

RANCANG BANGUN PENGENDALIAN LEVEL AIR OTOMATIS PADA TANGKI DENGAN SERVO VALVE BERBASIS PID CONTROLLER

Muchammad Nur Fatah Muizz

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya
e-mail : Muchammadmuizz@mhs.unesa.ac.id

Bambang Suprianto

Dosen Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya
e-mail : bambangsuprianto@unesa.ac.id

Abstrak

Air merupakan kebutuhan primer baik dalam dunia rumah tangga maupun dunia industri. Dalam setiap industri hampir seluruhnya membutuhkan air. Salah satu sistem pengendalian level air yang ada di industri adalah dengan menggunakan sistem manual. Kekurangan dari sistem manual adalah membutuhkan waktu yang lama dalam mencapai level air yang di inginkan dan rawan terjadi *human error* dalam pembacaan dari level air. Untuk memperbaiki masalah tersebut maka terdapat alternatif yaitu menggunakan sistem pengendali level otomatis menggunakan sensor untuk membaca level air dalam tangki yang lebih presisi dalam pembacaan. Tujuan penelitian ini adalah untuk merancang dan membuat sistem pengendalian level dari sebuah tangki dengan *servo valve* menggunakan kontroler PID. Dengan parameter K_p , K_i , dan K_d akan di hitung menggunakan pendekatan karakteristik respon sistem orde 1. Dalam penelitian ini digunakan software *labview* 2014 untuk mengontrol parameter K_p , K_d dan K_i dari kontroler PID. Hasil penelitian menunjukkan bahwa respon pengendalian level air dengan *servo valve* menggunakan kontroler PID mampu mencapai *setpoint* yang diinginkan dengan $\tau = 37,04$ detik dan *error steady state* 8,3% (dengan fungsi alih 12 cm) dan $\tau = 38,427$ detik dan *error steady state* 2,715% (dengan fungsi alih 4 cm). Respon ini menunjukkan perbaikan dari respon sistem tanpa kontroler dengan $\tau = 73,998$ detik dan *error steady state* 12,507%

Kata Kunci : *Servo Valve*, Kontroler PID, Labview.

Abstract

Water is a primary necessity in both the domestic and the industrial world. In any industry is almost entirely in need of water. One of the water level control system in the industry is to use a manual system. Disadvantages of a manual system is in need of a long time in reaching the desired water level and prone to human error in the reading of the water level. To fix these problems then there are alternatives that use automatic level control system uses sensors to read the water level in tangki. Tujuan this research is to design and create a level control system of a tank with a servo valve using a PID controller. With the parameter K_p , K_i , and K_d will be calculated approach-order system response characteristics 1. This study used a software *labview* 2014 to control the parameters K_p , K_d and K_i of the PID controller. The results showed that the water level control dengan response servo valve using a PID controller is able to achieve the desired setpoint by $\tau = 37.04$ second and error steady state 8.3% (With function over 12 cm) and $\tau = 38.427$ second and error steady state of 2,715% (with function over 4 cm). This response shows the improvement of the system response without controller with $\tau = 73.998$ second and error steady state 12.507%

Keywords: *Servo Valve*, PID Controller, Labview.

PENDAHULUAN

Air merupakan kebutuhan primer baik dalam rumah maupun industri. Dalam sebuah perusahaan yang memproduksi minuman, stok air merupakan kebutuhan yang vital maka dari itu harus ada sebuah tangki penampungan air yang di jaga volumenya agar dapat memenuhi kebutuhan bahan baku di industri dan juga tidak melebihi batas kapasitas dari tangki.

Control valve merupakan aplikasi motor servo DC sebagai aktuator dalam pengaturan *level* air. Plan tangki dalam industri membutuhkan sebuah aktuator input yang dapat menyesuaikan *level* air yang ada dalam tangki. Hal ini dapat mempengaruhi performansi dari

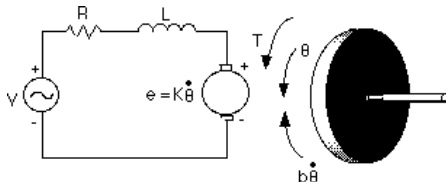
sistem dengan nilai parameter pengendalian yang digunakan bernilai tetap atau konstan. Pengendalian PID merupakan gabungan dari tiga pengendalian antara lain pengendalian *Proportional* (P), pengendalian *Integral* (I), dan pengendalian *Derivative* (D). Kelebihan dari pengendalian PID yaitu memiliki respon yang cepat, memiliki nilai overshoot dan nilai eror yang kecil. (Iwan ,2008).

