

Aplikasi Kendali PID pada Simulator *Water Level Control* Berbasis *TK-Series Controller*

Sarjono Wahyu Jadmiko[#], Luthfi Aditya Suyitno, Fadhlhan Muhammad Lukman, Benjamin Tazaka Geronimo Gibran

Jurusan Teknik Elektro, Politeknik Negeri Bandung
Jln. Gegerkalong Hilir Ds. Ciwaruga, Bandung, 40559, Indonesia
#sarjono_wahyu@polban.ac.id

Abstrak

Pengendalian air pada tandon secara manual sering kali tidak terkendali dan menyebabkan terbuangnya air. Kehandalan metode dan perangkat pengendalian dibutuhkan pada sistem ini, khususnya untuk aplikasi *plant* di industri. Penelitian ini bertujuan untuk merancang dan mengimplementasikan kendali Proporsional Integral Derivatif (PID) pada sebuah simulator *water level control* berbasis *TK-Series Controller*. Pada simulator ini, parameter PID diperhitungkan terlebih dahulu sebelum *input* ke dalam *tuning parameter* pada intruksi Autonics TK4S-T4CN. Pompa sebagai aktuator memompa air kedalam tandon berdasarkan ultrasonik sebagai sensor ketinggian level air. Sinyal ultrasonik yang berupa sinyal digital diubah ke analog oleh mikrokontroler Arduino sebagai *input*. Untuk pengujian sistem digunakan beberapa skenario konfigurasi pengendali yaitu P, PI, PD, dan PID. Pada pengujian dengan pengendali PID, konstanta pengendali ditentukan dengan tiga metode yaitu acak, kurva reaksi, dan *auto tuning*. Hasil pengujian secara eksperimen pada pengendalian simulator *water level control* ini menunjukkan bahwa respon terbaik dihasilkan pada pengendali PID dengan metode *auto tuning* dengan menghasilkan *error* kurang dari 1%.

Kata kunci: level air, PID, Autonics *TK-Series Controller*, kurva reaksi, *error*

Abstract

Manual water control in reservoirs is often uncontrolled and causes water to be wasted. Reliability of control methods and devices is needed in this system, especially for industrial plant applications. This study aims to design and implement Proportional Integral Derivative (PID) control on a water level control simulator based on TK-Series Controller. In this simulator, the PID parameter is calculated before input into the tuning parameter in the Autonics TK4S-T4CN instruction. The pump as an actuator pumps water into the reservoir based on ultrasonic as a water level sensor. The ultrasonic signal in the form of a digital signal is converted to analog by the Arduino microcontroller as input. For system testing, several controller configuration scenarios were used, namely P, PI, PD, and PID. In testing with PID controller, the control constant is determined by three methods, namely random, reaction curve, and auto tuning. The experimental test results on the control of the water level control simulator show that the best response is produced on the PID controller with the auto tuning method with an error of less than 1%.

Keywords: water level, PID, Autonics *TK-Series Controller*, reaction curve, *error*

I. PENDAHULUAN

Pengendalian level cairan pada suatu proses di industri sangat diperlukan. Jika tidak dikendalikan, maka level cairan dalam sebuah tangki dapat menjadi tidak stabil terlebih saat terjadi gangguan seperti pemanasan, kebocoran, dan pengisian dari luar. Sistem kendali yang diterapkan pada sistem ini dapat berupa kendali sederhana ON/OFF sampai metode kendali yang kokoh dan handal terhadap

gangguan. Namun, semakin baik metode kendali yang diterapkan akan berdampak pada semakin kompleksnya sistem komputasi yang dirancang.

Berbagai metode pengendalian level cairan telah banyak diusulkan pada penelitian-penelitian sebelumnya. Kendali konvensional Proporsional Integral Derivatif (PID) menjadi salah satu metode yang banyak digunakan, khususnya untuk aplikasi proses industri seperti pada penelitian [1]–[5]. Pengendali PID digunakan karena memiliki

