

1. Pendahuluan

1.1. Latar Belakang

Penggunaan Monolithic Architecture masih banyak di kalangan developer terutama developer pemula. Jika dilihat dari spesifikasinya, seluruh service yang ada pada suatu proyek dijadikan satu dalam sebuah codebase, dimana jika salah satu dari service tersebut mengalami kegagalan, maka semua service akan terpengaruh dan terjadi kegagalan juga. Oleh karena itu, untuk mengatasi masalah tersebut, para developer dapat menggunakan Microservices Architecture. Microservice Architecture merupakan sebuah teknologi arsitektur pengembangan aplikasi dimana aplikasi terdiri dari beberapa service berbeda[5]. Setiap service memiliki database tersendiri dan setiap service terhubung melalui sebuah saluran atau protokol yang biasanya menggunakan protokol HTTP[6].

Penggunaan Microservice Architecture untuk era sekarang cukup banyak dikarenakan beberapa kelebihan dari Microservice Architecture yang sangat menguntungkan developer. Namun di sisi lain, tidak sedikit juga developer yang menggunakan Monolithic Architecture dikarenakan lebih mudah implementasinya daripada Microservice Architecture. Banyak developer yang kurang memahami apa dan bagaimana cara kerja dari Microservice Architecture sehingga lebih memilih menggunakan Monolithic Architecture karena untuk implementasi dan bagaimana memulai pengembangan yang lebih mudah dari Microservices Architecture[6].

Kelebihan dari Microservices Architecture adalah mudah dalam hal pengaturan *service* karena setiap service dibuat terpisah yang berarti satu service memiliki ukuran yang kecil. Setiap service dapat menggunakan teknologi yang berbeda dengan service lain tergantung dengan kebutuhan bisnis dari service tersebut. Jika terdapat pembaruan pada salah satu service, kita tidak perlu deploy ulang seluruh service namun hanya service yang terdapat pembaruan saja. Terlepas dari kelebihan Microservice Architecture, masih ada kekurangan yang dimiliki arsitektur tersebut. Microservice Architecture merupakan sistem yang terdistribusi sehingga membuat Microservice Architecture lebih kompleks dan kompleksitas tersebut semakin bertambah seiring bertambahnya service. Selain itu memerlukan developer yang sudah memiliki keahlian lebih untuk mengembangkan Microservice Architecture[10]. Untuk kinerja dari Microservice Architecture pada penelitian terdahulu[11], Microservice Architecture dapat menyelesaikan permasalahan kinerja dari aplikasi yang diteliti saat menggunakan arsitektur Monolitik. Dapat disimpulkan bahwa aplikasi yang menggunakan Microservice Architecture memiliki kinerja yang lebih bagus daripada aplikasi yang menggunakan arsitektur Monolitik.

Tujuan dari penelitian ini adalah untuk mengetahui implementasi dan perbandingan kinerja dari *Microservice Architecture* pada *web app Agreeculture Market* (<https://market.agreeculture.id>) dengan *Agreeculture Market* dengan menggunakan *Monolithic Architecture* agar dapat diketahui mengapa penggunaan *Microservices Architecture* lebih baik daripada *Monolithic Architecture*. *Agreeculture Market* merupakan sebuah aplikasi web yang memiliki tujuan bisnis untuk digitalisasi pertanian di Indonesia terutama dalam hal jual beli komoditas pertanian. *Microservice Architecture* membagi seluruh komponen pada sebuah aplikasi menjadi beberapa komponen yang lebih kecil sehingga seharusnya kinerja dari tiap komponen *Agreeculture Market* memiliki kinerja yang lebih baik. Pengukuran bertujuan untuk mengetahui perbedaan kinerja antara dua arsitektur yaitu *Monolith* dan *Microservice* untuk mengetahui perbedaan kinerja aplikasi antara dua arsitektur tersebut yang kemudian dapat ditentukan arsitektur yang lebih efisien dengan kinerja yang lebih baik. Pada analisis ini, penulis akan melakukan implementasi *Microservice Architecture* pada aplikasi web *Agreeculture Market* berdasarkan codebase *Microservices* dan *Monolithic Architecture* yang diperoleh dari Telkom DBT lalu melakukan perbandingan analisis kinerja antara aplikasi *backend Agreeculture Market* ketika menggunakan teknologi *Microservice Architecture* dengan menggunakan teknologi *Monolithic Architecture*. Dengan implementasi *Microservice Architecture* diharapkan dapat meringankan beban aplikasi dan juga meningkatkan kinerja dari aplikasi tersebut.

1.2. Tujuan

A. Melakukan implementasi *Microservice Architecture* pada aplikasi *Agreeculture Market*

Implementasi *Microservice Architecture* pada aplikasi *Agreeculture Market* berguna untuk mengurangi beban dari aplikasi tersebut agar lebih ringan dan juga meminimalisir terjadinya suatu *error* maupun *bug* yang dapat mempengaruhi seluruh aplikasi sehingga aplikasi tidak dapat berjalan dengan baik.

B. Mengukur kinerja dari implementasi *Microservice Architecture* pada aplikasi *Agreeculture Market*

Pengukuran kinerja pada aplikasi *Agreeculture Market* berguna untuk mendapatkan data kinerja dan dapat digunakan untuk melakukan analisis kinerja dari *Microservice Architecture* pada aplikasi *Agreeculture Market*.

C. Melakukan analisis perbandingan kinerja antara *Microservice Architecture* dan *Monolithic Architecture*

Analisis perbandingan kinerja *Microservice Architecture* dan *Monolithic Architecture* pada aplikasi *Agreeculture Market* berguna untuk mengetahui perbandingan kinerja yang dihasilkan sebuah *service* untuk memulai sampai menyelesaikan tujuan bisnis *service* tersebut

1.3. Batasan Masalah

Masalah yang dibahas adalah implementasi dan analisis perbandingan kinerja *Microservices Architecture* dan *Monolithic Architecture* pada aplikasi web *Agreeculture Market*.