Fokus permasalahan yang ingin dipecahkan adalah pengaturan level air pada tangki agar mampu mencapai *set point* yang di inginkan dalam waktu yang cepat . Dan mampu menjaga tangki air pada level tertentu meski *valve output* dari tangki di buka.

KAJIAN PUSTAKA

Driven Servo Control Valve

Pada dasarnya *driven servo control valve* merupakan servo dengan sudut $0^\circ - 90^\circ$ yang diintegrasikan dengan valve. Motor servo disusun dari sebuah motor DC, *gearbox*, variabel resistor (VR) atau potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas maksimum putaran sumbu (*axis*) motor servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang pada pin kontrol motor servo. (Nanang dkk, 2009)



Gambar 1 Struktur *Servo Motor*
(Sumber : Nanang dkk,2009)

Keterangan :

Persamaan torsi yang dibangkitkan oleh Motor DC dapat didekati secara linear menurut persamaan berikut ini :

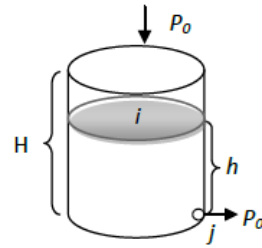
$$\frac{\theta}{V} = \frac{K}{S((Js+b)(Ls+R))+K^2} \quad (1)$$

Keterangan :

- V = Tegangan (v)
- K = Konstanta jangkar
- R = Resistansi jangkar (Ohm)
- L = Induktansi jangkar (H)
- T = Torsi motor (Nm)
- θ = Perpindahan sudut dari poros rotor ($^\circ$)
- J = Momen Inersia motor (kg m²)
- b = Koefisien gesekan viskos motor (Ns/m²)
- e = back emf (tegangan jangkar berputar) (v)

Tangki Buffer Softener

Buffer softener merupakan tempat penampungan air softener yang memiliki lubang masukan di bagian atas tangki dan lubang keluaran pada posisi bawah seperti yang di tampilkan pada gambar 2. Dalam *plant* di perusahaan, tangki *buffer softener* memiliki volume 8000 liter. Pada tangki ini sendiri tidak terdapat pompa air sehingga volume air pada tangki sangat berperan penting untuk menjaga debit aliran air menuju lini produksi. Proses produksi minuman teh dalam kemasan membutuhkan 1500 liter air untuk pelarutan gula, 3500 liter untuk penyeduhan teh dan 3000 liter untuk *bottle wash*. Pengaliran air dari tangki *buffer softener* menuju ke lini produksi dilakukan secara seri pada satu pipa. Pertama air akan mengalir untuk proses penyeduhan teh, setelah 3500 liter *valve* akan tertutup dan air akan mengalir untuk proses pelarutan gula, setelah 1500 liter *valve* akan tertutup dan selanjutnya air akan mengalir menuju ke proses *bottle wash*.



Gambar 2. Model matematika tangki
(Sumber : Irawati dkk,2009)

Keterangan :

- i : titik awal pada permukaan air,
- j : titik pada lubang.
- h : fungsi dari waktu
- H : ketinggian air

Karena permukaan air di i dan j langsung berhubungan dengan udara luar, maka tekanan di titik i (P_i) dan tekanan di titik j (P_j) akan sama yaitu sama dengan tekanan udara luar (P_o). Sehingga $P_i = P_j = P_o$. (Irawati dkk,2009)

Software Labview

Laboratory virtual instrument engineering workbench (LabVIEW) disebut instrumen virtual, atau VI, karena tampilannya dan pengoperasi yang mirip dengan instrumen fisik, seperti osiloskop dan multimeter. LabVIEW berisi seperangkat alat yang komprehensif untuk memperoleh, menganalisis, menampilkan, dan menyimpan data. Serta software untuk membantu dalam memecahkan masalah kode atau bahasa komputer. (National Instruments, 2013)

Arduino UNO

Arduino adalah mikrokontroler *single-board open source*. Mikrokontroler arduino dapat diprogram menggunakan bahasa pemrograman berbasis sintaks dan berdasarkan librari. Program berbasis ini tidak berbeda dengan C / C++, tetapi dengan beberapa penyederhanaan dan modifikasi. Untuk memfasilitasi perancangan aplikasi, Arduino mikrokontroler juga menggunakan pengolahan berbasis *Integrated Development Environment (IDE)*. Mikrokontroler Arduino dapat dipasangkan dengan bermacam-macam sensor dan aktuator lainnya. Adapun sensor dan aktuator yang dapat melekat pada Arduino seperti sensor gerak, ultrasonik, panas, suara, Ethernet Shield, LED display dan banyak lagi. (Bambang dkk,2017)

