

## SISTEM PENGENDALIAN WATER PUMP UNTUK MENGATUR TINGGI LEVEL AIR DENGAN ALGORITMA PID PADA PLANT WATER TREATMENT

Fahmi Fahroje Pane<sup>1</sup>, Hera Hikmarika<sup>1</sup>, Suci Dwijayanti<sup>1</sup>, Muhammad Yusup<sup>1</sup>, dan Bhakti Yudho Suprpto<sup>1</sup>

<sup>1</sup> Teknik Elektro, Universitas Sriwijaya, Palembang  
Corresponding author: bhakti@ft.unsri.ac.id

**ABSTRAK:** Salah satu prosedur yang digunakan pada *Plant Water treatment* adalah pengendalian level. Masalah yang sering timbul ketika level ketinggian air dalam tangki penampungan tidak diketahui, sehingga dimungkinkan terjadi keadaan tangki yang meluap atau kosong dikarenakan kurangnya pengawasan terhadap tangki penampungan. Maka dibutuhkannya pengendalian *water pump* untuk mengatur tinggi level air pada tangki penampungan. Untuk mengoptimalkan kinerja pompa pada saat pengisian tersebut diterapkan algoritma PID pada sistem kontrol untuk mengoptimalkan kinerja *water pump* agar dapat mengatur tinggi level air ketika proses pengisian air pada tangki penampungan berdasarkan *setpoint* yang telah ditentukan. Hasil penelitian menunjukkan bahwa parameter optimal untuk PID adalah nilai  $K_p : 0,7$ ,  $K_i : 0,001$ , dan  $K_D : 2$ . Penggunaan metode PID membuat pengisian air ditangki penampungan saat menuju *setpoint* menjadi lebih cepat dibandingkan dengan tidak menggunakan metode PID dan pompa mati saat mencapai *setpoint* yang telah ditentukan dan kinerja sensor dapat mempengaruhi data grafik PID yang diambil.

**Kata Kunci:** *Level*, PID, Ketinggian air, kontrol PID

**ABSTRACT:** One of the procedures used in *plant water treatment* is the level of control. The problem that often arises at the level of height in the storage tank is unknown, so it is possible to occur in a tank that is softened or empty due to consideration of the storage tank. It is necessary to control the water pump for high levels in the storage tank. To optimize pump performance when filling, a PID is needed in the control system to optimize the performance of the water pump so that a high level of compilation air filling process can be used in the specified storage tank. The results of the study show the optimal parameters for PID based on the  $K_p: 0,7$ ,  $K_i : 0.001$ , and  $KD: 2$ . The use of the PID method makes the delivery of air on the shelter when heading to the *setpoint* faster compared to not using the PID method and the pump dies when reaching the *setpoint* provided and using a sensor that can use PID graph data taken.

**Keywords:** *Level*, PID, Water level, PID control

### PENDAHULUAN

Air yang dikonsumsi setiap hari harus memenuhi standart kualitas air bersih. Namun tak jarang kita mendapati air yang belum memenuhi standart kualitas air bersih, terutama pada saat musim kemarau. Air sumur dan sumber lainnya menjadi keruh dan berbau. Ironisnya terkadang air tersebut tercampur dengan mikroorganisme yang dapat mengganggu fungsi tubuh pada seseorang. Solusinya adalah dengan menggunakan *Plant Water Treatment* yang mengolah air sungai menjadi air bersih yang siap digunakan untuk kebutuhan sehari-hari. Salah satu prosedur yang digunakan pada *Plant Water treatment* adalah pengendalian *level*. Pengendalian level di lakukan pada tangki penampungan air. Masalah yang sering timbul ketika level ketinggian air dalam tangki

penampungan tidak diketahui, sehingga dimungkinkan terjadi keadaan tangki yang meluap atau kosong dikarenakan kurangnya pengawasan terhadap tangki penampungan (Hadi 2013).

Asaad Ahmed Mohammedahmed Eltaieb, Zhang Jian Min. Dalam penelitian ini penulis menggunakan Arduino Uno untuk mengotomatisasi proses pemompaan air di sistem penyimpanan tangki *over-kepala* dan memiliki kemampuan untuk mendeteksi level air didalam tangki menggunakan *buzzer*. juga terdapat *switch on/off* pompa sesuai tingkat air. Kekurangan pada penilitian ini tidak menerapkan algoritma seperti *Fuzzy* atau PID agar meminimalisir tingkat *error* yang terjadi (Eltaieb 2015).

