

ABSTRAK

Microservice merupakan arsitektur yang sering digunakan untuk membangun sistem dengan performa tinggi. Desain arsitektur tertentu dapat digunakan dalam mengembangkan sistem *microservice*. Salah satunya adalah arsitektur *command-query responsibility segregation* (CQRS). *Microservice* CQRS yang dirancang pada penelitian ini menggunakan mekanisme penyimpanan *event sourcing* dan mengimplementasikan *in-memory cache* sebagai bentuk optimasi. Penelitian dilakukan untuk mengetahui cara mengimplementasikan *microservice* CQRS dengan *event sourcing* dan mengukur peningkatan performa sistem sebagai dampak optimasi menggunakan *in-memory cache* pada proses pembangunan ulang *application state*. Proses penelitian dimulai dengan merancang sistem *microservice*, melakukan pengujian performa, menganalisis hasil pengujian, dan membuat kesimpulan dari analisis. Hasil penggunaan metode statistik dengan analisis deskriptif dan tes hipotesis uji-t menunjukkan bahwa implementasi *in-memory cache* meningkatkan waktu operasi penyelesaian *payment* sebesar 379.972 milisekon. Probabilitas nilai ekstrem ($p = 7.672 \times 10^{-148}$) yang lebih kecil dari tingkat signifikansi ($\alpha = 0.01$) mengarah pada kesimpulan untuk menolak hipotesis nol dan menerima hipotesis alternatif, yaitu “implementasi *in-memory cache* mempercepat waktu penyelesaian pembayaran”. Sehingga kesimpulan dari uji hipotesis adalah implementasi *in-memory cache* signifikan dalam meningkatkan waktu penyelesaian pembayaran. Penyelesaian pembayaran yang semakin cepat menunjukkan peningkatan performa. Karena itu, kesimpulan akhir penelitian adalah implementasi *in-memory cache* dapat meningkatkan performa sistem *microservice* CQRS dengan *event sourcing*.

Kata kunci: *microservice*, CQRS, *event sourcing*, uji performa

ABSTRACT

Microservice is commonly used as an architecture to build system with high performance. Certain architectural design are mostly being used in developing a microservice system. One kind of architecture for microservice is command-query responsibility segregation (CQRS). The CQRS microservice that is built in this research uses event sourcing for storage and implement in-memory cache as an optimization. The research goal is to understand about the implementation of CQRS with event sourcing and to measure the impact of in-memory cache as an optimization approach for the application state rebuild process in order to improve system's performance. Research is conducted by developing the system of microservices, running performance testing, analyzing test result, and summarizing the analysis. Statistical methods of descriptive analysis and hypothesis test with t-test are showing that the implementation of in-memory cache improves the time to complete a business operation with the amount of 379.972 millisecond. The probability of extreme value ($p=7.672 \times 10^{-148}$) is known to be smaller than the significant level ($\alpha=0.01$) leads to rejection of the null hypothesis and acceptance of the alternative hypothesis defined as "payment completion time with in-memory cache is faster compared to without in-memory cache". Faster payment completion time indicates performance improvement. Therefore, the final conclusion of the research is the implementation of in-memory cache can improve the performance of a CQRS microservice system with event sourcing.

Keywords: microservice, CQRS, event sourcing, performance testing