Analisis Karakteristik Sistem Orde Satu

Dari model matematis sebuah sistem, orde dari suatu sistem dapat dilihat dari besar pangkat variabel s (dalam transformasi *Laplace*). Suatu sistem dikatakan ber-orde satu jika fungsi alihnya mempunyai variabel s dengan pangkat tertinggi satu. Bentuk fisisnya bisa berupa rangkaian listrik RC, sistem termal, atau sistem lainnya. (Ogata,1997)

$$\frac{C(s)}{R(s)} = \frac{k}{\tau s + 1} \quad (2)$$

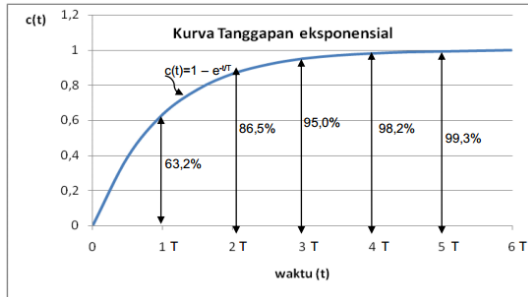
Keterangan :

C(s) = Sinyal Output

R(s) = Sinyal Input

k = Konstanta level air (cm)

τs = Konstanta Waktu (s)



Gambar 3. Karakteristik respon sistem orde satu terhadap masukan *unit-step* (Sumber: Ogata, 2010)

Kontroller PID

Suatu pengendalian otomatis membutuhkan beberapa parameter pengendalian . Di antaranya adalah parameter proporsional (P), parameter Integral (I), dan parameter derivatif (D). Kontrol proporsional berfungsi untuk yang berfungsi memberikan gain atau penguatan , sehingga akan mempercepat keluaran sistem mencapai titik referensi. Sedangkan kontrol integral berfungsi untuk menghilangkan kesalahan keadaan tunak (*offset*). Dan kontrol derivatif berfungsi untuk memberi redaman sehingga mengurangi lonjakan yang terdapat pada kontrol proporsional dan integral. (Ogata ,1997)

METODE PENELITIAN

Pendekatan Penelitian

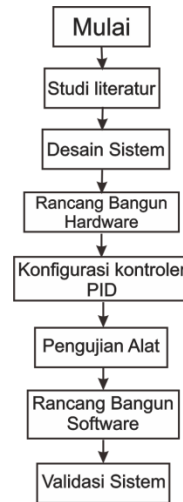
Pada penelitian ini menggunakan *software* Labview 2014 dan Matlab 2014 untuk simulasi menjalankan sistem PID yang akan digunakan dan merancang bangun *plant* yang akan menghasilkan respon pada *servo valve*.

Tempat dan Waktu Penelitian

Penelitian dilaksanakan di Laboratorium Kendali Jurusan Teknik Elektro Universitas Negeri Surabaya dan waktu pelaksanaannya dilakukan pada semester genap 2017/2018.

Rancangan Penelitian

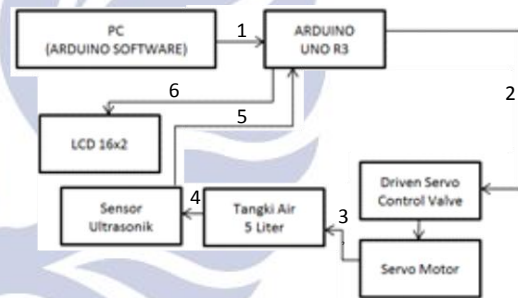
Rancangan penelitian ini bertujuan untuk menjawab permasalahan dalam rangka merumuskan kesimpulan, seperti dijelaskan pada Gambar 4. diagram alir sebagai berikut:



Gambar 4. Perancangan Penelitian (Sumber : Dokumen Pribadi, 2018)

Rancang Bangun Hardware

Rancang bangun *hardware* yang dimaksud adalah hubungan antar komponen, sehingga semua komponen yang digunakan dapat terhubung dan dapat beroperasi sesuai dengan desain sistem. Rancang bangun *hardware* pada penelitian ini ditunjukkan pada Gambar 5.



Gambar 5. Rancang Bangun Hardware (Sumber: Dokumen Pribadi, 2018)

Pada gambar di atas nampak rangkaian dan hubungan dari tiap komponen. PC yang terinstal *software* IDE Arduino akan memberikan perintah kepada Arduino UNO berupa *set point* lalu *servo valve* akan berputar menyesuaikan sudut untuk mencapai *set point* dan sensor ultrasonik akan membaca level pada tangki untuk di tampilkan di layar komputer dan juga LCD 16x2.