kemudahan dalam hal perancangan dan implementasi di berbagai perangkat pengendali. Seiring berkembangnya metode kendali, jenis pengendali cerdas seperti logika fuzzy juga telah diterapkan untuk pengendalian level fluida seperti pada [6] dan [7]. Perbedaan dengan kendali PID adalah pada penggunaan model matematis dari *plant* yang akan dikendalikan. Kendali logika fuzzy tidak memerlukan respon *open loop* dari *plant* yang akan dikendalikan, sedangkan kendali PID membutuhkan respon *open loop* untuk menentukan parameter pengendalian Kp, Ki, dan Kd. Penelitian [8] membandingkan kinerja pengendali PID dan logika fuzzy pada sebuah simulator *plant* orde dua. Hasilnya menunjukkan bahwa respon pengendali PID lebih baik jika diterapkan pada *plant* orde dua yang memiliki konstanta waktu yang kecil, sedangkan pengendali logika fuzzy lebih cocok digunakan pada *plant* orde dua dengan konstanta waktu yang besar. Selain metode kendali PID dan logika fuzzy, metode *Model Predictive Control* (MPC) yang memiliki karakteristik prediktif dan kokoh juga telah diterapkan seperti pada penelitian [9]. Pada penelitian ini MPC diimplementasikan pada *Programmable Logic Controller* (PLC) sehingga dihasilkan sebuah sistem pengendalian yang sangat handal untuk berbagai aplikasi di industri. PLC merupakan sebuah perangkat kendali yang banyak digunakan dalam proses industri. PLC digunakan karena memiliki desain perangkat yang lebih handal jika dibandingkan dengan perangkat pengendali lainnya seperti mikrokontroler. PLC dapat diaplikasikan untuk berbagai jenis pengendalian di industri, seperti kendali level cairan. Akan tetapi, tidak semua metode kendali dapat dapat dengan mudah diimplementasikan pada PLC karena keterbatasan komputasi [10].

Penelitian ini bertujuan untuk merancang sebuah sistem kendali level cairan berbasis pengendali PID menggunakan Autonics *TK Series Controller*. Perangkat ini digunakan karena sangat kompatibel diaplikasikan dan banyak digunakan untuk kebutuhan industri. Kendali PID juga telah terintegrasi pada perangkat ini sehingga lebih memudahkan dalam hal *tuning* pada masing-masing konstanta Kp, Ki, dan Kd untuk mendapatkan respon yang diinginkan. Seperti pada penelitian [11] dan [12], sistem kendali dirancang dengan umpan balik sensor ultrasonik HC-SR04. Perangkat keras dibangun dalam skala purwarupa untuk kebutuhan pengujian [13]. Berbeda dengan beberapa penelitian sebelumnya, kontribusi penelitian ini terletak pada implementasi pengendali PID pada perangkat pengendali Autonics *TK Series Controller* yang banyak digunakan untuk kebutuhan industri.

II. METODE PENELITIAN

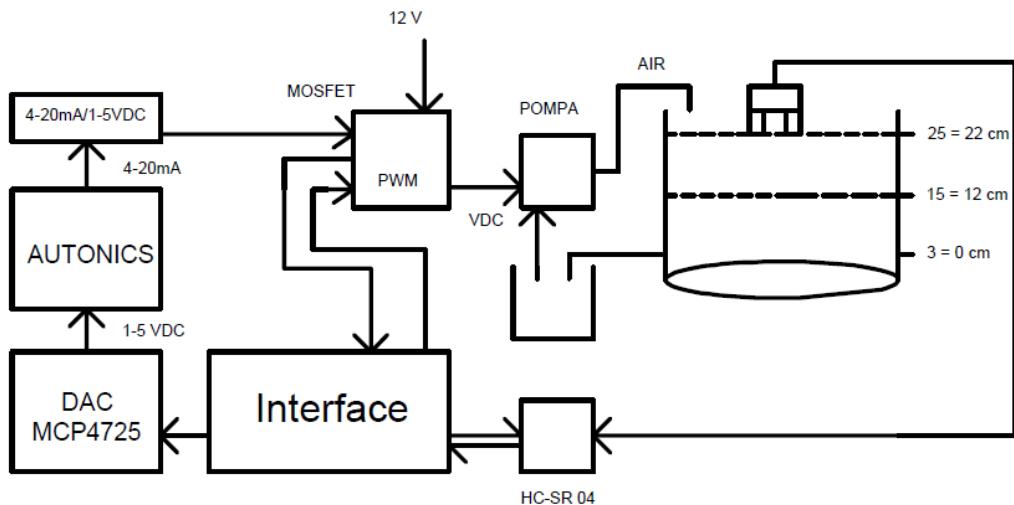
A. Perancangan Perangkat Keras

Pada bagian ini akan diuraikan perancangan perangkat keras dari mulai spesifikasi, diagram blok, dan desain purwarupa. Spesifikasi peralatan yang dipergunakan pada simulator *water level control* dapat dilihat pada Tabel 1. Pada spesifikasi terdiri dari beberapa bagian yaitu *plant* simulator, pengendali, jenis cairan, catu daya, serta debit dan sensor. Spesifikasi ini ditetapkan dalam skala laboratorium untuk mempermudah pengujian.

