

BAB 5. PENUTUP

5.1. Kesimpulan

Berdasarkan tahapan perancangan, implementasi, pengujian, dan pembahasan yang telah dilakukan, diperoleh kesimpulan sebagai berikut.

1. Pemisahan proses akuisisi dan komunikasi berhasil dirancang dan diimplementasikan menggunakan FreeRTOS Queue pada ESP32. SensorTask bertindak sebagai *producer* yang membaca sensor pH dan suhu secara periodik serta memasukkan data ke outgoingQueue, sedangkan NetSendTask bertindak sebagai *consumer* yang menangani pengiriman HTTP. Payload yang belum berhasil dikirim ditempatkan pada retryQueue dan dipindahkan kembali oleh ResendBufferTask ketika komunikasi tersedia. Pembagian tersebut membuat proses akuisisi, komunikasi, pemeliharaan koneksi, pencatatan status, dan pemulihan *payload* dapat dijalankan sebagai *task* dengan tanggung jawab serta prioritas yang berbeda.
2. *Firmware multitasking* FreeRTOS Queue lebih mampu mempertahankan periodisitas *sampling* daripada *firmware baseline* sekuensial ketika komunikasi lebih lama daripada periode *sampling*. Pada kondisi normal, *firmware baseline* sekuensial dan FreeRTOS Queue menghasilkan rata-rata interval masing-masing sebesar 1.999,408 ms dan 2.000,001 ms sehingga keduanya masih mendekati target 2.000 ms. Ketika diberikan delay komunikasi 3.000 ms, rata-rata interval *firmware* sekuensial meningkat menjadi 3.350,998 ms, sedangkan FreeRTOS Queue tetap menghasilkan rata-rata interval sebesar 1.999,992 ms. Hasil tersebut menunjukkan bahwa pemisahan task mempertahankan kesempatan akuisisi pada waktu yang direncanakan, sehingga pengukuran tidak tertunda oleh proses komunikasi yang sedang berlangsung. Konsekuensinya, keterlambatan komunikasi dialihkan menjadi waktu tunggu dan penumpukan paket pada queue yang tetap dibatasi oleh kapasitas buffer. Dengan demikian, FreeRTOS Queue mampu mempertahankan periodisitas *producer*, tetapi tetap memerlukan pengelolaan kapasitas *buffer* ketika laju pengiriman secara berkelanjutan lebih rendah daripada laju pembentukan data.
3. Alur telemetri *end-to-end* dari ESP32 menuju backend, database, dan *dashboard* berhasil diimplementasikan dan divalidasi. Pada kondisi jaringan normal, 155 data valid seluruhnya berhasil dikirim, tersimpan dengan waktu pengukuran unik, serta dapat disediakan kembali melalui endpoint data terbaru dan historis tanpa duplikasi. Seluruh 11 skenario *dashboard* dinyatakan lolos, meliputi penampilan data terbaru, grafik historis, tabel riwayat, filter waktu, reset, pembaruan otomatis, dan penanganan *backend* yang tidak tersedia. Mekanisme *retry-buffer-resend* juga berhasil memulihkan *payload* ketika gangguan bersifat sementara serta mempertahankan periodisitas *sampling* ketika komunikasi terputus. Pada

gangguan berkepanjangan, *buffer* mencapai kapasitas maksimum dan 75 *payload* dibuang melalui kebijakan drop-oldest, sehingga mekanisme ini meningkatkan toleransi terhadap gangguan sementara tetapi tidak menjamin seluruh data dapat dipertahankan apabila gangguan berlangsung melebihi kapasitas penyimpanan.

5.2. Saran

Berdasarkan hasil dan keterbatasan penelitian, saran untuk pengembangan selanjutnya adalah sebagai berikut.

1. Penelitian selanjutnya disarankan melakukan pengujian lapangan menggunakan badan buoy, enclosure kedap air, sumber daya mandiri, router, dan modem 4G agar ketahanan fisik serta komunikasi sistem dapat dievaluasi pada kondisi perairan yang sebenarnya.
2. Mekanisme penyimpanan *payload* tertunda dapat dikembangkan menggunakan media persisten, seperti flash atau kartu SD, agar data tidak langsung hilang ketika ESP32 kehilangan daya dan kapasitas penyimpanan dapat melebihi *retryQueue* yang berada pada RAM.
3. Pengembangan selanjutnya dapat menggunakan sinkronisasi waktu jaringan, identitas sesi perangkat, atau nomor urut persisten untuk menjaga keunikan data ketika RTC tidak tersedia dan perangkat mengalami reset.
4. Pengujian *firmware* perlu dilakukan dalam durasi yang lebih panjang, menggunakan beberapa pola gangguan, tingkat kualitas jaringan, periode *sampling*, ukuran *queue*, dan jumlah perangkat untuk mengetahui batas kemampuan sistem serta menentukan konfigurasi *buffer* yang lebih sesuai.
5. Kalibrasi dan validasi sensor dapat diperluas menggunakan lebih banyak larutan acuan, alat ukur referensi, pengulangan antarhari, dan sampel air dengan kondisi berbeda agar kinerja sensor dapat dievaluasi di luar rentang *buffer* yang digunakan dalam penelitian.
6. Backend dapat dikembangkan pada server daring dengan penerapan autentikasi perangkat, komunikasi HTTPS, pengelolaan beberapa perangkat, serta pemantauan kondisi perangkat agar sistem lebih siap digunakan pada implementasi operasional.