Identifikasi Plant

Pada penelitian ini, fungsi alih didapatkan dengan cara pendekatan sistem orde-1. Sehingga untuk mendapatkan parameter-parameter dari karakteristik sistem orde-1 yaitu K, dan τ dilakukan pengambilan data sebanyak 4 kali dengan *setpoint* bervariasi. Percobaan dilakukan dengan *loop* terbuka sehingga didapatkan respon sistem orde-1. Setelah didapatkan 4 fungsi alih dari setiap *setpoint*, selanjutnya fungsi alih tersebut akan disimulasikan dan dibandingkan dengan respon sistem sesungguhnya. Fungsi alih dengan *error*

rata-rata terkecil akan digunakan sebagai fungsi alih sistem dan menjadi acuan untuk mendesain kontroler PID.

Validasi model yang dimaksud adalah mencari fungsi alih dengan *error* terkecil. *Error* didapatkan dengan membandingkan output rata-rata *steady state* sistem sesungguhnya dengan output rata-rata *steady state* simulasi. *Input* pada proses tersebut adalah nilai level air dan outputnya adalah fungsi alih.

Setelah itu, data yang didapatkan akan diolah dan ditentukan parameter-parameter k dan τ berdasarkan karakteristik sistem orde-1 dari tiap *setpoint* dengan rumus sebagai berikut :

$$k = \frac{Y_{ss}}{X_{ss}} \quad (3)$$

$$C(\tau) = 0.632 \times Y_{ss} \quad (4)$$

Keterangan :

X_{ss} = Input level air (cm)

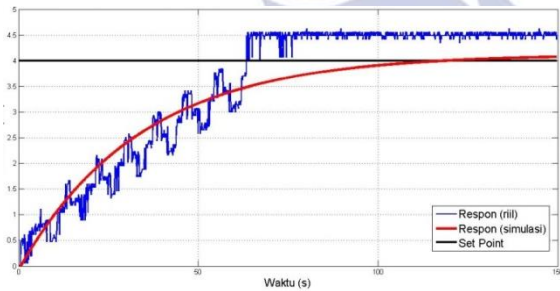
Y_{ss} = Output level air(cm)

$C(\tau)$ = 63,2% Keluaran Sistem

Konfigurasi Kontrol PID

Pemodelan Sistem

Pemodelan sistem dilakukan dengan cara memberikan *setpoint* level bervariasi yaitu 4cm, 7cm, 8cm, 12cm dengan *sampling time* 0.1 detik. Model matematika yang didapatkan merupakan hubungan antara level yang dibaca oleh *ultrasonik* dengan *setpoint* level yang diinginkan. Pemodelan dilakukan pada kondisi tanpa beban. Berikut adalah respon sistem dengan *setpoint* 4cm ditunjukkan pada Gambar 6.



Gambar 6. Respon Sistem Open Loop *setpoint* 4cm (Sumber : Dokumen Pribadi, 2018)

Berdasarkan Gambar 6, maka dapat dihitung parameter dari karakteristik sistem orde-1 yaitu K dan τ menggunakan persamaan (3), (4)

Pemilihan fungsi alih dilakukan dengan menghitung nilai *error* rata-rata dari setiap fungsi alih. Fungsi alih dengan nilai *error* rata-rata terkecil akan digunakan sebagai acuan untuk mendesain kontroler PID. Berikut persamaan yang digunakan untuk menghitung *error* :

$$Error = |Output Simulasi - Output Real| \quad (5)$$

Output pada detik ke 0.1 simulasi akan dikurangi dengan output real pada detik ke 0.1 juga. Begitu

seterusnya sampai detik ke-10. Sehingga didapatkan nilai *error* tiap kelipatan 0.1 detik. Setelah itu, nilai-nilai *error* tersebut akan di rata-ratakan dan dibandingkan dengan *error* rata-rata fungsi alih yang lain.

Desain Kontroler PID

Fungsi dari perancangan kontroler adalah memperbaiki respon sistem. Oleh karena itu, karakteristik *output* yang akan diperbaiki ditunjukkan pada Tabel 1.

