# INTISARI

**Implementasi *Load Balancing* Pada Raspberry Pi Dengan *Monitoring* Performa Berbasis Elastic Stack**

Oleh

Nindya Salsabilla
21/474165/PA/20472

Seiring berkembangnya teknologi, kebutuhan akan informasi melalui internet kini menjadi kebutuhan pokok. Peningkakan permintaan pada layanan *website* seringkali menyebabkan *server* mengalami *overload* hingga *crash* sehingga mengganggu ketersediaan layanan. Untuk mengatasi masalah ini, diterapkan sistem *load balancing* untuk mendistribusikan permintaan secara merata ke beberapa *server worker*. Namun, solusi yang tersedia di pasaran umumnya memiliki biaya implementasi yang tinggi, sehingga dikembangkan alternatif berupa implementasi sistem *load balancing* berbasis Raspberry Pi yang didukung dengan pemantauan performa perangkat menggunakan Elastic Stack.

Dalam penelitian ini, sistem dirancang menggunakan HAProxy yang mengelola tiga *server worker* berbasis Node.js. Pengujian performa dilakukan menggunakan Apache JMeter dengan parameter *response time, throughput,* dan *error rate.* Hasil menunjukkan bahwa penerapan *load balancing* menghasilkan *response time* dan *throughput* yang lebih baik pada jumlah *request* rendah hingga menengah, dibandingkan dengan sistem tanpa *load balancing*. Namun, ditemukan *error* sebesar 0,01% pada 8.000 *request* dan meningkat hingga 96,16% pada 10.000 *request,* yang kemungkinan disebabkan oleh keterbatasan sumber daya *server.* Selain itu, performa Raspberry Pi sebagai perangkat *load balancer* berhasil dipantau melalui *dashboard* Elastic Stack, yang menyajikan data *real-time* mengenai penggunaan CPU, memori, disk, dan jaringan.

**Kata kunci:** *load balancing*, pemantauan, *Raspberry Pi, Elastic Stack*

# ABSTRACT

***Implementation of Load Balancing on Raspberry Pi with Elastic Stack-Based Performance Monitoring***

by

Nindya Salsabilla
21/474165/PA/20472

*As technology advances, the demand for information via the internet has now become a primary need. Increasing demand for website services often leads to server overload and crashes, disrupting service availability.To solve this problem, a load balancing system is implemented to distribute requests evenly to several worker servers. However, commercially available solutions generally come with high implementation costs, which led to the development of an alternative solution in the form of a Raspberry Pi-based load balancing system, supported by device performance monitoring using Elastic Stack.*

*In this research, the system is designed using HAProxy, which manages three Node.js-based worker servers. Performance testing was conducted using Apache JMeter with parameters including response time, throughput, and error rate. The results showed that the implementation of load balancing produces better response time and throughput for low to medium request volumes compared to a system without load balancing. However, errors were found at 0.01% for 8,000 requests, increasing to 96.16% at 10,000 requests, likely due to server resource limitations.Additionally, the performance of Raspberry Pi as a load balancer device is successfully monitored through the Elastic Stack dashboard, which presents real-time data on CPU, memory, disk, and network usage.*

***Keywords:*** *load balancing*, *monitoring*, *Raspberry Pi, Elastic Stack*