

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam perkembangan teknologi industri saat ini telah menuntut adanya sistem kendali yang dapat bekerja secara otomatis, akurat, dan juga stabil dalam mengendalikan berbagai variabel proses seperti suhu, tekanan, level, dan laju aliran fluida. Salah satu parameter yang sangat penting dalam berbagai proses industri pengolahan air, makanan dan minuman, serta industri kimia adalah pengendalian laju aliran fluida, dimana parameter tersebut berpengaruh langsung terhadap kestabilan proses produksi dan efisiensi sistem secara keseluruhan [1].

Pengaturan laju aliran fluida pada suatu proses industri dapat berdampak sangat serius pada hasil hingga dapat beresiko kerusakan selama produksi berlangsung. Pengaturan laju aliran yang tidak tepat dapat menyebabkan ketidakseimbangan proses, penurunan kualitas produk, hingga potensi kerusakan pada sistem perpipaan dan peralatan industri [2]. Oleh karena itu, sistem pengendalian yang mampu menjaga laju aliran fluida sangat diperlukan agar tetap berada pada nilai yang diinginkan dan sesuai dengan nilai setpoint yang telah ditentukan.

Pengendalian proses pada sistem otomasi industri umumnya dilakukan menggunakan *Programmable Logic Controller* (PLC). PLC merupakan perangkat kontrol berbasis mikroprosesor yang dirancang untuk mengendalikan proses industri secara *real-time* dengan tingkat keandalan yang tinggi [3]. Penggunaan PLC pada sistem otomasi industri dikarenakan PLC memiliki kemampuan untuk memproses input dari berbagai sensor serta mengendalikan aktuator diantaranya motor, pompa dan katup kontrol. PLC menjadi salah satu perangkat utama yang banyak digunakan dalam sistem otomasi industri modern karena memiliki beberapa keunggulan diantaranya dapat mendukung berbagai bahasa pemrograman standar industri seperti *Ladder Diagram*, *Function Block Diagram* dan *Structured Text* [4].

Meskipun demikian, penggunaan PLC komersial masih memiliki beberapa keterbatasan, khususnya pada bidang pendidikan dan penelitian. Harga perangkat PLC yang relatif tinggi menyebabkan pengadaan media praktikum maupun simulator industri menjadi kurang ekonomis. Kondisi tersebut mendorong

berkembangnya konsep OpenPLC, yaitu perangkat lunak PLC berbasis *open-source* yang mampu menjalankan standar IEC 61131-3 pada berbagai platform perangkat keras, seperti komputer, Raspberry Pi, maupun mikrokontroler ESP32. Dengan memanfaatkan OpenPLC, sistem kendali dapat dibangun dengan biaya yang lebih rendah tanpa menghilangkan konsep dasar pemrograman PLC yang digunakan di lingkungan industri [5], [6].

Selain perangkat pengendali, pemilihan metode kontrol juga berpengaruh terhadap performa sistem. Salah satu metode yang banyak digunakan pada proses industri adalah *Proportional-Integral-Derivative* (PID) karena mampu memberikan aksi koreksi berdasarkan selisih antara setpoint dan variabel proses. Pada sistem pengendalian laju aliran air, PID digunakan untuk mengatur kecepatan pompa berdasarkan selisih antara setpoint laju aliran dan nilai laju aliran aktual. Pada penelitian ini algoritma PID diimplementasikan menggunakan *Function Block CTRL_PID* pada program *Function Block Diagram* (FBD) OpenPLC sehingga perhitungan sinyal kendali dilakukan secara siklik oleh OpenPLC *Runtime*.

Berdasarkan hasil kajian terhadap beberapa penelitian terdahulu, implementasi OpenPLC telah diterapkan pada simulator PLC, pengendalian motor, dan sistem monitoring berbasis mikrokontroler. Namun, kajian yang secara khusus mengimplementasikan OpenPLC berbasis ESP32 untuk pengendalian laju aliran air menggunakan PID serta mengevaluasi respons transien dan *steady-state* masih relatif terbatas. Oleh karena itu, penelitian ini memfokuskan kajian pada implementasi OpenPLC *Runtime* pada ESP32 sebagai platform pengendali laju aliran air skala laboratorium dengan evaluasi berdasarkan karakteristik *open-loop*, parameter awal Ziegler-Nichols, dan performa *closed-loop*.

Pada penelitian ini digunakan dua buah mikrokontroler ESP32 yang memiliki fungsi berbeda. ESP32 pertama digunakan sebagai OpenPLC *Runtime* yang menjalankan algoritma PID untuk mengendalikan laju aliran air, sedangkan ESP32 kedua digunakan untuk menangani proses pembacaan sensor serta pengendalian perangkat pendukung. Kedua mikrokontroler berkomunikasi menggunakan metode *pin-to-pin* melalui GPIO sehingga proses pertukaran data dapat dilakukan secara sederhana dengan waktu respons yang cepat. Arsitektur ini

diharapkan mampu meningkatkan fleksibilitas sistem sekaligus membagi beban pemrosesan sehingga implementasi OpenPLC dapat berjalan lebih optimal. Pembagian fungsi dilakukan agar ESP32 OpenPLC dapat difokuskan pada eksekusi program FBD dan pengendalian PID, sedangkan ESP32 pendukung menangani akuisisi sensor dan penerapan sinyal kendali pada perangkat keras.