Tabel 1. Karakteristik *Output* yang akan diperbaiki

Karakteristik yang akan diperbaiki
Waktu Tunak (<i>Settling Time</i>)
Waktu Naik (<i>Rise Time</i>)
Simpangan Maksimum (<i>Overshoot Maximum</i>)
Error <i>Steady State</i>

Sedangkan untuk desain parameter kontroler PID didapatkan melalui metode analitik untuk mencari nilai parameter τ_i , τ_d dan K_p sebagai berikut :

$$\tau^* = \tau \times 0,5 \quad (6)$$

$$\tau_i = \tau_d = \tau^* \quad (7)$$

Untuk mendapatkan nilai K_p , K_i dan K_d , digunakan persamaan sebagai berikut

$$K_p = \frac{\tau_i}{\tau^* \cdot k} \quad (8)$$

$$K_i = \frac{K_p}{\tau_i} \quad (9)$$

$$K_d = K_p \times \tau_d \quad (10)$$

Keterangan :

τ_i = Waktu Integral (s)

τ_d = Waktu Derivatif (s)

τ^* = Waktu yang diinginkan (s)

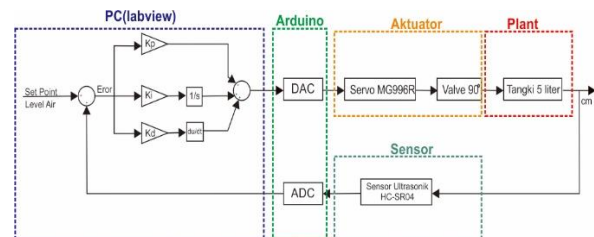
K_p = Konstanta Proporsional

K_i = Konstanta Integral

K_d = Konstanta Derivatif

Rancang Bangun Software

Software yang digunakan pada penelitian ini adalah Labview 2017. Labview 2017 berfungsi sebagai pusat pengendali dan juga berfungsi untuk menyimulasikan fungsi alih yang telah didapatkan sebelumnya. Hubungan antara *software* dan *hardware* dapat dilihat pada Gambar 7.

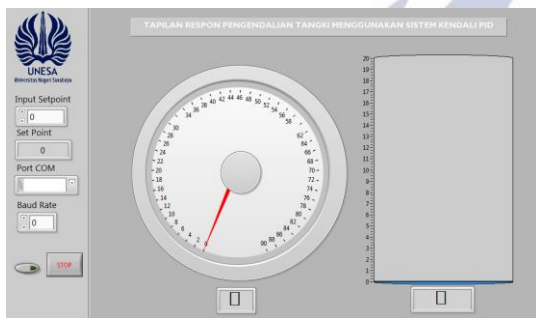


Gambar 7. Hubungan antara *software* dan *hardware* pada pengendalian level air menggunakan kontroler PID (Sumber : Dokumen Pribadi, 2018)

Berdasarkan Gambar 7, secara umum dapat dijelaskan bahwa parameter kontroler PID yaitu K_p , K_i dan K_d sudah dihitung pada proses desain kontroler PID. Sehingga pada penelitian ini, parameter kontroler PID ditentukan oleh penulis dan tidak berubah saat proses berjalan. Nilai *setpoint* juga ditentukan oleh penulis antara 1cm sampai 22cm. Level air yang dibaca oleh sensor *ultrasonic* akan diubah menjadi nilai tegangan. Nilai tegangan akan dikirim ke NI Elvis dan langsung diterima arduino setelah proses ADC. Nilai tersebut akan dibandingkan dengan nilai *setpoint* dan didapatkan nilai error.

Perancangan Tampilan pada Labview

Tampilan labview yang akan digunakan ditunjukkan pada Gambar 8.



Gambar 8. Rancangan Tampilan pada Labview (Sumber : Labview, 2017)

Nilai *setpoint* berupa nilai level air yang akan ditampilkan. Untuk sudut servo didapatkan setelah dilakukan kalibrasi.

HASIL DAN PEMBAHASAN

Pada bagian ini akan dibahas proses kalibrasi sensor *ultrasonic*, pemodelan sistem, validasi model, perancangan kontroler PID, dan pengujian respon sistem.

1) Kalibrasi sensor *ultrasonic HC-SR0*.

Proses kalibrasi sensor *Ultrasonic* dilakukan agar didapatkan nilai level ketinggian air dalam setiap satuan centimeter yang dikeluarkan oleh sensor *ultrasonic*. Alat bantu kalibrasi sensor *Ultrasonic* menggunakan penggaris manual dan juga roll meter. Tabel kalibrasi ditunjukkan pada Tabel 2.

Tabel 2. Data kalibrasi *ultrasonic*.

No	Sensor <i>Ultrasonic</i>	Penggaris manual	Error
1.	4	4	0%
2.	6,5	6,65	2,3%
3.	7,9	8	1,25%
4.	9,25	9	2,7%
5.	12,21	12	1,75%
6.	13,18	13	1,38%

No	Sensor <i>Ultrasonic</i>	Penggaris manual	Error
7.	16	16	0%
8.	18,09	18	0,5%
9.	19,24	19	1,26%
10.	20,14	20	0,7%

Setelah mendapatkan data kalibrasi, menggunakan persamaan 5 maka dapat di hitung nilai rata-rata dari sensor sebesar 1,18% atau 0,15 cm antara sensor ultrasonik dan penggaris manual.

2) Pemodelan Sistem

Pemodelan sistem dilakukan menggunakan metode pendekatan karakteristik sistem orde-1 dengan parameter yang dibutuhkan adalah K dan τ . Metode ini dilakukan bertujuan untuk mendapatkan model atau fungsi alih dari sistem. Fungsi alih tersebut akan digunakan sebagai acuan untuk merancang parameter dari kontroler PID dan simulasi untuk mengetahui respon sistem. Pada skripsi ini dilakukan pemodelan dengan 4 setpoint berbeda, yaitu 4cm, 7cm, 8cm, dan 12cm. Berdasarkan Gambar 6, dan menggunakan persamaan (2), (3), (4), (5) maka dapat dihitung parameter dari karakteristik sistem orde-1 yaitu $k = 1.312$, $\tau = 33.678$.

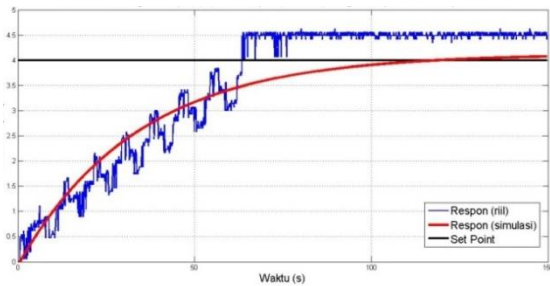
3) Validasi Sistem

Validasi model dilakukan untuk mendapatkan model atau fungsi alih dari sistem yang memiliki respon dengan tingkat *error*. Validasi model juga berfungsi untuk menentukan apakah fungsi alih yang telah didapatkan benar dan dapat digunakan sebagai model dari *plant* riil. Fungsi alih tiap *setpoint* beserta *error* rata-rata ditunjukkan pada Tabel 3.

Tabel 3. Fungsi alih masing-masing set point.

Setpoint	Fungsi Alih	Error Rata-Rata
4cm	$G_s = \frac{1,312}{33,678 s + 1}$	0.273
7cm	$G_s = \frac{1,05}{43,88 s + 1}$	0.92
8cm	$G_s = \frac{1,0754}{54,6 s + 1}$	0.37
12 cm	$G_s = \frac{1,0215}{83,298 s + 1}$	0.485

Dari data fungsi alih yang diperoleh dipilih fungsi alih dengan error rata-rata terkecil yaitu 0,273. Fungsi alih dengan *setpoint* 4cm akan dipilih sebagai acuan untuk mendesain parameter dari kontroler PID. Perbandingan respon dari fungsi alih dengan respon sistem (riil) ditunjukkan pada Gambar 9. :



Gambar 9. Perbandingan Respon Sistem (Riil) dengan Respon Fungsi Alih (4 cm)
(Sumber : Dokumen Pribadi, 2018)

Sebelum merancang parameter dari kontroler PID, akan dihitung parameter dari respon sistem riil *plant* secara *open loop* pada *setpoint* 4cm V dengan nilai $\tau = 33,6$ detik. Parameter respon *plant* riil dengan *setpoint* 4vm secara *open loop* ditunjukkan pada Tabel 4.

Tabel 4. Parameter Respon *Plant* Riil dengan *setpoint* 4cm (*Open Loop*)

Parameter Respon	Nilai
Waktu Tunda (τ_d)	23,35 detik
Waktu Naik ($\tau_r = 5\%$ - 95%)	99,162 detik
Waktu Naik ($\tau_r = 10\%$ - 90%)	73,998 detik
Waktu Tunak ($\tau_s = 5\%$)	101,304 detik
Waktu Tunak ($\tau_s = 2\%$)	134,712 detik
Waktu Tunak ($\tau_s = 0.5\%$)	168,39 detik
<i>Error SteadyState</i> (Ess)	12,5002%
<i>Maximum Overshoot</i> (MO)	—

4) Perancangan Kontroler PID

Perancangan kontroler PID dilakukan untuk mendapatkan parameter dari kontroler PID yaitu K_p , K_i , dan K_d . Pada pembahasan sebelumnya telah ditentukan bahwa fungsi alih dengan *setpoint* 4 cm memiliki *error* rata-rata terkecil dan dijadikan acuan untuk menentukan parameter dari kontroler PID. Berdasarkan fungsi alih *setpoint* 4cm, maka dapat dihitung parameter kontroler PID menggunakan persamaan (8), (9), dan (10)

