ABSTRAK

Pemantauan maritim berbasis Internet of Things (IoT) terkendala oleh pemborosan daya akibat pengiriman data mentah pada perangkat dengan sumber daya terbatas. Penelitian ini membandingkan metode reduksi data lossy *Thresholding* dan kompresi lossless *Gzip* untuk optimalisasi transmisi *firmware Buoy IoT* . Eksperimen dilakukan melalui akuisisi data di perairan terbuka selama 3 jam (5.400 sampel, interval 2 detik) menggunakan *firmware* terpadu dengan empat skenario pengujian simultan pada ESP32 DevKit V1 dan sensor DHT22. Hasil pengujian menunjukkan Skenario A (Raw Baseline) menghasilkan ukuran *file* 97.084 *Bytes* (RR = 0%) dengan waktu eksekusi 0 µs dan memori 0 *Bytes*. Skenario B (*Gzip* murni) mereduksi data menjadi 22.375 *Bytes* (RR = 76,95%) dengan waktu proses 272,4 ms per batch dan memori 102 KB. Skenario C (*Thresholding*) menyaring data menjadi 10.652 *Bytes* (RR = 89,03%, waktu proses 15,0 µs, memori 8 *Bytes*). Skenario D (*Hybrid*) menghasilkan ukuran *file* terkecil yaitu 3.432 *Bytes* (RR = 96,46%) dengan waktu proses 209,1 µs dan memori 20 KB. Karena tingginya variabilitas data lapangan, *Thresholding* menghasilkan rasio reduksi lebih tinggi dibanding *Gzip*. Berdasarkan analisis *Trade-off* multi-dimensi, Skenario C direkomendasikan untuk aplikasi monitoring standar karena menawarkan keseimbangan terbaik antara efisiensi, kecepatan, dan keamanan RAM, sedangkan Skenario D direkomendasikan jika rekonstruksi data penuh diperlukan.

Kata Kunci: *Buoy IoT* , ESP32, *Thresholding*, *Gzip*, reduksi data.

ABSTRACT

IoT-based maritime monitoring is constrained by excessive power consumption due to raw data transmission on resource-limited devices. This study compares lossy Thresholding data reduction and lossless Gzip compression methods for optimizing firmware transmission in Buoy IoT . Experiments were conducted through data acquisition in open water for 3 hours (5,400 samples, 2-second interval) using a unified firmware running four test scenarios simultaneously on an ESP32 DevKit V1 and a DHT22 sensor. Results show Scenario A (Raw Baseline) produces a 97,084-byte file (RR = 0%) with an execution time of 0 µs and 0 Bytes of memory. Scenario B (Gzip) compresses data to 22,375 Bytes (RR = 76.95%) with a processing time of 272.4 ms per batch and 102 KB of memory. Scenario C (Thresholding) filters data to 10,652 Bytes (RR = 89.03%, latency of 15.0 µs, 8 Bytes of memory). Scenario D (Hybrid) yields the smallest file of 3,432 Bytes (RR = 96.46%) with a latency of 209.1 µs and 20 KB of memory. Due to high data variability, Thresholding yields a higher RR than Gzip. Based on multi-dimensional Trade-off analysis, Scenario C is recommended for standard monitoring due to its optimal balance of efficiency, speed, and RAM safety, while Scenario D is recommended when full data reconstruction is required.

***Keywords**: Buoy IoT , ESP32, Thresholding, Gzip, data reduction.*