

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil penelitian yang telah dilakukan didapatkan beberapa kesimpulan diantaranya:

1. Sistem pengaturan temperatur berhasil dirancang dan diimplementasikan menggunakan dua ESP32 dengan fungsi yang berbeda. ESP32 Arduino IDE digunakan untuk membaca sensor PT100 dan menangani antarmuka perangkat keras, sedangkan ESP32 *OpenPLC* digunakan untuk menjalankan logika pengendalian. Pembagian fungsi tersebut dilakukan karena pada konfigurasi *OpenPLC* yang digunakan, pembacaan PT100 belum dapat diterapkan secara langsung sesuai kebutuhan sistem.
2. Pertukaran data antara ESP32 *Arduino IDE* dan ESP32 *OpenPLC* dapat dilakukan melalui komunikasi *pin to pin* dengan tahapan PWM, *RC low pass filter*, ADC, dan *scaling*. Pengujian menunjukkan perubahan nilai dapat diteruskan dari kondisi minimum hingga maksimum, tetapi masih terdapat selisih dan fluktuasi nilai yang memengaruhi temperatur yang diterima *OpenPLC* sebagai *process variable*.
3. Pengujian Open Loop dengan masukan Heater 69% menghasilkan kenaikan temperatur dari 23,06 °C hingga 72,51 °C. Dari karakteristik respons diperoleh dead time 2,79 menit dan konstanta waktu 42,98 menit. Hasil tuning Ziegler–Nichols menghasilkan parameter $K_P = 40,09$, $T_N = 5,59$ menit, dan $T_V = 1,40$ menit yang kemudian digunakan pada Function Block CTRL_PID di OpenPLC. Ketika data Open Loop yang sama dianalisis pada target *setpoint* 35 °C, 45 °C, dan 55 °C, diperoleh Rise Time masing-masing 14,55 menit, 24,99 menit, dan 35,65 menit. Sepuluh menit setelah target pertama kali terlewati, Error terhadap target masing-masing sebesar 24,74%, 16,27%, dan 10,69%.
4. *OpenPLC* ESP32 dapat menjalankan logika pengendalian yang telah dirancang dengan menerima nilai *process variable*, membandingkannya dengan *setpoint*, menjalankan Function Block CTRL_PID, serta menghasilkan keluaran pengendalian. Hal ini menunjukkan bahwa dari sisi eksekusi program, logika

kontrol dapat dijalankan pada ESP32. Namun, hasil keluaran PID tetap bergantung pada kualitas nilai temperatur yang diterima *OpenPLC*.

5. Pada pengujian *Closed Loop* dengan *setpoint* 35 °C, 45 °C, dan 55 °C diperoleh *Rise Time* masing-masing 19,65 menit, 31,29 menit, dan 47,52 menit, dengan *Overshoot* 8,57%, 8,89%, dan 7,27%. *Settling time* pada ketiga pengujian belum tercapai, sedangkan *Error* akhir terhadap *setpoint* masing-masing sebesar 2 °C, 3 °C, dan 3 °C. Hasil tersebut menunjukkan temperatur dapat mencapai *setpoint*, tetapi belum dapat bertahan stabil pada pita toleransi $\pm 5\%$. Perbandingan pada target temperatur yang sama menunjukkan bahwa *Open Loop* memiliki *Rise Time* lebih cepat, yaitu 14,55 menit, 24,99 menit, dan 35,65 menit, sedangkan *Closed Loop* membutuhkan 19,65 menit, 31,29 menit, dan 47,52 menit. Namun, 10 menit setelah target tercapai, rata-rata persentase *Error* berkurang dari 17,23% pada *Open Loop* menjadi 5,34% pada *Closed Loop* atau turun sekitar 69,02%. Hasil tersebut menunjukkan bahwa PID pada penelitian ini lebih berperan dalam mengurangi penyimpangan temperatur terhadap target daripada mempercepat proses pemanasan.
6. Hasil keseluruhan penelitian menunjukkan bahwa *OpenPLC* berbasis ESP32 dapat digunakan untuk menjalankan logika PID pada sistem pengaturan temperatur yang dibuat, tetapi penerapannya masih memiliki beberapa keterbatasan. Keterbatasan yang ditemukan meliputi pembacaan sensor yang memerlukan ESP32 tambahan, adanya selisih nilai pada komunikasi pin-to-pin dan proses scaling, perbedaan kompatibilitas fungsi pada versi *OpenPLC* yang digunakan selama penelitian, serta distribusi temperatur air yang belum merata karena tangki tidak menggunakan mixing. Kondisi tersebut menyebabkan hasil pengendalian closed-loop belum dapat mempertahankan temperatur secara stabil pada nilai *setpoint*.

5.2 Saran

Berdasarkan hasil pengujian dan keterbatasan yang ditemukan pada sistem, saran untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Penelitian selanjutnya dapat menggunakan metode komunikasi lain antara perangkat pembacaan sensor dan *OpenPLC*, seperti komunikasi digital yang mampu mengirimkan data secara langsung tanpa melalui tahapan PWM, RC

Filter, ADC, dan *scaling*. Penggunaan metode komunikasi lain dapat dibandingkan dengan komunikasi *pin-to-pin* yang digunakan pada penelitian ini untuk mengetahui apakah nilai temperatur yang diterima OpenPLC dapat menjadi lebih konsisten.

2. Sistem dapat dikembangkan dengan menambahkan mekanisme *mixing* atau pengadukan pada tangki agar distribusi temperatur air lebih merata. Dengan distribusi temperatur yang lebih seragam, nilai yang dibaca sensor diharapkan lebih mewakili kondisi temperatur air secara keseluruhan.
3. Versi OpenPLC yang digunakan sebaiknya dicatat dan dipertahankan selama proses penelitian apabila sistem sudah dapat berjalan dengan baik. Apabila dilakukan pembaruan versi, fungsi dan fitur yang digunakan pada program perlu diuji kembali karena terdapat kemungkinan perubahan kompatibilitas antarversi.