Tabel 5. Parameter Kontroler PID

Parameter Kontroler PID	Nilai
K_p	0,969
K_i	0,057
K_d	16,65

5) Implementasi Kontroler PID dan Pengujian Respon Sistem

Setelah mendapatkan nilai K_p , K_i dan K_d , selanjutnya adalah mengimplementasikan kontroler PID. Sebelum mengimplementasikan kontroler PID langsung ke *plant* riil, akan disimulasikan terlebih dahulu dengan *software Arduino UNO*. Dan selanjutnya akan dilakukan pengujian pada respon sistem riil dengan beberapa metode. Pertama implementasi dan Pengujian pada *Plant Riil*

Setelah melakukan simulasi, maka selanjutnya adalah mengimplementasikan kontroler PID pada *plant* riil dan melakukan pengujian respon sistem. Script pada *arduino uno* ditunjukkan pada Gambar 10.

```

#include <PID_v1.h>
#include <NewPing.h>
#include <Servo.h>
long waktu;
#define TRIGGER_PIN 12
#define ECHO_PIN 11
#define MAX_DISTANCE 35

NewPing sonar (TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
Servo servo;

float Kp = 0.969;
float Ki = 0.11164;
float Kd = 43.89;
double Setpoint, Input, Output;

PID myPID (&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);

float setpointcm = 0;
float timpilair = 0;
float volumeair = 0;
float sudutservo = 0;
float bacaaservo = 0;

void setup() {
  myPID.SetMode(AUTOMATIC);
  
```

Gambar 10. Script Arduino dengan PID
(Sumber : Dokumen Pribadi, 2018)

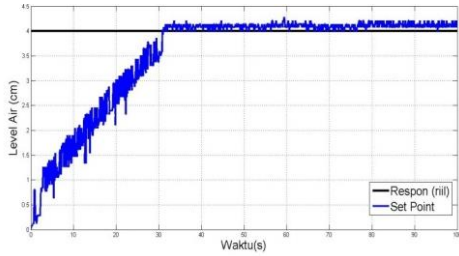
Untuk mengetahui perbedaan respon sistem sebelum dan sesudah ditambahkan kontroler PID, maka dilakukan beberapa metode pengujian sebagai berikut :

a. Pengujian Respon Sistem Riil dengan *Setpoint* Tetap.

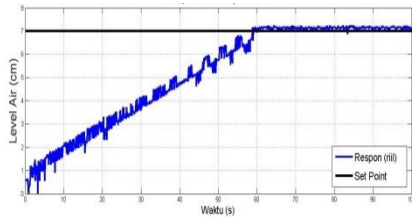
Dilakukan pengujian dengan *setpoint* tetap (tanpa beban) yaitu 4cm, 7cm, 8cm, dan 12cm dengan *sampling time* 0.01 detik. Bentuk fisik dari pengujian ditunjukkan pada Gambar 12. Respon sistem riil dengan *setpoint* 4cm, 7cm, 8cm, dan 12cm berturut-turut ditunjukkan pada Gambar 13, 14, 15, dan 16.



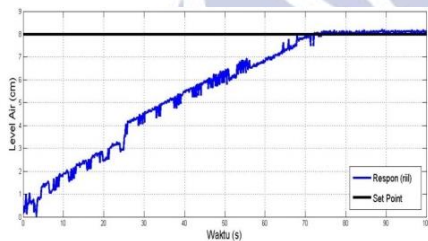
Gambar 12. Pengujian dengan *Setpoint* Tetap
(Sumber : Dokumen Pribadi, 2018)



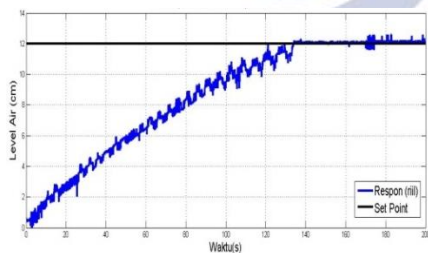
Gambar 13. Respon sistem riil *setpoint* 4cm menggunakan Kontroler PID
(Sumber : Dokumen Pribadi, 2018)



Gambar 14. Respon sistem riil *setpoint* tetap 7cm menggunakan Kontroler PID
(Sumber : Dokumen Pribadi, 2018)



Gambar 15. Respon sistem riil *setpoint* tetap 8cm menggunakan Kontroler PID
(Sumber : Dokumen Pribadi, 2018)



Gambar 16. Respon sistem riil *setpoint* tetap 12cm menggunakan Kontroler PID
(Sumber : Dokumen Pribadi, 2018)