Berdasarkan hal tersebut, penelitian ini mengusulkan implementasi sistem pengendalian laju aliran air menggunakan metode PID berbasis OpenPLC pada ESP32 sebagai platform pengendalian *open-source* untuk skala laboratorium. Selain merancang sistem pengendalian, penelitian ini mengevaluasi performa sistem berdasarkan karakteristik respons, kestabilan *process variable*, dan nilai kesalahan terhadap *setpoint*. Hasil penelitian diharapkan dapat menjadi referensi bagi pengembangan media pembelajaran otomasi industri berbasis OpenPLC serta memberikan gambaran implementasi *soft PLC* pada sistem kendali proses.

Dalam implementasi sistem kendali pada plant nyata, kemampuan pengendali dalam mengikuti nilai *setpoint* perlu dibedakan dengan akurasi sistem pengukuran. Kemampuan *tracking* menunjukkan seberapa dekat *process variable* yang digunakan oleh pengendali terhadap nilai *setpoint*, sedangkan akurasi pengukuran menunjukkan seberapa dekat nilai *process variable* terhadap laju aliran fisik yang sebenarnya. Oleh karena itu, pada penelitian ini selain dilakukan pengujian respons *closed-loop* pada *setpoint* 1000 mL/min, 2000 mL/min, dan 3000 mL/min, dilakukan pula pengujian volumetrik untuk memverifikasi kesesuaian nilai pembacaan laju aliran terhadap kondisi fisik sistem.

Sistem yang dikembangkan pada penelitian ini merupakan bagian dari suatu sistem otomasi terintegrasi yang mencakup pengendalian laju aliran, pengaturan temperatur, dan sistem komunikasi dan monitoring, dimana pada pengaturan temperatur menjadi fokus penelitian Muhammad Haikal, sedangkan sistem komunikasi menjadi fokus penelitian Afdhol Arafat.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang sudah diuraikan, terdapat beberapa rumusan masalah yakni:

1. Bagaimana implementasi OpenPLC berbasis ESP32 sebagai pengendali pada sistem laju aliran air?
2. Bagaimana karakteristik plant berdasarkan pengujian *open-loop* dan bagaimana parameter awal PID diperoleh menggunakan metode Ziegler–Nichols *open-loop*?
3. Bagaimana implementasi *Function Block* CTRL_PID pada OpenPLC berbasis ESP32 dalam mengendalikan laju aliran air?
4. Bagaimana kinerja sistem *closed-loop* pada *setpoint* 1000, 2000, dan 3000 mL/min berdasarkan parameter respons dan *tracking process variable* terhadap *setpoint*?

1.3. Tujuan

1. Mengimplementasikan OpenPLC berbasis ESP32 sebagai platform pengendali pada sistem laju aliran air.
2. Mengidentifikasi karakteristik plant melalui pengujian *open-loop* dan memperoleh parameter awal PID dengan metode Ziegler–Nichols *open-loop*.
3. Mengimplementasikan *Function Block* CTRL_PID pada OpenPLC dan mengintegrasikannya dengan sistem aktuasi pompa.
4. Mengevaluasi performa *closed-loop* pada *setpoint* 1000, 2000, dan 3000 mL/min berdasarkan *rise time*, *maximum overshoot*, *bias/steady-state error*, *MAE*, *RMSE*, serta kestabilan *process variable*.

1.4. Manfaat

1. Manfaat Akademis

Penelitian ini diharapkan dapat memberikan kontribusi terhadap pengembangan ilmu pengetahuan di bidang sistem kontrol, khususnya mengenai implementasi OpenPLC berbasis ESP32 sebagai pengendali pada sistem pengendalian laju aliran air menggunakan metode PID. Selain itu, penelitian ini diharapkan dapat menjadi referensi bagi penelitian selanjutnya yang berkaitan dengan penerapan *soft PLC* pada sistem kendali proses.

2. Manfaat Praktis

Penelitian ini diharapkan menghasilkan prototipe sistem pengendalian laju aliran air berbasis OpenPLC dan metode PID yang dapat menjadi alternatif implementasi *soft* PLC pada sistem kendali proses skala laboratorium.

3. Manfaat bagi Pengembangan Penelitian

Penelitian ini diharapkan dapat menjadi referensi bagi mahasiswa dalam mempelajari implementasi OpenPLC berbasis ESP32, metode PID, serta penerapan sistem pengendalian laju aliran air pada skala laboratorium.