Vardan Mittal. Penelitian ini bertujuan mengukur tingkat air menggunakan sensor ultrasonik. Untuk membangun sebuah sistem secara otomatis

menghidupkan motor pompa. Kontrol otomatis hidup ketika air pada tingkat yang rendah. Kekurangan pada penelitian ini tidak diterapkan algoritma cerdas seperti *fuzzy* atau PID agar meminimalisir tingkat *error* yang terjadi (Mittal 2017).

Ditya Satria Nugroho Hadi, Aris Triwiyatno, Budi Setiyono. Pada penelitian ini telah dilakukan rancang bangun sistem pengontrolan level air pada tangki penampungan *water treatment* sistem berbasis mikrokontroler. Pengontrolan ini digunakan untuk mengetahui seberapa tinggi level air yang ada pada tangki penampungan dan mengontrol level air pada ketinggian tertentu dengan menggunakan sensor jarak (Ping) dan pompa penghisap sebagai aktuaternya serta menampilkannya pada LCD (*Liquid Crystal Display*). Pengendalian dilakukan melalui metode PI Ziegler Nichols I. Dari hasil pengujian didapatkan bahwa kontrol PI dapat diaplikasikan dengan baik untuk mengontrol level air dengan menggunakan metode penalaan Ziegler Nichols. Kekurangan pada penelitian ini adalah respon sistem yang belum baik karena hanya menggunakan tuning PI dan motor pompa yang bekerja kurang maksimal. Untuk mengoptimalkan kinerja pompa pada saat pengisian tersebut dibutuhkan sebuah algoritma cerdas pada sistem pengontrolannya, apabila terjadi sebuah gangguan atau terdapat sinyal *error* pada sistem algoritma PID akan menyelesaikan permasalahan tersebut. Proses pengontrolan di industri-industri pun banyak menggunakan kendali konvensional seperti PID karena kesederhanaan struktur serta kemudahan dalam melakukan *tuning parameter* kontrolnya. Penentuan parameter-parameter yang sesuai agar mendapatkan respon keluaran *system* yang stabil dapat dilakukan dengan metode tuning PID (Santoso 2015).

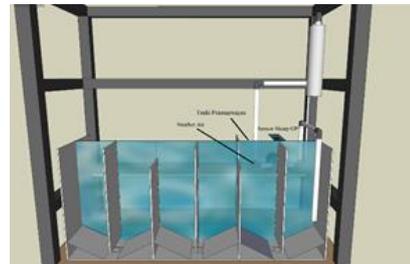
Sistem kendali di perusahaan air minum daerah sudah menggunakan Sistem kendali terdistribusi dimana semua kendali sudah bisa dimonitoring melalui aplikasi yang dibangun sehingga semua aktifitas produksi dipantau oleh satu operator di ruang control. Pada bagian penyedotan air dari sungai pdam menggunakan sebuah pompa, namun kinerja pompa tersebut secara konstan. Kinerja pompa tersebut apabila terjadi gangguan pada saat penyedotan yang menyebabkan mengurangi volume air kinerjanya tidak berpengaruh akan gangguan tersebut. Untuk mengoptimalkan kinerja pompa tersebut dibutuhkan sebuah algoritma cerdas pada sistem kontrolnya apabila terjadi sebuah gangguan atau terdapat sinyal *error* pada sistem algoritma tersebut akan menyelesaikan permasalahan tersebut. Dengan bertujuan menerapkan algoritma PID pada sistem kontrol untuk mengoptimalkan kinerja *water pump* agar dapat mengatur tinggi *level* air ketika proses pengisian air pada tangki penampungan

berdasarkan *setpoint* yang telah ditentukan. (Simanjuntak 2013).

