdasarkan tujuan dari tabel ERD terhadap sistem. Apabila terdapat domain bisnis di dalam *context* yang berbeda membutuhkan tabel tertentu sebagai *entity* acuan, maka *entity* acuan tersebut akan dimanfaatkan di tiap domain bisnis dalam *context* yang berbeda. Hal ini sebagai langkah agar kebutuhan sistem terisolasi dengan baik. Setelah, domain bisnis berhasil untuk dirancang atas ERD, maka kebutuhan selanjutnya adalah merefleksikan domain bisnis ke dalam arsitektur mikroservis. Arsitektur mikroservis yang sesuai dengan rancangan ERD tersebut ada 6 yaitu '*auth*', '*user*', '*product*', '*chat*', '*order*', dan '*review*'. Selain itu, peneliti juga menambahkan mikroservis lainnya seperti API *gateway* dan '*notification*' sebagai penunjang kebutuhan sistem.
2. Perancangan dan penerapan arsitektur mikroservis dapat diimplementasikan sesuai dengan kondisi proyek yang ada dengan menggunakan paradigma DDD (Domain Driven-Design) dan prinsip bounded context yang memberikan batasan agar dapat mengurangi kompleksitas dari sistem dengan membagi atau mengelompokkan domain bisnis menjadi bagian yang lebih kecil, data yang digunakan adalah ERD yang telah dirancang sebelumnya untuk memberikan fokus yang lebih ideal. Dengan adanya DDD dan prinsip bounded context, proyek ini dipermudah atas perancangan arsitektur mikroservis dan pengimplementasian multi *database*.

3. Penerapan Kubernetes sebagai manajemen *workload* berhasil diterapkan dengan menyesuaikan kondisi proyek yang ada yang secara keseluruhan implementasinya dipaparkan pada bab 4 dengan menggunakan *cluster* Minikube. Proses *deploy* mikroservis ke dalam Kubernetes membutuhkan bantuan *container registry* yang menyimpan *image container* yang akan dilakukan proses *pull image container* ke dalam Kubernetes versi lokal.

5.2. Saran

Dalam penelitian ini terdapat beberapa kekurangan dan untuk lebih mengoptimalkan penelitian ini terdapat beberapa saran sebagai berikut:

1. Melakukan integrasi Kubernetes versi lokal ke dalam Kubernetes versi *cloud* seperti AWS, GCP, dan lainnya.
2. Melakukan proses unit testing atau integration testing terhadap sistem.
3. Memecah mikroservis *user* menjadi 2 mikroservis lainya seperti pembeli dan penjual.

Sebagai catatan, apabila peneliti selanjutnya berupaya untuk melaksanakan beberapa saran yang telah disampaikan oleh peneliti saat ini, maka peneliti selanjutnya wajib untuk bekerja sama dengan M-Knows Consulting karena sumber kode yang telah dibangun oleh peneliti saat ini merupakan data kredensial atau data penting dari pihak perusahaan.