

ABSTRAK

Salah satu permasalahan yang sering dihadapi ketika melakukan pengambilan dan penyaringan data adalah duplikasi data. Hal ini dapat mengakibatkan waktu pengiriman data menjadi lebih lama. Pada GraphQL kemungkinan duplikasi data akan semakin besar karena fleksibilitas dalam pengambilan data yang diminta oleh klien. Penelitian ini bertujuan untuk meningkatkan performa GraphQL dengan menggunakan metode deduplikasi objek untuk mengurangi data yang akan dikirim oleh server dan diparsing oleh client.

Metode yang akan digunakan dalam penelitian ini adalah deduplikasi objek untuk mengurangi data duplikat pada respon GraphQL. Penelitian ini diawali dari pengumpulan data, desain arsitektur, peningkatan performa, pengujian arsitektur, dan analisis hasil performa. Pada penelitian ini akan dibuat arsitektur tanpa perlakuan, kompresi HTTP, dan deduplikasi objek untuk mengetahui perbandingan performa pada arsitektur deduplikasi objek. Parameter yang akan diuji yaitu *throughput*, *response time*, *response size*, dan *parsing time*.

Penerapan deduplikasi objek dapat digunakan untuk meningkatkan performa dari GraphQL API dengan cara menerapkan deduplikasi objek pada respon GraphQL sehingga data yang dikirim oleh *server* dan yang diparsing oleh *client* menjadi lebih kecil. Deduplikasi objek memiliki peningkatan performa pada semua parameter pengujian jika dibandingkan dengan arsitektur tanpa perlakuan yaitu peningkatan pada pengujian *throughput* sebesar 0,33% pada pengujian 100 halaman, peningkatan pada pengujian *response time* sebesar 11,04% pada pengujian 10 halaman, 62,53% pada pengujian 20 halaman, dan 95,22% pada pengujian 100 halaman, peningkatan pada pengujian *parsing time* sebesar 75,78% pada pengujian 10 halaman, 276,38% pada pengujian 50 halaman, dan 309,35% pada pengujian 100 halaman. Jika dibandingkan dengan kompresi HTTP, deduplikasi objek memiliki kelebihan pada *parsing time* yaitu peningkatan performa sebesar 64,61% pada pengujian 10 halaman, 193,76% pada pengujian 50 halaman, dan 218,07% pada pengujian 100 halaman. Pada pengujian *throughput* memiliki performa sama, dan pada pengujian *response time* tidak memiliki perbedaan yang cukup signifikan yaitu peningkatan performa sebesar 0,66% pada pengujian performa 10 halaman, penurunan performa sebesar 0,12 pada pengujian 50 halaman, dan penurunan performa sebesar 1,45% pada pengujian 100 halaman.

Kata Kunci: Duplikasi Data, GraphQL, Deduplikasi Objek

ABSTRAC

One of the problems that are often encountered when collecting and filtering data is duplication of data. This can result in a longer data transmission time. In GraphQL the possibility of data duplication will be even greater because of the flexibility in retrieving data requested by the client. This study aims to improve the performance of GraphQL by using the object deduplication method to reduce data to be sent by the server and parsed by the client.

The method to be used in this study is object deduplication to reduce duplicate data in GraphQL responses. This research begins with data collection, architectural design, performance improvement, architectural testing, and analysis of performance results. In this research, an architecture without treatment, HTTP compression, and object deduplication will be made to compare the performance of object deduplication architectures. The parameters to be tested are throughput, response time, response size, and parsing time

The application of object deduplication can be used to improve the performance of the GraphQL API by applying object deduplication to the GraphQL response so that the data sent by the server and parsed by the client becomes smaller. Object deduplication has increased performance in all test parameters when compared to an untreated architecture, namely an increase in throughput testing of 0.33% on a 100-page test, an increase in response time testing of 11.04% on a 10-page test, 62.53% on testing 20 pages, and 95.22% on testing 100 pages, an increase in parsing time testing of 75.78% on testing 10 pages, 276.38% on testing 50 pages, and 309.35% on testing 100 pages. When compared to HTTP compression, object deduplication has advantages in parsing time, namely a performance increase of 64.61% on a 10-page test, 193.76% on a 50-page test, and 218.07% on a 100-page test. The throughput test has the same performance, and the response time test does not have a significant difference, namely an increase in performance of 0.66% on a 10 page performance test, a decrease in performance of 0.12 on a 50 page test, and a decrease in performance of 1.45 % on the 100 page test.

Keywords: Data Duplication, GraphQL, Object Deduplication