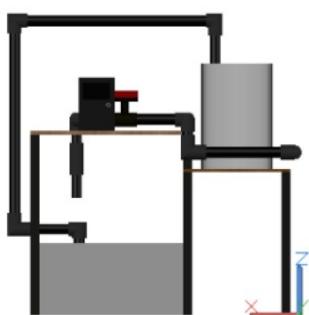
Adapun diagram blok perangkat keras yang dirancang dapat dilihat pada Gambar 1. Dari gambar tersebut terdapat *plant* yang merupakan tempat dimana level ketinggian air dikendalikan atau biasa disebut juga dengan tangki yang dilengkapi skala ketinggian. Lalu bukaan aliran air yang masuk dan keluar dari tangki tersebut dapat dikendalikan secara manual dengan menggunakan *valve*. Di atas tangki tersebut terdapat sensor ultrasonik HC-SR04 yang berfungsi mendekripsi level ketinggian air dan hasil ketinggian air itulah yang digunakan sebagai *input* untuk Autonics *TK-Series Controller* yang sebelumnya diubah terlebih dahulu sinyalnya menjadi analog oleh MCP4725. Autonics *TK-Series Controller* mengendalikan dan melakukan *tuning* untuk jenis pengendali PI. *Output* Autonics yang merupakan *current* diubah menjadi tegangan sebelum masuk ke *driver* MOSFET untuk mengaktifkan motor pompa DC. MOSFET berperan sebagai saklar untuk menerima sinyal *Pulse Width Modulation* (PWM) yang dapat mengendalikan kecepatan motor DC atau dalam sistem ini yaitu pompa celup Taffware 12 VDC yang berfungsi untuk mensirkulasi air ke *plant* dengan VDC sebagai *supply* untuk pompa tersebut.

Tabel 1. Spesifikasi alat

Jenis komponen	Keterangan
Plant	
Volume tangki	12 liter
Tinggi tangki	40 cm
Diameter tangki	20 cm
Pengendali	
Metode	PID
Perangkat Keras	Autonics
Zat cair yang dikendalikan	
Air	6 Liter
Catu daya	
Sumber AC	220V/50Hz
Sumber DC	12VDC/5A
Lainnya	
Debit pengisian air	9,1liter/menit
Sensor	HC-SR04



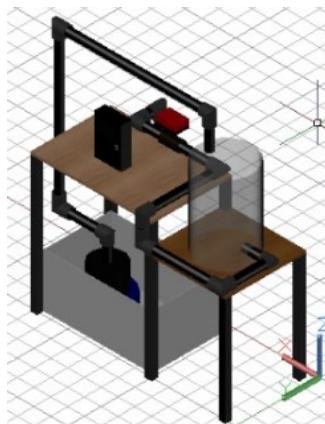
Gambar 1. Diagram blok simulator



Gambar 2. Rancangan *plant* tampak depan



Gambar 5. Rancangan *plant* tampak samping



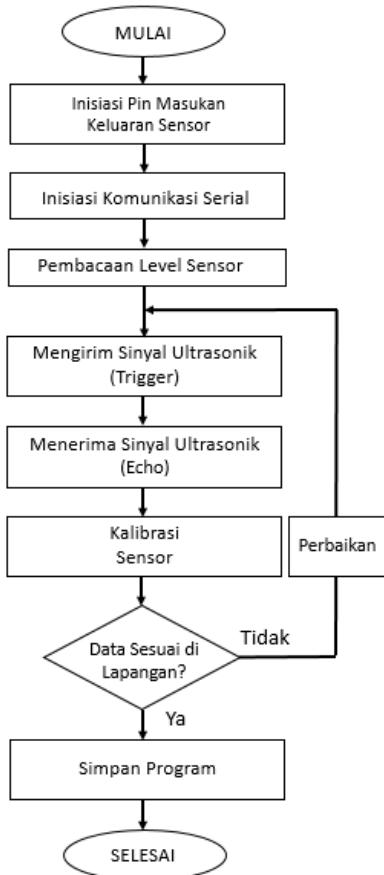
Gambar 3. Rancangan *plant* tampak samping

Setelah mengetahui blok diagram pada simulator *water level control* dan dapat menetapkan spesifikasi alat yang digunakan, maka berikutnya adalah merancang perangkat keras dan mekanik untuk simulator tersebut. Perancangan dilakukan terhadap dimensi, penempatan komponen, dan material mekanik agar dapat menopang peralatan yang digunakan dalam sistem. Gambar 3 dan Gambar 4 menunjukkan rancangan desain perangkat keras elektrik dan mekanik dari simulator *water level control*.