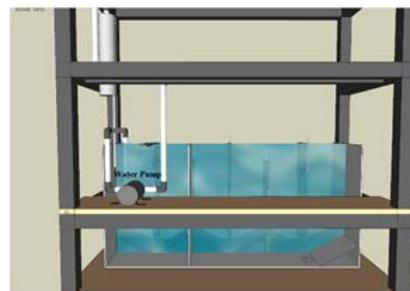
## METODE PENELITIAN

Terdapat beberapa tahapan dalam penelitian ini yang membutuhkan perancangan dan persiapan untuk nantinya diimplementasikan dalam bentuk fisik. Persiapan-persiapan ini dilakukan untuk memudahkan pencapaian tujuan penelitian. Sistem Pengendalian *Water pump* untuk Mengatur Tinggi Level Air dengan Algoritma PID pada *Plant Water Treatment*

Untuk mengimplementasikan pengendalian *water pump* untuk mengatur tinggi level air ini dibutuhkan 2 buah tangki, tangki yang pertama sebagai *water source*/sumber air dan tangki kedua sebagai tangki penampungan air yang telah siap pakai atau siap distribusikan. Hasil dari perancangan mekanik dari sistem pengendalian *water pump* untuk mengatur tinggi level air dengan algoritma PID dapat dilihat pada gambar 1 pada tampilan depan alat terdapat tangki sumber air, tangki penampungan dan sensor *Sharp GP*, kemudian di gambar 2 dengan tampilan belakang alat terdapat motor pompa air.



Gambar 1. Tampilan depan



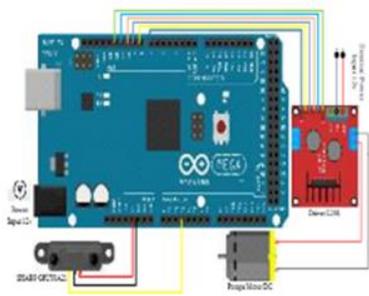
Gambar 2. Tampilan belakang

*Water pump* diletakan didekat *water source* agar dalam pengambilan air lebih maksimal dalam mendistribusikan air menuju tangki penampungan. Letak sensor *sharp gp* harus diatas tangki penampungan agar kinerja sensor *sharp gp* dapat mendeteksi langsung ketinggian dari level air yang berada di tangki penampungan tersebut. *Box* komponen harus diletakan diposisi aman dari gangguan yang dihasilkan dari proses

produksi. Posisi *Box* komponen harus strategis dengan komponen lain seperti sensor *flow* dan *water pump* agar dalam manajemen *wiring* rapi dan mudah dalam mendeteksi gangguan pada komponen tersebut.

#### A. Perancangan Elektrikal Arduino dengan Sensor Sharp GP

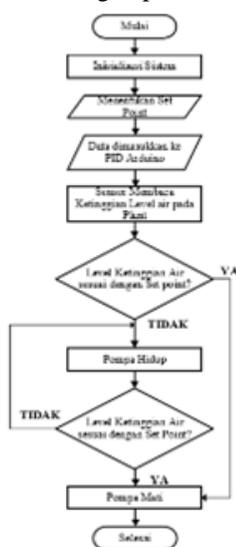
Perancangan sistem elektrikal Arduino dengan *Sharp GP* sensor digunakan untuk mendeteksi ketinggian level air agar nilai *setpoint* yang ditentukan telah terpenuhi maka *water pump* akan mati jika *setpoint* belum terpenuhi maka *water pump* akan tetap hidup. Pada gambar 3 adalah skema rangkaian arduino dengan sensor *sharp gp*.



Gambar 3. Skema rangkaian arduino dengan sensor *sharp gp*

#### B. Perancangan Flowchart Perangkat

Gambaran *Flowchart* sistem pengendalian *water pump* untuk mengatur Tinggi Level Air dengan Algoritma PID yang telah dirancang dapat dilihat pada gambar 4.



Gambar 4. *Flowchart* sistem

Berdasarkan Diagram alir perangkat diatas Parameter *input* adalah sensor *Sharp GP* yang akan dikontrol oleh PID dengan menentukan *setpoint* ketinggian level pada air dalam tangki penampungan, setelah menentukan

*setpoint* maka sensor akan membaca ketinggian level air pada *plant* yang telah di *set*. Jika ketinggian level air belum sesuai dengan *setpoint* maka pompa akan hidup sampai mencapai titik *setpoint*, sedangkan jika ketinggian level air pada tangki penampungan telah mencapai titik *setpoint* maka pompa akan otomatis mati dan proses pengisian air ke tangki penampungan telah selesai.

