

## INTISARI

Perkembangan teknologi belakangan ini sangatlah cepat, terutama pada teknologi perangkat lunak. Perkembangan ini membuat beberapa konsep dan cara pengembangan perangkat lunak yang sebelumnya sering digunakan menjadi tidak relevan. Arsitektur *monolith* yang sebelumnya sering digunakan pada pengembangan perangkat lunak mulai digantikan oleh arsitektur *microservices*. Arsitektur *microservices* memiliki masalah dan tantangan terutama pada bagian *monitoring* dan memahami keadaan internal sistem secara utuh. Oleh karena itu, dibutuhkan suatu metode untuk mengatasi permasalahan tersebut.

Tujuan penelitian ini adalah menerapkan *distributed tracing* pada sistem berbasis *microservices* menggunakan standar Opentelemetry di lingkungan Kubernetes untuk membantu dalam memahami keadaan internal sistem secara utuh. Metode penelitian melibatkan studi literatur, perbandingan metode dan alat yang dapat digunakan, perancangan infrastruktur *distributed tracing*, instrumentasi sistem *microservices*, dan analisis untuk mengevaluasi apakah penerapan *distributed tracing* berjalan sesuai rancangan dan tujuan.

Hasil penelitian ini menunjukkan bahwa penerapan *distributed tracing* mampu memahami keadaan internal sistem secara utuh serta dapat membantu dalam melakukan *debugging* dan analisis terkait performa pada sistem *microservices* melalui deteksi *error* dan *bottleneck*. Namun, penerapan *distributed tracing* memberikan dampak negatif pada performa sistem *microservices* yang tercermin pada penurunan rata-rata *response time* dan *throughput*. Penurunan tersebut bergantung pada bahasa pemrograman yang digunakan serta berbanding lurus terhadap jumlah *span* yang dibuat.

Kata kunci : *Distributed Tracing*, Arsitektur *Microservices*, *Opentelemetry*, Arsitektur *Monolith*, Kubernetes.

## ABSTRACT

*The recent advancement in technology has been incredibly rapid, especially in software technology. This progress has rendered several previously common concepts and software development approaches irrelevant. The previously prevalent monolithic architecture in software development has started to be replaced by microservices architecture. The microservices architecture presents challenges, particularly concerning monitoring and comprehending the holistic internal state of the system. Consequently, a method is needed to address these issues.*

*The objective of this research is to implement distributed tracing in microservices-based systems using the Opentelemetry standard in a Kubernetes environment to aid in comprehending the complete internal state of the system. The research methodology involves a literature review, a comparison of applicable methods and tools, the design of a distributed tracing infrastructure, the instrumentation of microservices systems, and an analysis to evaluate whether the implementation of distributed tracing aligns with the design and objectives.*

*The results of this research demonstrate that the implementation of distributed tracing enables a comprehensive understanding of the internal system state and facilitates debugging and performance analysis of microservices systems by detecting errors and bottlenecks. However, the implementation of distributed tracing has a negative impact on the performance of microservices systems, manifested by a decrease in average response time and throughput. This decline is dependent on the programming language used and directly proportional to the number of spans created.*

**Keywords :** *Distributed Tracing, Microservices Architecture, Opentelemetry, Monolith Architecture, Kubernetes.*