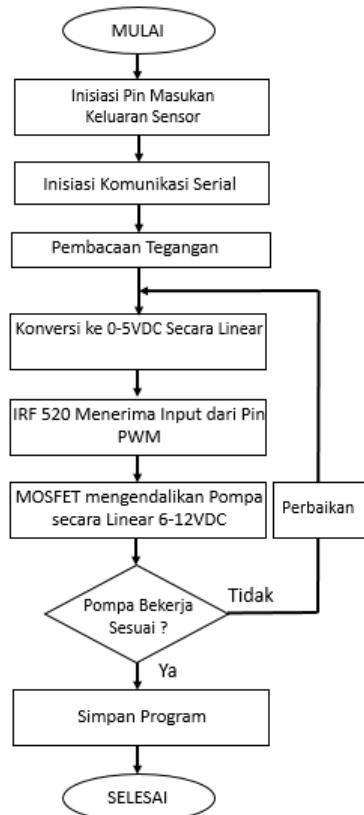
Rancangan *plant* kemudian direalisasikan ke dalam bentuk perangkat keras dalam bentuk purwarupa skala laboratorium seperti yang ditunjukkan Gambar 5. Pada gambar tersebut terdapat sebuah tangki kendali yang berfungsi sebagai tempat cairan yang akan dikendalikan. Kemudian di bagian bawah terdapat *reservoir* sebagai tempat sumber air. Selanjutnya terdapat pula modul perangkat pengendali yang didalamnya terdapat Autonics. Sebagai umpan balik, sensor ultrasonik HC-SR04 ditempatkan di bagian atas tangki untuk mendeteksi level cairan yang ada di dalamnya.

B. Perancangan Perangkat Lunak

Pada bagian ini, akan dibahas mengenai perangkat lunak dan algoritmanya. Perangkat lunak yang dirancang terdiri dari program pembacaan sensor, program konversi PWM ke sinyal analog, dan program penggerak pompa. Ketiga program tersebut diimplementasikan pada mikrokontroler Arduino Uno yang berperan sebagai antarmuka pengkondisi sinyal. Adapun diagram alir pemrograman dapat dilihat pada Gambar 6 dan Gambar 7.



Gambar 6. Flowchart program HC-SR04 dan MCP4725



Gambar 7. Flowchart program penggerak pompa

III. HASIL DAN PEMBAHASAN

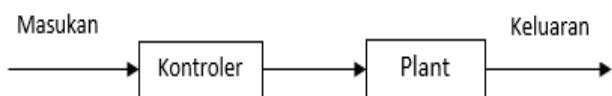
Setelah melakukan perancangan dan realisasi baik perangkat keras maupun perangkat lunak, selanjutnya rancangan sistem digabungkan dalam bentuk realisasi sistem keseluruhan. Langkah selanjutnya yaitu melakukan pengujian dan analisia. Tujuan dari melakukan pengujian adalah untuk mengetahui apakah sistem bekerja sesuai perancangan atau masih terdapat kegagalan dalam sistem. Pengujian dapat juga digunakan untuk menjadi alat pembanding apakah kinerja sistem sesuai dengan teori atau tidak. Dalam hal ini pengujian dilakukan dengan bagian pengujian perangkat keras dengan alat ukur Fluke 179 dan pengujian perangkat lunak menggunakan DAQMaster sebagai penampil grafik.

Pengujian yang dilakukan dalam hal ini yaitu pengujian kendali PID pada simulator *water level control* berbasis *TK-Series Controller*. Pengujian dilakukan untuk mengetahui karakteristik respon tanpa pengendali dan dengan kendali. Jenis pengendali yang diujikan adalah P, PD, PI, PID, dan juga *auto tuning*. Pengujian dilakukan untuk mengetahui karakteristik kendali agar selanjutnya kita dapat menentukan dan memilih tipe pengendalian yang paling baik. Seperti kita ketahui bahwa tipe pengendalian yang baik yaitu meliputi *rise time* yang cepat, *overshoot* yang kecil, dan juga *error* yang minimal.

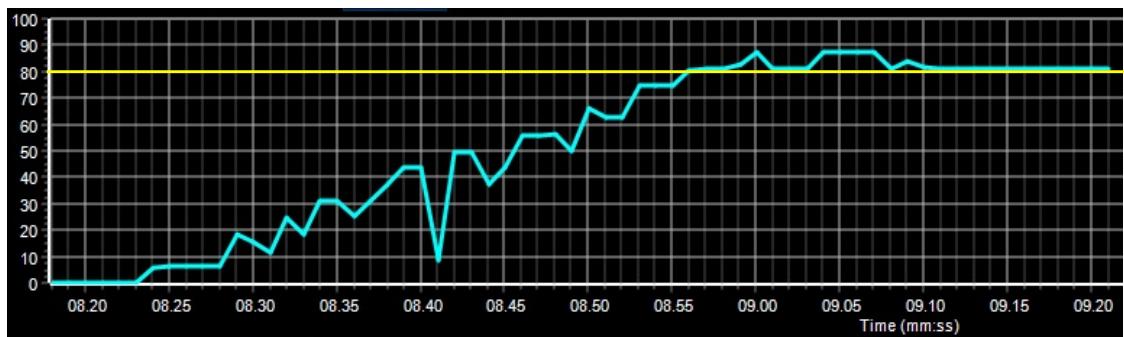
A. Pengujian Tanpa Kendali

Pada skenario ini, pengujian dilakukan menggunakan skema *loop* terbuka dengan gambar rangkaian seperti yang ditunjukkan Gambar 8. Seperti terlihat pada rangkaian, pada skenario ini tidak terdapat sensor sebagai umpan balik. Dengan demikian kita hanya dapat mengamati respon ketika diberikan masukan.

