RANCANG BANGUN PENGENDALIAN LAJU ALIRAN PADA SISTEM PERPIPAAN DENGAN METODE TUNING ZIEGLER NICHOLS BERBASIS ESP32

Zafira Rakhmatillah[1], Mochamad Rizky Pradana[1*], Suka Handaja Budi[1]
[1]Teknik Instrumentasi Kilang, PEM Akamigas, Jl. Gajah Mada No.37, Blora, 55312
**E-mail*: *Mochamad.Pradana@esdm.go.id*

ABSTRAK

Pengendalian laju aliran fluida dalam sistem perpipaan diperlukan untuk menjaga stabilitas dan efisiensi proses industri. Penelitian ini merancang sistem pengendalian aliran berbasis ESP32 yang terintegrasi dengan sensor aliran YF-S201, motor servo MG996R sebagai aktuator katup, serta web server untuk pemantauan dan penyesuaian setpoint secara real-time. Metode tuning Ziegler–Nichols digunakan untuk menentukan parameter kontroler PI (Kp = 2,7; Ti = 3,3) dan PID (Kp = 3,6; Ti = 2; Td = 0,5. Hasil menunjukkan bahwa kontroler PI memiliki performa paling stabil, ditandai dengan rise time 5 detik, settling time 5 detik, serta kemampuan memulihkan aliran tanpa osilasi meskipun terdapat gangguan. Sebaliknya, kontroler PID menunjukkan respons yang tidak stabil dan menghasilkan osilasi berkelanjutan akibat sensitivitas komponen derivatif terhadap noise sensor. Temuan ini mengindikasikan bahwa kontroler PI lebih sesuai digunakan pada sistem perpipaan berbasis servo yang memiliki karakteristik nonlinier dan tingkat noise pengukuran yang tinggi.

Kata kunci: Pengendalian Flow, PID Controller, ESP32

1. PENDAHULUAN

Pengendalian laju aliran fluida dalam sistem perpipaan diperlukan untuk menjaga stabilitas dan efisiensi proses industri, terutama pada proses yang rumit seperti industri migas[1]. Hal ini dilakukan untuk mengatur feed yang masuk kedalam sebuah proses ataupun mencegah terjadinya kebocoran pipa[2], [3], [4] yang dapat berakibat fatal secara keamanan, ekonomi maupun lingkungan[5]. Informasi dari laju aliran fluida dapat pula digunakan untuk mengetahui profil aliran[6] yang berguna untuk analisis kinerja pompa, pengukuran NPSH, maupun estimasi umur dari pipa[7]. Sehingga pengukuran dan pengendalian laju aliran menjadi bagian penting dari keseluruhan industri proses.

Pada penelitian ini digunakan metode pengendalian laju aliran menggunakan kontroler PID[8], [9], [10]. Kontroler PID[11] merupakan metode yang paling umum digunakan dalam pengaturan industri proses karena penggunaannya yang sederhana. Paremeter pengaturan dari kontroler PID merupakan kombinasi dari gain proporsional, integral, dan derivative[12]. Pengaturan yang baik dari ketiga parameter tersebut adalah proses yang penting untuk optimasi dari sistem pengendalian. Pengaturan yang salah dapat mengakibatkan sistem yang tidak efektif dan mengurangi performa keseluruhan. Oleh karena itu, diperlukan kombinasi yang terbaik untuk menjamin proses pengendalian laju aliran bersifat optimal[13].

Beberapa metode digunakan untuk mencari kombinasi dari parameter kontroler PID. Metode tersebut dinamakan sebagai proses tuning kontroler PID. Tuning PID memiliki banyak metode, mulai dari metode konvensional seperti Ziegler Nichols[14], Cohen Coon[14], dan Direct Sythesis[13], ataupun penggunaan kecerdasan buatan seperti Metode Fuzzy[15] maupun Algoritma Genetik[16]. Penelitian yang dilakukan oleh Hambali[14] menunjukan pengendalian laju aliran menggunakan metode tuning Ziegler Nichols menghasilkan rise time sebesar 35,45 detik dengan settling time sebesar 70,91 detik. Penelitian yang dilakukan oleh Argo[17]

menggunakan metode Particle Swarm Organization sebagai metode tuning untuk pengendalian laju aliran. Hasil dari penelitian tersebut menunjukan bahwa rise time pada pengendalian laju aliran dapat mencapai 0,01 detik dan settling time dapat mecapai 6,1 detik, namun memiliki overshoot hingga mencapai 3.3%

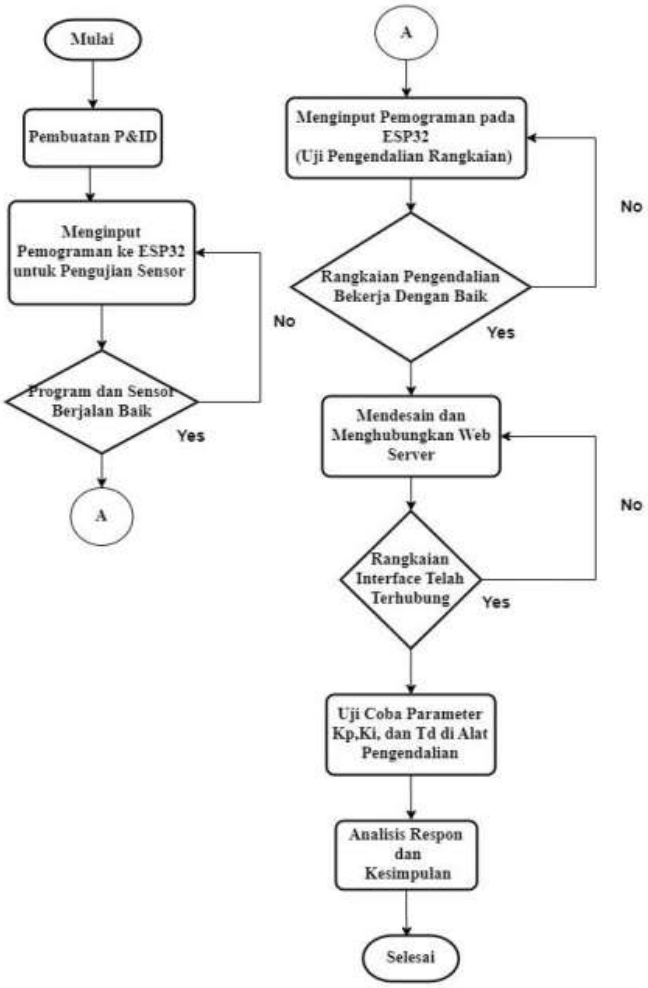
Pengendalian laju aliran dilakukan pada mikrokontroler seperti Arduino, PLC[18], [19], [20], dan ESP32[21]. ESP32 digunakan sebagai mikrokontroler pada penelitian ini. Alasan penggunaan ESP32 sebagai mikrokontroler karena selain harga yang murah dibandingkan dengan mikrokontroler seperti Arduino UNO, STM32, dan PLC, juga ESP32 dapat menggunakan bahasa pemograman yang sama seperti Arduino UNO. Hal ini merupakan suatu keunggulan karena perangkat lunak dan bahasa pemograman Arduino IDE bersifat freeware atau gratis. Selain itu, ESP32 dilengkapi dengan fitur komunikasi berbasis WIFI sehingga dapat berkomunikasi dengan webserver[22]. Oleh karena itu fokus penelitian ini adalah pengendalian laju aliran dengan menggunakan metode PID controller dilengkapi oleh tuning Ziegler Nichols dan mikrokontroler ESP32.

2. METODE

A. Alur Penelitian

Penelitian dimulai dari perancangan P&ID dari sistem pengendalian aliran. Air dialirkan dari tangki penyimpanan menuju pipa untuk dihitung laju aliran menggunakan sensor laju aliran. Kendali laju aliran dilakukan oleh servo valve sebagai aktuator. Error dari pengendalian dikirim kepada mikrokontroler lalu dilakukan koreksi. Pengendalian menggunakan PID Controller dengan optimasi Ziegler Nichols pada paremeter kendali. Untuk melakukan pengendalian menggunakan PID Controler, dilakukan bump test terlebih dahulu untuk mendapatkan parameter Time Constant dan Dead Time. Parameter tersebut digunakan agar tuning dapat dilakukan dan sistem dapat dievaluasi berdasarkan respon transien terbaik.

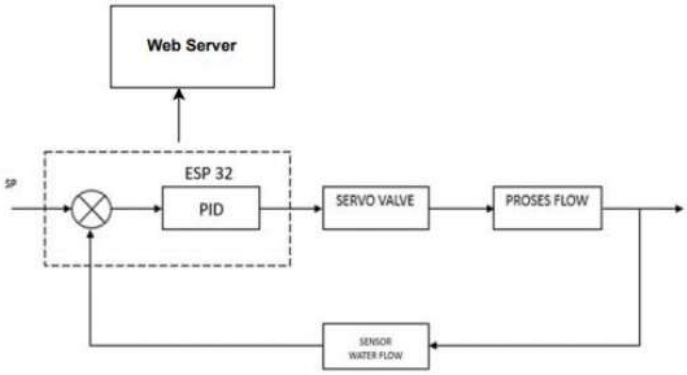
Setelah koreksi error dilakukan dan setpoint tercapai, maka nilai dari laju aliran dikirimkan oleh mikrokontroler kepada webserver untuk ditampilkan sebagai HMI. Penelitian dilakukan menggunakan fluida air dengan suhu ruangan dengan tekanan atmosfir pada tangki terbuka. Penelitian berfokus pada perubahan manipulated variable 50% ke 70% bukaan dari servo valve. Flowchart penelitian dapat dilihat pada gambar 1.



Gambar 1. Flow Chart Penelitian Pengendalian Laju Aliran

B. Blok Diagram Sistem

Sistem yang dirancang adalah sebuah sistem kontrol closed-loop yang berfungsi untuk mempertahankan laju aliran rate air pada nilai setpoint yang telah ditentukan, seperti yang ditunjukkan pada Gambar 2. Dalam sistem ini, terdapat beberapa komponen utama yang bekerja secara terintegrasi. Laju aliran air diukur oleh sensor water flow, yang kemudian mengirimkan data ke mikrokontroler ESP32. Mikrokontroler ini berperan sebagai otak pengendali, yang membandingkan nilai terukur dengan setpoint yang diinginkan. Berdasarkan error yang terdeteksi, algoritma kontrol (PI atau PID) akan menghitung sinyal kontrol yang sesuai dan mengirimkannya ke aktuator. Aktuator dalam sistem ini adalah motor servo yang terhubung ke sebuah ball valve, berfungsi untuk mengatur bukaan katup guna menyesuaikan laju aliran air. Feedback dari process variable (laju aliran) secara terus-menerus kembali ke sensor, membentuk siklus kontrol tertutup. Selain itu, sebuah web server diintegrasikan sebagai antarmuka Human Machine Interface (HMI) yang memungkinkan operator untuk memantau flow rate secara real-time dan mengubah setpoint dari jarak jauh.



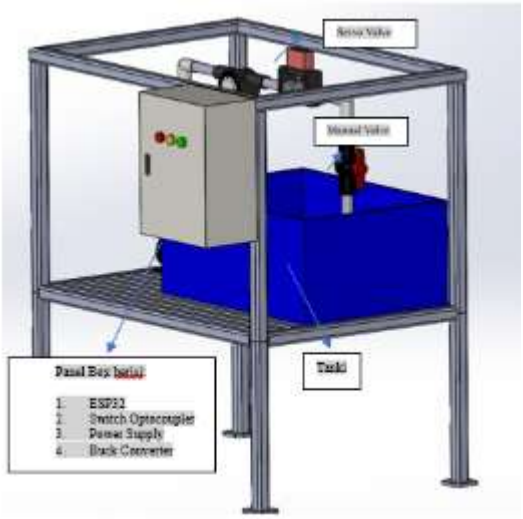
Gambar 2. Blok Diagram Sistem Pengendalian

C. Perancangan Hardware

Gambar 3 menunjukan rangkaian perangkat keras dari sistem pengendalian laju aliran. ESP32 merupakan mikrokontroler dengan kelengkapan berupa 448 kB internal ROM, 32-dual bit core, kecepatan proses data 160/240 MHz, WiFi Module, dan Bluetooth 4.2. Hal ini memungkinkan ESP32 untuk digunakan sebagai mikrokontroler pada sistem Internet of Things (IoT). Selain itu ESP32 dapat menggunakan bahasa pemograman Arduino IDE hingga Micropyhton.

Gambar 3. Mikrokontroler ESP32

Sensor *water flow* YF-S201 diintegrasikan untuk mengukur debit air. Sensor ini mengukur Laju aliran air dengan prinsip kerja mengubah energi mekanik aliran menjadi sinyal digital. Air yang mengalir akan memutar rotor internal yang dilengkapi magnet. Setiap putaran magnet melewati sensor *hall effect*, menghasilkan pulsa Listrik. Motor servo MG996R digunakan sebagai aktuator, yang secara presisi menggerakkan *ball valve* untuk mengatur laju aliran. Pompa air dipilih untuk menciptakan aliran fluida dalam sistem perpipaan.



Gambar 4. Rangkaian Alat Sistem Pengendalian Flow

D. Perancangan Software

Rancangan *software* dibagi menjadi dua bagian utama yang saling berinteraksi: program yang berjalan pada mikrokontroler ESP32 dan *platform web server*. Program pada ESP32 dikembangkan menggunakan Arduino IDE, di mana algoritma kontrol PI dan PID diimplementasikan. Kode ini bertanggung jawab untuk membaca data pulsa dari sensor aliran, menghitung laju aliran, menerapkan perhitungan kontroler untuk menghasilkan *output* yang sesuai bagi motor servo, serta mengelola koneksi Wi-Fi untuk komunikasi dengan *web server*. Sementara itu, *web server* dibangun menggunakan XAMPP (sebagai *environment* yang mencakup Apache untuk *web server* dan MySQL untuk basis data) serta PHP sebagai bahasa pemrograman *server-side*. *Web server* ini berfungsi sebagai antarmuka *Human Machine Interface* (HMI) yang memungkinkan pengguna untuk memonitor nilai *flow rate* secara *real-time*, mengatur *setpoint* yang diinginkan, dan melihat status operasional sistem. Data aliran yang dikirimkan oleh ESP32 akan disimpan dalam basis data MySQL dan kemudian divisualisasikan dalam bentuk grafik pada halaman *web*, meskipun *histeresis* pada grafik *real-time* memerlukan analisis data lebih lanjut di luar *web server* (menggunakan Microsoft Excel) untuk akurasi yang lebih tinggi.