#### C. Pengujian dan Pengambilan Data

Sebelum melakukan pengujian terhadap sistem ditentukan terlebih dahulu *setpoint* sensor *Sharp GP* untuk mendapatkan *error* yang mendekati *Setpoint* saat pengujian. Setelah mendapatkan *setpoint* peneliti menentukan  $K_p$ ,  $K_i$ ,  $K_d$  untuk mendapatkan *output* dari PID. Pengujian *variable* PID untuk mendapatkan *steady state error* ditampilkan menggunakan grafik. Setelah data tersebut ditampilkan grafik diharapkan nilai *error* sekecil mungkin atau mendekati *Setpoint*, dan osilasi yang sedikit mungkin.

### HASIL DAN PEMBAHASAN

Bagian ini akan membahas pengujian dan analisa sistem kendali terhadap setiap proses-proses yang dilakukan pada sistem pengendalian *water pump* dengan Algoritma PID. Selanjutnya, akan dibandingkan perbedaan kinerja *water pump* yang telah diterapkan Algoritma PID dan kinerja *water pump* tanpa diterapkannya Algoritma PID.

Kinerja *water pump* sangat berhubungan dengan ketinggian level air yang akan dibaca oleh sensor *infra red sharp dp* karena sensor tersebut membaca ketinggian level air di tangki penampungan hasil pengisian dari *water pump*.

#### A. Pengujian Ketinggian Lever air berdasarkan Pembacaan Sensor

Pengujian dilakukan setiap proses pada sistem secara urut sesuai dengan ketentuan dari apa yang telah dirancang sebelumnya. Pada pengujian ini, data akan diambil adalah data berupa hasil pembacaan ketinggian level air dari sensor *sharp dp* pada Arduino Mega 2560, dimana pada pengujian akan dilakukan pada tangki penampungan yang berukuran panjang 20 cm lebar 11 cm tinggi 30 cm.

Pada pengujian sensor tangki penampungan diisi air dengan ketinggian 5 cm sehingga didapat waktu pengisian menuju ketinggian 5cm yaitu sekitar 1 menit. Pada pengujian ini disimulasikan *water pump* mengambil air dari *water source* yang berisi 3 liter air yang telah bersih

siap didistribusikan menuju tangki penampungan. Sensor akan membaca ketinggian air yang berada di tangki penampungan tersebut melalui gabus pembatas. Pengambilan data sensor untuk mengukur ketinggian level air perlu dilakukan agar mendapatkan hasil pengukuran yang lebih mendekati pada hasil pengukuran yang sebenarnya. Berikut adalah data dari pembacaan dari sensor *sharp dp* untuk mengukur ketinggian Air yang di *set* pada ketinggian 6 cm dalam waktu 1 menit yang terbaca dengan arduino mega 2560. Data pengujian pembacaan sensor dapat dilihat pada tabel 1.

Tabel 1 Data Pengujian Pembacaan Sensor

Waktu (s)	Ketinggian Air (cm)
00:01	1cm
00:02	1cm
00:03	1cm
00:04	1cm
00:05	1cm
00:06	1cm
00:07	1cm
00:08	1cm
00:09	1cm
00:10	1cm
00:11	1cm
00:12	1cm
00:13	1cm
00:14	2cm
00:15	1cm
00:16	2cm
00:17	2cm
00:18	2cm
00:19	1cm
00:20	1cm
00:21	2cm
00:22	2cm
00:23	2cm
00:24	2cm
00:25	2cm
00:26	2cm
00:27	3cm
00:28	2cm
00:29	3cm
00:30	3cm
00:31	3cm
00:32	3cm
00:33	2cm
00:34	3cm
00:35	3cm
00:36	3cm
00:37	3cm
00:38	3cm
00:39	3cm
00:40	3cm
00:41	3cm
00:42	4cm
00:43	4cm

00:44	4cm
00:45	4cm
00:46	4cm
00:47	3cm
00:48	4cm
00:49	4cm
00:50	4cm
00:51	4cm
00:52	5cm
00:53	6cm
00:54	5cm
00:55	5cm
00:56	5cm
00:57	5cm
00:58	5cm
00:59	6cm
00:60	6cm

### B. Pengujian Motor dan Driver Motor DC

Tujuan dari pengujian ini adalah mengetahui untuk mengetahui *output* dari *driver motor BTS L298* apabila diberi *input* yang berbeda-beda. Pada pengujian ini akan mengambil data *Pulse Width Modulation (PWM)* yang berfungsi sebagai mengatur daya yang akan diberikan kepada *water pump*. Berikut adalah data pengujian *driver motor L298*:

