TELEKONTRAN, VOL. 10, NO. 2, OKTOBER 2022
DOI : 10.34010/telekontran.v10i2.9330
p-ISSN : 2303 – 2901
e-ISSN : 2654 – 7384

Pengaplikasian Kontroler PID Pada Sistem Kontrol Level Ketinggian Air Menggunakan MATLAB

Application of PID Controller in Water Level Control System Using MATLAB

Salma Rahmani, Suci Aulia Rosana, Ghina Hanin Tian
Teknik Elektro, Universitas Komputer Indonesia
Jalan Dipati Ukur No 112 – 116 Bandung, Indonesia
Email : salma.13120022@mahasiswa.unikom.ac.id

Abstrak – Salah satu masalah dalam dunia industri adalah sistem pengontrolan ketinggian air. Kontrol level air dilakukan pada tangki penampung air, yang merupakan masalah umum ketika level air pada tangki penampungan tidak diketahui, dan kurangnya pemantauan tangki penampungan dapat menyebabkan luapan atau pengosongan tangki. Penelitian ini bertujuan untuk mengontrol sistem ketinggian air agar mencapai titik level ketinggian air yang ditetapkan. Metodologi yang digunakan dengan memanfaatkan sistem fluida yang masuk ke tangki dan dapat memvariasikan kecepatan pompa air dengan motor DC, penelitian ini disimulasikan menggunakan aplikasi MATLAB. Telah dilakukan 4 jenis pengontrolan yaitu pengontrol P, PI, PD, dan PID, hal ini untuk mengetahui pengontrol yang paling baik diterapkan pada sistem kontrol level air. