

## PENERAPAN KONTROLER PID PADA SISTEM PENGATUR KETINGGIAN AIR BERBASIS LABVIEW

Arjuna Sutanto<sup>1</sup>, Harlianto Tanudjaja<sup>2</sup>  
<sup>1,2</sup>Jurusan Teknik Elektro - Fakultas Teknik  
 Universitas Katolik Indonesia Atma Jaya – Jakarta  
 email: <sup>1</sup>arjuna.sutanto@atmajaya.ac.id, <sup>2</sup>harlianto.tanudjaja@atmajaya.ac.id

### ABSTRAK

Perkembangan teknologi yang pesat pada zaman sekarang berimbas pula terhadap perkembangan alat di sebuah industri. Peralatan di sebuah industri yang dulunya digerakkan manual oleh manusia kini mulai dikendalikan secara otomatis. Salah satu sistem kontroler otomatis yang banyak digunakan di sebuah industri adalah sistem kontroler *proportional integral derivative* (PID). kontroler PID terdiri dari tiga macam metode kontroler yaitu kontroler proporsional (*proportional controller*), kontrolerintegral (*integral controller*), dan kontroler turunan (*controller derivative*).

Pengontrolan sistem ketinggian dilakukan melalui LabView 2013. Boards Arduino UNO digunakan untuk mengolah data pada sensor *flow* dan sensor ketinggian. Data yang diperoleh dari sensor ketinggian dan sensor *flow* digunakan untuk mengatur masukan dan keluaran tangki air agar ketinggian air sesuai dengan ketinggian yang diinginkan.

Dari hasil pengujian, dengan menggunakan parameter  $T_i$  sebagai input aksi integral dalam metode zigler-nichols. Kondisi osilasi stabil dan konstan pada keluaran level ketinggian akan diperoleh bila kondisi parameter nilai  $T_i$  yang diberikan semakin lebih besar dan nilai parameter  $K_p$  yang diberikan semakin lebih kecil. Dengan menggunakan parameter yang didapat dari perhitungan menggunakan metode zigler – Nichols, diperoleh hasil keluaran lebih cepat stabil bila menggunakan kontroler PI.

**Kata kunci:** Kontroler, *Proportional Integral Derivative*(PID), Ziegler-Nichols, LabView,

### ABSTRACT

*Rapid technological developments in contemporary times also affected the development of the tools in an industry. Equipment in an industry that was once driven manually by humans are now beginning to be controlled automatically. One of the automatic controller system that is widely used in an industrial controller system is proportional integral derivative (PID). PID controller consists of three kinds of methods that the controller proportional controller, the integral controller, and derivative controller. Controlling the height levels of the system is done through LabView 2013. Arduino UNO board is used to process the data on the flow sensor and height sensor. Data obtained from the height sensor and flow sensor is used to set the input and output water tank so that the water level in accordance with the desired height. From the test results, using  $T_i$  as an input parameter in the integral action zigler – Nichols method. Condition stable and constant oscillation at the output level elevation will be obtained, when the conditions given parameter value  $T_i$  is getting bigger and parameter values  $K_p$  given increasingly smaller. By using the parameters obtained from the calculation using the method zigler – Nichols, obtained results faster output stable when using a PI controller.*

**Keywords ;** Controller, *Proportional Integral Derivative*, Ziegler-Nichols, LabView,

### PENDAHULUAN

Perkembangan teknologi yang pesat pada zaman sekarang

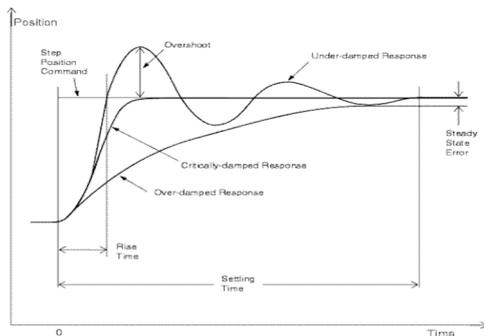
menyebabkan peralatan di sebuah industri yang dulunya digerakkan manual oleh manusia kini mulai

dikendalikan secara otomatis. Salah satu sistem kontroler otomatis yang banyak digunakan di sebuah industri adalah sistem kontroler *proportional integral derivative* (PID). kontroler PID terdiri dari tiga macam metode kontroler yaitu kontroler proporsional (*proportional controller*), kontrolerintegral (*integral controller*), dan kontroler turunan (*controller derivative*).

## Landasan Teori

### A. Respon system

Respon sistem adalah perubahan perilaku keluaran terhadap perubahan sinyal masukan menjadi dasar untuk meng-analisa karakteristik sistem selain menggunakan persamaan atau model matematika.



Gambar 1 Bentuk sinyal tanggapan system

### B. *Proportional Integral Derivative* (PID)

*Proportional Integral Derivative* (PID) merupakan salah satu metode dalam sistem kontrol. PID terdiri dari 3 kontroler yaitu kontroler proporsional, kontroler integral, dan kontroler turunan. Masing-masing aksi kontrol mempunyai keunggulan-keunggulan tertentu, dimana aksi kontrol proporsional mempunyai keunggulan *rise time* yang cepat, aksi

kontrol integral mempunyai keunggulan untuk memperkecil *error* dan aksi kontrol turunan mempunyai keunggulan untuk meredam *overshoot* atau *undershoot*. Untuk itu agar kita dapat menghasilkan keluaran dengan *rise time* yang cepat dan *error* yang kecil didapat dengan menggabungkan ketiga aksi kontrol ini menjadi aksi kontrol PID. Model matematikanya seperti terlihat pada persamaan 1.