Tabel 2. Data Pengujian *Driver Motor* Terhadap Keluaran *Waterpump*

PWM	Voltage
10	0,43 V
20	0.88 V
30	1,38 V
40	1,84 V
50	2.32 V
60	2.79 V
70	3.25 V
80	3.71 V
90	4.17 V
100	4.63 V
110	5.10 V
120	5.56 V
130	6.02 V
140	6.48 V
150	6.94 V
160	7.40 V
170	7.84 V
180	8.33 V
190	8.79 V
200	9.26 V
210	9.72 V
220	10.17 V
230	10.61 V
240	11.05 V
250	11.59 V
255	11.73 V

Berdasarkan Tabel 2. dapat dilihat setiap *pwm* akan memberikan tegangan yang berbeda kepada *water pump*. Tabel memudahkan untuk menentukan *setpoint* yang dibutuhkan dalam proses pembuatan program di *arduino IDE*.

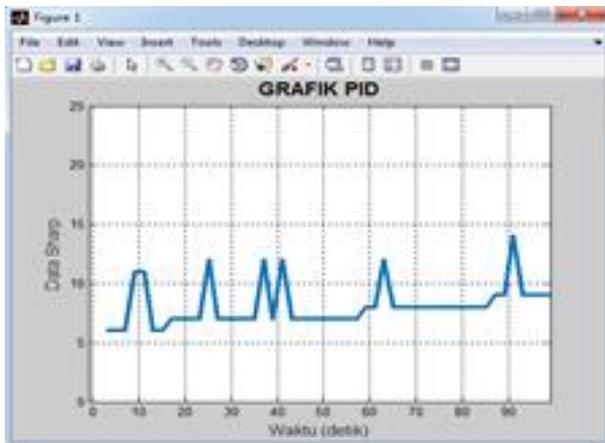
### C. Percobaan-percobaan penentuan nilai $K_p$ , $K_i$ , dan $K_d$

Penentuan nilai  $K_p$ ,  $K_i$  dan  $K_d$  dilakukan dengan metode *try and error*. Parameter akan dikatakan optimal apabila osilasi pada data *sharp* semakin berkurang atau dapat diminimalisasikan.

#### 1. Percobaan Pertama

Pada percobaan pertama dengan memasukkan nilai  $K_p : 1$ ,  $K_i : 1$ , dan  $K_d : 1$  maka dapat dilihat hasilnya pada gambar 5 masih terlalu banyaknya osilasi yang terdapat pada grafik PID yang mana membuat sistem belum stabil. Sehingga membutuhkan nilai  $K_p, K_i$ , dan  $K_d$  yang lain untuk meminimalisir osilasi yang terjadi pada grafik PID.

Percobaan ini dimulai dengan ketinggian air pada posisi 6cm dan di *setpoint* pada 8cm. Pada gambar 5 adalah grafik percobaan PID pertama.

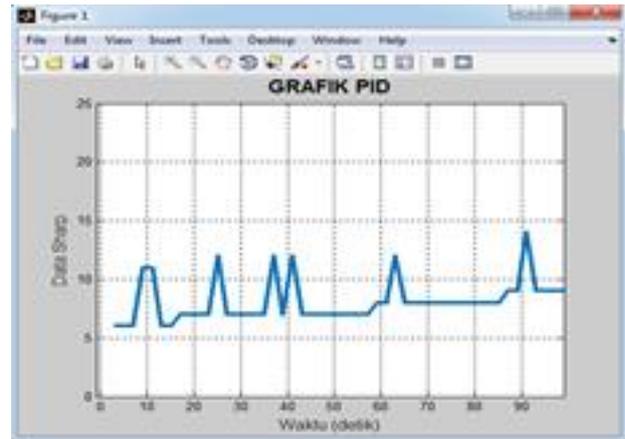


Gambar 5. Grafik percobaan PID pertama

#### 2. Percobaan Kedua

Di percobaan kedua penulis mencoba dengan nilai  $K_p : 2$ ,  $K_i : 0,1$ ,  $K_d : 2$  terdapat perubahan pada grafik PID dengan berkurangnya osilasi yang terjadi pada sistem dan membuat sistem lebih stabil dari sebelumnya dapat dilihat pada gambar 6.

