UNIVERSITAS
GADJAH MADA
Analisis Performa dan Pemanfaatan Bloom Filter untuk Whitelisting pada Arsitektur Microservices
Tantowi Shah Hanif, Dr. Irkham Huda, S.Kom., M.Cs.
Universitas Gadjah Mada, 2026 | Diunduh dari http://etd.repository.ugm.ac.id/

INTISARI

ANALISIS PERFORMA DAN PEMANFAATAN BLOOM FILTER UNTUK *WHITELISTING* PADA ARSITEKTUR *MICROSERVICES*

Tantowi Shah Hanif
22/502562/SV/21429

Pada arsitektur *microservices*, proses verifikasi keanggotaan pengguna sebelum suatu permintaan diteruskan ke layanan umumnya dibebankan langsung kepada database sebagai sumber data utama, sehingga setiap permintaan menambah beban kueri dan berisiko memperlambat latensi ketika jumlah permintaan meningkat secara signifikan. Penelitian ini mengkaji penerapan *Bloom Filter* sebagai mekanisme *pre-filtering* pada Whitelist Service, yaitu layanan yang menangani verifikasi keanggotaan pengguna pada platform Social Network Service (SNS), sebuah platform jejaring sosial dengan fitur pengelolaan akses berbasis status keanggotaan, dengan konfigurasi kapasitas 10.000 entri, *false positive rate* 1%, tujuh fungsi *hash*, faktor penyangga 1,3×, dan ambang pembaruan ulang 80%.

Evaluasi dilakukan melalui tiga skenario pembebanan, yaitu kenaikan beban bertahap langsung pada Whitelist Service, kenaikan beban bertahap melalui SNS sebagai pemanggil layanan, dan lonjakan trafik secara tiba-tiba, dengan membandingkan kondisi sistem tanpa dan dengan Bloom Filter terhadap waktu respons, *throughput*, pemanfaatan sumber daya, serta jumlah kueri ke database per detik (DB QPS). Hasil menunjukkan bahwa Bloom Filter menurunkan DB QPS sebesar 73,8%–74,4%, latensi p50 berkurang hingga 62,7%, p95 hingga 73,8%, dan p99 hingga 79,6%, penggunaan CPU turun hingga 56,8% saat lonjakan trafik, serta *throughput* meningkat hingga 3,9% dengan penggunaan memori yang stabil, sehingga *Bloom Filter* terbukti mampu meningkatkan efisiensi dan stabilitas Whitelist Service pada konfigurasi dan skenario yang diuji.

Kata kunci: bloom filter*, microservices, whitelisting,* gRPC*, false positive rate,* performa, database

UNIVERSITAS
GADJAH MADA

Analisis Performa dan Pemanfaatan Bloom Filter untuk Whitelisting pada Arsitektur Microservices
Tantowi Shah Hanif, Dr. Irkham Huda, S.Kom., M.Cs.
Universitas Gadjah Mada, 2026 | Diunduh dari http://etd.repository.ugm.ac.id/

ABSTRACT

PERFORMANCE ANALYSIS AND UTILIZATION OF BLOOM FILTER FOR WHITELISTING IN MICROSERVICES ARCHITECTURE

Tantowi Shah Hanif

22/502562/SV/21429

In microservices architecture, verifying a user's membership before a request is forwarded to a service is commonly delegated directly to the database as the primary data source, causing every incoming request to add query load and risking slower response times as traffic increases significantly. This study examines the application of Bloom Filter as a pre-filtering mechanism in the Whitelist Service, a dedicated microservice for verifying user membership on a Social Network Service (SNS) platform featuring membership-based access management, configured with a capacity of 10,000 entries, a 1% false positive rate, seven hash functions, a buffer factor of 1.3×, and a rebuild threshold of 80%.

The evaluation is conducted through three load scenarios: a gradual load increase directed to the Whitelist Service, a gradual load increase through the SNS as the calling service, and a sudden traffic spike, comparing the system with and without Bloom Filter in terms of response time, throughput, resource utilization, and database queries per second (DB QPS). The results show that Bloom Filter reduces DB QPS by 73.8%–74.4%, decreases p50 latency by up to 62.7%, p95 by up to 73.8%, and p99 by up to 79.6%, lowers CPU usage by up to 56.8% during traffic spikes, and improves throughput by up to 3.9% with stable memory usage, demonstrating that Bloom Filter is an effective pre-filtering solution for improving the efficiency and stability of the Whitelist Service under the tested configuration and scenarios.

Keywords: bloom filter, microservices, whitelisting, gRPC, false positive rate, performance, database