ABSTRAK

Pemantauan parameter air secara periodik memerlukan *firmware* yang mampu menjaga keteraturan akuisisi ketika komunikasi jaringan mengalami keterlambatan. Penelitian ini merancang dan membangun prototipe *monitoring* pH dan suhu pada studi kasus *sea buoy* berbasis ESP32 menggunakan *multitasking* FreeRTOS Queue. SensorTask menjalankan akuisisi periodik, sedangkan NetSendTask menangani komunikasi melalui pola *producer–consumer*. Evaluasi membandingkan *firmware baseline* sekuensial dan FreeRTOS Queue pada kondisi normal serta keterlambatan komunikasi terkontrol sebesar 3.000 ms dengan periode *sampling* 2.000 ms. Pada kondisi keterlambatan, rata-rata interval *firmware* sekuensial meningkat menjadi 3.350,998 ms, sedangkan FreeRTOS Queue mempertahankan rata-rata interval sebesar 1.999,992 ms. Pemisahan *task* tersebut menimbulkan konsekuensi berupa keterisian *queue* hingga 33 paket dan persentil ke-95 waktu tunggu sebesar 95.087,035 ms. Sistem juga berhasil mengintegrasikan ESP32, *backend* API, database, dan *dashboard* untuk pengiriman, penyimpanan, pencegahan duplikasi, serta penyajian data terbaru dan historis. Mekanisme *retry-buffer-resend* berhasil memulihkan *payload* pada gangguan sementara, tetapi membuang 75 *payload* tertua melalui kebijakan drop-oldest ketika gangguan melebihi kapasitas *buffer*.

Kata kunci: ESP32, FreeRTOS Queue, *multitasking*, periodisitas *sampling*, *sea buoy*.

ABSTRACT

Periodic water-parameter monitoring requires firmware that can maintain regular acquisition when network communication is delayed. This study designs and develops an ESP32-based pH and temperature monitoring prototype for a sea buoy case study using FreeRTOS Queue multitasking. SensorTask performs periodic acquisition, while NetSendTask handles communication through a producer–consumer pattern. The evaluation compares sequential baseline firmware and FreeRTOS Queue firmware under normal conditions and a controlled 3,000 ms communication delay with a 2,000 ms sampling period. Under the delayed condition, the sequential firmware's average interval increased to 3,350.998 ms, whereas the FreeRTOS Queue firmware maintained an average interval of 1,999.992 ms. Task separation resulted in a queue occupancy of up to 33 packets and a 95th-percentile waiting time of 95,087.035 ms. The system also successfully integrates the ESP32, backend API, database, and dashboard for data transmission, storage, duplicate prevention, and presentation of recent and historical data. The retry-buffer-resend mechanism recovered payloads during temporary disruptions but discarded 75 oldest payloads through a drop-oldest policy when the disruption exceeded the buffer capacity.

Keywords: *ESP32, FreeRTOS Queue, multitasking, sampling periodicity, sea buoy.*