$$CO = K_p \cdot E + K_I \int E \cdot dt + K_d \frac{dE}{dt} \quad (1)$$

Kontroler proporsional pada dasarnya merupakan penguatan dengan penguatan yaang dapat diatur. Hubungan antara keluaran kontroler  $y(t)$  dan sinyal kesalahan penggerak  $e(t)$  diperlihatkan pada persamaan 2.

$$y(t) = K_p \cdot e(t) \quad (2)$$

Kontroler integral berfungsi menghasilkan respon sistem yang memiliki kesalahan keadaan mantap nol. Kalau sebuah plant tidak memiliki unsur *integrator* ( $1/s$ ), kontroler *proporsional* tidak akan mampu menjamin keluaran sistem dengan kesalahan keadaan mantabnya nol. Pada kontroler integral, harga keluaran kontroler  $y(t)$  diubah dengan laju yang sebanding dengan sinyal kesalahan penggerak  $e(t)$ . Modelnya dapat dilihat pada persamaan 3. Konstanta integral  $K_i$  yang berharga besar akan mempercepat hilangnya *offset*. Tetapi semakin besar nilai konstanta  $K_i$  akan mengakibatkan peningkatan osilasi dari sinyal keluaran kontroler

$$y(t) = K_i \int_0^t e(t) \quad (3)$$

Kontroler turunan memiliki sifat seperti halnya suatu operasi

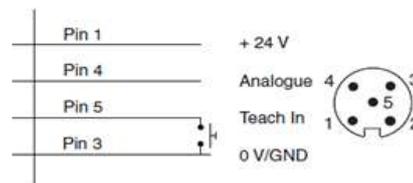
diferensial. Perubahan yang mendadak pada masukan kontroler, akan mengakibatkan perubahan yang sangat besar dan cepat. Kontroler turunan umumnya dipakai untuk mempercepat respon awal suatu sistem, tetapi tidak memperkecil kesalahan pada keadaan tunaknya. Kerja kontroler turunan hanyalah efektif pada lingkup yang sempit, yaitu pada periode peralihan. Oleh sebab itu kontroler diferensial tidak pernah digunakan tanpa ada kontroler lain sebuah sistem.

$$y(t) = Kd \frac{d}{dt} e(t) \quad (4)$$

### C. Sensor

*Level* sensor yang digunakan dalam percobaan ini adalah sensor ultrasonik. Sensor ultrasonik merupakan sensor yang memanfaatkan pancaran gelombang ultrasonik. Sensor ultrasonik terdiri dari rangkaian pemancar ultrasonik yang disebut *transmitter* dan rangkaian penerima ultrasonik yang disebut *receiver*. Sinyal ultrasonik yang dibangkitkan akan dipancarkan dari *transmitter* ultrasonik. Ketika sinyal mengenai benda penghalang, maka sinyal ini dipantulkan, dan diterima oleh *receiver* ultrasonik. Sinyal yang diterima oleh rangkaian *receiver* dikirimkan ke rangkaian mikrokontroler untuk selanjutnya diolah untuk menghitung jarak terhadap benda di depannya (bidang pantul). Gambar 2 menunjukkan susunan kaki sensor ketinggian yang digunakan Untuk menghitung jarak yang terukur dari waktu terima sensor dapat menggunakan Persamaan 5.

$$S = \frac{V \cdot t}{2} \quad (5)$$



Gambar 2 Susunan kaki sensor ketinggian tipe PK104141

Sensor *Flow* adalah sensor yang ber-fungsi mengukur laju/kecepatan aliran dari suatu cairan. Laju aliran adalah kecepatan *fluida* yang mengalir dalam suatu penampang tertentu. Laju aliran dapat dicari dengan Persamaan 6,

$$A_1 v_1 = A_2 v_2 \quad (6)$$

Keterangan:

$A_1$  = Luas penampang 1 ( $m^2$ )

$v_1$  = Laju aliran 1 ( $m/s$ )

$A_2$  = Luas penampang 2 ( $m^2$ )

$v_2$  = Laju aliran 2 ( $m/s$ )

Dengan mengetahui laju aliran, dapat menghitung debit air dengan menggunakan Persamaan 8

$$Q = vA \quad (7)$$

$$Q = V/t \quad (8)$$

Keterangan:

$Q$  = Debit air ( $m^3/s$ )

$A$  = Luas penampang pipa ( $m^2$ )

$v$  = Laju aliran air ( $m/s$ )

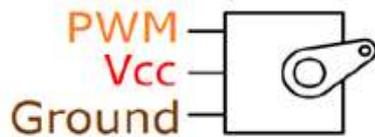
$V$  = Volume air ( $m^3$ )

$t$  = Waktu (s)

### D. MG995 Servo Motor

Servo motor adalah sebuah motor yang dirancang dengan sistem control umpan balik *loop* tertutup, sehingga dapat diatur untuk menentukan dan memastikan posisi

sudut dari poros keluaran motor dan digunakan sebagai pengendali buka atau tutup keran (valve). MG995 merupakan servo yang bisa berputar  $120^\circ$  ( $60^\circ$  disetiap arah). MG995 memiliki tiga keluaran yaitu, PWM, Vcc dan Ground. Gambar 3 menunjukkan rangkaian servo motor MG995.

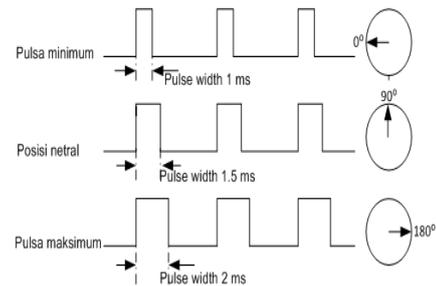


Gambar 3 Rangkaian servo motor MG995

Motor servo dikendalikan dengan memberikan sinyal modulasi lebar pulsa (*Pulse Width Modulation* (PWM)) melalui kabel kontrol. Lebar pulsa sinyal kontrol yang diberikan akan menentukan posisi sudut putaran dari poros motor servo. Sebagai contoh, lebar pulsa dengan waktu 1,5 ms (mili detik) akan memutar poros motor servo ke posisi sudut  $90^\circ$ . Bila pulsa lebih pendek dari 1,5 ms maka akan berputar ke arah posisi  $0^\circ$  atau ke kiri (berlawanan dengan arah jarum jam), sedangkan bila pulsa yang diberikan lebih lama dari 1,5 ms maka poros motor servo akan berputar ke arah posisi  $180^\circ$  atau ke kanan (searah jarum jam). Lebih jelasnya perhatikan Gambar 4.

Ketika lebar pulsa kendali telah diberikan, maka poros motor servo akan bergerak atau berputar ke posisi yang telah diperintahkan, dan berhenti pada posisi tersebut dan akan tetap bertahan pada posisi tersebut. Jika ada kekuatan eksternal yang mencoba memutar atau mengubah posisi tersebut, maka

motor servo akan mencoba menahan atau melawan dengan besarnya kekuatan torsi yang dimilikinya (rating torsi servo). Namun motor servo tidak akan mempertahankan posisinya untuk selamanya, sinyal lebar pulsa kendali harus diulang setiap 20 ms (mili detik) untuk menginstruksikan agar posisi poros motor servo tetap bertahan pada posisinya.

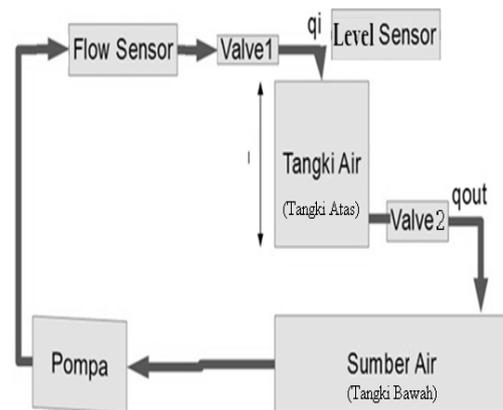


Gambar 4 *Pulse Wide Modulation*

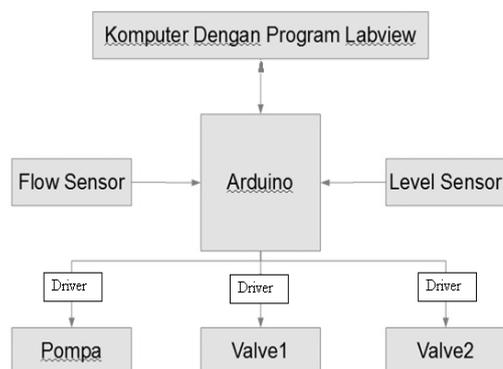
## PERANCANGAN SISTEM

### A. Blok Diagram Sistem

Simulasi sistem PID akan direalisasikan dengan model seperti pada Gambar 5. Gambar 6 menunjukkan blok diagram sistem.



Gambar 5 Model Sistem



Gambar 6 Blok Diagram Sistem

Sistem terdiri dari 3 masukan dan 3 keluaran. Masukan terdiri dari sensor *flow*, sensor ketinggian dan LabView. Keluaran terdiri dari pompa, valve1, valve2.

Cara kerja sistem:

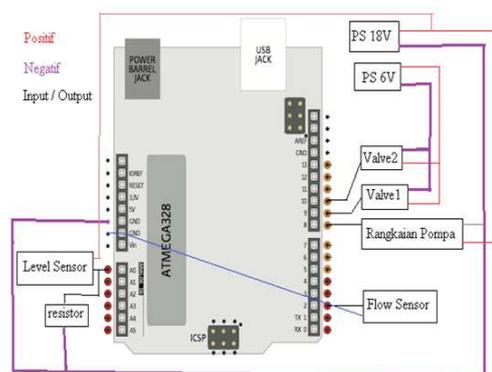
1. Proses akan dijalankan saat tombol start ditekan. Ketika tombol start ditekan, pompa akan dinyalakan.
2. Sistem akan melakukan pengecek-an terhadap set air. Bila set air lebih besar ketinggian air, Valve1 akan terbuka. Bila set air lebih kecil atau sama dengan ketinggian air, Valve1 akan tetap tertutup.
3. Sistem akan mengecek Valve2. Jika Valve2 dalam keadaan terbuka, sistem akan mengecek set air seperti langkah no. 2. Jika Valve2 dalam keadaan tertutup, sistem akan melakukan langkah no. 2 hingga program dimatikan

## B. Perancangan Perangkat Keras

### B.1 Rangkaian Pemroses Arduino

Rangkaian Arduino Uno mempunyai fungsi sebagai pengendali dan pengolah data keseluruhan sistem. Arduino Uno adalah sebuah modul mikrokontroler yang dibuat berdasarkan ATmega328. Modul ini terdiri dari

14 pin digital masukan atau keluaran (yang 6 diantaranya dapat digunakan sebagai keluaran *Pulse Width Modulation* (PWM)), 6 analog masukan. Gambar 7 menunjukkan rangkaian arduino.



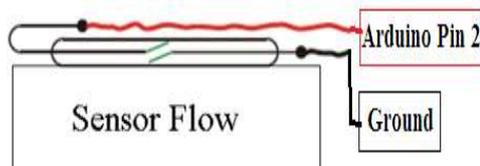
Gambar 7 Rangkaian arduino

*Port-port* yang digunakan dalam Gambar 7:

1. *Port2* dihubungkan dengan keluar-an *flow sensor* / sensor aliran
2. *Port 8* dihubungkan dengan keluar-an dari SSR yang akan mengaktif-kan atau mematikan pompa
3. *Port9* dan *port 10* dihubungkan dengan keluaran dari motor servo untuk mengatur posisi *valve*
4. *Port analog0* dihubungkan dengan keluaran *level sensor* / sensor ketinggian

### B.2 Rangkaian Sensor *Flow*

Sensor *flow* berfungsi untuk memberikan informasi kecepatan aliran air yang masuk kedalam tangki air.

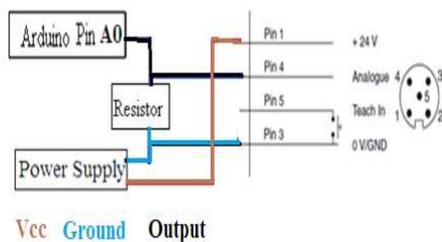
Gambar 8 Rangkaian sensor *flow* tipe *WTM multi jet water meter*.

Rangkaian sensor terdiri dari 2 keluaran, yaitu:

1. Kabel merah sebagai *pulse*. Kabel merah dihubungkan pada Arduino Pin 2.
2. Kabel hitam sebagai *ground*. Kabel hitam dihubungkan pada Arduino Pin ground.

### B.3 Rangkaian Sensor Level

Sensor ketinggian berfungsi untuk memberikan informasi ketinggian air yang ada didalam tangki air. Gambar 9 menunjukkan rangkaian sensor ketinggian.



Vcc Ground Output

Gambar 9 Rangkaian sensor ketinggian

Rangkaian sensor terdiri dari 4 keluaran, yaitu:

1. Kabel merah sebagai Vcc (+24V). Kabel hitam dihubungkan pada *power supply*
2. Kabel hitam sebagai *pulse*. Kabel hitam dihubungkan pada Arduino Pin analog 0.
3. Kabel merah muda sebagai *Teach in*.
4. Kabel biru sebagai *ground*. Kabel biru dihubungkan pada *power supply*.

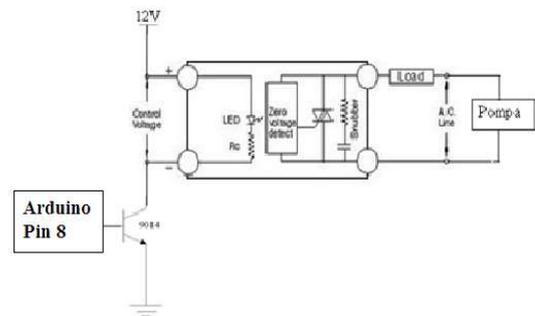
### B.4 Rangkaian Pompa

**Solid state relay (SSR)** adalah sebuah saklar elektronik yang tidak memiliki bagian yang bergerak. SSR dapat didiskripsikan sebagai berikut:

1. Mempunyai empat buah terminal, 2 masukan terminal dan 2 buah keluaran terminal

2. Tegangan masukan dapat berupa tegangan AC atau DC
3. Antara keluaran dan masukan diisolasi dengan sistem optikal
4. Keluaran menggunakan keluarga thyristor, SCR untuk beban DC dan TRIAC untuk beban AC

Rangkaian pompa terdiri dari SSR. SSR digunakan untuk menghidupkan atau mematikan pompa. SSR akan mendapat masukan dari Arduino Pin 8. Gambar 10 menunjukkan rangkaian pompa.



Gambar 10 Rangkaian pompa

### B.4 Rangkaian Valve1

Rangkaian valve1 menggunakan motor servo MG995 untuk menggerakkan valve1. Motor servo untuk valve1 mendapat masukan dari Arduino Pin 9. Gambar 11 menunjukkan rangkaian valve1.



Gambar 11 Rangkaian valve1

### B.5 Rangkaian Valve2

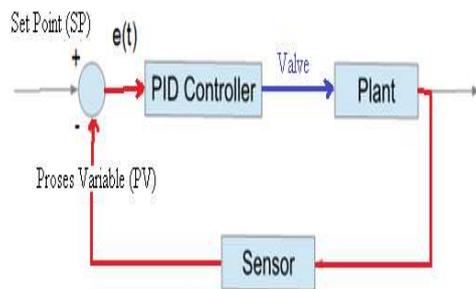
Rangkaian valve2 menggunakan motor servo MG995 untuk menggerakkan valve2. Motor servo untuk valve2 mendapat masukan dari Arduino Pin 10. Gambar 12 menunjukkan rangkaian valve2.



Gambar 12 Rangkaian valve2

### B.6 Metode PID

Metode PID dapat bekerja dengan baik ditentukan oleh nilai-nilai parameter  $K_p$ ,  $K_i$ ,  $K_d$  yang sesuai. Untuk mendapatkan parameter yang sesuai perlu digunakan metode *tuning*. Metode *tuning* PID yang paling umum digunakan adalah metode Ziegler-Nichols metode 1 dan metode 2. Ziegler-Nichols metode 1 digunakan untuk sistem siklus terbuka dan Ziegler-Nichols metode 2 digunakan untuk sistem siklus tertutup. Blok diagram PID siklus tertutup dapat dilihat pada Gambar 13.



Gambar 13 Blok Diagram PID Sistem Siklus Tertutup

Pada penelitian ini digunakan metode Ziegler-Nichols metode 2 dan Langkah-langkah dalam menggunakan Ziegler-nichols metode 2:

1. Proses dalam keadaan stabil (tidak ada perubahan pada proses variabel)
2. Hentikan osilasi dengan menggunakan mode manual jika proses variabel dalam keadaan osilasi

3. Hilangkan kontroler *integral*
4. Jika menggunakan *integraltime* ( $T_i$ ), berikan nilai yang sangat besar pada parameter *integral* (contoh: 9999). Jika menggunakan *integralgain* ( $K_i$ ), berikan nilai nol pada parameter *integral*
5. Hilangkan kontroler *derivative* ( $K_d=0$  atau  $T_d=0$ )
6. Buat perubahan pada *set point* dan catat hasil keluaran
7. Naikkan nilai  $K_p/K_c$
8. Jika proses variabel tidak beresilasi, perbesar nilai  $K_p/K_c$
9. Jika ketinggian puncak dari proses variabel menurun, perbesar nilai  $K_p/K_c$
10. Jika ketinggian puncak dari proses variabel meningkat, perkecil nilai  $K_p/K_c$
11. Jika proses variabel atau kontroler keluaran menyentuh batas maksimum dan minimum, perkecil nilai  $K_p/K_c$
12. Jika osilasi berhenti, ulangi langkah 6
13. Jika terjadi osilasi tetapi tidak dengan ketinggian konstan, ulangi langkah 8 dan 9
14. Jika terjadi osilasi dengan ketinggian konstan, dan proses variabel dan kontroler keluaran tidak menyentuh batas maksimum dan minimumnya, catat nilai  $K_u$  dan  $T_u$ . (Nilai  $K_u$  dan  $T_u$  yang didapat dalam percobaan dimasukkan kedalam Tabel 2)

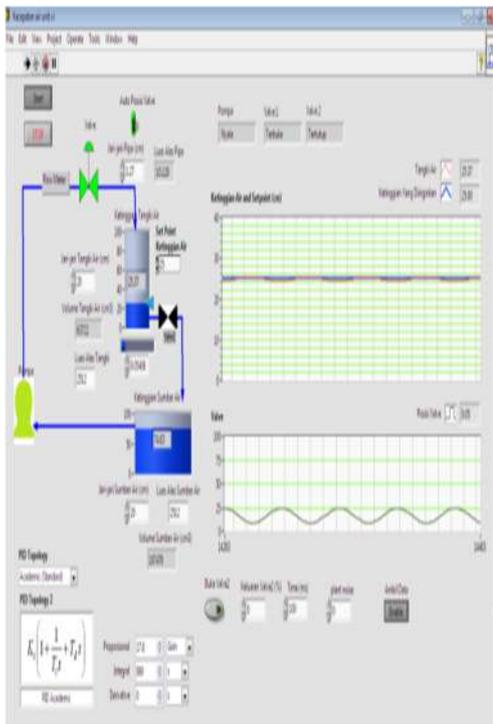
Tabel 2 Ziegler-Nichols Metode 2

Control Tipe	$K_p$	$T_i$	$T_d$
P	$0.5K_u$	$\infty$	0
PI	$0.45K_u$	$0.83T_u$	0
PID	$0.6K_u$	$0.5T_u$	$0.125T_u$

**PENGUJIAN SISTEM**

Pada percobaan ini nilai  $T_i$  yang diberikan adalah 999, 9999, 99999 dan 999999. Perbedaan dari nilai  $T_i$  pada percobaan dilakukan untuk melihat pengaruh yang ditimbulkan nilai  $T_i$ .

Pada percobaan 1 yang dapat dilihat pada Gambar 14 dengan  $T_i = 999s$ , keluaran mengalami osilasi konstan pada  $K_c = 17.8$  dengan  $T_u = 4.5s$ . Data pengujian dimasukkan kedalam tabel Ziegler-Nichols didapatkan nilai  $K_p$ ,  $T_i$  dan  $T_d$  yang dapat dilihat pada Tabel 3.



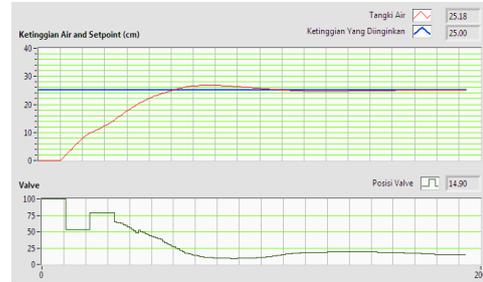
Gambar 14 Percobaan 1 dengan  $T_i = 999s$

Tabel 3  $K_p$ ,  $T_i$ ,  $T_d$  percobaan 1

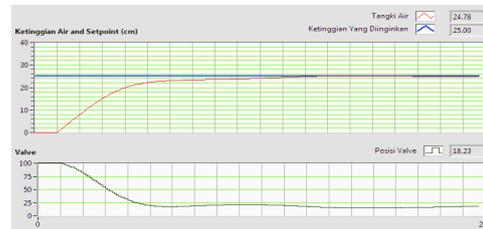
PID Tipe	$K_p$	$T_i$	$T_d$
P	8.9	$\infty$	0
PI	8.01	3.735	0
PID	10.68	2.25	0.5625

Pengambilan data baru dengan data dari Tabel 2. Data hasil PID *ideal*

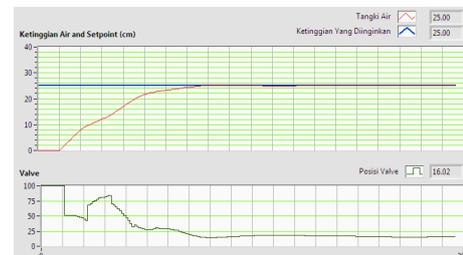
dapat dilihat pada Gambar 15 dan Gambar 16. Data hasil PID *series* dapat dilihat pada Gambar 17 dan Gambar 18.