Percobaan ini dimulai dengan ketinggian air pada posisi 6cm dan di *setpoint* pada 8cm. Pada gambar 6 adalah grafik percobaan PID kedua.

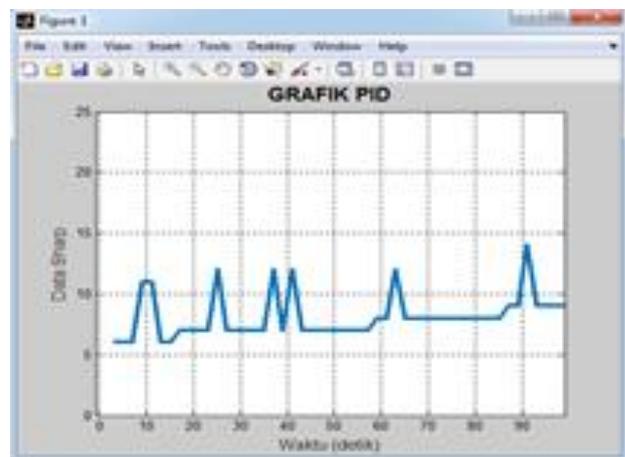


Gambar 6. Grafik percobaan PID kedua

#### 3. Percobaan Ketiga

Pada percobaan ketiga penulis mencoba dengan nilai  $K_p : 4$ ,  $K_i : 0,1$ ,  $K_d : 2$  terdapat perubahan pada grafik PID dengan berkurangnya osilasi yang terjadi pada sistem dan lonjakan osilasi yang tidak terlalu tinggi membuat sistem lebih stabil dari sebelumnya dapat dilihat pada gambar 7.

Percobaan ini dimulai dengan ketinggian air pada posisi 6cm dan di *setpoint* pada 8cm. Pada gambar 7 adalah grafik percobaan PID ketiga.

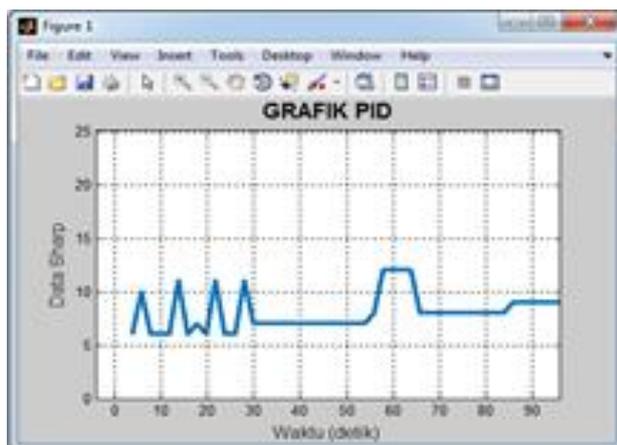


Gambar 7. Grafik percobaan PID ketiga

#### 4. Percobaan Keempat

Pada percobaan keempat penulis mencoba dengan nilai  $K_p : 4$ ,  $K_i : 1$ ,  $K_d : 2$  terdapat perubahan pada Grafik PID dengan osilasi yang hampir sama seperti percobaan ketiga tetapi osilasi yang terjadi seakan menumpuk kemudian berangsur stabil menuju ke *setpoint* yang telah ditentukan dapat dilihat pada gambar 8.

Percobaan ini dimulai dengan ketinggian air pada posisi 6cm dan di *setpoint* pada 8cm. Pada gambar 8 adalah grafik percobaan PID keempat.

