

## SIMULASI PENGENDALIAN LEVEL AIR DALAM TANKI BERBASIS LABVIEW MENGGUNAKAN FACTORY IO

<sup>1</sup>Zaenal Abubakar Siddik, <sup>1</sup>Nanang Roni Wibowo, dan <sup>1</sup>Sigit Agung Purnomo

<sup>1</sup>Teknologi Rekayasa Mekatronika, Politeknik Enjinereng Indorama, Purwakarta

e-mail: [zaenalabubakarsiddik@gmail.com](mailto:zaenalabubakarsiddik@gmail.com)

### Abstrak

*Factory IO merupakan aplikasi simulasi 3D yang memberikan visualisasi real plant seperti dalam dunia industri, kesulitan untuk melaksanakan simulasi pada real plant yang ada di industri menjadi landasan virtual plant Factory IO dipilih sebagai simulator, Factory IO memiliki virtual plant yang dapat digunakan untuk simulasi seperti plant tangki berisi air, Factory IO juga memiliki kemampuan untuk berintegrasi dengan software controller seperti LabVIEW, dengan menggunakan software LabVIEW memungkinkan perancangan sistem control yang selanjutnya controller diaplikasikan terhadap virtual plant Factory IO, pada penelitian ini virtual plant yang digunakan yakni plant berupa tangki berisi air, dengan pengendalian PID, kendalier PID dibuat dengan software LabVIEW dengan tujuan untuk melakukan pengendalian level air dalam tangki, penelitian ini dilakukan dengan bertujuan untuk mendapatkan nilai parameter KP, TI dan TD yang sesuai untuk diterapkan pada virtual plant Factory IO. Dalam penelitian dilakukan simulasi pengendalian PID dengan metode trial and error. Berdasarkan hasil simulasi dengan pengendalian PID, dengan parameter kendali  $K_p = 475$ ,  $T_i = 0.1$ ,  $T_d = 0.0001$ , terlihat bahwa respons sistem dapat mengikuti set point yang diberikan oleh controller.*

**Kata kunci:** factory IO, PID, LabVIEW, level

### Abstract

*Factory IO is a 3D simulation application that provides real-world plant visualization, such as those found in industry. The difficulty of conducting simulations on real plants in industry is the basis for choosing Factory IO as a simulator. Factory IO has a virtual plant that can be used for simulations such as plant tank filled with water, Factory IO also can integrate with controller software such as LabVIEW, using LabVIEW software makes it possible to design a control system which then the controller is applied to the virtual plant Factory IO, in this research the virtual plant used is a plant in the form of a tank filled with water, with PID control, the PID controller was created with LabVIEW software to control the water level in the tank. This research was carried out to obtain KP, TI, and TD parameter values suitable for application to the virtual plant Factory IO. In the research, a PID control simulation was carried out using the trial-and-error method. Based on the simulation results with PID control ( $K_p = 475$ ,  $T_i = 0.1$ ,  $T_d = 0.0001$ ), the system response follows the controller-set point.*

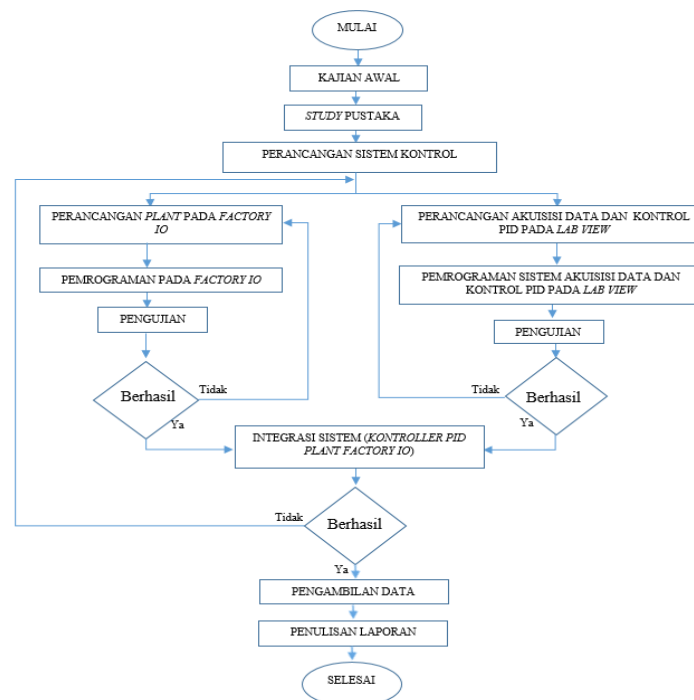
**Keywords:** factory IO, PID, LabVIEW, level