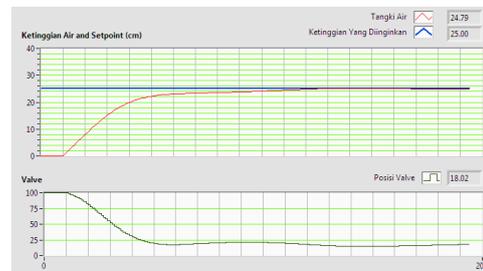
Gambar 15 Hasil pengujian PID tipe Ideal



Gambar 16 Hasil percobaan PID *ideal* tipe PI

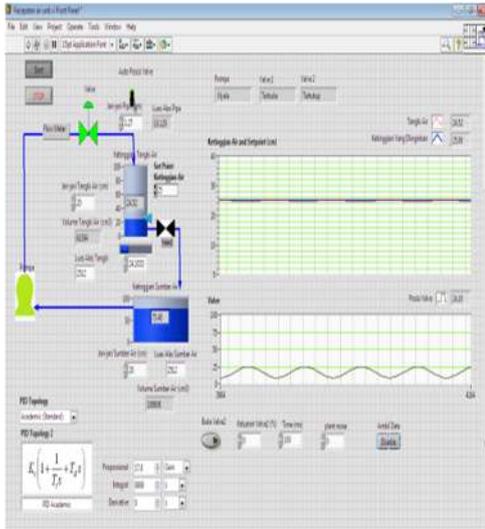


Gambar 17. Hasil Percobaan PID *series* tipe PID



Gambar 18 Hasil percobaan PID *series* tipe PI

Pada percobaan 2 yang dapat dilihat pada Gambar 19 dengan nilai  $T_i = 9999s$  dan  $K_c = 17.8$  diperoleh  $T_u = 6.4s$ . Data pengujian dimasukkan kedalam tabel Ziegler-Nichols didapatkan nilai  $K_p$ ,  $T_i$  dan  $T_d$  yang dapat dilihat pada Tabel 4.

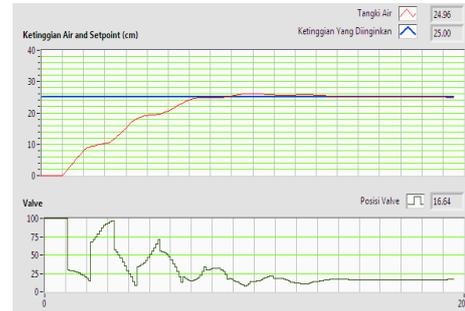


Gambar 19 Percobaan 2 dengan  $T_i = 9999s$

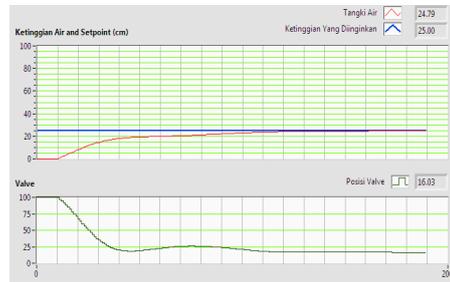
Tabel 4  $K_p$ ,  $T_i$ ,  $T_d$  percobaan 2

PID Tipe	$K_p$	$T_i$	$T_d$
P	8.9	$\infty$	0
PI	8.01	5.312	0
PID	10.68	3.2	0.8

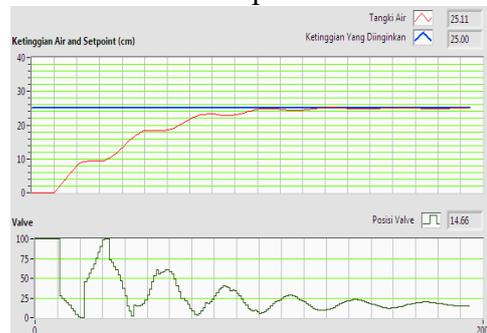
Dilakukan pengambilan data baru dengan menggunakan data dari Tabel 3. Data dimasukkan pada algoritma PID *ideal* dan PID *series*. Data hasilnya PID *ideal* dapat dilihat pada Gambar 23 dan Gambar 24. Data hasilnya PID *series* dapat dilihat pada Gambar 25 dan Gambar 26.



