



INTISARI

MeetingYuk merupakan aplikasi *e-commerce* yang menyediakan layanan pemesanan ruang pertemuan secara daring. Sistem *backend* yang terdapat pada aplikasi MeetingYuk versi terdahulu menggunakan teknologi yang memiliki keterbatasan dalam hal skalabilitas, responsivitas, dan efisiensi penggunaan sumber daya. Oleh karena itu, dilakukan pengembangan ulang sistem *backend* yang sepenuhnya mengimplementasikan arsitektur *microservices* dengan menggunakan bahasa pemrograman Go yang selanjutnya diteliti untuk mengevaluasi kelebihan dan kekurangan *backend microservices* yang dikembangkan.

Metodologi yang digunakan meliputi metode dekomposisi aplikasi berdasar sub-domain, pola integrasi antar-*service choreography*, metode komunikasi antar-*service asynchronous* menggunakan Apache Kafka, dan Go sebagai bahasa pemrograman. Pengujian yang dilakukan meliputi pengujian *latency*, *response per second*, *response time*, dan konsumsi memori.

Hasil pengujian menunjukkan bahwa *backend microservices* yang dikembangkan menghasilkan peningkatan performa sebesar 70% pada *latency*, peningkatan *response per second* sebesar 1.598%, dan peningkatan kecepatan *response time* sebesar 54%. Pada konsumsi memori *service-service* yang dikembangkan mampu menghasilkan meningkatkan efisiensi konsumsi memori sebesar 754% tetapi lebih berkurang sebesar 349% jika dihitung secara keseluruhan termasuk layanan *message broker*.

Kata kunci: *backend microservices*, arsitektur *microservices*, Go, *e-commerce*.



ABSTRACT

MeetingYuk is an e-commerce application that provides online meeting room booking services. The backend system developed in the previous version of the MeetingYuk application uses technology that has limitations in terms of scalability, responsiveness, and efficient use of resources. Therefore, a re-development of the backend system is carried out which fully implements the microservices architecture using the Go programming language which is then examined to evaluate the advantages and disadvantages of the developed backend microservices.

The methodology used includes the application subdomain-based decomposition method, choreography service integration pattern, the asynchronous service communication method using Apache Kafka, and Go as the programming language. Tests carried out include testing latency, response per second, response time, and memory consumption.

The test results show that the developed backend microservices resulted in a 70% performance increase in latency, a 1,598% response per second increase, and a response time speed increase of 54 %. The developed service-service memory consumption is capable of increasing the efficiency of memory consumption by 754% but it is reduced by 349% if calculated as a whole including the message broker service.

Keywords : *backend microservices, microservices architecture, Go, e-commerce.*