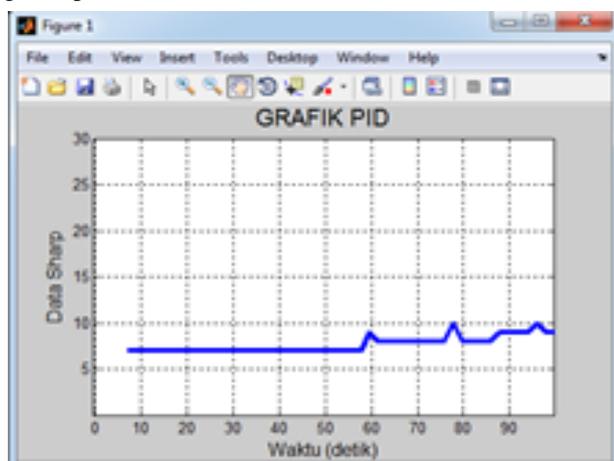


Gambar 8. Grafik percobaan PID keempat

### 5. Percobaan Kelima

Pada percobaan kelima sekaligus terakhir penulis mendapatkan nilai  $K_p$ ,  $K_i$ , dan  $K_d$  yang lebih stabil dari sebelumnya dengan lebih sedikitnya osilasi yang terjadi dan tidak terlalu tinggi dari percobaan-percobaan sebelumnya dengan memasukkan nilai  $K_p : 0,7$ ,  $K_i : 0,001$ ,  $K_d : 2$  dapat dilihat pada gambar 9.

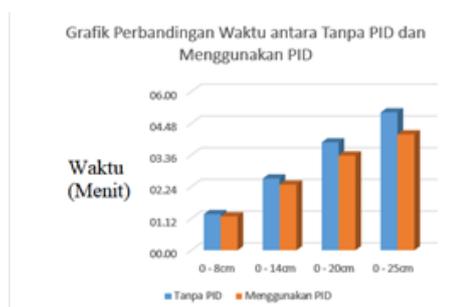
Percobaan ini dimulai dengan ketinggian air pada posisi 6cm dan di *setpoint* 8cm. Pada gambar 9 adalah grafik percobaan PID kelima.



Gambar 9. Grafik percobaan PID kelima

### D. Perbandingan Sistem Pengendalian Water pump dengan PID dan Tanpa PID

Perbandingan dibutuhkan untuk melakukan analisa perbedaan dari pengukuran sistem pengendalian *water pump* dengan menggunakan PID dan tanpa menggunakan PID disini perbandingan yang diambil adalah waktu tempuh menuju ke *setpoint* yang ditentukan bisa dilihat grafik perbandingan pada gambar 10.



Gambar 10. Grafik perbandingan waktu antara tanpa PID dan menggunakan PID

Berdasarkan grafik diatas dengan menggunakan pengendali PID maka waktu tempuh menuju ke *setpoint* lebih cepat dari pada tanpa menggunakan pengendali PID,  $K_p$ ,  $K_i$ , dan  $K_d$  yang di set berdasarkan *try and error* adalah  $K_p : 0,7$ ,  $K_i : 0,001$ ,  $K_d : 2$ , dari  $k_p$ ,  $k_i$  dan  $k_d$  yang telah di *set* maka didapat perbedaan antara menggunakan PID dan tanpa menggunakan PID yaitu waktu tempuh menuju ke *setpoint* lebih cepat dengan menggunakan pengedali PID. Rata-rata perbedaan waktu tempuh dari beberapa *setpoint* yaitu pada *setpoint* 8cm berbeda 5s, *setpoint* 14cm berbeda 14s, *setpoint* 20cm berbeda 30s, *setpoint* 25cm berbeda 50s.

### KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan, maka dapat diperoleh beberapa kesimpulan sebagai berikut:

1. Penggunaan PID membuat pengisian air ditangki saat menuju *setpoint* menjadi lebih cepat dibandingkan dengan tidak menggunakan metode PID dan pompa mati saat mencapai *setpoint* yang telah ditentukan.
2. Kinerja sensor dapat mempengaruhi data grafik PID yang diambil.

### DAFTAR PUSTAKA

- Eltaieb, A.A.M., Min, Z.J. (2015). Automatic Water Level Control System. International Journal of Science and Research. 4(12): 1505-1509.
- Hadi, D.S.N., Triwiyatno, A., and Setiyono, B. (2013). Pengendalian Level Air pada Plan Tangki Penampungan Sistem Pengolahan Air Limbah Menggunakan Metode Kontrol PI. Transmisi. 15(1): 28-35.
- Mittal, V. (2017). Automatic Water Level Controller. International Journal of Science and Research. 6(1): 136-138.
- Santoso, H. (2015). Panduan Praktis Arduino Untuk Pemula. Elangsakti.com: Jakarta.
- Simanjuntak, M.G. (2013). Perancangan Prototipe Smart Building berbasis Arduino UNO. Skripsi Ilmu Komputer 2(6): 6-14.