

ABSTRAK

Server merupakan perangkat yang menyediakan layanan, baik dalam bentuk *hardware* atau *software*. Adanya server memungkinkan untuk penyediaan sumber daya seperti data. Fungsi lain dari server ialah menjadi tempat melakukan *deploy* aplikasi web. Suatu server, sebaiknya diatur supaya mempunyai *availability* atau ketersediaan maka ketika *client* membutuhkan data atau ingin mengakses aplikasi web, layanan akan tetap tersedia. Salah satu cara untuk menjaga *availability* server ialah dengan mengimplementasikan *load balancer*. *Load balancer* merupakan layanan yang bisa mendistribusikan *traffic* ke beberapa server maka server tertentu tidak akan kelebihan beban. *Load balancer* yang layanannya bisa dipakai antara lain ialah NGINX, dengan metode *round-robin*, *least-connection*, dan *ip-hash*. Tiap metode mempunyai cara kerja yang berbeda dan bisa dianalisis *Quality of Service*-nya untuk menentukan metode yang cocok. Selain itu, ketersediaan server bisa dimonitoring secara *real-time* memakai *Grafana* dan *Prometheus* untuk memastikan bila server bekerja dengan baik atau ada kendala yang perlu diatas. Untuk itu, dilakukan implementasi *load balancer* pada *flask web app* untuk tingkatan performansi web server, serta mengintegrasikan dengan layanan *Grafana* dan *Prometheus* supaya bisa dimonitoring. Sesuai hasil penelitian, diketahui *load balancer* bisa membagi *traffic* ke 3 VM memakai metode *round robin*, *least connection*, dan *IP hash*, serta dimonitoring pada dashboard *Grafana* dengan data source *Prometheus*. Dari ketiga metode, *throughput* terbesar memakai metode *IP hash* (0,571 Mbps Flask C) dan *round robin* (0,215 Mbps Flask A, 0,281 Flask B, 0,241 Flask C). Lalu *packet loss* terkecil memakai metode *least connection* (0,5% Flask A, 0,3% Flask B, 0,4% Flask C). Kemudian *delay* terkecil memakai metode *round robin* (7,362 ms Flask A, 5,549 ms Flask B, 7,992 Flask C). Terakhir *jitter* terkecil memakai metode *round robin* (4,092 ms Flask A, 1,735 ms Flask B, 3,822 Flask C).

Kata kunci: *load balancer, Flask, nginx, grafana, Prometheus*