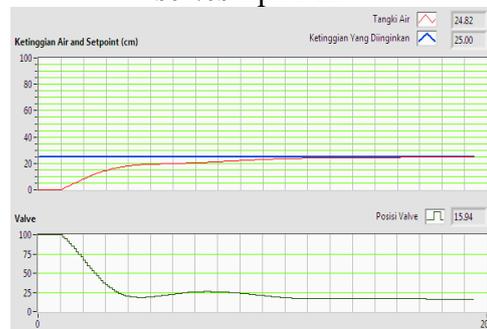
Gambar 20 Hasil percobaan PID *ideal* tipe PID



Gambar 21 Hasil percobaan PID *ideal* tipe PID

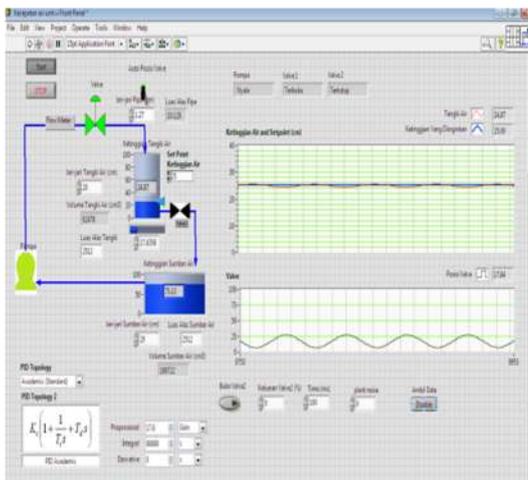


Gambar 22 Hasil percobaan PID *series* tipe PID



Gambar 23 Hasil percobaan PID *series* tipe PI

Pada percobaan 3 yang dapat dilihat pada Gambar 27 dengan nilai  $T_i = 99999s$  dan  $K_c = 17.8$  diperoleh  $T_u = 5.5s$ . Data pengujian dimasukan kedalam tabel Ziegler-Nichols di-dapatkan nilai  $K_p$ ,  $T_i$  dan  $T_d$  yang dapat dilihat pada Tabel 4.

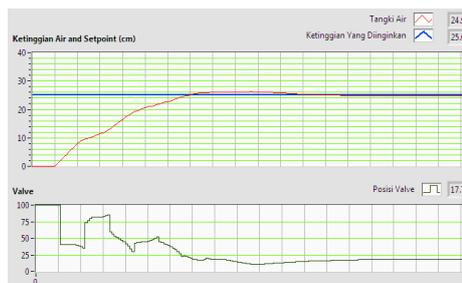


Gambar 24 Percobaan 3 dengan  $T_i = 99999$

Tabel 5  $K_p$ ,  $T_i$ ,  $T_d$  percobaan 3

PID Tipe	$K_p$	$T_i$	$T_d$
P	8.9	$\infty$	0
PI	8.01	4.565	0
PID	10.68	2.75	0.6875

Dilakukan pengambilan data baru dengan menggunakan data dari Tabel 5. Data dimasukan pada algoritma PID *ideal* dan PID *series*. Data hasilnya PID *ideal* dapat dilihat pada Gambar 25 dan Gambar 26. Data hasilnya PID *series* dapat dilihat pada Gambar 27 dan Gambar 28.

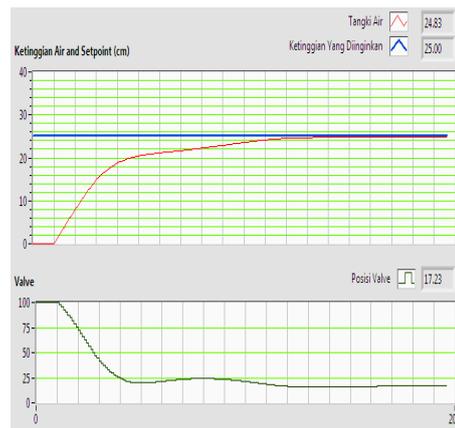


Gambar 25 Hasil percobaan PID *ideal* tipe PID

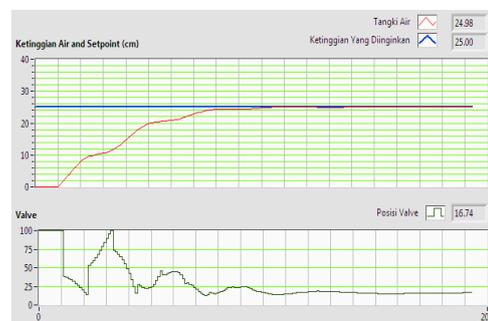
Pada percobaan yang dapat dilihat pada Gambar 26 dengan nilai  $T_i = 999999s$  dan  $K_c = 17.8$  diperoleh  $T_u = 6.5s$ . Data pengujian dimasukan kedalam tabel Ziegler-Nichols di-dapatkan nilai  $K_p$ ,  $T_i$  dan  $T_d$  yang dapat dilihat pada Tabel 4.

Tabel 6  $K_p$ ,  $T_i$ ,  $T_d$  percobaan 4

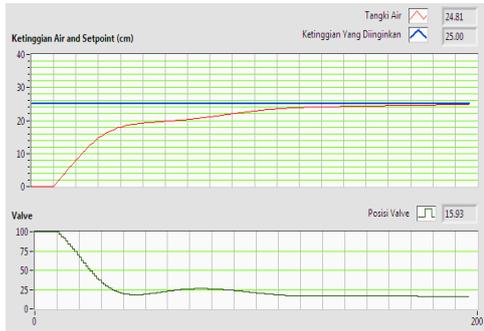
PID Tipe	$K_p$	$T_i$	$T_d$
P	8.9	$\infty$	0
PI	8.01	5.395	0
PID	10.68	3.25	0.8125