## 1. PENDAHULUAN

Perkembangan dunia industri disertai berkembangnya sistem kendali yang memiliki keluaran semakin baik, pengendalian sistem berbasis *PID* merupakan salah satu sistem kendali yang memiliki keandalan yang baik sehingga banyak digunakan pada plant di banyak industri, Pengendalian berbasis *PID* terdiri atas 3 parameter yakni  $K_p$ ,  $T_i$ ,  $T_d$  yang memungkinkan koreksi terhadap output berdasarkan umpan balik yang didapatkan dari sensor, sehingga output dapat sesuai setpoint dalam jangka waktu yang relative cepat dan stabil, hal tersebut menyebabkan kendali *PID* dinilai lebih tepat digunakan untuk sistem kendali ketinggian yang perlu menjaga nilai ketinggian (continuous control), namun perlu penentuan nilai  $K_p$ ,  $T_i$ ,  $T_d$  yang tepat guna memaksimalkan hasil dari pengendalian untuk itu perlu dilakukan percobaan [1][2][3]. Percobaan yang dilakukan pada real plant, selain memerlukan biaya yang tinggi, juga sangat berpotensi menimbulkan masalah terhadap plant yang sedang beroperasi. Hal tersebut dapat menimbulkan kerugian bagi perusahaan. Untuk kepentingan percobaan maka diperlukan bantuan software yang akan digunakan sebagai alat untuk menjalankan percobaan, Factory IO merupakan software simulasi yang memiliki virtual plant 3 dimensi berupa tangki yang dapat digunakan untuk menjalankan simulasi pengendalian ketinggian air di dalam tangki, dengan menggunakan software ini pengendalian, serta respon real time dapat dilakukan seolah pengendalian dilakukan pada real plant [4][5][6], guna pengaplikasian kendaliler *PID* pada proses percobaan, software *LabVIEW* dapat digunakan sebagai alat untuk membuat kendaliler yang akan melakukan kendali terhadap virtual plant serta menjalankan proses akuisisi data yang dihasilkan oleh virtual plant, software *LabVIEW* menggunakan bahasa pemrograman berbasis grafis. Hal ini memberikan kemudahan dalam merancang/membuat program kendali untuk virtual plant [7][8][9][10][11].

## 2. METODE PENELITIAN

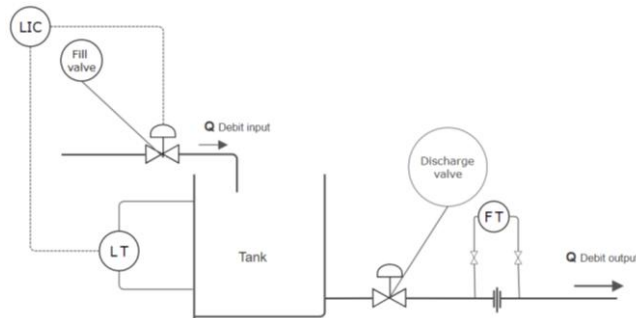
Penelitian ini dilakukan dengan metode eksperimental, dimana proses perancangan sistem kendali dilakukan dengan merancang virtual plant berupa tangki berisi air pada software Factory io dan kendaliler *PID* menggunakan software LabView, adapun langkah penelitian yang akan dilakukan mengacu kepada diagram alir sebagai berikut:



Gambar 2.1. Diagram Alir Penelitian

### 2.1 Perancangan *Virtual Plant*

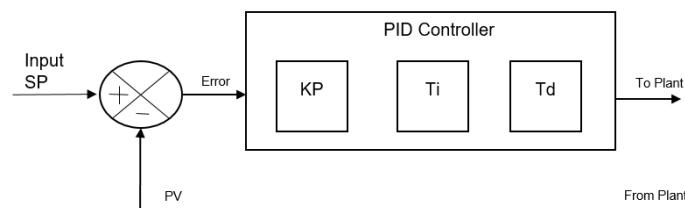
Perancangan virtual plant dilakukan dengan terlebih dahulu membuat diagram *P&ID*. Diagram ini akan memuat informasi terkait komponen-komponen serta loop control dari sebuah plant. Gambar 2.2 merupakan gambar *P&ID* dari virtual plant pengendalian level cairan pada tangki dalam *Factory IO* [12].



Gambar 2.2. Rancangan Diagram *P&ID Virtual Plant*

### 2.2 Perancangan *PID Controller*

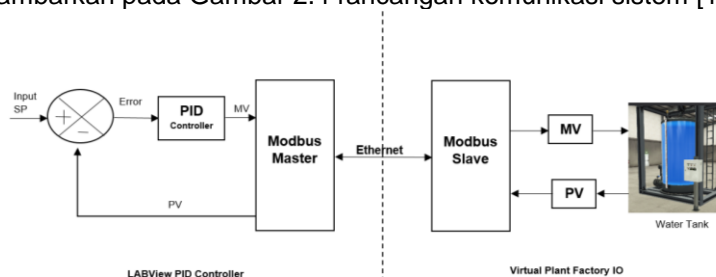
*PID controller* akan melaksanakan proses perhitungan guna menentukan output yang akan menjadi sinyal koreksi pada sistem pengendalian. *PID controller* terdiri atas algoritma pengendalian  $K_p$ ,  $T_i$ ,  $T_d$ . Algoritma tersebut memerlukan nilai untuk menjalankan proses perhitungan. Nilai untuk masing-masing algoritma pengendalian diberikan oleh user. Nilai-nilai tersebut akan dihitung beserta masukan setpoint (SP) yang diinginkan oleh user/operator serta *process value* (PV) yang didapat dari virtual plant yang dikendalikan. Gambar 2.3 merupakan diagram blok perancangan *PID controller* [1][3][13].