E. Tuning PID Controller Open Loop Ziegler Nichols

PID Controller merupakan metode pengendalian yang berbasis pada pengendalian error dan pengaplikasin koreksi terhadap error tersebut dengan menggunakan gain proporsional, integral dan derivative. Metode ini merupakan metode yang umum digunakan pada industri proses, termasuk didalamnya industri migas. PID controller menghasilkan keluaran $u(t)$ yang nantinya akan bekerja pada plant menggunakan persamaan koreksi error $e(t)$[23], yaitu:

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

Metode tuning Ziegler-Nichols merupakan pendekatan empiris yang luas digunakan untuk menentukan parameter optimal kontroler PI dan PID (Kp, Ki, Kd) pada sistem kontrol single-input single-output (SISO). Proses tuning diawali dengan pengujian open loop yang dikenal sebagai bump test. Dalam pengujian ini, kontroler diatur dalam mode open loop, dan perubahan langkah (step change) kecil diberikan pada input proses (misalnya, bukaan katup). Respon process variable (pV) kemudian diamati hingga mencapai kondisi steady-state baru. Dari kurva

respons ini, parameter Time Constant (T) dan Dead Time (L) sistem dapat diidentifikasi. Time Constant (T) adalah waktu yang diperlukan sistem untuk mencapai 63,2% dari setpoint, dan Dead Time (L) adalah waktu jeda antar perubahan yang terjadi pada sistem. Setelah T dan L diperoleh, parameter Kp, Ti, dan Td untuk kontroler PI dan PID dihitung berdasarkan tabel aturan Ziegler-Nichols yang telah ditetapkan.

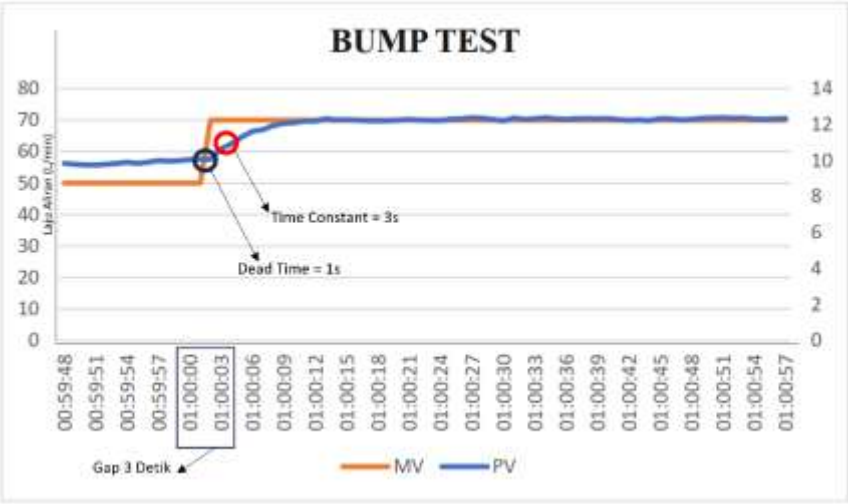
Tabel 1. Metode Ziegler Nichols

Tipe	Kp	Ti	Td
P	$\frac{T}{L}$		
PI	$0{,}9\frac{T}{L}$	$\frac{L}{0.3}$	
PID	$1{,}2\frac{T}{L}$	$2L$	$0{,}5L$

3. PEMBAHASAN

A. Pemodelan Proses Laju Aliran

Pengujian bump test dilakukan untuk memverifikasi bahwa seluruh komponen, baik perangkat keras maupun perangkat lunak, berfungsi sesuai dengan rancangan. Pengujian dilakukan dengan bukaan awal *servo valve* sebesar 50% sehingga nilai PV stabil pada 9,94 L/min. Selanjutnya, diberikan masukan berupa *step input* pada sinyal MV dengan mengubah bukaan *servo valve* dari 50% menjadi 70%. Perubahan MV tersebut menyebabkan respons pada PV, karena *servo valve* merupakan elemen pengendali utama pada jalur aliran. Kenaikan bukaan sebesar 20% menghasilkan peningkatan laju aliran hingga 12,32 L/min. Data hasil bump test kemudian digunakan untuk menentukan parameter dinamika proses, meliputi *Process Dead Time* (L), *Process Time Constant* (T), dan *Process Static Gain* (K). Berikut adalah gambar hasil bump test dari sistem kendali laju aliran:



Gambar 5. Hasil Pengujian Bump Test Pengendalian Laju Aliran

Berdasarkan gambar 5, sistem kontrol merespon perubahan MV dengan jeda 1 detik, sehingga didapatkan Dead Time (L) sebesar 1 detik. Selanjutnya, untuk proses mencapai 63,2% dari setpoint diperlukan waktu sebesar 3 detik sehingga didapatkan nilai Time Constant (T) sebesar 3 detik. Dan terakhir, nilai gain perubahan MV dari 50% hingga mencapai 70% ber-

ubah dari 9,97 L/min menjadi 12,32 L/min, sehingga didapatkan gain DC dapat dihitung dengan rasio ΔPV dan ΔCO dengan nilai akhir sebesar 0,119

Tabel 2. Parameter Sistem Pengendalian Level

Parameter	Nilai
Range Process Variable (ΔPV)	2.38
Range Controller Output (ΔCO)	20
Dead Time (L)	1 s
Time Constant (T)	3 s
Gain (K_c)	0,119

B. Pengujian Metode *Open Loop Ziegler Nichols* Tanpa *Disturbance*

Berdasarkan nilai time constant (T), dead time (L), dan gain (K_c) yang didapatkan dari Bump Test, dapat ditentukan nilai proportional gain (K_p), integral time (T_i), dan derivative time (T_d) untuk Tuning PID Controller dengan Metode Open Loop Ziegler Nichols sebagai berikut:

Tabel 3. Nilai Kp, Ti, dan Td pada Metode Tuning Open Loop Ziegler Nichols

Tipe	Kp	Ti	Td
PI	2,7	3,3	
PID	3,6	2	0,5

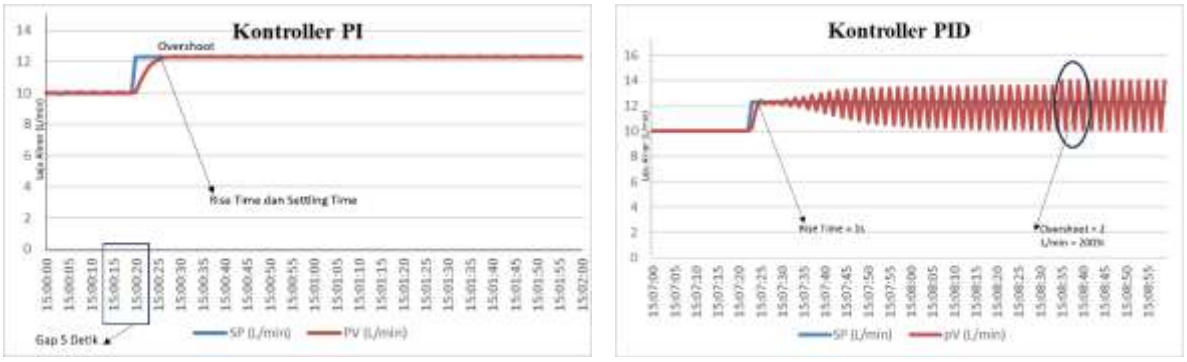
Evaluasi kinerja PID controller pada sistem pengendalian laju aliran dilakukan dengan menganalisis respons transien, meliputi *rise time*, *settling time*, dan *overshoot*. Pengujian dilakukan dengan memberikan perubahan setpoint dari 10 L/min menjadi 12,3 L/min. Seluruh konfigurasi kontroler PID diuji untuk memperoleh parameter respons transien tersebut. Pada pendekatan *Open-Loop Ziegler–Nichols*, dua konfigurasi diuji, yaitu PI dan PID, dengan hasil ditunjukkan pada Gambar 6.

Pada kondisi tanpa gangguan eksternal, kontroler PI menunjukkan performa pengendalian yang stabil dan sesuai dengan karakteristik respons industri. Ketika setpoint dinaikkan dari 10 L/min menjadi 12,3 L/min, *process variable* (PV) merespons dengan *rise time* 5 detik dan *initial time constant* 2 detik. Respons yang dihasilkan tidak menunjukkan *overshoot*, dan PV mencapai *settling time* 5 detik dengan kondisi tunak yang stabil tanpa fluktuasi atau osilasi. Hal ini menunjukkan bahwa kombinasi aksi proporsional dan integral pada kontroler PI menghasilkan respons transien yang cepat sekaligus menjaga stabilitas steady-state. Penggunaan kontroler PI pada sistem ini menghasilkan error yang sangat kecil dan hanya terjadi pada fasa transien aliran, ditambah dengan sistem yang tidak memiliki overshoot dan langsung mencapai kestabilan. Error yang tidak ada tersebut membuat parameter IAE dan ISE menjadi nol karena sistem langsung menuju kestabilannya.

Sebaliknya, kontroler PID menunjukkan respons yang tidak stabil. Meskipun perubahan setpoint menghasilkan *rise time* 2 detik dan *initial time constant* 1 detik, respon tersebut disertai osilasi besar dan berkelanjutan. Dalam seluruh durasi pengujian, PV tidak pernah mencapai *settling time* karena respons terus berfluktuasi di sekitar setpoint. Osilasi ini menunjukkan bahwa komponen derivatif atau pengaturan *gain* proporsional dan integral yang terlalu agresif memperkuat *noise* pada sinyal pengukuran aliran. Aksi derivatif yang diterapkan pada *noise* tersebut menghasilkan perubahan sinyal kendali yang besar dan cepat, sehingga memicu osilasi yang tidak dapat diredam. Osilasi yang tidak dapat diredam tersebut membuat nilai IAE dan ISE Respons kontroler PI dan PID hasil tuning Open-Loop Ziegler–Nichols ditunjukkan pada Gambar 6.

Tabel 4. Respons Transien pada PID Controller dengan Tuning Open Loop Ziegler Nichols

Tipe	*Time Constant (s)*	*Rise Time (s)*	*Overshoot (%)*	*Settling Time (s)*
PI	2	5	0	5
PID	1	2	200%	∞



(a) (b)

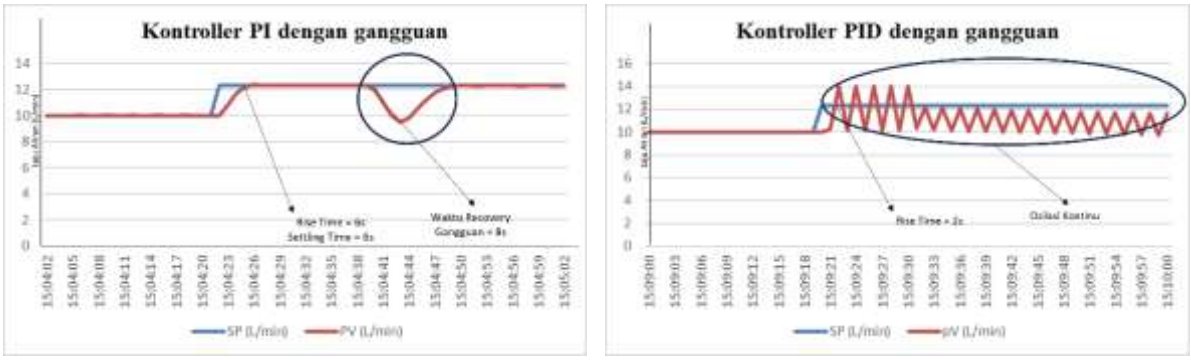
Gambar 6. Respons Transien pada PID Controller dengan Tuning Open Loop Ziegler Nichols
(a) Respons dengan Konfigurasi PI
(b) Respons dengan Konfigurasi PID

C. Pengujian Metode Open Loop Ziegler Nichols Dengan Disturbance

Gangguan pada sistem diberikan untuk mengevaluasi kemampuan kontroler PID dalam mempertahankan kestabilan laju aliran. Gangguan diaplikasikan dengan menutup manual valve secara tiba-tiba selama 10 detik ketika sistem berada pada kondisi *steady state*. Tindakan ini menghasilkan hambatan konkret pada jalur aliran dan mensimulasikan perubahan tekanan mendadak yang harus diatasi oleh kontroler.

Kontroler PI menunjukkan ketahanan (*robustness*) yang baik terhadap gangguan tersebut. Respons awal terhadap perubahan setpoint identik dengan kondisi tanpa gangguan, dengan respons yang cepat dan stabil. Ketika gangguan diberikan, *process variable* (PV) menurun secara langsung dari setpoint. Kontroler PI segera mendeteksi error tersebut dan memberikan aksi korektif yang efektif. Meskipun terjadi penurunan PV yang signifikan, kontroler secara konsisten mengembalikannya ke setpoint dengan waktu pemulihan 8 detik dari awal gangguan. Proses pemulihan berlangsung stabil tanpa muncul osilasi lanjutan yang besar. Hal ini menunjukkan bahwa kontroler PI dengan parameter tuning yang digunakan mampu menangani gangguan eksternal secara efisien sekaligus mempertahankan stabilitas keseluruhan sistem.

Sebaliknya, kontroler PID menunjukkan performa yang paling tidak stabil pada kondisi ini. Osilasi besar dan berkelanjutan yang telah muncul pada kondisi tanpa gangguan tetap terjadi dan meningkat ketika gangguan diberikan. Meskipun respons awal menghasilkan *rise time* 2 detik dan *initial time constant* 1 detik, gangguan menyebabkan fluktuasi PV yang semakin ekstrem. PV tidak pernah mencapai kondisi *settling* dalam batas toleransi selama seluruh durasi pengamatan. Hal ini menunjukkan bahwa konfigurasi PID dengan parameter tuning yang digunakan tidak mampu menghasilkan performa yang stabil dalam menghadapi gangguan dan *noise* secara simultan. Respons PI dan PID terhadap gangguan pada tuning Open-Loop Ziegler–Nichols ditunjukkan pada Gambar 7.



(a) (b)

Gambar 7. Respons Transien pada PID Controller dengan Tuning Open Loop Ziegler Nichols dengan Disturbance
(a) Respons dengan Konfigurasi PI
(b) Respons dengan Konfigurasi PID

4. SIMPULAN

Sistem pengendalian laju aliran berbasis mikrokontroler ESP32 telah berhasil direalisasikan dan berfungsi sesuai rancangan. Integrasi sensor aliran YF-S201, aktuator servo MG996R, dan pompa air membentuk sistem kendali tertutup yang mampu meregulasi laju aliran secara real-time. Hasil uji *bump test* menunjukkan bahwa proses memiliki *process gain* sebesar 0,119 dengan dead time (L) sebesar 1 detik, dan time constant (T) sebesar 3 detik. Model ini menjadi dasar penentuan parameter kontrol menggunakan metode *Open-Loop Ziegler–Nichols*.

Metode tuning tersebut menghasilkan parameter kontroler PI (Kp = 2.7, Ti = 3.3) dan PID (Kp = 3.6, Ti = 2, Td = 0.5). Evaluasi performa menunjukkan bahwa kontroler PI memberikan hasil paling stabil dan efektif untuk sistem aliran ini. Pada perubahan setpoint dari 10 L/min ke 12,3 L/min, kontroler PI menghasilkan *rise time* 5 detik, *settling time* 5 detik, tanpa overshoot, serta error yang menurun secara monoton. Nilai IAE dan ISE pada kontroler PI sangat rendah karena error hanya muncul selama fase transien singkat tanpa osilasi. Ketika diberikan gangguan berupa penutupan katup manual selama 10 detik, kontroler PI tetap mempertahankan stabilitas dan mampu memulihkan *process variable* ke setpoint dalam 8 detik tanpa menghasilkan fluktuasi lanjutan. Hal ini menunjukkan bahwa kontroler PI memiliki robustness yang baik terhadap perubahan kondisi proses dan gangguan eksternal.

Sebaliknya, kontroler PID menunjukkan performa yang tidak stabil. Meskipun menghasilkan *rise time* dan *time constant* yang lebih cepat, kontroler PID memunculkan osilasi besar dan berkelanjutan baik pada kondisi normal maupun saat gangguan diberikan. PV tidak pernah mencapai keadaan tunak, dan nilai IAE serta ISE meningkat signifikan karena error berosilasi terus-menerus. Hal ini mengindikasikan bahwa komponen derivatif atau parameter P/I yang terlalu agresif memperkuat noise bawaan dari sensor aliran, sehingga menimbulkan respons kontrol yang tidak terkendali.

Secara keseluruhan, hasil pengujian menunjukkan bahwa kontroler PI merupakan konfigurasi yang paling sesuai, memberikan stabilitas terbaik, error minimum, dan ketahanan yang baik terhadap gangguan. Dengan demikian, implementasi kontrol PI direkomendasikan sebagai strategi pengendalian utama untuk sistem pengaturan laju aliran berbasis ESP32 ini.

5. DAFTAR PUSTAKA

[1] S. N. Patrialova, D. H. Wardhana, L. P. Rahayu, and T. Agasta, "The Characteristic of Variable Frequency Drive for Water Flow Control in the SFC Plant," *IPTEK The Journal of Engineering*, vol. 7, no. 1, pp. 6–10, 2021.

[2] C. A. De Sousa and O. J. Romero, "Influence of oil leakage in the pressure and flow rate behaviors in pipeline," *Latin American Journal of Energy Research*, vol. 4, no. 1, pp. 17–29, Aug. 2017, doi: 10.21712/lajer.2017.v4.n1.p17-29.

[3] G. Yong, "Research on AI-Based Oil and Gas Pipeline Leak Detection and Localization Technology," in *2023 International Conference on Intelligent Computing, Communication & Convergence (ICI3C)*, IEEE, Dec. 2023, pp. 418–423. doi: 10.1109/ICI3C60830.2023.00085.

[4] R. F. Rahmat, I. S. Satria, B. Siregar, and R. Budiarto, "Water Pipeline Monitoring and Leak Detection using Flow Liquid Meter Sensor," *IOP Conf Ser Mater Sci Eng*, vol. 190, p. 012036, Apr. 2017, doi: 10.1088/1757-899X/190/1/012036.

[5] C. M. Biegel, L. S. Lee, J. G. Graveel, and J. J. Vorst, "The Midwest Oil Pipeline Leak: A Decision Case Study," *Journal of Natural Resources and Life Sciences Education*, vol. 27, no. 1, pp. 122–128, Jan. 1998, doi: 10.2134/jnrlse.1998.0122.

[6] A. Weissenbrunner, A. Fiebach, S. Schmelter, M. Bär, P. U. Thamsen, and T. Lederer, "Simulation-based determination of systematic errors of flow meters due to uncertain inflow conditions," *Flow Measurement and Instrumentation*, vol. 52, pp. 25–39, Dec. 2016, doi: 10.1016/j.flowmeasinst.2016.07.011.

[7] S. Yin, X. Ji, X. Lu, and Q. Zhang, "Simulation study of steam flow behavior and piping mechanical response analysis in safety valve discharge pipes with fluid-structure interaction," *International Journal of Advanced Nuclear Reactor Design and Technology*, vol. 7, no. 3, pp. 308–318, Sep. 2025, doi: 10.1016/j.jandt.2025.07.006.

