

Rancang Sistem Kontrol Suhu pada Inkubasi Telur Berbasis PID

Madyono¹, Dareh Muhammad², Dedid Cahya Happyanto³
^{1,2,3}Teknik Elektronika, Jl. Raya ITS, Surabaya, Indonesia

email: madyono@pens.ac.id¹, darehspeed@gmail.com², dedid@pens.ac.id³

Abstrak - Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem kontrol suhu berbasis PID pada mesin penetas telur. Sistem ini dikembangkan untuk menjaga suhu di dalam inkubator tetap stabil pada setpoint 37,5 °C, guna meningkatkan keberhasilan penetasan telur. Hasil penelitian menunjukkan bahwa kontrol PID mampu menjaga suhu inkubator dengan deviasi yang minimal dari setpoint. Implementasi PID dengan parameter yang telah ditala menghasilkan respon sistem yang stabil dengan waktu yang optimal terhadap setpoint dan overshoot serta fluktuasi suhu yang minimal. Metode yang digunakan dalam penelitian ini meliputi perancangan perangkat keras berupa sensor suhu, aktuator pemanas berbasis lampu yang terhubung dengan dimmer AC, dan mikrokontroler sebagai unit pemroses. Selain itu, dilakukan pula penalaan parameter PID dengan menggunakan metode Trial and Error dan analisis performansi sistem melalui simulasi dan pengujian langsung. Dengan penerapan kontrol PID, sistem inkubasi dapat mempertahankan suhu yang optimal dan stabil, sehingga mendukung proses penetasan telur yang lebih efektif. Penelitian ini menunjukkan bahwa penggunaan kontrol PID dapat meningkatkan efisiensi sistem penetasan dibandingkan dengan metode konvensional.

Kata Kunci - Kontrol PID, Mesin Penetas Telur, Kontrol Suhu, Sistem Otomasi.

Abstract - This research aims to design and implement a PID-based temperature control system in an egg incubator. This system was developed to keep the temperature in the incubator stable at a setpoint of 37.5 °C, in order to increase the success rate of egg hatching. The results show that PID control is able to maintain the incubator temperature with minimal deviation from the setpoint. Implementation of PID with tuned parameters results in a stable system response with optimal time to setpoint and minimal overshoot and temperature fluctuation. The methods used in this research include hardware design in the form of temperature sensors, lamp-based heating actuators connected to AC dimmers, and microcontrollers as processing units. In addition, PID parameter tuning using the Trial and Error method and system performance analysis through simulation and direct testing are carried out. With the application of PID control, the incubation system can maintain an optimal and stable temperature, thus supporting a more effective egg hatching process. This research shows that the use of PID control can increase the efficiency of the incubation system compared to conventional methods.

Keywords - PID Control, Egg Incubator, Temperature Control, Automation System.

I. PENDAHULUAN

Unggas merupakan kelompok hewan jenis burung yang dimanfaatkan untuk diambil daging dan telurnya [1]. Secara taksonomi, unggas yang umum ditanakkan termasuk dalam ordo Galliformes (contohnya ayam dan kalkun) dan Anseriformes (contohnya bebek) [2]. Sebagai hewan ovipar atau yang berkembang biak dengan cara bertelur, unggas menghasilkan dua jenis telur, yaitu telur fertil dan infertil. Telur fertil, yang telah dibuahi oleh sel jantan, memiliki potensi untuk menetas menjadi individu baru dan disebut sebagai telur tetas. Sebaliknya, telur infertil yang tidak dibuahi hanya dapat dimanfaatkan sebagai telur konsumsi [3]. Proses penetasan telur yang berhasil bergantung pada beberapa faktor lingkungan kritis, di

antaranya adalah suhu, kelembapan, konsentrasi oksigen (O₂), konsentrasi karbon dioksida (CO₂), aliran udara, dan pemutaran telur secara berkala [4].