Gambar 2.3. Rancangan *PID controller*.

### 2.3 Perancangan Modul Komunikasi

Modul komunikasi digunakan untuk menghubungkan dua komponen penting yang digunakan dalam penelitian ini, yakni *PID controller* pada software LabVIEW serta virtual plant pada *FactoryTalk IO*. Modul komunikasi yang digunakan yakni protokol *Modbus TCP/IP*. Dengan menggunakan protokol *Modbus TCP/IP*, kedua komponen akan bertindak sebagai *master* dan *slave* seperti digambarkan pada Gambar 2.4 rancangan komunikasi sistem [14][15].



Gambar 2.4. Rancangan Modul Komunikasi *Modbus TCP/IP*

4

Dalam prosesnya modul komunikasi *Modbus* memiliki beberapa *function code* yang berbeda, Setiap *function code* mempunyai fungsi yang berbeda sesuai dengan tipe data dan jenis perintah [14], Tabel 2.1 menunjukkan *Function code* beserta variabel kendali yang digunakan dalam penelitian:

**Tabel 2.1.** Konfigurasi fungsi *Modbus TCP/IP*

No	Variabel	Function Code
1	Set point	Holding register
2	Fill valve	Holding register
3	Discharge valve	Holding register
4	Point Value	Holding register
5	Sensor ketinggian	Input register
6	Set point local	Input register
7	Flow sensor	Input register
8	Start light button	Coil
9	Stop light button	Coil
10	Alarm siren	Coil
11	Stack light green	Coil
12	Stack light yellow	Coil
13	Stack light red	Coil
14	Emergency button	Discrete input
15	Start button	Discrete input
16	Stop button	Discrete input
17	Selector remote	Discrete input
18	Selector local	Discrete input

### 3. HASIL DAN PEMBAHASAN

Dilakukan pengujian sistem untuk mengetahui performa dari sistem rancangan. Analisis yang dilakukan adalah dengan menguji respons tiap komponen sistem serta sistem secara keseluruhan dengan masukan sinyal step, kemudian menganalisis grafik respons dinamik yang terjadi.

#### 3.1 Pengujian *Virtual Plant* pada *Factory IO*

Pengujian rancangan *virtual plant* bertujuan untuk memastikan bahwa rancangan *virtual plant* dapat berjalan serta mendapatkan data-data terkait *virtual plant*. Pengujian *virtual plant* dilaksanakan dengan menerapkan 2 kondisi berbeda terhadap peralatan dalam *virtual plant*. Gambar 3.1 merupakan rancangan *virtual plant*. Rancangan *virtual plant* ini memiliki data spesifikasi sebagaimana Tabel 2.2.



**Gambar 3.1.** *Virtual Plant* Pengendalian Level Air dalam Tanki

**Tabel 3.1. Spesifikasi Virtual Plant Level Air**

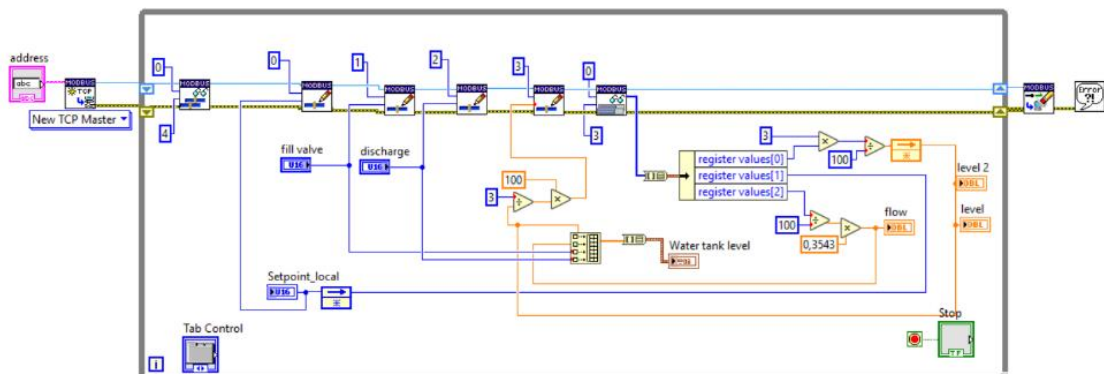
No	Spesifikasi	Nilai
1	Tinggi Tangki	3 meter
2	Diameter Tangki	2 meter
3	Maksimum <i>flow input</i>	0.25 m <sup>3</sup> /s
4	Maksimum <i>flow output</i>	0.354 m <sup>3</sup> /s
5	Sinyal <i>control valve input</i>	0 – 10 V
6	Sinyal <i>control valve output</i>	0 – 10 V
7	Sinyal sensor level	0 – 10 V
8	Sinyal sensor <i>flow output</i>	0 – 10 V / 0 - 0.354 m <sup>3</sup> /s

Pengujian terhadap *virtual plant* akan dilakukan dengan menggunakan software *LabVIEW* dengan protokol komunikasi *Modbus TCP/IP*, oleh karena itu perlu dilakukan pemrograman pada *Factory IO* dan *LabVIEW* sebelum dilakukan pengujian, Gambar 3.2 merupakan hasil pemrograman *Modbus slave* pada *virtual plant Factory IO*.



