Jurnal JEETech: Vol. 6 No. 2 Tahun 2025
Journal Of Electrical Engineering And Technology
https://ejournal.ft-undar.ac.id/index.php/jeetech
DOI: https://doi.org/10.32492/jeetech.v6i2.62005
e-ISSN: 2722-5321
p-ISSN: 2964-7320

Sistem Kendali Kecepatan Turbin Menggunakan PID Kontrol Berbasis Arduino

[1]Alfa Crysto Anugrah, [2] Widodo Pudji Muljanto, [3] Aryuanto Soetedjo
[1,2,3] Teknik Elektro S1, Institut Teknologi Nasional Malang, Malang Indonesia
[1] alfa.crysto.45@gmail.com, [2] widodo_pm@lecturer.itn.ac.id, [3] aryuanto@lecturer.itn.ac.id

Article Info

Article history:

Received: 24 February 2025
Revised: 26 April 2025
Accepted: 11 November 2025

Keyword:

PID control
turbine speed
tuning
arduino
automatic control

ABSTRACT

Turbine speed control plays a crucial role in maintaining the stability and efficiency of power generation systems. This study develops a turbine speed control system utilizing a Proportional-Integral-Derivative (PID) control method implemented on an Arduino platform, with the tuning process conducted through a trial-and-error approach. This method is chosen due to its simplicity in gradually adjusting the Kp, Ki, and Kd parameters based on the system's response to turbine speed variations. The tuning process begins by determining the initial PID parameter values, followed by iterative testing and adjustments until optimal performance is achieved. The system is evaluated by analyzing key performance metrics, including delay time, rise time, settling time, overshoot, and steady-state error. Experimental results demonstrate that trial-and-error tuning enables the system to achieve a faster response with minimal error, although the adjustment process requires time and repeated testing to attain the desired stability. With Kd set to 0.0005, Kp to 0.04, and Ki to 0.005, the system exhibits improved performance, characterized by reduced overshoot and faster stabilization. Experimental tests recorded a rise time of 2 seconds, a settling time of 17 seconds, and an overshoot of 115%.

Corresponding Author:
Alfa Crysto Anugrah,
Electrical Engineering, National Institute of Technology Malang,
Karanglo Highway, Kilometer 2, Tasikmadu, Lowokwaru District, Malang City, East Java.
Email: alfa.crysto.45@ gmail.com

Abstrak— Kontrol kecepatan turbin memainkan peran krusial dalam menjaga stabilitas dan efisiensi sistem pembangkitan energi. Penelitian ini mengembangkan sistem pengendalian kecepatan turbin menggunakan metode Proportional-Integral-Derivative (PID) berbasis Arduino, dengan proses tuning yang dilakukan melalui pendekatan trial-and-error. Metode ini dipilih karena kemudahannya dalam menyesuaikan parameter Kp, Ki, dan Kd secara bertahap berdasarkan respons sistem terhadap perubahan kecepatan turbin. Tahapan tuning diawali dengan penentuan nilai awal parameter PID, kemudian dilakukan pengujian serta penyesuaian secara bertahap hingga diperoleh kinerja optimal. Evaluasi dilakukan dengan menganalisis karakteristik sistem, termasuk waktu tunda (delay time), waktu naik (rise time), waktu tunak (settling time), overshoot, dan error steady-state. Hasil pengujian menunjukkan bahwa metode tuning trial-and-error memungkinkan sistem mencapai respons yang lebih cepat dengan tingkat kesalahan minimal, meskipun memerlukan waktu serta pengujian berulang untuk memperoleh kestabilan yang diinginkan. Dengan nilai Kd sebesar 0,0005, Kp sebesar 0,04, dan Ki sebesar 0,005, sistem menunjukkan performa yang lebih baik, ditandai dengan overshoot yang lebih kecil serta waktu stabilisasi yang lebih cepat. Pengujian juga mencatat bahwa sistem memiliki rise time sebesar 2 detik, settling time 17 detik, dan overshoot sebesar 115%.

Pendahuluan

Pengendalian kecepatan turbin sangat penting untuk menjaga stabilitas dan efisiensi sistem pembangkitan energi. Ketidakstabilan kecepatan dapat menyebabkan fluktuasi daya, mengurangi efisiensi, dan berisiko merusak komponen. Oleh karena itu, diperlukan sistem kendali yang efektif, seperti kontrol Proportional-Integral-Derivative (PID), yang mampu mengoreksi kesalahan dan meningkatkan stabilitas dengan menyesuaikan parameter Kp, Ki, dan Kd[1] [2] [3].

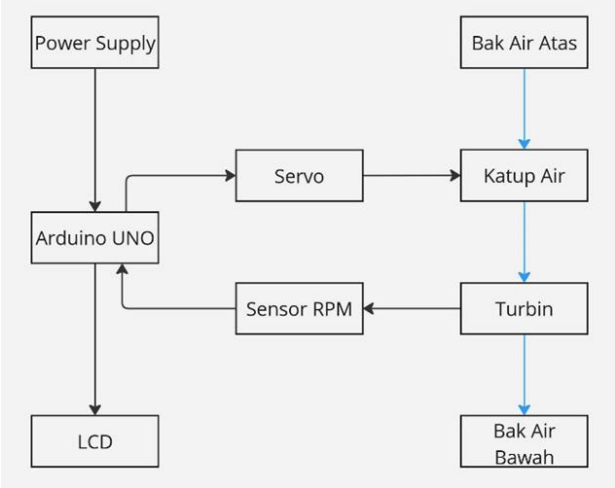
Proses tuning PID menjadi faktor kunci dalam menentukan performa sistem. Metode trial-and-error sering digunakan karena kesederhanaannya, dengan penyesuaian bertahap berdasarkan respons sistem terhadap perubahan kecepatan turbin[4] [5]. Evaluasi dilakukan dengan mengamati karakteristik sistem seperti delay time, rise time, settling time, overshoot, dan steady-state error. Meskipun memerlukan waktu dan pengujian berulang, metode ini tetap efektif, terutama dalam sistem berbasis Arduino yang fleksibel dan memungkinkan pemantauan serta penyesuaian real-time[6] [7] [8].

Arduino dipilih karena kemudahan integrasi dengan sensor dan aktuator serta biayanya yang lebih terjangkau dibandingkan sistem kendali industri. Dengan tuning PID yang tepat, sistem ini dapat meningkatkan efisiensi operasi dan kestabilan turbin[9] [10]. Penelitian ini bertujuan untuk mengembangkan sistem kendali kecepatan turbin berbasis Arduino dengan metode tuning trial-and-error, yang dapat diterapkan dalam berbagai sektor industri yang membutuhkan regulasi kecepatan presisi[9][11] [12].

Penelitian sebelumnya menunjukkan efektivitas PID berbasis Arduino dalam berbagai aplikasi, seperti pengendalian kecepatan motor DC dan integrasi dengan LabVIEW untuk pemantauan real-time[13]. Namun, tuning optimal sering kali masih mengandalkan trial-and-error, terutama dalam sistem yang kompleks. Dengan mempertimbangkan hal ini, penelitian ini berfokus pada pengembangan sistem kendali kecepatan turbin yang andal dan aplikatif dalam industri[14] [15].

I. Metode Penelitian

A. Blok Diagram Alat



Gambar 1. Blok Diagram Alat

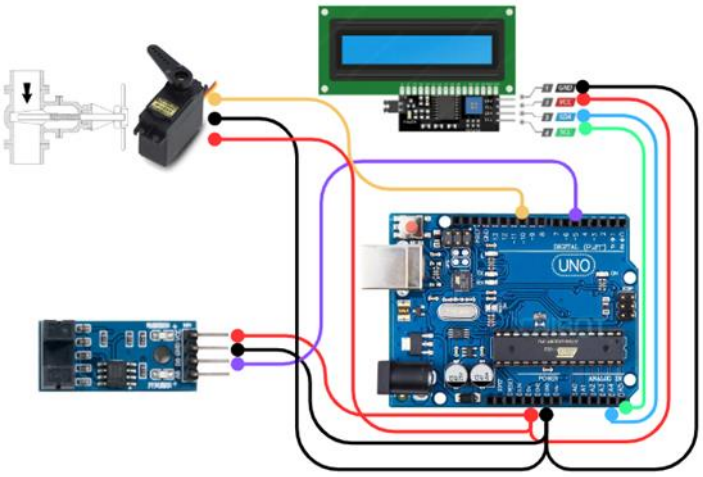
Blok diagram pada gambar 1 menggambarkan skema alur sistem kontrol berbasis Arduino UNO untuk mengatur aliran air pada turbin air menggunakan sensor dan aktuator. Sistem ini diawali dengan sumber daya listrik yang menyuplai energi ke Arduino UNO, servo motor, dan sensor RPM. Arduino bertindak sebagai pengendali utama, memproses data sensor dan mengatur aktuator untuk menjaga stabilitas sistem.

Air dari bak atas dialirkan ke turbin melalui katup yang dikendalikan oleh servo motor. Sensor RPM membaca kecepatan putaran turbin dan mengirimkan data ke Arduino. Jika kecepatan tidak sesuai, Arduino menyesuaikan posisi servo untuk membuka atau menutup katup, sehingga aliran air dapat dikontrol demi

menjaga stabilitas kecepatan turbin. Air yang telah melewati turbin dikumpulkan di bak bawah untuk didaur ulang, menjadikan sistem ini ideal untuk simulasi atau edukasi energi terbarukan.

Sistem ini juga dilengkapi layar LCD yang menampilkan data seperti RPM turbin dan status sistem secara real-time, memudahkan pemantauan dan pengoperasian. Dengan kendali berbasis Arduino, sistem menjadi fleksibel dan dapat disesuaikan untuk berbagai kebutuhan. Implementasi ini menunjukkan efisiensi teknologi kontrol otomatis dalam pengelolaan energi air yang ramah lingkungan..

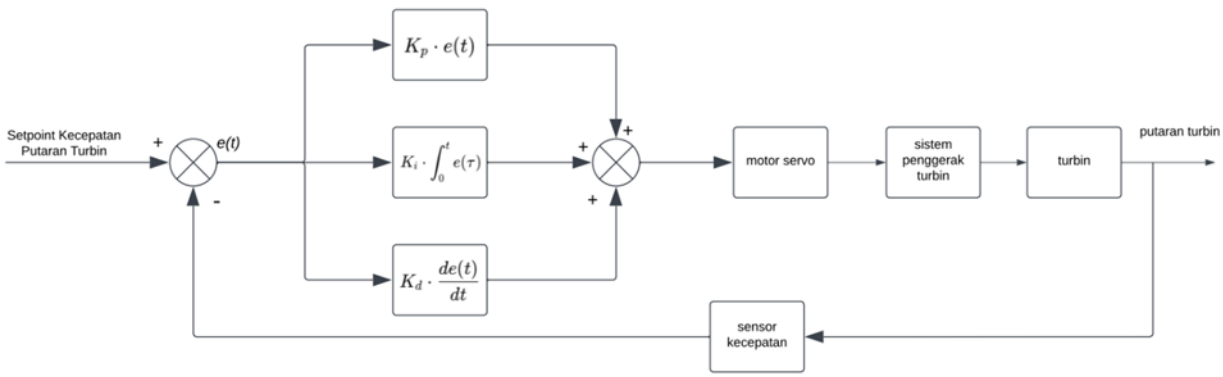
B. Wiring Alat



Gambar 2. Wiring Alat

Wiring diagram pada gambar 2 menunjukkan integrasi komponen utama dalam sistem kendali kecepatan turbin berbasis Arduino, di mana Arduino Uno sebagai pengendali utama menghubungkan servo motor untuk mengatur katup aliran air, sensor kecepatan untuk membaca putaran turbin dan menghitung error dalam algoritma PID, serta LCD melalui modul I2C untuk menampilkan informasi sistem secara real-time, dengan tambahan resistor dan modul pendukung untuk stabilitas koneksi, serta sumber daya yang disuplai melalui USB atau adaptor eksternal guna memastikan semua komponen berfungsi dengan tegangan yang sama, sehingga sistem dapat beroperasi secara optimal dengan hubungan input (sensor) dan output (servo) yang terintegrasi secara logis dan fisik.

C. Blok Diagram Sistem Kedali

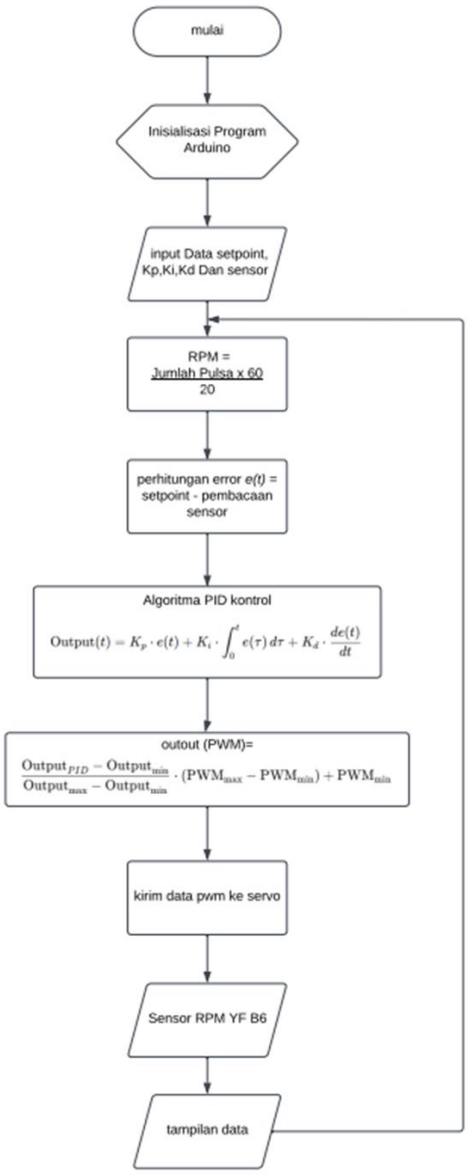


Gambar 3. Blok Diagram Sistem Kendali

Blok diagram sistem kendali pada gambar 3 menunjukkan mekanisme kontrol loop tertutup pada sistem kendali kecepatan turbin menggunakan algoritma PID. Proses dimulai dengan menetapkan setpoint kecepatan turbin, yang dibandingkan dengan kecepatan aktual dari sensor. Selisihnya menghasilkan error yang diproses melalui tiga komponen PID: Proportional (P) yang merespons error saat ini, Integral (I) yang mengakumulasi error dari waktu ke waktu, dan Derivative (D) yang mengantisipasi perubahan error untuk meningkatkan stabilitas.

Hasil dari ketiga komponen dijumlahkan menjadi sinyal kontrol yang dikirim ke servo motor untuk mengatur bukaan katup dan mengendalikan aliran air ke turbin. Kecepatan aktual kemudian diukur kembali sebagai umpan balik untuk terus menyesuaikan kontrol. Proses ini berlangsung secara otomatis dalam loop tertutup, menjaga kecepatan turbin tetap stabil meskipun terjadi perubahan beban atau gangguan, sehingga sistem beroperasi lebih efisien dan stabil.

D. Flowchart Perancangan Software



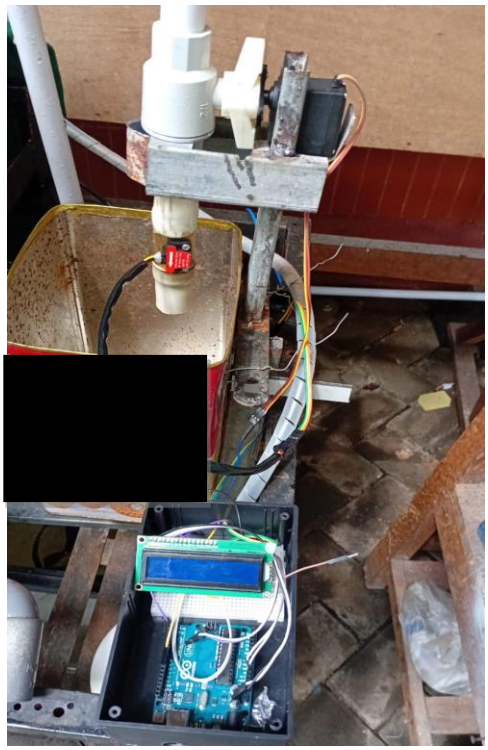
Gambar 4. Flowchat Program

Flowchart program pada gambar 4 menggambarkan alur kerja program Arduino dalam mengontrol kecepatan motor menggunakan algoritma PID. Proses dimulai dengan inisialisasi perangkat keras dan variabel, kemudian pengguna memasukkan setpoint kecepatan motor serta parameter PID (Kp, Ki, Kd). Sensor RPM membaca kecepatan motor dan menghitung error sebagai selisih dari setpoint.

Algoritma PID memproses error melalui tiga komponen: proportional (merespons error saat ini), integral (mengakumulasi error), dan derivative (mengantisipasi perubahan error). Hasil perhitungan dikonversi menjadi sinyal PWM yang dikirim ke motor driver untuk menyesuaikan kecepatan motor. Proses ini berlangsung secara terus-menerus dalam loop, di mana sensor terus memperbarui data RPM dan sistem secara otomatis menyesuaikan kecepatan agar tetap stabil sesuai setpoint, meskipun ada gangguan atau perubahan beban.

II. Hasil dan Pembahasan

A. Perancangan Perangkat Keras



Gambar 5. Perangkat Keras

Pada gambar 5 menunjukan hasil perancangan sistem kontrol otomatis yang mengatur aliran air dalam pipa untuk menjaga kecepatan turbin tetap stabil. Air mengalir menuju turbin sebagai sumber energi mekanis, dengan katup yang dikendalikan oleh servo motor untuk mengatur debit air. Sensor kecepatan membaca putaran turbin secara real-time dan mengirim data ke Arduino sebagai pengontrol utama. Menggunakan algoritma PID, Arduino menghitung error antara kecepatan aktual dan target, lalu menyesuaikan bukaan katup melalui servo motor. Dengan pendekatan loop tertutup, sistem ini secara otomatis memastikan turbin beroperasi dengan stabil dan presisi.

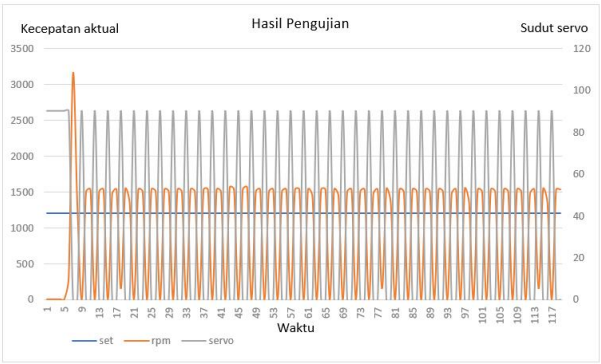
B. Pengujian Trial dan Error Tuning PID pada Hardware

1) Hasil Pengujian Nilai Kp

Tabel 1. Pengujian nilai Kp

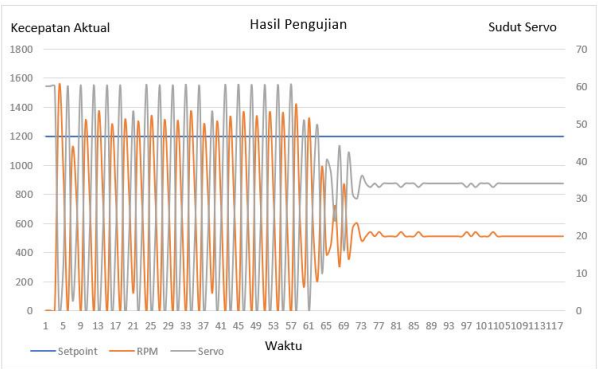
No.	Nilai Tuning			Respon			
	Kp	Ki	Kd	Tr(s)	Ts(s)	Ov(%)	Error %
1	0,1	0	0	2	-	162,5	-
2	0,04	0	0	2	73	27,5	-57,,5
3	0,02	0	0	2	9	-47	-82,5

Pada tabel 1 telah dilakukan percobaan nilai Kp dengan trial error 3 percobaan untuk menentukan nilai Kp yang tepat yang selanjutnya di masukan ke percobaan berikutnya menggunakan nilai ki. Pada percobaan ini menggunakan nilai kp 0.1 , 0.05 dan 0.03 unuk mengetahui respon sistem dan hasil kontrol dengan nilai setpoint 1200 rpm.



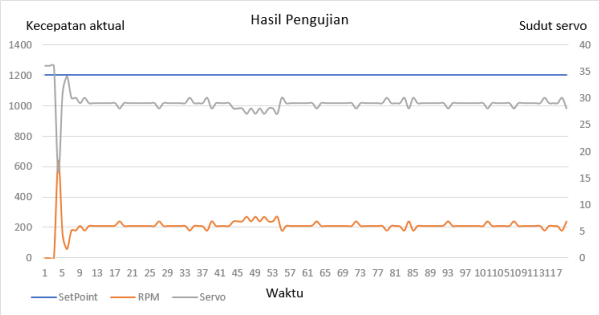
Gambar 6. Grafik respon waktu dengan nilai Kp 0,1

Pada gambar 6 grafik di atas merupakan percobaan pertama untuk mencari nilai Kp yang baik. Pada percobaan tersebut menggunakan nilai kp 0,1 mendapatkan hasil yang tidak stabil, nilai overshoot di 162,5% melebihi set piont yang telah di tentukan yaitu 1200 rpm dan terus berosilasi di kecepatan 1500 ke 0 dan sebaliknya secara terus menerus dan tidak mendapatkan nilai yang stabil. Nilai rise time yang di dapat 2 detik dimulai dari detik ke 3, nilai settling time tidak ada dikarenakan sistem yang terus ber osilasi jauh dari batas error maksimal di 10% (1140 < 1200 < 1260) dan overshoot di 162,5% (3150 rpm).



Gambar 7. Grafik respon waktu dengan nilai Kp 0,04

Pada gambar 7 grafik di atas merupakan percobaan kedua dengan nilai Kp 0,05 mendapatkan hasil sistem yang berosilasi hingga detik ke 72 yang kemudiaan stabil di 500 rpm overshoot awal di 27.5% dari setpoint yaitu di nilai 1500 rpm. Nilai rise time waktu yang di butuhkan sistem untuk mendekati setpoint 2 detik dimulai dari setik ke 3, nilai settling time 72 detik dan overshoot di 1530 rpm. Dalam percobaan tersebut sistem sudah dapat stabil tetapi membutuhkan waktu yang lama untuk stabil dan juga stabil di nilai yang jauh dari setpoint, nilai eroro yang didapat dari respon sistem saat mencapai stabil di – 57,5% dari set point yang bernilai 500 rpm.



Gambar 8. Grafik respon waktu dengan nilai Kp 0,02

Pada gambar 8 diatas merupakan grafik respon waktu untuk mencari nilai tuning Kp yang tepat. dengan nilai Kp 0,02 di dapatkan hasil sistem dengan respon yang berosilasi hingga detik ke 6 yang kemudian stabil di 210 rpm, overshoot awal tidak ada dikarenakan respon sistem tidak melebihi batas setpoint tetapi di 650 rpm sistem sidah berusaha menstabilkan osilasi. Nilai rise time di 2 detik dimulai dari detik ke 3, nilai sttling time di 9 detik tetapi nilai stabil jauh dari nilai rp, yaitu dengan nilai error -82,5% dari nilai setpoint yang bernilai 210 rpm. Ditengah sistem yang sudah stabil terdapat gangguan debit air yang menyebabkan terjadinya osilasi tetapi masih di batas overshot nilai stabil 10%

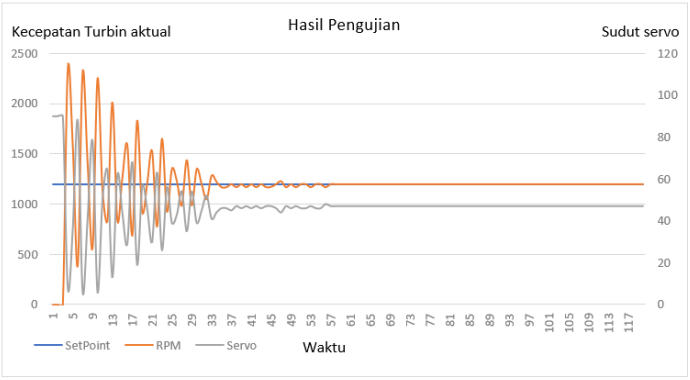
Pada 3 percobaan tersebut dapat di simpulkan bahwa nilai Kp yang tinggi dapat mengakibatkan sistem yang tidak stabil atau sistem yang terus berosilasi, jika nilai kp yang terlalu kecil dapat mengakibatkan nilai stabilitas sistem atau nilai steady state jauh dari nilai setpoint sehingga nilai error lebih tinggi tetapi memiliki waktu stabil yang lebih cepat, hanya membutuhkan waktu 9 detik untuk sistem dapat stabil. Pada percobaan kedua dengan niilai kp 0,05 hasil respon sistem membutuhkan waktu yang lama tetapi memiliki nilai error dari setpoint yang lebih kecil di 57,5% error. Dari ketiga percobaan tersebut maka nilai kp yang selanjutnya digunakan untuk percobaan nilai kp dan ki yaitu 0,04 dikarenakan yang kita inginkan waktu stabil yang lebih cepat dan nilai error yang lebih kecil dari setpoint.

2) Hasil Pengujian Kp dan Ki

Tabel 2. Pengujian nilai Ki

No.	Nilai Tuning			Respon			
	Kp	Ki	Kd	Tr(s)	Ts(s)	Ov(%)	Error %
4	0,04	0,01	0	2	47	122,5	0
5	0,04	0,005	0	2	34	95	0
6	0,04	0,002	0	2	55	155	0

Pada tabel 2 telah dilakukan percobaan nilai ki dengan trial error 3 percobaan untuk menentukan nilai Ki yang tepat dan selanjutnya di masukan ke percobaan berikutnya menggunakan nilai Kp, Ki dan Kd. Pada percobaan ini menggunakan nilai Kp 0,04 dengan nilai Ki 0.01, 0,005 dan 0,002. unuk mengetahui respon sistem dan hasil kontrol dengan nilai setpoint 1200 rpm.



Gambar 9. Grafik respon waktu dengan nilai Ki 0,005

Pada gambar 9 diatas merupakan grafik respon waktu dengan nilai Kp 0,04 dan nilai Ki 0,005 didapatkan hasil respon sistem yang berosilasi hingga 34 detik yang kemudian stabil di 1200 rpm, sesuai dengan nilai setpoint yang di tentukan dengan sudut servo yang terbuka di 47 derajat. Nilai rise time yang di dapat ketika sistem mencapai nilai setpoint di 2 detik, nilai settling time ketika sistem sudah stabil dengan batas toleransi 10% di waktu 34 detik dengan overshoot awal 95% di nilai 2340 rpm dan error dari hasil sistem tidak ada dikarenaakan sistem sidah stabil dengan batas toleransi dari nilai setpoint 1200 rpm.

.

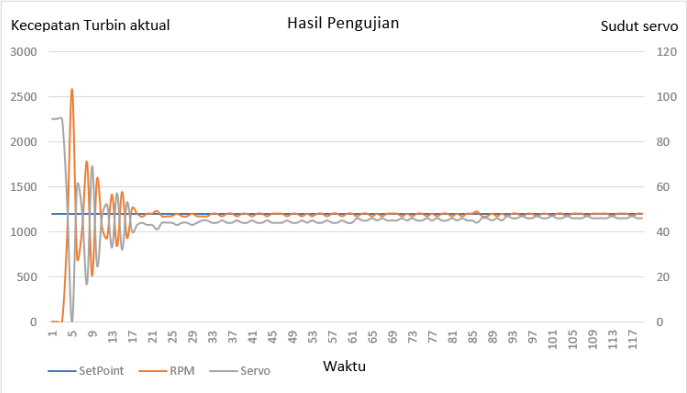
Dari 3 percobaan dengan nilai Ki yang berbeda dapat di simpulan bahwa nilai Ki yang tinggi dapat mengakibatkan waktu nilai stabil yang lama di 47 detik dan overshoot 122,5%, pada percobaan ke dua dengan nilai Ki 0,005 di dapatkan waktu stabil yang lebih cepat di 35 detik dan overshoot di 95%, dan pada percobaan ketiga nilai Ki 0,02 waktu stabil lebih lama di 55 detik dan juga nilai overshoot 155% dari nilai setpoint. Dapat disimpulkan bahwa nilai Ki 0,005 dengan nilai Kp 0,04 memiliki nilai respon yang lebih baik dimana nilai overshoot lebih kecil dan juga waktu sistem untuk stabil lebih cepat. Pada percobaan selanjutnya untuk mencari nilai Kd yang tepat digunakan nilai Kp 0,04 dan nilai Ki 0,005.

3) Hasil Pengujian Kp, Ki Dan Kd

Tabel 3. pengujian nilai Kd

No.	Nilai Tuning			Respon			
	Kp	Ki	Kd	Tr(s)	Ts(s)	Ov(%)	Error %
7	0,04	0,005	0,0015	2	21	117,5	0
8	0,04	0,005	0,001	2	32	135	0
9	0,04	0,005	0,0005	2	17	115	0

Pada tabel 3 telah dilakukan percobaan nilai Kd dengan trial error 3 percobaan untuk menentukan nilai Kd yang tepat yang selanjutnya di digunakan untuk mendapatkan hasil tuning yang terbaik. Pada percobaan ini menggunakan nilai Kp 0,04 , nilai Ki 0,5 dengan nilai Kd 0.0015, 0,001 dan 0,0005. unuk mengetahui respon sistem dan hasil kontrol dengan nilai setpoint 1200 rpm.



Gambar 10. Grafik respon waktu dengan nilai Kd 0,0005

Pada gambar 10 diatas merupakan grafik respon waktu dengan nilai Kp 0,04 nilai Ki 0,005 dan nilai Kd 0,0005 didapatkan hasil respon sistem yang berosilasi hingga 17 detik yang kemudian stabil di 1200 rpm, sesuai dengan nilai setpoint yang di tentukan dengan sudut servo yang terbuka di 47 derajat. Nilai rise time yang di dapat ketika sistem mencapai nilai setpoint di 2 detik dimulai dari detik ke 3, nilai settling time ketika sistem sudah stabil dengan batas toleransi 10% di waktu 17 detik dengan overshoot awal 115% di nilai 2580 rpm dan error dari hasil sistem tidak ada dikarenaakan sistem sidah stabil dengan batas toleransi dari nilai setpoint 1200 rpm.

Dari 3 percobaan dengan nilai Kd yang berbeda dapat di simpulan bahwa nilai Kd yang tinggi dapat mengakibatkan waktu nilai stabil yang lama di 22 detik dan overshoot 117%, pada percobaan ke dua dengan nilai Kd 0,001 di dapatkan waktu stabil di 32 detik dan overshoot di 135%, dan pada percobaan ketiga nilai Kd 0,0005 waktu stabil lebih cepat di 17 detik dan juga nilai overshoot 115% dari nilai setpoint. Dapat

disimpulkan bahwa nilai Kd 0,0005 dengan nilai Kp 0,04 dan nilai Ki 0,005 memiliki nilai respon yang lebih baik dimana nilai overshoot lebih kecil dan juga waktu sistem untuk stabil lebih cepat.

III. Kesimpulan

1. Sistem kendali PID yang dirancang terbukti efektif dalam mengatur kecepatan turbin melalui pengendalian debit air, dengan koreksi error yang memungkinkan turbin mencapai set point secara stabil.
2. Pengujian menunjukkan adanya hubungan linier antara sudut servo, debit air, dan kecepatan turbin, di mana peningkatan sudut servo secara proporsional meningkatkan debit air dan kecepatan turbin, sehingga memberikan respons yang dapat diprediksi dan dikendalikan dengan baik.
3. Parameter PID yang optimal (Kp = 0,04, Ki = 0,005, Kd = 0,005) berhasil meningkatkan stabilitas sistem dengan mengurangi osilasi dan menghindari overshoot yang signifikan. Selain itu, penggunaan LCD sebagai media monitoring memberikan kemudahan bagi pengguna untuk memantau kinerja sistem secara real-time.

IV. Daftar Pustaka

[1] W. Waluyo, A. Fitriansyah, and S. Syahrial, "Analisis Penalaan Kontrol PID pada Simulasi Kendali Kecepatan Putaran Motor DC Berbeban menggunakan Metode Heuristik," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 1, no. 2, p. 79, Jul. 2013, doi: 10.26760/elkomika.v1i2.79.
[2] F. Azim, A. Ullah, J. Jufrizel, and A. Faizal, "Implementasi Kendali PID Pada Kecepatan Motor DC Sebagai Media Pembelajaran Berbasis Arduino Dan LabVIEW," *J. Telecommun. Electron. Control Eng. JTECE*, vol. 6, no. 2, pp. 139–151, Jul. 2024, doi: 10.20895/jtece.v6i2.1461.
[3] H. Supriyanto, F. Suryatini, A. R. H. Martawireja, and H. Rudiansyah, "Implementasi Kontroler PID Dengan Metode Tuning ZIEGLER-NICHOLS DAN COHEN-COON Pada Sistem SCADA Kendali Level Air," *JTT J. Teknol. Terap.*, vol. 8, no. 2, p. 149, Oct. 2022, doi: 10.31884/jtt.v8i2.410.
[4] Y. Tds and A. Mulya, "Pengaturan Level Ketinggian Air Menggunakan Kontrol PID," vol. 4, no. 2.
[5] Rayhan Al Hayubi, Salsabila Aulia, Dafairro Abbil Gunawan, Syarif Hidayatullah, and Didik Aribowo, "Implementasi Sistem Penggerak Servo SG 90 Berbasis Arduino Uno dengan Kontrol Sudut Dinamis," *Mars J. Tek. Mesin Ind. Elektro Dan Ilmu Komput.*, vol. 2, no. 6, pp. 130–140, Dec. 2024, doi: 10.61132/mars.v2i6.535.
[6] J. A. Prihantono, "Pengaturan Suhu Dengan Menggunakan Kontrol PID".
[7] S. F. Anggraini, A. Ma'arif, and R. D. Puriyanto, "Pengendali PID pada Motor DC dan Tuning Menggunakan Metode Differential Evolution".
[8] T. D. Waya, K. Siringo-ringo, and M. E. Pangaribuan, "Perancangan Sistem Kendali Motor Penggerak arah Pantau Kamera CCTV Menggunakan Android Berbasis Nodemcu," vol. 3, no. 1, 2023.
[9] N. Haryanti, F. L. Sanjaya, and A. Suprihadi, "Rancang Bangun Kerangka Turbin Ulir Archimedes Untuk Pembangkit Listrik Tenaga Mikrohidro Berbantu Perangkat Lunak Solidworks 2016," 2021.
[10] A. D. S. Sukowati, "Sistem Kendali PID Aplikasi Mini Plant Water Flow Berbasis Arduino," *J. Elektron. Dan Otomasi Ind.*, vol. 10, no. 3, Nov. 2023, doi: 10.33795/elkolind.v10i3.3677.
[11] F. Isdaryani, M. F. V. Hesya, and F. Feriyonika, "Sintesis Kendali PID Digital dengan Diskritisasi Langsung dan Backward Difference," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 9, no. 2, p. 467, Apr. 2021, doi: 10.26760/elkomika.v9i2.467.
[12] A. Rahmat and R. Arman, "Pembuatan Turbin Ulir Archimedes Tenaga Mikrohidro".
[13] P. P. Kalatiku, "Sistem Pengendalian PID Yang Diaplikasikan Pada Pengendalian Steam Turbin Dengan Single Variable Input Dan Single Output," vol. 9, no. 2.
[14] R. Solekha and U. Latifa, "Sistem Kendali Proportional Integral Derivative (PID) Menggunakan Mikrokontroler Arduino Pada Thinkercad," *ELECTRON J. Ilm. Tek. Elektro*, vol. 5, no. 1, pp. 89–97, May 2024, doi: 10.33019/electron.v5i1.108.
[15] F. Hardianto, M. Shofani, and H. H. Kusuma, "Utilization of IP LM393 Sensor Module as an Automation System for a Portable Gallon Pump," *Phys. Educ. Res. J.*, vol. 5, no. 2, pp. 91–96, Dec. 2023, doi: 10.21580/perj.2023.5.2.11099.