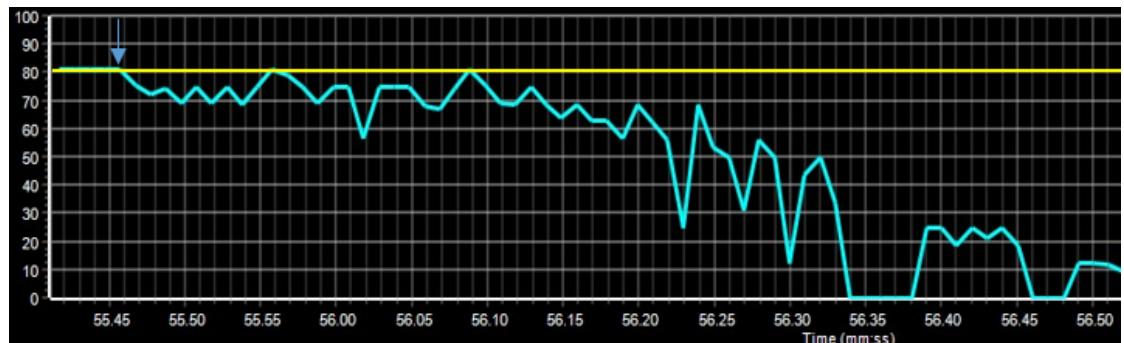
Pengujian ini dilakukan dengan mematikan seluruh kendali dan mengatur *setpoint* sebesar 80%. Pada pengujian ini juga dilakukan skenario tanpa adanya gangguan dan dengan adanya gangguan sebesar 50%. Gangguan dalam hal ini adalah bukaan *valve* pada bagian bawah tangki yang menyebabkan cairan di dalam tangki menjadi berkurang. Dengan mengamati respon *loop* terbuka dari *plant* saat tidak ada gangguan dan diberi gangguan, maka kita dapat mengetahui penentuan konstanta PID pada skema *loop* tertutup.



Gambar 8. Rangkaian pengujian dengan *loop* terbuka



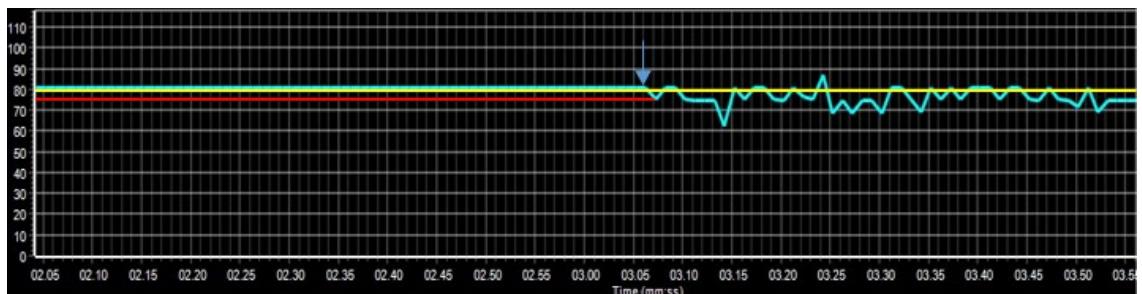
Gambar 9. Pengujian tanpa kendali dan tanpa gangguan



Gambar 10. Pengujian tanpa kendali dan dengan gangguan 50%



Gambar 11. Kendali P dengan $K_p = 10$ dan gangguan 50%



Gambar 12. Kendali P dengan $K_p = 20$ dan gangguan 50%

Gambar 9 dan Gambar 10 menunjukkan hasil pengujian *loop* terbuka tanpa pengendali dengan skenario tanpa dan dengan gangguan. Dari pengujian tanpa kendali dan tanpa gangguan, dapat dilihat respon dari *loop* terbuka mencapai *setpoint* dengan waktu 50 detik. Pada pengujian dengan gangguan terlihat bahwa respon ketinggian cairan terus

menurun ketika *loop* terbuka tanpa kendali dan diberi gangguan 50%.

B. Pengujian Kendali P

Pada skenario ini, pengujian dilakukan dengan cara *loop* tertutup dengan memasukan nilai kendali P yang berbeda beda untuk mengamati respon

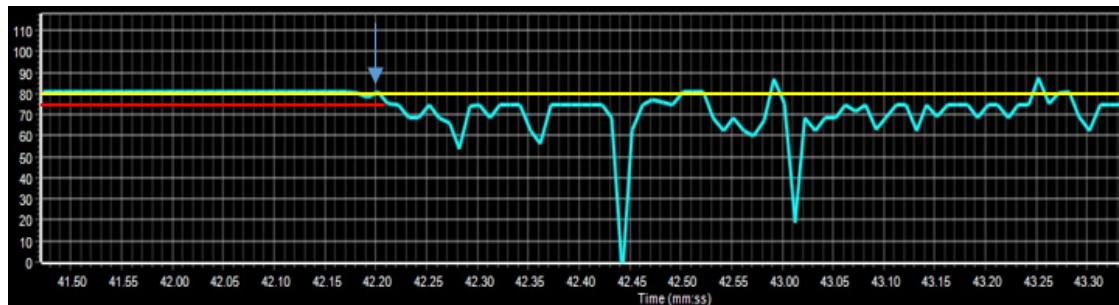
keluarannya. Pengujian dilakukan dengan *setpoint* 80% dan diberi gangguan 50%. Gangguan diberikan dengan cara membuka *valve* pada *plant* ketika keadaan telah *steady state*. Gambar 11 dan Gambar 12 menunjukkan hasil pengujian kendali P dengan $K_p = 10$ dan $K_p = 20$.

Dari hasil pengujian, analisa yang bisa diambil dari respon kendali P tersebut menunjukkan karakteristik kendali P. Semakin tinggi nilai K_p , maka semakin kecil pula *error*-nya. Dapat dilihat bahwa $K_p = 10$ menghasilkan *error* sebesar 25%

dan $K_p = 20$ menghasilkan *error* sebesar 6,25%. Namun, kendali P ini tidak membuat *error* sama dengan 0, atau dengan kata lain masih terdapat perbedaan antara *setpoint* dan hasil keluaran pengendalian tersebut.

C. Pengujian Kendali PD

Pengujian dilakukan dengan cara *loop* tertutup dengan memasukan nilai kendali PD yang berbeda beda untuk mengamati respon keluarannya. Sama



Gambar 13. Kendali PD dengan $K_p = 5$ dan $K_d = 5$ serta gangguan 50%



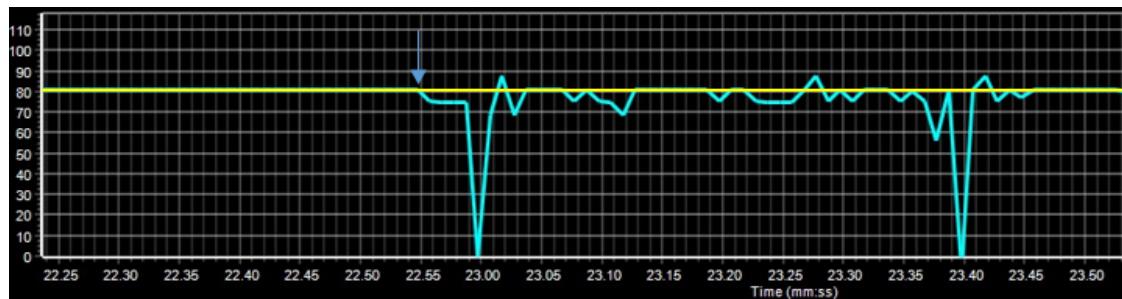
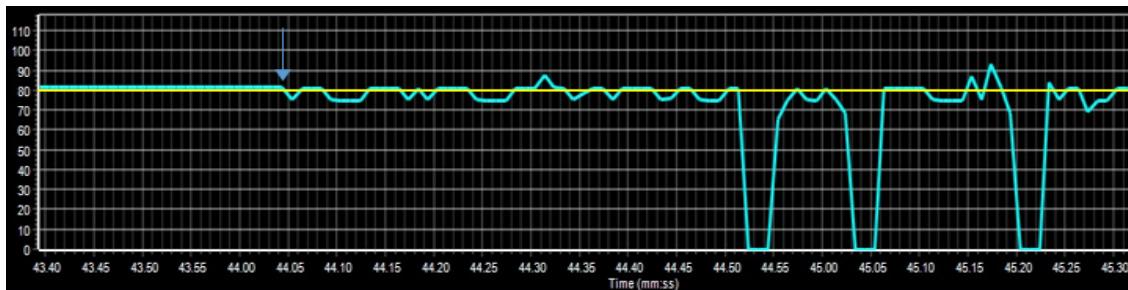
Gambar 14. Kendali PD dengan $K_p = 10$ dan $K_d = 5$ serta gangguan 50%