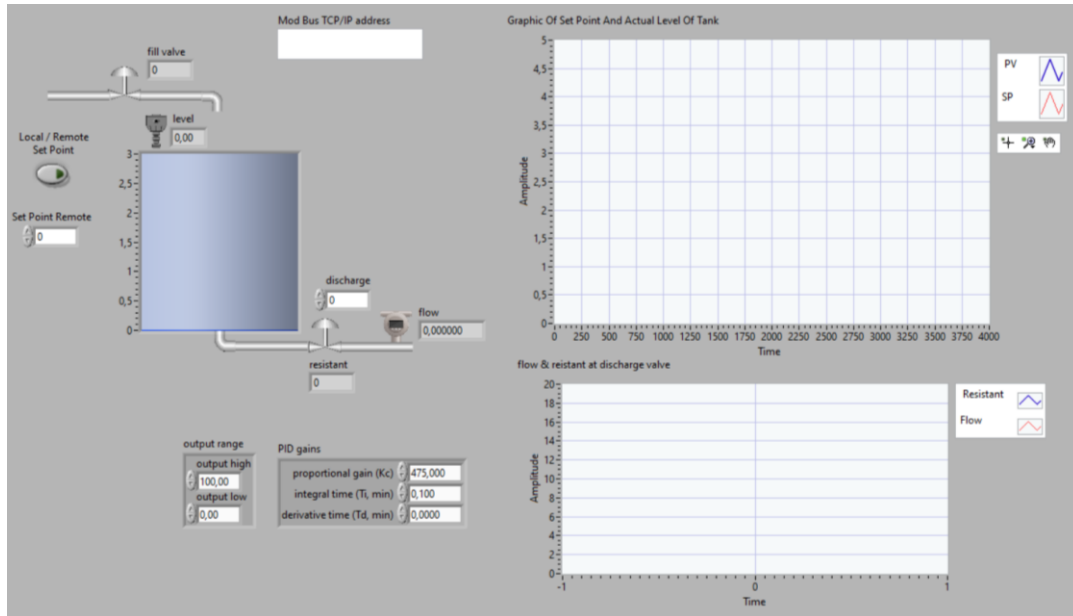
**Gambar 3.2. Konfigurasi Modbus Slave pada Virtual Plant di Factory IO**

Sedangkan program *LabVIEW* yang akan digunakan untuk pengujian *virtual plant* kondisi 1 dan 2, yakni seperti diperlihatkan pada Gambar 3.3, dirancang sebagai *Modbus master* agar dapat melakukan kendali serta akuisisi data terhadap rancangan *virtual plant* yang akan diuji.



**Gambar 3.3. Program LabVIEW Pengujian Virtual Plant Kondisi 1 dan 2**

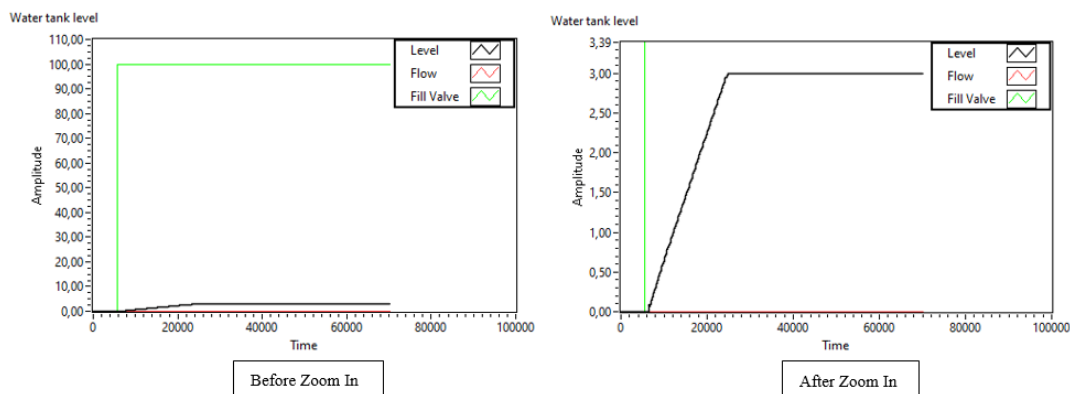
Data-data yang diterima dari *virtual plant* akan terekam oleh *LabVIEW* dan dimuat dalam bentuk grafik pada *front panel LabVIEW* seperti terlihat pada Gambar 3.4, yang selanjutnya data tersebut dapat di manfaatkan sebagai bahan analisa terkait sistem pengendalian level air dalam *virtual plant Factory IO*, selain itu proses *input set point*, parameter, *addressing modbus*, serta data *output* juga dilaksanakan dan ditampilkan pada *front panel LabVIEW* sehingga dalam hal ini *front panel LabVIEW* bertindak sebagai *HMI (human machine interface)* pada sistem pengendalian level air dalam *virtual plant Factory IO*.