Untuk meningkatkan populasi unggas, diperlukan metode pengeraman telur yang efektif. Pengeraman alami oleh induk memiliki keterbatasan dalam jumlah telur yang dapat dierami. Sebagai solusi untuk penetasan skala besar, dikembangkan sebuah teknologi berupa mesin tetas atau inkubator [4]. Inkubator adalah sebuah perangkat yang dirancang untuk menciptakan dan menjaga kondisi lingkungan buatan yang optimal bagi perkembangan embrio.

Salah satu parameter terpenting dalam inkubasi adalah stabilitas suhu. Fluktuasi suhu di luar rentang ideal dapat mengakibatkan kegagalan perkembangan embrio atau bahkan kematian [4]. Seiring perkembangan teknologi otomasi, sistem kendali *Proportional-Integral-Derivative* (PID) menawarkan solusi yang efektif untuk menjaga stabilitas suhu secara presisi. Kontroler PID mampu memberikan respons yang cepat dan akurat terhadap perubahan suhu, sehingga memastikan lingkungan inkubasi tetap berada pada kondisi optimal [5]. Implementasi sistem kontrol suhu berbasis PID pada inkubator diharapkan dapat meningkatkan laju keberhasilan penetasan dan kualitas anakan yang dihasilkan.

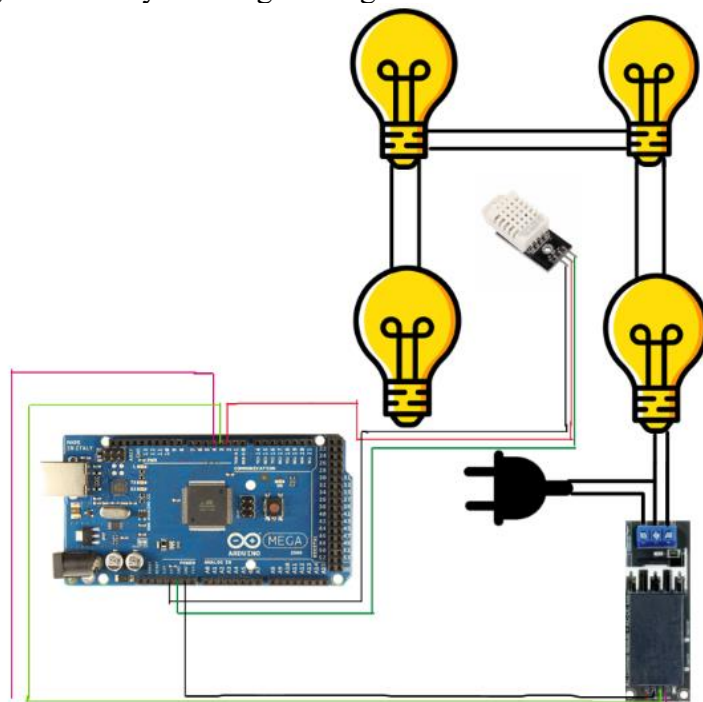
Tujuan dari penelitian ini adalah merancang sistem kontrol suhu otomatis berbasis PID untuk inkubasi telur. Penggunaan sistem ini dengan tujuan agar mengevaluasi kinerja sistem kontrol PID dalam menjaga kestabilan suhu inkubasi telur dan menghasilkan pertumbuhan embrio yang baik.

II. METODE

Metodelogi pada penelitian ini meliputi:

A. Konfigurasi Sistem

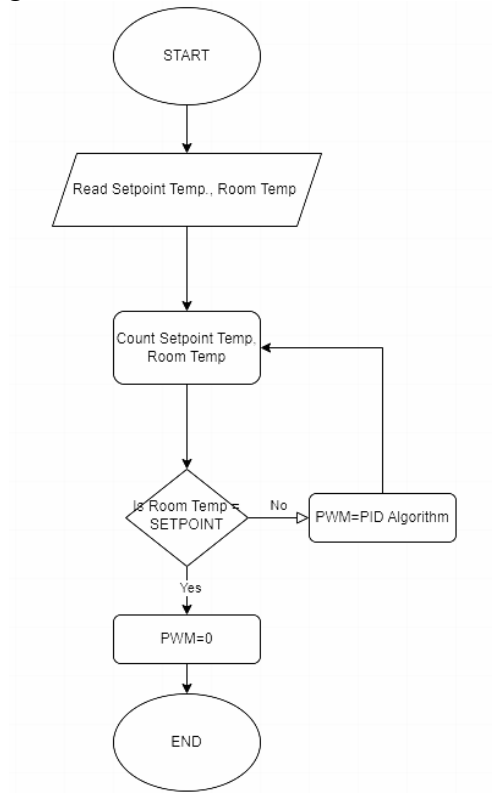
Dari konfigurasi pada gambar dibawah dapat dilihat terdapat empat lampu yang dirangkai secara paralel, kemudian dihubungkan ke *dimmer*. Untuk menyalakan *dimmer* alat harus menghubungkan kabel *power* dahulu. Segala pin yang berhubungan dengan pengumpulan data seperti pin VCC, data, GND, PSM dan Zero-Cross akan dihubungkan langsung ke mikrokontroler dengan alamatnya masing-masing.



Gambar 1. Konfigurasi Sistem

B. Algoritma PID

Berikut *flowchart* pada algoritma PID:



Gambar 2. Algoritma PID

Dari *flowchart* di atas disimpulkan bagaimana kerja mesin inkubator telur. Pertama alat tersebut akan membaca *setpoint* yang ditentukan oleh pengguna dengan suhu ruangan inkubator. Lalu proses perhitungan akan dimulai sampai mencapai *setpoint* tertentu. Jika suhu ruangan sudah mencapai yang diatur tersebut maka PWM pada *dimmer* lampu akan bernilai 0, tidak ada sinyal yang akan diatur. Jika belum mencapai *setpoint* maka PWM akan menyala sesuai dengan perhitungan PID untuk mencapai dan menstabilkan *setpoint* suhu. Program tersebut akan dijalankan melalui Arduino dan hasil keluarannya akan menuju ke *dimmer*, lalu akan diterapkan ke lampu.

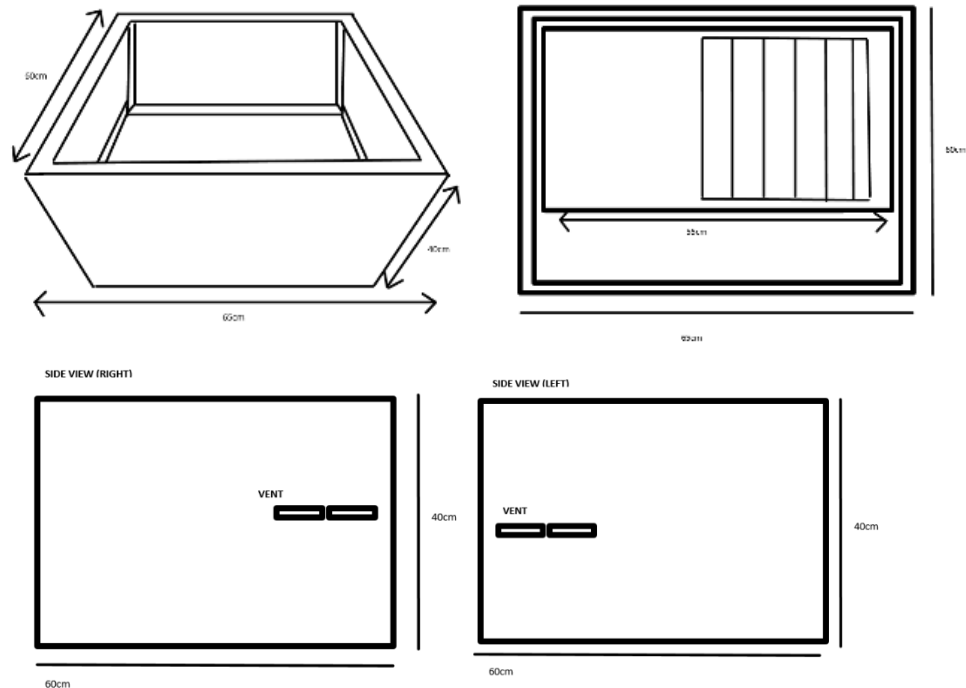
Kontrol PID bekerja dengan menghitung kesalahan antara nilai yang diinginkan (*setpoint*) dan nilai aktual dari suatu proses. Kontrol ini kemudian mengoreksi kesalahan tersebut menggunakan tiga aksi kontrol: Proporsional (P), Integral (I), dan Derivatif (D) [5]. PID yang digunakan dalam sistem menggunakan rumus berikut:

$$mv(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (1)$$

Di mana $mv(t)$ adalah *output* dari pengontrol PID atau *Manipulator Variable*, K_p adalah konstanta proporsional, T_i adalah konstanta integral, T_d adalah konstanta derivatif, dan $e(t)$ adalah selisih kesalahan antara titik setel dan level aktual.

C. Perancangan Wadah Pengetesan

Pada gambar dibawah dapat dilihat desain wadah telur yang akan dibuat. Hal ini akan memudahkan pengujian inkubator. Untuk bahannya, digunakan triplek sebagai wadah etsa dan mengecatnya dengan warna putih. Desain tersebut memiliki ukuran panjang 65 cm, lebar 60 cm dan tinggi 40 cm. Selain itu, terdapat wadah telur dengan ukuran 40 cm x 30 cm.



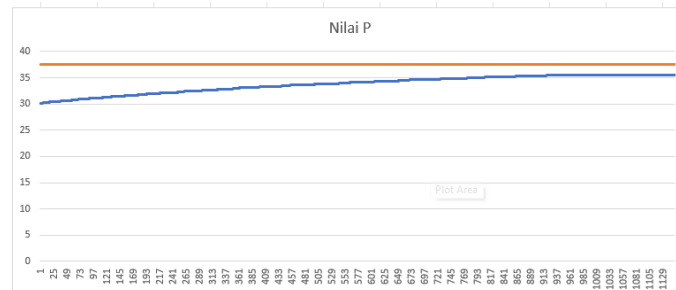
Gambar 3. Desain Wadah Pengetesan

D. Penentuan Parameter P , I dan D

Penentuan parameter PID (K_p , K_i , K_d) pada penelitian ini tidak menggunakan metode teoretis atau analitis, melainkan menggunakan metode penyetelan eksperimental secara manual (*manual tuning method*). Metode ini dipilih karena implementasi kode pada kelas *PIDController* menyediakan fungsi-fungsi setter (*setKp*, *setKi*, *setKd*) yang memungkinkan perubahan nilai konstanta PID secara langsung pada saat sistem sedang berjalan. Pendekatan ini bersifat iteratif dan didasarkan pada observasi respons sistem terhadap perubahan nilai parameter.

Adapun beberapa prosedur penyetelan manual yang dilakukan mengikuti langkah-langkah sistematis sebagai berikut:

1. Inisiasi Parameter: Pada tahap awal, nilai konstanta proporsional (K_p), integral (K_i), dan derivatif (K_d) diatur ke nilai nol. Hal ini dilakukan untuk memastikan bahwa pada mulanya tidak ada aksi kontrol yang diberikan oleh kontroler PID.
2. Penyetelan Konstanta Proporsional: Pertama, nilai K_i dan K_d dipertahankan pada nilai nol. Lalu, nilai K_p dinaikkan secara bertahap dari nol hingga sistem mulai menunjukkan respons osilasi yang stabil dan konstan terhadap suatu *setpoint* yang diberikan. Kemudian, pengamatan difokuskan pada kecepatan respons sistem (*rise time*) dan kestabilannya. Nilai K_p akan terus ditingkatkan hingga respons mencapai *setpoint* dengan cepat, namun tanpa osilasi yang berlebihan atau ketidakstabilan. Nilai K_p di mana sistem berosilasi secara stabil dicatat sebagai $K_{p\text{ kritis}}$ ($K_{p\text{cr}}$).