Gambar 26 Hasil percobaan PID *ideal* tipe PI

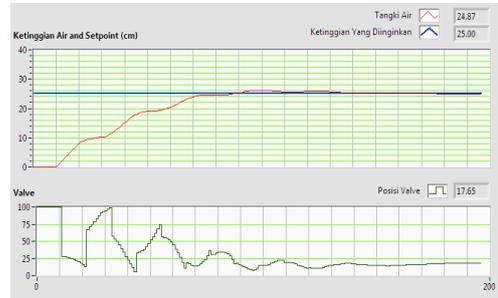


Gambar 27 Hasil percobaan PID *series* tipe PID

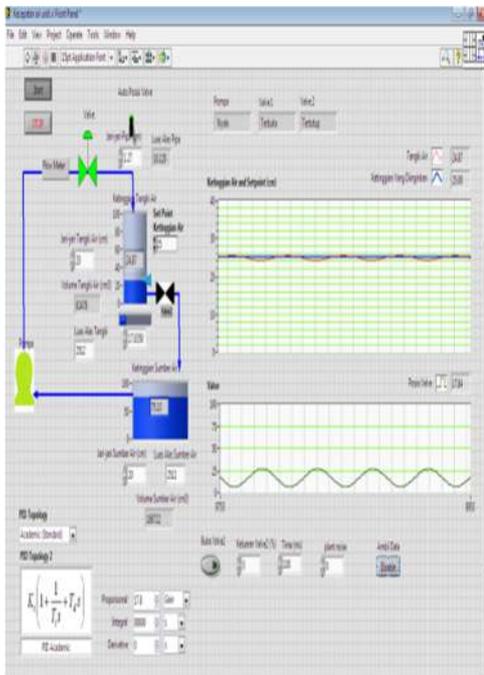


Gambar 28 Hasil percobaan PID series tipe PI

Data hasilnya PID series dapat dilihat pada Gambar 32 dan Gambar 33.

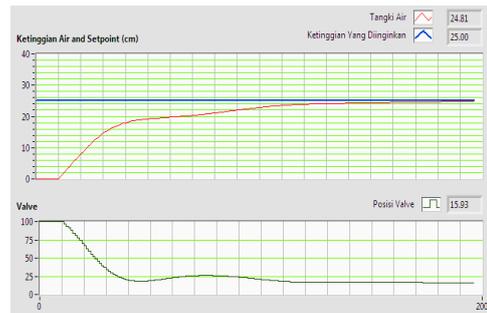


Gambar 30 Hasil percobaan PID ideal tipe PID



Gambar 29 Percobaan 4 dengan  $T_i = 999999s$

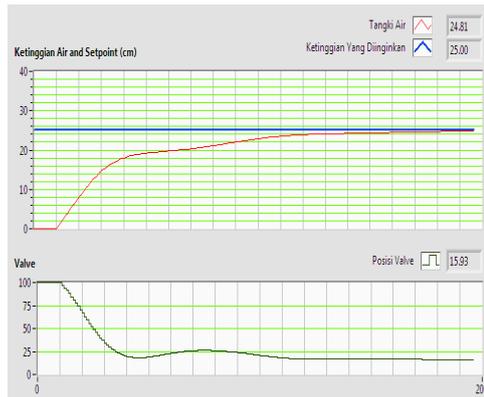
Dilakukan pengambilan data baru dengan menggunakan data dari Tabel 5. Data dimasukkan pada algoritma PID ideal dan PID series. Data hasilnya PID ideal dapat dilihat pada Gambar 30 dan Gambar 31.



Gambar 31 Hasil percobaan PID ideal tipe PI



Gambar 32 Hasil PID series tipe PID



Gambar 33 Hasil percobaan PID series tipe PI

## 5. SIMPULAN

Berdasarkan pada hasil pengujian maka diperoleh simpulan sebagai berikut:

1. Bila menggunakan  $T_i$  sebagai aksi integral dalam metode zigler-nichols, semakin besar nilai  $T_i$ , semakin kecil nilai  $K_p$  yang dibutuhkan agar proses variabel beresilasi dengan ketinggian konstan.
2. Keluaran lebih cepat stabil dengan menggunakan kontroler PI.
3. Algoritma PID *ideal* dan PID *series* tidak berbeda saat tidak menggunakan kontroler *derivative*.
4. Dengan adanya kontroler *derivative*, PID *ideal* dan *series* memiliki keluaran yang berbeda. PID *ideal* menghasilkan *rise time* yang lebih cepat daripada PID *series* dan PID *ideal* menghasilkan *overshoot* sedangkan PID *series* tidak menghasilkan *overshoot*.

## DAFTAR PUSTAKA

- [1] Artanto, D. 2012. *Interaksi Arduino dan LabVIEW*. Jakarta: PT Elex Media Komputindo.
- [2] Chipdip. 2010. *G 1/2 Water Flow Sensor*. ([lib.chipdip.ru/583/DOC000583441.pdf](http://lib.chipdip.ru/583/DOC000583441.pdf) diakses 7 Juni 2014).
- [3] Nasional Instrument. 2013. *Introduction To LabVIEW Six-Hour Course*. Texas: Nasional Instrument.
- [4] Ogata, K. 1997. *Modern control engineering, 3rd*. New Jersey: Prentice Hall.
- [5] Parallax. 2013. *Ping))) Ultrasonic Distance Sensor*. ([www.parallax.com/sites/default/files/downloads/28015-PING-Sensor-Product-Guide-v2.0.pdf](http://www.parallax.com/sites/default/files/downloads/28015-PING-Sensor-Product-Guide-v2.0.pdf) diakses 7 Juni 2014).
- [6] Syahrul. 2012. *Mikrokontroler AVR ATmega 8535*. Bandung: Informatika.
- [7] Rangan, C.S, et all. 1992. *Instrumentation Devices and Systems, 10th*. New Delhi: Tata MCGraw-Hill.