[8] Y. Kh. Yaseen, A. H. Mhmood, M. R. Subhi, A. B. Rakan, and H. A. Mohammed, "Advanced Flowrate Control of Petroleum Products in Transportation: An Optimized Modified Model Reference PID Approach," *Journal of Robotics and Control (JRC)*, vol. 4, no. 5, pp. 591–599, Sep. 2023, doi: 10.18196/jrc.v4i5.19177.

[9] R. A. Hidayat and D. Figana, "Feed Pump Flow Rate Adjustment in Depth Filtration Process with PID Control," in *Seminar Nasional Teknik Elektro (SNTE)*, Jakarta: Politeknik Negeri Jakarta, 2024, pp. 93–99.

[10] S. Razvarz, C. Vargas-Jarillo, R. Jafari, and A. Gegov, "Flow Control of Fluid in Pipelines Using PID Controller," *IEEE Access*, vol. 7, pp. 25673–25680, 2019, doi: 10.1109/ACCESS.2019.2897992.

[11] Z. Li, "Review of PID control design and tuning methods," *J Phys Conf Ser*, vol. 2649, no. 1, p. 012009, Nov. 2023, doi: 10.1088/1742-6596/2649/1/012009.

[12] J. G. Ziegler and N. B. Nichols, "Optimum Settings for Automatic Controllers," *J Fluids Eng*, vol. 64, no. 8, pp. 759–765, Nov. 1942, doi: 10.1115/1.4019264.

[13] G.-B. So, "Design of Linear PID Controller for Pure Integrating Systems with Time Delay Using Direct Synthesis Method," *Processes*, vol. 10, no. 5, p. 831, Apr. 2022, doi: 10.3390/pr10050831.

[14] N. Hambali, A. Masngut, A. A. Ishak, and Z. Janin, "Process controllability for flow control system using Ziegler-Nichols (ZN), Cohen-Coon (CC) and Chien-Hrones-Reswick (CHR) tuning methods," in *2014 IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, IEEE, Nov. 2014, pp. 1–6. doi: 10.1109/ICSIMA.2014.7047432.

[15] E. B. Priyanka, C. Maheswari, and S. Thangavel, "Online Monitoring and Control of Flow rate in Oil Pipelines Transportation System by using PLC based Fuzzy-PID Controller," *Flow Measurement and Instrumentation*, vol. 62, pp. 144–151, Aug. 2018, doi: 10.1016/j.flowmeasinst.2018.02.010.

[16] K. Ujjwal, M. Sreejeth, and A. Anupama, "Performance Optimization of PID Controllers in Flow-Rate Control Systems Using Hybrid Algorithm Model (Genetic Algorithm - Grey Wolf Optimizer)," in *2025 7th International Conference on Energy, Power and Environment (ICEPE)*, IEEE, May 2025, pp. 1–6. doi: 10.1109/ICEPE65965.2025.11139785.

[17] B. Dwi Argo, Y. Hendrawan, D. Firmanda Al Riza, and A. N. Jaya Laksono, "Optimization of PID Controller Parameters on Flow Rate Control System Using Multiple Effect Evaporator Particle Swarm Optimization," *Int J Adv Sci Eng Inf Technol*, vol. 5, no. 2, p. 62, Apr. 2015, doi: 10.18517/ijaseit.5.2.491.

[18] P. H. Suharti, H. M. Kristiana, and R. N. Amalia, "Design of flow control system based on PLC armfield pressure control module in chemical engineering laboratory," *IOP Conf Ser Mater Sci Eng*, vol. 1073, no. 1, p. 012053, Feb. 2021, doi: 10.1088/1757-899X/1073/1/012053.

[19] E. B. Priyanka, K. Krishnamurthy, and C. Maheswari, "Remote monitoring and control of pressure and flow in oil pipelines transport system using PLC based controller," in *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*, IEEE, Nov. 2016, pp. 1–6. doi: 10.1109/GET.2016.7916754.

[20] S. Włodarczak *et al.*, "Flow Rate Control by Means of Flow Meter and PLC Controller," *Sensors*, vol. 21, no. 18, p. 6153, Sep. 2021, doi: 10.3390/s21186153.

[21] A. Majid and J. Jamaaluddin, "Water Outlet Control Using ESP32 Based Water Flow Sensor," Feb. 24, 2023. doi: 10.21070/ups.421.

[22] S. Handaja, A. K. Dewi, and R. H. Triyanto, "Wireless Volume Corrector for Natural Gas Flow Metering Using ESP32 Microcontroller and Open-Source Web Server," *JOIV : International Journal on Informatics Visualization*, vol. 8, no. 3, p. 1303, Sep. 2024, doi: 10.62527/joiv.8.3.2160.

[23] N. S. Nise, *Control System Engineering*, 8th ed. Wiley, 2019.