1.5. Batasan Masalah

1. Sistem yang digunakan sebagai objek pengujian berupa prototipe sistem pengendalian laju aliran air skala laboratorium.
2. Platform pengendali utama yang diteliti adalah OpenPLC Runtime yang diimplementasikan pada mikrokontroler ESP32, sedangkan OpenPLC *Editor* digunakan sebagai perangkat lunak untuk pengembangan program kontrol.
3. Penelitian difokuskan pada implementasi dan evaluasi OpenPLC berbasis ESP32 dalam menjalankan fungsi pengendalian laju aliran air, meliputi pembacaan variabel proses, pengolahan kontrol, serta pemberian keluaran kendali terhadap aktuator.
4. Metode kontrol *Proportional-Integral-Derivative* (PID) digunakan sebagai metode pengendalian untuk menguji implementasi fungsi kontrol pada OpenPLC. PID tidak menjadi objek utama penelitian, tetapi digunakan sebagai algoritma kontrol yang dijalankan pada platform OpenPLC.
5. Penentuan parameter awal PID dilakukan menggunakan metode Ziegler-Nichols berdasarkan respons *open loop*, dan penelitian tidak melakukan perbandingan dengan metode tuning PID lainnya.
6. Variabel utama yang dikendalikan adalah laju aliran air (*flow rate*) dalam satuan mL/min.

7. Sensor yang digunakan untuk mengukur laju aliran adalah sensor water *flow* YF-S201, sedangkan aktuator utama berupa pompa DC yang dikendalikan menggunakan *driver* BTS7960.
8. Implementasi OpenPLC pada penelitian ini dibatasi pada fungsi yang berkaitan dengan sistem pengendalian, meliputi pemetaan *input/output*, eksekusi program FBD pada OpenPLC *Runtime*, pengolahan *process variable*, pembentukan keluaran kendali, serta implementasi Function Block CTRL_PID. Penelitian tidak membandingkan kinerja OpenPLC dengan PLC komersial atau platform kontrol lainnya.
9. Sistem menggunakan dua mikrokontroler ESP32. ESP32 OpenPLC menjalankan OpenPLC *Runtime* dan *Function Block* CTRL_PID, sedangkan ESP32 pendukung menangani akuisisi sensor dan penerapan sinyal kendali pada perangkat keras sesuai rancangan sistem.
10. Komunikasi utama antar kedua ESP32 dilakukan secara *pin-to-pin* melalui GPIO dengan representasi nilai menggunakan sinyal PWM yang dilewatkan melalui RC *low-pass filter* dan dibaca kembali melalui ADC. Penelitian tidak membandingkan metode tersebut dengan protokol komunikasi industri lainnya.
11. Evaluasi kinerja sistem dilakukan berdasarkan respons *open-loop* dan *closed-loop*. Pada pengujian *closed-loop*, parameter yang dianalisis meliputi *rise time*, *maximum overshoot*, nilai rata-rata *process variable*, bias atau *steady-state error*, standar deviasi, koefisien variasi, *Mean Absolute Error* (MAE), *Root Mean Square Error* (RMSE), serta persentase data yang berada dalam batas toleransi $\pm 5\%$ dan $\pm 10\%$.
12. Sensor JSN-SR04T tidak digunakan sebagai *process variable* PID laju aliran, tetapi sebagai variabel pendukung dan *interlock* untuk menghentikan pompa ketika level mencapai LEVEL_OK.
13. Antarmuka HMI hanya digunakan sebagai fasilitas pendukung pemberian setpoint dan monitoring dan tidak menjadi objek analisis penelitian ini.

14. Perhitungan kebutuhan daya hanya dilakukan untuk memperkirakan kapasitas catu daya. Analisis efisiensi energi, konsumsi daya jangka panjang, dan ketahanan komponen tidak menjadi objek penelitian.
15. Pengujian *closed-loop* pada penelitian ini dilakukan menggunakan tiga variasi setpoint laju aliran, yaitu 1000 mL/min, 2000 mL/min, dan 3000 mL/min. Evaluasi kinerja pengendalian dilakukan berdasarkan kemampuan *process variable* (PV) yang digunakan oleh OpenPLC dalam mengikuti nilai *setpoint*. Kesesuaian *process variable* terhadap laju aliran fisik dievaluasi secara terpisah melalui pengujian volumetrik.

1.6. Sistematika Penulisan Laporan Tugas Akhir

Dalam penulisan laporan tugas akhir ini, sistematika penyusunan laporan dijelaskan sebagai berikut:

BAB I PENDAHULUAN

Bagian ini berisi latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan laporan tugas akhir sesuai topik penelitian.

BAB II DASAR TEORI

Bagian ini berisi mengenai teori yang mendukung topik yang diangkat oleh penulis.

BAB III METODE PENELITIAN

Bagian ini berisi mengenai metode yang digunakan dalam penelitian penulis.

BAB IV PENGUJIAN DAN ANALISA

Bagian ini berisi hasil pengujian, pengolahan data, analisa performa, serta pembahasan hasil pengujian sistem yang telah dibuat.

BAB V PENUTUP

Bagian ini berisi kesimpulan yang diperoleh dari hasil penelitian serta saran untuk pengembangan sistem selanjutnya.