Gambar 3.4. Front Panel LabVIEW Sistem Kendali Level Air dan HMI

### 1. Pengujian Kondisi 1

Pengujian kondisi 1 dilaksanakan terhadap *virtual plant* dengan memberikan kondisi *full closed* terhadap *discharge valve* dan kondisi *full open* terhadap *fill valve*.



Gambar 3.5. Grafik Pengujian Kondisi 1

Gambar 3.5 merupakan grafik data (program *LabVIEW*) yang di dapat ketika *virtual plant* diberikan sinyal *unit step* (bukaan penuh 100%) terhadap *fill valve* dan *discharge valve* ditutup, Setelah di amati terdapat reaksi yang lebih lambat dari pembacaan sensor ketika *fill valve* diberikan sinyal *unit step*, hal ini disebabkan konstruksi dari tangki dan *discharge valve* pada *virtual plant* yang memiliki jarak, sehingga fluida yang masuk pada tangki terlebih dahulu mengisi ruang antara tangki dan *discharge valve*, selain itu perubahan ketinggian terlihat linear seiring

dengan penambahan waktu yang dibutuhkan ketika fluida memenuhi tangki. Dengan menggunakan data spesifikasi dari *Factory I/O* serta data yang dihasilkan dari pengujian *virtual plant*, diperoleh data terkait kapasitas tangki dalam virtual plant seperti diperlihatkan pada Tabel 3.2.

**Tabel 3.1. Data Sampel Pengujian Virtual Plant Kondisi 1**

No	h1(m)	t1(ms)	h2(m)	t2(ms)	dh/dt	qi(m3/s)	kapasitas
1	0,03	151	0,3	442	0,092783505	0,25	2,694444444
2	0,21	345	0,42	577	0,090517241	0,25	2,761904762
3	0,42	577	0,6	791	0,08411215	0,25	2,972222222
4	0,6	791	0,81	1056	0,079245283	0,25	3,154761905
5	0,81	1056	1,02	1318	0,080152672	0,25	3,119047619
6	1,02	1318	1,2	1540	0,081081081	0,25	3,083333333
7	1,2	1540	1,41	1803	0,079847909	0,25	3,130952381
8	1,02	1318	2,01	2553	0,080161943	0,25	3,118686869
9	0,03	151	1,02	1318	0,084832905	0,25	2,946969697
10	0,03	151	3	3803	0,081325301	0,25	3,074074074
Nilai rata-rata kapasitas tanki							3,005639731

**2. Pengujian Kondisi 2**

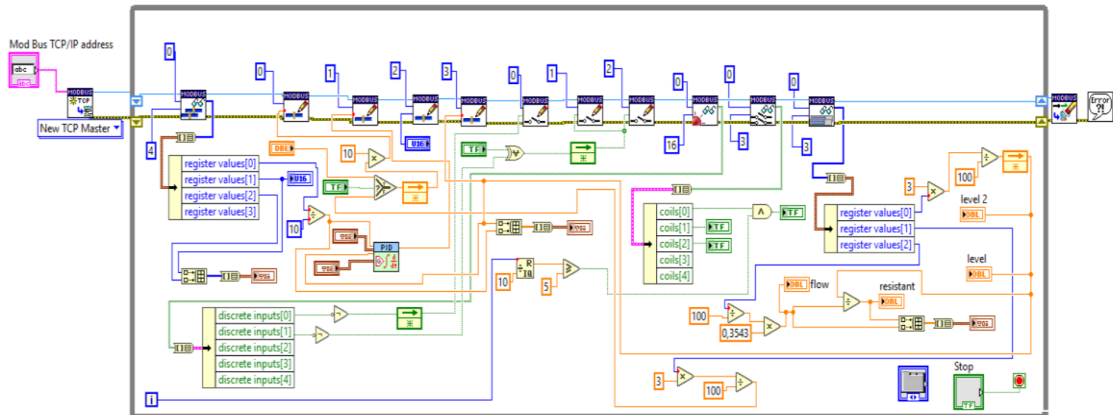
Pengujian terhadap *virtual plant* dilakukan dengan memberikan kondisi discharge valve terbuka dengan nilai bukaan 10% sampai dengan 100%, kemudian diberikan debit input dengan bukaan 100% terhadap fill valve. Pengujian dengan kondisi 2 akan memberikan hasil berupa data terkait respons valve pada rancangan virtual plant, serta dari data hasil pengujian akan didapat nilai resistansi pada discharge valve sesuai dengan posisi bukaannya. Pengujian dengan kondisi ini dilaksanakan sebanyak 10 kali, seperti diperlihatkan pada Tabel 3.3.

**Tabel 3.3. Hasil Pengujian Kondisi 2**

Pengujian ke-	Fill valve (%)	Discharge valve (%)	R ( $s/m^2$ )
1	100	10	101,21
2	100	20	54,079
3	100	30	38,681
4	100	40	26,816
5	100	50	21,534
6	100	60	18,223
7	100	70	15,918
8	100	80	13,203
9	100	90	10,336
10	100	100	8,3172

**3.2. Pengujian PID Controller**

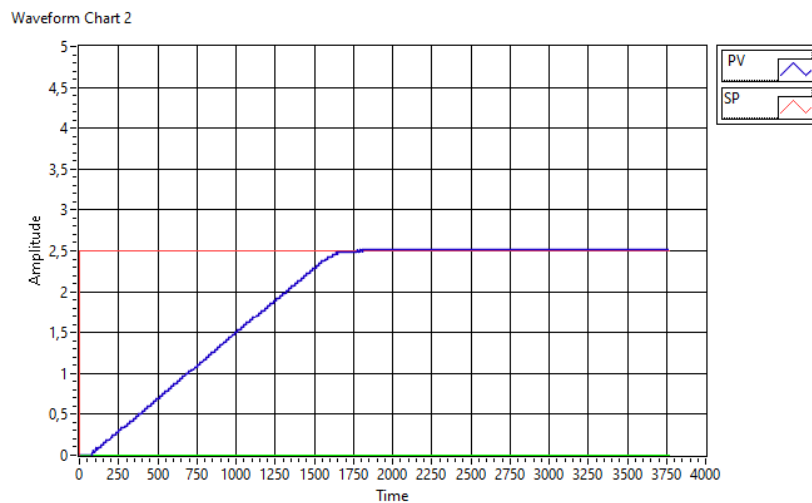
Pengujian *PID controller* terhadap *virtual plant* dilaksanakan dengan menggunakan metode *trial and error*, yaitu metode tuning parameter dengan menggunakan uji coba untuk memberikan penguatan dan melihat respons dari *virtual plant*. Parameter  $K_p$ ,  $T_i$ , dan  $T_d$  akan diberikan nilai, dan selanjutnya sistem kendali dijalankan dengan kondisi *setpoint* level sebesar 2,5 m. Pengujian dilakukan dalam 3 tahapan, yakni pengujian dengan nilai  $K_p$ ,  $K_p + T_i$  dan  $K_p + T_d$ . Gambar 3.6 merupakan program *LabVIEW* yang sudah dilengkapi dengan *PID controller*, program tersebut merupakan program yang akan di gunakan dalam pengendalian level air di dalam tangki pada *virtual plant Factory IO*.



Gambar 3.6. Program LabVIEW dengan PID Controller

### 1. Pengujian Pengendali P

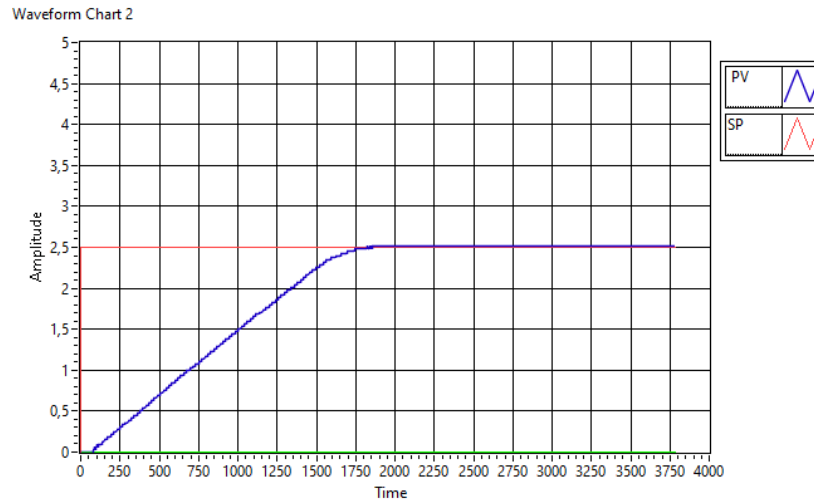
Pengujian ini dilaksanakan dengan memberikan nilai  $K_p$  terhadap Pengendali PID, selanjutnya dilakukan pengamatan terhadap respons *virtual plant*. Gambar 3.7 merupakan grafik hasil pengujian nilai  $K_p$  yang menghasilkan respon *virtual plant* terbaik.