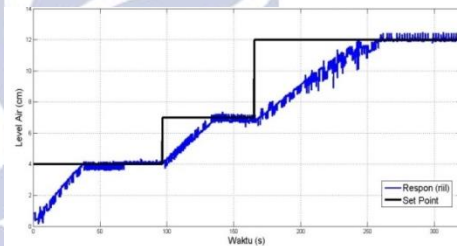
Setelah ditambahkan kontroler PID, maka respon sistem riil akan mengalami perubahan. Untuk mengetahui perubahan tersebut, maka dilakukan perhitungan parameter dari salah satu respon sistem riil yaitu pada saat *setpoint* 4cm. Parameter yang akan dihitung adalah τ_d (*Time Delay*), τ_r (*Rise Time*), τ_s (*Settling Time*), E_{ss} (*Error Steady State*) dengan nilai $\tau = 16,8$ detik. Parameter respon *plant* riil dengan *setpoint* 4cm menggunakan kontroler PID ditunjukkan pada Tabel 6.

Tabel 6. Parameter Respon *plant* riil dengan *setpoint* 4cm menggunakan kontroler PID

Parameter Respon	Nilai
Waktu Tunda (τ_d)	12,122 detik
Waktu Naik ($\tau_r = 5\%$ - 95%)	51,495 detik
Waktu Naik ($\tau_r = 10\%$ - 90%)	38,427 detik
Waktu Tunak ($\tau_s = 5\%$)	52,467detik
Waktu Tunak ($\tau_s = 2\%$)	69,957 detik
Waktu Tunak ($\tau_s = 0.5\%$)	87,445 detik
<i>Error SteadyState</i> (E_{ss})	2,715%

b. Pengujian Respon Sistem Riil dengan *Setpoint* Bervariasi

Dilakukan pengujian dengan memberikan *setpoint* bervariasi yaitu 4cm,7cm dan 12cm dengan *sampling time* 0.1 detik. Respon sistem riil dengan *setpoint* bervariasi ditunjukkan pada Gambar 17.



Gambar 17. Respon sistem riil *setpoint* bervariasi menggunakan Kontroler PID
(Sumber : Dokumen Pribadi, 2018)

PENUTUP
Simpulan

Berdasarkan penelitian yang dilakukan, maka didapatkan simpulan bahwa pengujian yang dilakukan menggunakan *software* Labview mampu mengolah data respon sistem riil dengan *real time*. Dengan pendekatan karakteristik respon sistem orde satu, menggunakan fungsi alih *set point* 4 cm didapatkan parameter kontroler PID seperti yang tercantum pada Tabel 5, yaitu $K_p = 0.969$, $K_i = 0.057$, dan $K_d = 16,65$. Kontroler PID yang dirancang untuk mengendalikan respon dari *servo valve* dalam mengisi level air dari tangki. Kendali PID mampu mengurangi E_{ss} (*Error Steady State*) dari respon sistem riil seperti yang di tunjukan pada Tabel 4 dan Tabel 6 dari 12,507% sebelum menggunakan kontroler PID , menjadi 2,715% setelah menggunakan kontroler PID.

Saran

Berdasarkan simpulan diatas ada beberapa saran yang dapat dilakukan untuk pengembangan sistem agar hasil lebih maksimal sehingga dapat dilakukan metode pendekatan yang lain agar mendapatkan fungsi alih yang lebih baik dari penelitian ini, menggunakan sensor yang lebih akurat dan presisi dan sistem ini masih dapat dikembangkan dengan menggunakan metode kontroler yang lain seperti *fuzzy logic*, *model predictive control* (MPC), dan *Linear Quadratic Regulator control* (LQR) untuk mendapatkan hasil respon yang lebih baik.

DAFTAR PUSTAKA

- Hartono, Nanang Budi dkk. 2009. Pengaturan Posisi Motor Servo DC Dengan metode P,PI, dan PID. Jurnal Teknik Elektronika PENS Surabaya.
- Irawati, Kuntjoro adji Sidarto. 2008. Model Matematika Waktu Pengosongan Tangki Air. Jurnal Fakultas MIPA, Universitas Negeri Yogyakarta.
- National Instruments, 2013. Getting Started with LabVIEW, National Instruments Corporate Headquarters, Texas - USA.
- Ogata, Katsuhiko. 1997. "Teknik Kontrol Automatik". Terjemahaan Edi Laksono .Jakaeta : Erlangga .
- Setiawan, Iwan .2008. Kontrol PID untuk Proses Industri. Jakarta :PT. Elex Media Komputindo.
- Suprianto, Bambang , Lucky Aggazi Subagyo .2017. Control System Of Current Flow 3-Phase Unbalanced Based On Arduino Uno. International Journal of Engineering and Applied Sciences (IJEAS).