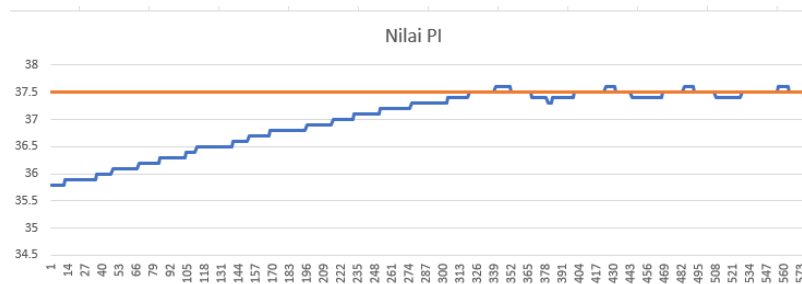
Gambar 4. Pengetesan Nilai P

Dari grafik tersebut dengan nilai $K_p = 0,5$, bisa dilihat bahwa suhu akhir yang dicapai oleh sistem tidak menyentuh nilai *setpoint*. Suhu maksimum yang tercapai dan cenderung stabil (kondisi *steady-state*) adalah sekitar $35,5\text{ }^{\circ}\text{C}$. Hal ini menunjukkan adanya *error steady-state*, yaitu selisih antara nilai *setpoint* dengan nilai aktual yang dicapai sistem. Perhitungan *error steady-state* adalah sebagai berikut:

- Setpoint* (SP) = $37,5\text{ }^{\circ}\text{C}$
- Nilai Proses (PV) = $35,5\text{ }^{\circ}\text{C}$
- Error* (e) = $\text{SP} - \text{PV} = 37,5\text{ }^{\circ}\text{C} - 35,5\text{ }^{\circ}\text{C} = 2,0\text{ }^{\circ}\text{C}$

Adanya *error steady-state* sebesar $2,0\text{ }^{\circ}\text{C}$ ini merupakan karakteristik dari penggunaan kontroler proporsional murni. Kontroler P menghasilkan sinyal keluaran yang sebanding dengan besarnya *error*. Ketika *error* mengecil, sinyal keluaran dari kontroler juga akan mengecil hingga pada satu titik tidak cukup kuat untuk mendorong sistem mencapai *setpoint*. Untuk menghilangkan *error* ini dan mencapai *setpoint* yang diinginkan, diperlukan penyesuaian lebih lanjut, dengan menambahkan komponen Integral (I) pada kontroler.

- Penyetelan Konstanta Integral: Setelah nilai K_p yang optimal diperoleh (biasanya sedikit di bawah nilai yang menyebabkan osilasi), nilai K_i dinaikkan secara bertahap. Tujuan dari penambahan komponen integral adalah untuk menghilangkan *steady-state error*, yaitu selisih antara *setpoint* dengan nilai keluaran sistem setelah kondisi stabil tercapai. Nilai K_i dinaikkan perlahan sambil mengamati respons sistem. Nilai K_i yang terlalu besar dapat menyebabkan *overshoot* (nilai keluaran melebihi *setpoint*) dan osilasi, bahkan hingga ketidakstabilan. Oleh karena itu, nilai K_i yang dipilih adalah nilai terkecil yang efektif untuk mengoreksi *steady-state error* dalam rentang waktu yang dapat diterima.

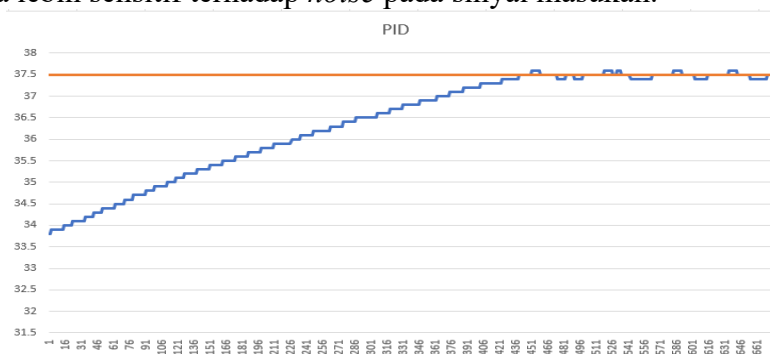


Gambar 5. Pengetesan Nilai P dan I

Dari grafik tersebut dengan nilai $K_p = 0,5$ dan $K_i = 0,1$, dapat diamati bahwa *error steady-state* yang sebelumnya ada pada kontroler P telah berhasil dieliminasi. Nilai proses mampu mencapai *setpoint* $37,5\text{ }^{\circ}\text{C}$. Fungsi dari aksi Integral adalah mengakumulasi *error* dari waktu ke waktu. Akumulasi ini akan terus meningkatkan *output* kontroler hingga *error* menjadi nol, sehingga memaksa sistem mencapai targetnya.

Namun, respons sistem menunjukkan adanya osilasi di sekitar nilai *setpoint* setelah target tercapai. Suhu terlihat sedikit berfluktuasi naik-turun di sekitar 37,5 °C. Osilasi ini menandakan bahwa meskipun *error steady-state* hilang, sistem menjadi kurang stabil atau memiliki redaman yang buruk. Hal ini merupakan efek samping yang umum terjadi jika nilai K_i terlalu besar atau tidak diimbangi dengan parameter lainnya secara tepat.

4. **Penyetelan Konstanta Derivatif:** Setelah K_p dan K_i diatur, nilai K_d dinaikkan secara bertahap untuk meningkatkan stabilitas sistem dan mengurangi *overshoot*. Komponen derivatif bekerja dengan cara memprediksi *error* di masa depan berdasarkan laju perubahannya saat ini. Aksi ini memberikan efek redaman (*damping*) pada sistem. Nilai K_d dinaikkan perlahan hingga *overshoot* berkurang ke tingkat yang dapat diterima dan sistem mencapai *setpoint* dengan lebih mulus tanpa osilasi berlebih. Perlu diperhatikan bahwa nilai K_d yang terlalu besar dapat memperlambat respons sistem dan membuatnya lebih sensitif terhadap *noise* pada sinyal masukan.



Gambar 6. Pengetesan Nilai P, I dan D

Dari grafik tersebut dengan nilai $K_p = 0,5$, $K_i = 0,1$ dan $K_d = 0,8$, terlihat bahwa sistem berhasil mencapai *setpoint* 37,5 °C, yang mengonfirmasi bahwa aksi integral (I) tetap efektif menghilangkan *error steady-state*. Lebih penting lagi, osilasi yang sebelumnya muncul pada respons kontroler PI kini telah berhasil diredam. Respons sistem menjadi jauh lebih stabil ketika berada di sekitar nilai *setpoint*.

Aksi derivative (D) bekerja dengan cara mengantisipasi *error* di masa depan berdasarkan laju perubahan *error* saat ini. Ketika suhu mendekati *setpoint* dengan cepat, komponen D akan memberikan aksi pengereman untuk mencegah terjadinya *overshoot* dan osilasi. Hasilnya adalah transisi yang lebih mulus dan stabil menuju kondisi *steady-state*. Fluktuasi kecil yang tersisa masih dapat diamati, namun amplitudonya jauh lebih kecil dan berada dalam rentang yang dapat diterima untuk sebagian besar aplikasi.

III. HASIL DAN PEMBAHASAN

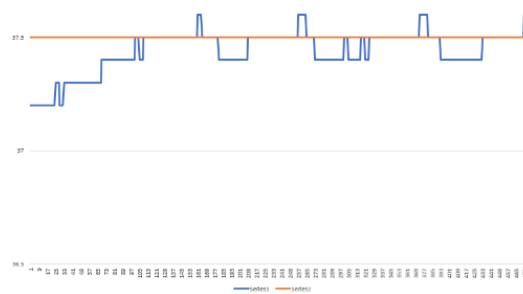
Bagian ini mengevaluasi kinerja inkubator telur yang mana pengaturan PID yang terbaik dan bagaimana PID dapat mempertahankan suhu dan bagaimana setiap pengaturan dapat memiliki hasil yang berbeda.

A. Hasil Setting PID

Tujuan dari pengujian ini adalah untuk memeriksa pengaturan PID mana yang paling efektif untuk mengendalikan suhu, untuk mengamati bagaimana algoritma PID (Proporsional, Integral, Derivatif) merespons perubahan suhu untuk mencapai *setpoint* dan memastikan

pengaturan PID mana yang memiliki fluktuasi yang baik dan seimbang. Pengujian akan dicoba berulang-ulang selama 5 menit dan akan diatur berulang-ulang dengan mengkonfigurasi nilai K_p , K_i dan K_d hingga mencapai *setpoint* dengan kestabilan yang baik.

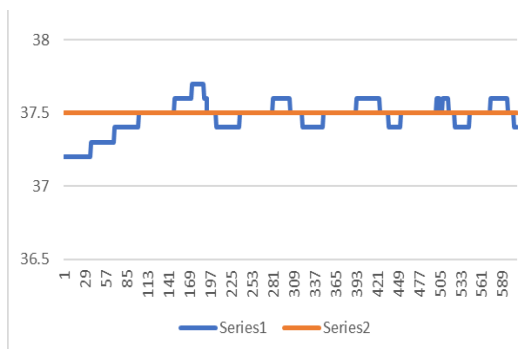
1. $K_p = 0,5$; $K_i = 0,1$; $K_d = 0,8$



Gambar 7. Hasil Pengetesan pada $K_p = 0,5$; $K_i = 0,1$; $K_d = 0,8$

Pada *setting* tersebut, terlihat memiliki respons yang paling lambat mencapai *setpoint* (*rise time* lama). Setelah mendekati *setpoint*, fluktuasi atau osilasinya terlihat kecil dan cenderung stabil (mendekati *critically damped* atau *overdamped*), namun proses pencapaian *setpoint* awalnya lambat.

2. $K_p = 1,0$; $K_i = 0,2$; $K_d = 1,6$



Gambar 8. Hasil Pengetesan pada $K_p = 1,0$; $K_i = 0,2$; $K_d = 1,6$

Pada grafik tersebut, terlihat memiliki respon yang lebih cepat mencapai *setpoint* dibandingkan grafik pada Gambar 7 (*rise time* lebih pendek). Pada grafik tersebut, terlihat ada sedikit *overshoot* (nilai melewati *setpoint* sesaat), namun memiliki fluktuasi yang moderat di sekitar *setpoint*. Osilasi lebih terlihat dibandingkan grafik pada Gambar 3, namun tidak berlebihan. Respons ini cenderung *underdamped* tetapi terkontrol.

3. $K_p = 2,0$; $K_i = 0,4$; $K_d = 3,2$



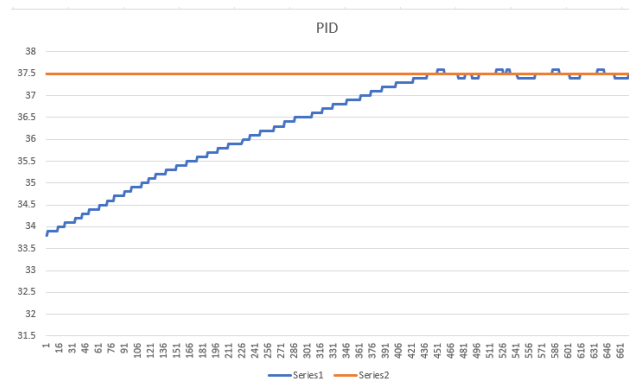
Gambar 9. Hasil Pengetesan pada $K_p = 2,0$; $K_i = 0,4$; $K_d = 3,2$

Pada grafik tersebut, terlihat memiliki respons *rise time* yang paling cepat diantara dari grafik yang lain, namun memiliki *overshoot* yang signifikan dan berulang. Terlihat terjadi fluktuasi yang besar dan osilasi yang terus-menerus (*persistent oscillation*) di sekitar *setpoint*. Sistem tampak kesulitan untuk stabil dan cenderung tidak stabil (*highly underdamped* atau bahkan menuju *unstable*).

Oleh karena itu, grafik pada Gambar 8 ($K_p=1,0$, $K_i=0,2$, $K_d=1,6$) menunjukkan fluktuasi yang paling bagus karena mencapai keseimbangan terbaik antara kecepatan respons, *overshoot*, dan stabilitas (osilasi) dibandingkan dua grafik lainnya. Tetapi dalam kasus penelitian ini, grafik pada Gambar 7 bisa menjadi pilihan yang cocok karena sistem pada inkubator memiliki performa kenaikan suhu yang relatif lambat dan lebih cocok untuk pemakaian jangka lama.

B. Hasil Stabilitas PID

Pengujian ini dilakukan untuk memeriksa seberapa stabil suhu yang dihasilkan dalam jangka waktu lama. Percobaan ini akan dilakukan selama 30 menit. Berikut hasil pada grafik pada percobaan berikut:



Gambar 10. Hasil Pengujian Stabilitas PID

Grafik yang ditampilkan menunjukkan performa kontrol PID untuk menjaga suhu pada *setpoint* 37,5 °C. Setelah mencapai *setpoint*, sistem PID dapat mempertahankan suhu di sekitar 37,5 derajat dengan cukup baik. Hal ini terlihat dari fluktuasi suhu yang relatif kecil di sekitar *setpoint*. Setelah mencapai *setpoint*, terdapat sedikit osilasi kecil di sekitar 37,5 °C, namun masih dalam kondisi yang bisa diterima dalam aplikasi yang umum. Grafik menunjukkan bahwa sistem berhasil mencapai stabilitas setelah beberapa iterasi, tanpa osilasi besar atau ketidakstabilan. Ini menandakan kombinasi parameter PID sudah cukup efektif untuk menjaga suhu dalam batas yang diinginkan. Dengan sedikit penyempurnaan tuning parameter PID akan menghasilkan hasil yang lebih stabil dengan *error* yang sangat minimal bahkan tidak ada *error* sama sekali.



Gambar 11. Dokumentasi Pengujian pada Alat

IV. KESIMPULAN

Berdasarkan analisa yang telah dilakukan, dapat disimpulkan bahwa sistem PID yang digunakan untuk mengontrol suhu pada grafik tersebut memiliki performa yang baik. Secara keseluruhan, grafik mengindikasikan bahwa sistem PID yang digunakan dapat mengontrol suhu dengan baik dan mempertahankan *setpoint* yang diinginkan. Pengaturan parameter PID yang tepat telah menghasilkan performa sistem yang efektif dan stabil.

UCAPAN TERIMA KASIH

Terima kasih kepada Bunda dan Ayah yang telah memberikan dukungan dan mendoakan, dosen pembimbing dan penguji proyek akhir Tim *Jurnal Elektro Polbeng* yang telah meluangkan waktu untuk membuat template ini dan pembaca yang telah membaca penelitian ini.

REFERENSI

- [1] I. Pazerangi and Tahang, "Rancang Bangun Mesin Penetas Telur (Inkubasi) Dengan Teknologi Tepat Guna (TTG) Yang Ekonomis Dan Serbaguna," *J. Techno Entrep. Acta*, vol. 4, no. 1, pp. 23–29, 2019.
- [2] M. Rasyaf, *Beternak Ayam Kampung*. Jakarta: Penebar Swadaya, 2011.
- [3] E. Suprijatna, U. Atmomarsono, and R. Kartasudjana, *Ilmu Dasar Ternak Unggas*. Jakarta: Penebar Swadaya, 2005.
- [4] D. Sudrajat, *Petunjuk Praktis Menetas Telur*. Jakarta Selatan: Agromedia Pustaka, 2003.
- [5] W. Bolton, *Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering (6th ed.)*. London: Pearson, 2015.