Gambar 3.7. Grafik Pengujian dengan  $K_p=475$

### 2. Pengujian Pengendali PI

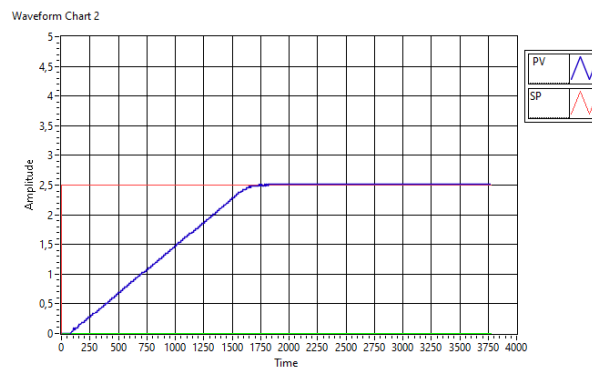
Pengujian ini dilaksanakan dengan memberikan nilai  $K_p$  457 dan nilai  $T_i$  terhadap *PID controller*, selanjutnya dilakukan pengamatan terhadap respons *virtual plant*. Gambar 3.8 merupakan grafik hasil pengujian nilai  $K_p$  yang menghasilkan respons *virtual plant* terbaik, dengan nilai  $K_p$  475 dan  $T_i$  0,1.



Gambar 3.8. Grafik Pengujian dengan  $K_p=475$ ,  $T_i=0.1$

### 3. Pengujian Pengendali PD

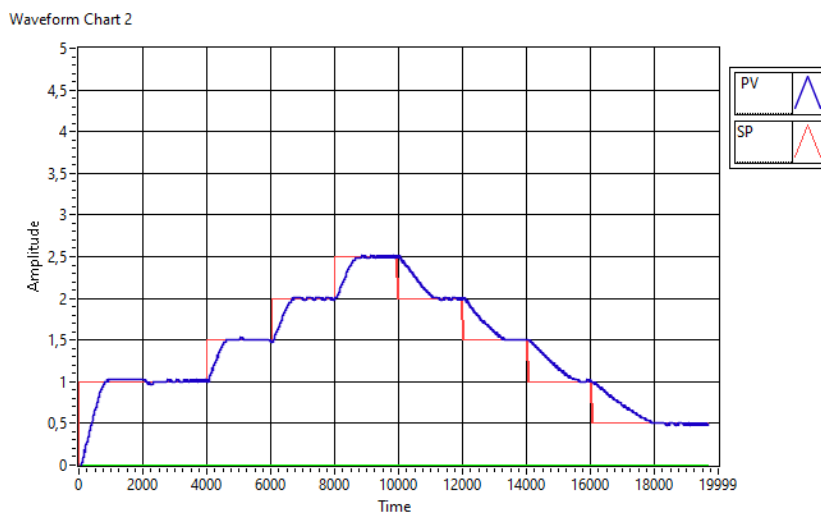
Pengujian ini dilaksanakan dengan memberikan nilai  $K_p$  dan nilai  $T_d$  terhadap  $PID$  controller, selanjutnya dilakukan pengamatan terhadap respons *virtual plant*. Gambar 3.9 merupakan grafik hasil pengujian nilai  $K_p$  dan  $T_d$  yang menghasilkan respons *virtual plant* terbaik, dengan nilai  $K_p$  475 dan  $T_d$  0,0001.



Gambar 3.9. Grafik Pengujian dengan  $K_p=475$ ,  $T_d=0.0001$

### 3.3 Pengujian Tracking Setpoint dengan Kendali PID

Pengujian *tracking setpoint* dilakukan agar diketahui keandalan dari suatu kendali PID apakah ketika *setpoint* sistem pengendalian tersebut dinaikkan atau diturunkan, sistem masih dapat mengikuti *setpoint*. Perbedaan beberapa *setpoint* pada pengujian yaitu 0,5 meter. Pengujian ini menetapkan nilai parameter untuk  $K_p$  sebesar 475,  $T_i$  sebesar 0,1 dan  $T_d$  sebesar 0,0001. Nilai tersebut didapat berdasarkan pengujian PID controller yang sudah dilakukan sebelumnya. Terdapat gangguan yang berupa bukaan tetap *discharge valve* sebesar 30% yang diterapkan setelah sistem mencapai *setpoint* pertama. Gambar 3.10 menunjukkan hasil pengujian.



Gambar 3.10. Grafik Pengujian Sistem dengan Tracking Setpoint

#### 4. KESIMPULAN

Berdasarkan penelitian yang dilaksanakan, dapat disimpulkan bahwa virtual plant pada FactoryTalk IO dapat digunakan sebagai simulator yang mampu menerima pengendalian berbasis PID melalui komunikasi Modbus TCP/IP. Komunikasi data antara virtual plant FactoryTalk IO (sebagai *slave*) dan *PID controller LabVIEW* (sebagai *master*) berjalan dengan baik secara *real time*, sehingga informasi yang berada dalam *virtual plant* dapat diterima dan diolah dengan baik oleh *PID controller LabVIEW*. Melalui *PID controller LabVIEW*, pengendalian terhadap *virtual plant* dapat dioptimalkan dengan memberikan nilai pada parameter  $K_p$ ,  $T_i$ , dan  $T_d$ , berdasarkan hasil penelitian dengan metode *trial and error*. Data terkait *respons plant* dapat diketahui, sehingga kita dapat menentukan nilai untuk parameter  $K_p$ ,  $T_i$ , dan  $T_d$  yang paling sesuai dengan *plant* yang akan dikontrol. Setelah ditemukan nilai yang sesuai untuk pengendali PID, metode *tracking setpoint* dapat digunakan untuk menguji keandalan dari *PID controller* yang digunakan dalam sistem. Berdasarkan hasil simulasi dengan pengendalian PID, dengan parameter kendali  $K_p = 475$ ,  $T_i = 0.1$ ,  $T_d = 0.0001$ , terlihat bahwa *respons sistem* dapat mengikuti *setpoint* yang diberikan oleh pengendali PID.