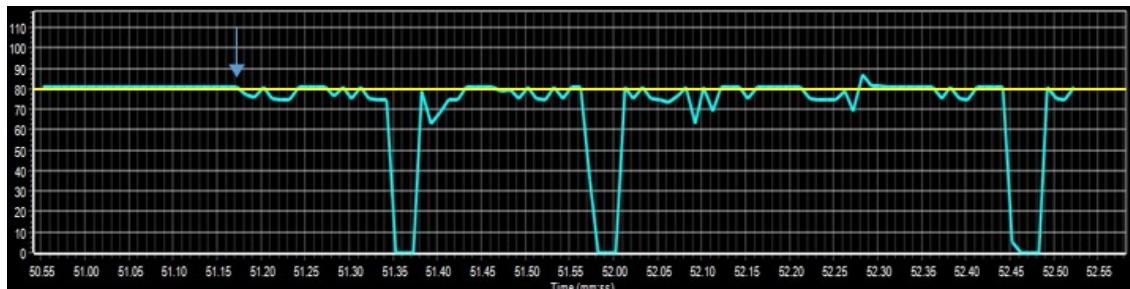
Gambar 15. Kendali PI dengan $K_p = 5$ dan $K_i = 60$ serta gangguan 50%



Gambar 16. Kendali PI dengan $K_p = 5$ dan $K_i = 100$ serta gangguan 50%

Gambar 17. Kendali PID dengan $K_p = 15$, $K_i = 30$, dan $K_d = 5$ serta gangguan 50%

Gambar 18. Kendali PID dengan kurva reaksi dan gangguan 50%

Gambar 19.5 Kendali PID dengan *auto tuning* dan gangguan 50%

halnya dengan pengujian pada kendali P, pengujian dilakukan dengan *setpoint* 80% dan diberi gangguan 50%. Hasil pengujian dapat dilihat pada Gambar 13 dan Gambar 14.

Analisa yang bisa diambil dari respon kendali PD tersebut menunjukkan karakteristik kendali PD. Dapat dilihat bahwa $K_p = 5$ dan $K_d = 5$ menghasilkan *error* sebesar 6,25%, sedangkan $K_p = 10$ dan $K_d = 5$ menghasilkan *error* sebesar 3,75%. Dengan adanya nilai K_d akan membuat *overshoot* lebih kecil dan menambah kestabilan.

D. Pengujian Kendali PI

Pada bagian ini, pengujian juga dilakukan dengan cara *loop* tertutup dengan memasukan nilai K_p dan K_i yang berbeda. Hasil pengujian dapat dilihat pada Gambar 15 dan Gambar 16.

Hasil pengujian dari respon kendali PI tersebut menunjukkan bahwa dengan menambahkan parameter K_i pada pengendalian, pengendali PI memungkinkan sistem memiliki *error* = 0. Nilai $K_i = 60$ menghasilkan *error* sebesar 3,75%, sedangkan

nilai $K_i = 100$ menghasilkan *error* sebesar < 1%. Pada kendali PI, makin besar K_i maka akan makin cepat juga mencapai *setpoint* setelah *overshoot*, dan juga semakin besar nilai I , maka akan lebih cepat kembali pada nilai *setpoint* setelah diberi gangguan.

E. Pengujian Kendali PID

Sama halnya dengan pengujian skenario sebelumnya, pengujian dilakukan dengan cara *loop* tertutup dengan memasukan nilai K_p , K_i , dan K_d secara acak untuk mengamati respon keluarannya. Hasil pengujian dapat dilihat pada Gambar 17.

Untuk mengoptimasi konstanta PID, pengujian kendali PID juga dilakukan dengan perhitungan sesuai dengan metode kurva reaksi (Ziegler Nichols 1). Perhitungan parameter kendali dilakukan berdasarkan respon *loop* terbuka yang telah dilakukan pengujian sebelumnya. Setelah dilakukan perhitungan, maka diperoleh nilai $K_p = 7,2$, $T_i = 10,6$, dan $T_d = 2,65$. Hasil perhitungan tersebut lalu diuji dalam pengujian kendali PID dengan hasil seperti ditunjukkan pada Gambar 18.

Terakhir, pengujian kendali PID juga dilakukan dengan menyetel kendali secara *autotuning* yang tersedia pada perangkat *TK-Series controller*. Dengan fitur ini, parameter PID secara otomatis diganti sesuai dengan perhitungan dari *TK-Series controller*. Hasil pengujiannya pengujian dengan *auto tuning* ditampilkan pada Gambar 19.

