

INTISARI

PERANCANGAN SISTEM MONITORING PERFORMA APLIKASI MENGGUNAKAN OPENTELEMETRY DAN GRAFANA STACK

Semakin banyaknya penggunaan teknologi digital mengharuskan arsitektur aplikasi didesain agar memiliki ketersediaan dan keandalan yang tinggi. Hal ini karena ketika sebuah aplikasi tidak dapat diakses, akan menyebabkan kerugian yang tidak sedikit bagi organisasi. Oleh karena itu tim *developer* maupun *operation* harus bisa mendeteksi ketika sistem mereka sedang tidak baik-baik saja. Untuk itulah diperlukan sebuah sistem yang dapat memonitor performa aplikasi. Pada penelitian kali ini dikembangkan sebuah sistem yang dapat mengumpulkan data telemetri yaitu *metrics* dan *traces* dari sebuah aplikasi *backend* donasi online yang berbasis REST API. Telemetri tersebut dihasilkan oleh OpenTelemetry sebagai alat instrumentasi telemetri yang *open source*. Kemudian data telemetri tersebut dikumpulkan oleh OpenTelemetry Collector yang selanjutnya disimpan pada *backend* masing-masing telemetri. *Metrics* dikirimkan menuju Prometheus dan *traces* dikirimkan ke Jaeger. Data *metrics* yang dikumpulkan adalah *throughput*, *request latency*, dan *error rate* yang divisualisasikan menggunakan Grafana *dashboard*. Hasil pengujian menunjukkan bahwa sistem monitoring dapat mengumpulkan data *metrics* secara *realtime* dengan waktu tunda rata-rata 15,4 detik dari server *staging* dan 12,2 detik dari server *production*. Sistem juga dapat mendeteksi ketika terjadi anomali pada aplikasi dan mengirimkan pemberitahuan melalui Slack. Selain itu, data *traces* yang dikumpulkan dapat digunakan untuk mempermudah proses *debugging* ketika terjadi kesalahan pada aplikasi. Namun, implementasi OpenTelemetry dalam aplikasi *backend* berbasis REST API untuk mengumpulkan *metrics* dan *traces* memberikan dampak negatif pada performa aplikasi itu sendiri yaitu dapat menurunkan jumlah *request throughput* dengan penurunan rata-rata 23,31% dan menaikkan *request latency* dengan kenaikan rata-rata 22,80%.

Kata kunci: *monitoring*, APM, *system observability*, deteksi anomali, OpenTelemetry, Prometheus, Grafana

ABSTRACT

DESIGN OF APPLICATION PERFORMANCE MONITORING USING OPENTELEMETRY AND GRAFANA STACK

Abstract -- The increasingly massive use of digital technology requires that the application architecture be designed to have high availability and reliability. This is because when an application cannot be accessed, it will cause no small loss to the organization. Therefore, the development and operation teams must be able to detect when their system is not working well. For that, we need a system that can monitor application performance. In this research, a system is developed that can collect telemetry data, namely metrics and traces from an online donation backend application based on the REST API. The telemetry is produced by OpenTelemetry as an open-source telemetry instrumentation tool. Then the telemetry data is collected by the OpenTelemetry Collector which is then stored on the backend of each telemetry. Metrics are sent to Prometheus and traces are sent to Jaeger. The data metrics collected are throughput, request latency, and error rate which are visualized using the Grafana dashboard. The test results show that the monitoring system can collect metrics data in real-time with an average delay of 15,4 seconds from the staging server and 12,2 seconds from the production server. The system can also detect when an anomaly occurs in the app and sends notifications via Slack. In addition, the trace data collected can be used to simplify the debugging process when an error occurs in the application. However, the implementation of OpenTelemetry in a REST API-based backend application to monitor metrics and traces has a negative impact on the performance of the application itself, which can reduce the number of request throughput with an average decrease of 23.31% and increase request latency with an average increase of 22.80 %.

Keywords: monitoring, APM, system observability, anomaly detection, OpenTelemetry, Prometheus, Grafana