#### DAFTAR PUSTAKA

- [1] A. S. Maharani, Sumardi, and B. Setiyono, "Aplikasi Kontrol Pid Untuk Pengendalian Ketinggian Level Cairan Dengan Menggunakan Tcp/Ip," *Tek. Elektro Univ. Diponegoro*, pp. 1–7, 2007.
- [2] J. Dewanto, Y. T. D. Suprpto, and H. R. Candra, "Pemodelan Sistem Pengatur Ketinggian Air pada Sebuah Tangki Tunggal," *J. Tek. Mesin*, vol. 2, no. 1, pp. 35–40, 2000.
- [3] N. R. Wibowo and R. Praminasari, "Implementasi Pengendali PID Digital pada FPGA," no. November. pp. 106–110, 2016.
- [4] H. Vargas, R. Heradio, M. Donoso, and G. Farias, "Teaching automation with *Factory I/O* under a competency-based curriculum," *Multimed. Tools Appl.*, vol. 82, no. 13, pp. 19221–19246, 2023, doi: 10.1007/s11042-022-14047-9.
- [5] K. J. Alaameri, A. J. Ramadhan, A. S. Maklakov, and V. R. Gasiyarov, "Design of PID Controller for Liquid Level in Tanks Using Tia Portal and *FACTORY I/O* Programs," *AIP Conf. Proc.*, vol. 2977, no. 1, 2023, doi: 10.1063/5.0181881.
- [6] B. Salah, A. M. Alsamhan, W. Saleem, R. Khan, and A. T. A. Soliman, "3D Simulation of a Yogurt Filling Machine Using Graficet Studio and *Factory Io*: Realization of Industry 4.0," *Trans. Famena*, vol. 47, no. 3, pp. 15–30, 2023, doi: 10.21278/TOF.473049922.
- [7] J. D. Wagoner and N. F. Macia, "Automatic liquid level controller using a *LabView* based PC," *ASEE Annu. Conf. Proc.*, pp. 1021–1035, 2004.

11

- [8] Ardilessi and J. G. Kara, "Pengendalian Volume Tangki Menggunakan *LabView* dan Arduino UNO Tank Volume Control Using *LabView* and Arduino UNO," *Telekontran*, vol. 3, no. 2, pp. 28–35, 2015.
- [9] S. Buwarda, O. Sistem Permesinan, and P. ATI Makassar, "Development of DCS SCADA Teaching Module," pp. 89–96, 2022.
- [10] I. D. Pranowo and D. Artanto, "Improved control and monitor two different PLC using *LabView* and NI-OPC server," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 4, pp. 3003–3012, 2021, doi: 10.11591/ijece.v11i4.pp3003-3012.
- [11] D. S. Pongoh, M. Budiman, D. Kantohe, and V. Lumentut, "Implementasi Software *LabView* untuk Mendukung Sistem Smart Home Berbasis Mikrokontroler," *Pros. Ind. Res. Work. Natl. Semin.*, vol. 12, pp. 394–400, 2021, [Online]. Available: <https://jurnal.polban.ac.id/ojs-3.1.2/proceeding/article/view/2725>
- [12] S. Rahmani, S. A. Rosana, and G. H. Tian, "Sistem Kontrol Level Air Dengan Pengontrol PID," *Telekontran J. Ilm. Telekomun. Kendali dan Elektron. Terap.*, vol. 10, no. 2, pp. 174–181, 2023, doi: 10.34010/telekontran.v10i2.9330.
- [13] B. N. Getu, "Water level controlling system using PID controller," *Int. J. Appl. Eng. Res.*, vol. 11, no. 23, pp. 11223–11227, 2016.
- [14] T. Tosin, "Perancangan dan Implementasi Komunikasi RS-485 Menggunakan Protokol Modbus RTU dan Modbus TCP Pada Sistem Pick-By-Light," *Komputika J. Sist. Komput.*, vol. 10, no. 1, pp. 85–91, 2021, doi: 10.34010/komputika.v10i1.3557.
- [15] B. Y. Pratama and E. Anisa, "Analisis Implementasi Komunikasi Modbus TCP/IP dalam Penerapan Visualisasi Data Hasil Produksi pada Sistem Andon Line Production," *J. Asimetrik J. Ilm. Rekayasa Inov.*, vol. 5, pp. 43–52, 2023, doi: 10.35814/asimetrik.v5i1.3791.