Hasil analisa yang dapat diambil dari respon kendali PID tersebut menunjukkan bahwa dengan menggabungkan setiap konstanta K_p , K_i , dan K_d baik dengan cara acak, metode kurva reaksi, dan *auto tuning* respon sistem menghasilkan $error < 1\%$. Kendali PID berpengaruh pada ketahanan terhadap gangguan dengan *overshoot* yang minimal. Dari ketiga metode pengujian kendali PID, metode *auto tuning* memiliki ketahanan terhadap gangguan dan respon yang paling cepat sehingga *overshoot* minimal. Walaupun terdapat riak dari ketiga metode PID tersebut, tetapi itu bukan dihasilkan dari parameter PID tetapi dikarenakan tidak adanya filter pada pembacaan sensor ultrasonik.

IV. KESIMPULAN

Pengujian dan analisa yang telah dilakukan dengan kendali PID pada simulator *water level control* secara keseluruhan berjalan dengan baik dan sesuai dengan tujuan dan deskripsi sistem. Kesimpulan yang dapat diambil adalah simulator *water level control* dapat dikendalikan dengan kendali PID berbasis *TK-Series Controller*. Kendali PID yang diaplikasikan dengan metode acak, kurva reaksi, dan *auto tuning* menghasilkan respon terbaik jika dibandingkan dengan kendali P, PI, dan PD dengan menghasilkan $error < 1\%$. Metode *auto tuning* yang dimiliki oleh *TK-Series Controller* mampu menghasilkan ketahanan terhadap gangguan yang paling baik dan respon yang cepat sehingga *overshoot* minimal. Untuk pengembangan penelitian berikutnya dapat diterapkan filter pada sensor sehingga mampu meminimalkan adanya riak dan menghasilkan respon yang lebih baik.

REFERENSI

- [1] A. Hartanto and K. Kamarudin, “Analisa Pressure dan Level pada CE33 Electronic Process Control Berbasis PID,” *J. Appl. Sci. Electr. Eng. Comput. Technol.*, vol. 1, no. 2, pp. 13–18, 2020.
- [2] T. Pujiati, “Penerapan Kontroler PID pada Sistem Kendali Level Cairan dengan Metode Ziegler-Nichols Berbasis Arduino,” *JTEIN J. Tek. Elektro Indones.*, vol. 2, no. 1, pp. 55–60, 2021.
- [3] I. Iswanto and Jamaluddin, “Studi Pengendalian Level Steam Drum 62-FA-102 Menggunakan PID Aplikasi PT. Pupuk Iskandar Muda.,” *J. Tektro*, vol. 1, no. 1, pp. 1–8, 2017.
- [4] 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, 2021.
- [5] S. A. Simanullang, P. S. Rudati, and F. Feriyonika, “Sistem PID Pengendali Level Ketinggian Air Berbasis Modbus / TCP - LCU dan Industrial Field Control Node - RTU,” *Industrial Reasearch Work. Natl. Semin. Pol. Negri Bandung*, pp. 1–9, 2017.
- [6] A. R. Al Tahtawi, S. Yahya, B. Setiadi, and C. Marsya, “The Implementation of Embedded Fuzzy Logic Controller on Liquid Level Control System,” *International Seminar of Science and Applied Technology (ISSAT 2020)*, 2020, pp. 161–166.
- [7] C. Wida, “Implementasi Fuzzy Logic Controller untuk Pengendalian Level Air,” *Jurnal ELEKTRAN*, vol. 2, no. 1, pp. 1–5, 2012.
- [8] S. W. Jadmiko, S. Yahya, S. Sudrajat, and F. Azizah, “Komparasi Kinerja Kendali PID dan Logika Fuzzy pada Simulator Plant Orde Dua,” *JTERA (Jurnal Teknol. Rekayasa)*, vol. 5, no. 2, p. 237, 2020.
- [9] S. Aslam, S. Hannan, M. U. Sajjad, and M. W. Zafar, “PLC based model predictive control for industrial process control,” *Int. J. Adv. Appl. Sci.*, vol. 4, no. 6, pp. 63–71, 2017.
- [10] A. Venuprasad and Y. S. Narayan, “Automatic Monitoring and Controlling of Pressure using PLC and SACDA,” *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 5, no. July, pp. 13671–13677, 2016.
- [11] A. Alawiah and A. Rafi Al Tahtawi, “Sistem Kendali dan Pemantauan Ketinggian Air pada Tangki Berbasis Sensor Ultrasonik,” *KOPERTIP J. Ilm. Manaj. Inform. dan Komput.*, vol. 1, no. 1, pp. 25–30, 2017.
- [12] D. A. Insantama and B. Suprianto, “Rancang Bangun Kendali Level Air Otomatis Pada Tangki Dengan Servo Valve Berbasis Fuzzy Logic Controller Menggunakan Arduino,” *J. Tek. Elektro*, vol. 08, no. 01, pp. 143–151, 2019.
- [13] K. Karwati and J. Kustija, “Prototype of Water Level Control System,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 384, no. 1, 2018.