

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengendalian level air merupakan aspek penting dalam berbagai bidang, seperti industri, pertanian, dan sistem penyediaan air bersih. Ketidak stabilan level air dapat menyebabkan gangguan operasional, pemborosan energi, bahkan kerusakan peralatan. Oleh karena itu, diperlukan suatu sistem kontrol yang mampu menjaga level air tetap stabil sesuai dengan nilai yang diinginkan.

Kontroler *Proportional-Integral-Derivative* (PID) merupakan metode pengendalian yang banyak digunakan karena kemampuannya dalam menghasilkan respon sistem yang stabil dan akurat. Namun, sistem kontrol konvensional masih memiliki keterbatasan dalam hal pemantauan jarak jauh dan pencatatan data secara *real-time*. Perkembangan teknologi *internet of things* (IoT) memungkinkan sistem kontrol terintegrasi dengan jaringan internet sehingga data level air dapat dipantau secara *real-time* dari jarak jauh. Dengan menggabungkan kontroler PID dan teknologi IoT, diharapkan sistem memantau dan pengendalian level air menjadi lebih efektif, efisien, dan mudah diakses.

Berdasarkan latar belakang tersebut, maka dilakukan penelitian dengan judul “*Monitoring pada Kontroler PID Level Air Berbasis IoT*”.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dalam penelitian ini adalah:

1. Bagaimana merancang sistem kontrol PID untuk mengendalikan level air?
2. Bagaimana mengintegrasikan sistem kontrol PID dengan teknologi IoT untuk memantau jarak jauh?
3. Bagaimana kinerja sistem dalam menjaga kestabilan level air dan menampilkan data secara *real-time*?
4. Bagaimana akurasi sensor level air yang digunakan terhadap perubahan ketinggian air dalam tangki.

1.3 Batasan Masalah

Agar penelitian ini lebih terarah dan tidak menyimpang dari tujuan yang ingin dicapai, maka diperlukan batasan masalah sebagai berikut:

1. Merancang dan mengimplementasikan kontroler PID untuk pengendalian level air.
2. Mengembangkan sistem memantau level air berbasis IoT.
3. Menguji kinerja sistem kontrol dan memantau dalam menjaga kestabilan level air.
4. Pemantauan ini dilakukan secara *real-time* berbasis IoT, terbatas pada pengiriman dan penampilan data melalui jaringan internet.
5. Media komunikasi yang digunakan adalah WiFi, tidak membahas jaringan lain seperti LoRa, NB-IoT, atau GSM secara detail.
6. Gangguan yang diperhitungkan hanya berupa perubahan debit air, tidak mencakup gangguan mekanik atau kerusakan perangkat keras.
7. Sistem yang dibahas hanya memantau dan pengendalian level air pada satu tangki menggunakan kontroler PID.
8. Sensor yang digunakan terbatas pada sensor level air (misalnya sensor *ultrasonic waterproof*), tanpa membahas jenis sensor lain secara mendalam.

1.4 Tujuan dan Manfaat Penelitian

Tujuan yang diharapkan pada penelitian ini antara lain:

1. Merancang dan mengimplementasikan sistem memantau level air berbasis *internet of things* (IoT) yang terintegrasi dengan kontroler PID.
2. Menganalisis kinerja kontroler PID dalam menjaga kestabilan level air sesuai *setpoint* yang ditentukan.
3. Mengembangkan sistem pemantauan *real-time* level air yang dapat diakses dari jarak jauh melalui aplikasi atau *web*.
4. Mengetahui efektivitas penggunaan IoT dalam meningkatkan akurasi, *respon*, dan efisiensi pengendalian level air.
5. Menguji keandalan sistem dalam minimalkan *error*, *overshoot*, dan *fluktuasi* pada level air.

Manfaat yang diharapkan dari penelitian ini antara lain:

1. Memberikan solusi sistem kontrol level air yang stabil dan efisien.
2. Memudahkan pemantauan level air secara *real-time* dan jarak jauh.
3. Menjadi referensi bagi pengembangan sistem kontrol berbasis IoT di bidang serupa.
4. Mengembangkan sistem memantau berbasis IoT yang dapat menampilkan data level air secara *real-time* melalui jaringan internet.
5. Menguji keandalan sistem memantau IoT dalam mengirim dan menampilkan data level air secara jarak jauh.

1.5 Metode Penyelesaian Masalah

Dalam penelitian “*Monitoring Pada Kontroler PID Level Air Berbasis IoT*”, metode yang digunakan meliputi beberapa tahapan sebagai berikut:

1. Studi literatur
 - a. Beberapa penelitian memaparkan penggunaan NodeMCU ESP8266 sebagai mikrokontroler IoT untuk memantau level air, menunjukkan bagaimana ESP8266 digunakan untuk membaca sensor ultrasonik dan mengirimkan data level ke aplikasi atau *platform cloud*.
 - b. Penelitian serupa juga memaparkan desain *prototype* memantau level air berbasis ESP8266 dengan tampilan data melalui aplikasi IoT. Sistem ini mampu memantau level air pada tangki secara jarak jauh, menggunakan sensor *ultrasonic waterproof* untuk deteksi ketinggian dan koneksi WiFi untuk pengiriman data.
2. Perancangan sistem
 - a. Sistem dirancang untuk mengendalikan dan memantau ketinggian air pada sebuah tangki secara otomatis menggunakan kontroler PID, dengan ESP8266 sebagai pengendali utama dan media komunikasi IoT.
 - b. Pada sisi perangkat keras, sistem terdiri dari sensor level air, mikrokontroler ESP8266, aktuator, *driver* aktuator, dan catu daya. Sensor level air, seperti sensor ultrasonik, digunakan untuk mengukur jarak permukaan air terhadap sensor sehingga diperoleh nilai ketinggian air aktual.
3. Pemilihan dan integrasi komponen

- a. Pemilihan komponen pada sistem *monitoring* kontroler PID level air berbasis *internet of things* (IoT) menggunakan ESP8266 dilakukan dengan mempertimbangkan fungsi, kompatibilitas, dan kemudahan integrasi antar perangkat.
 - b. Mikrokontroler ESP8266 (NodeMCU) dipilih sebagai pusat kendali sistem karena telah dilengkapi dengan modul WiFi internal yang mendukung implementasi IoT.
4. Perancangan perangkat lunak (*software*)
- a. Proses utama perangkat lunak dimulai dengan inisialisasi sensor, modul WiFi, dan pengaturan parameter PID. ESP8266 secara berkala membaca nilai level air dari sensor ultrasonik, kemudian menghitung *error* antara level aktual dan nilai *setpoint* yang diinginkan.
 - b. Perancangan perangkat lunak pada sistem *monitoring* kontroler PID level air berbasis IoT menggunakan ESP8266 dirancang untuk mengintegrasikan pengolahan data sensor, pengendalian aktuator, dan pengiriman data ke *platform* IoT secara *real-time*.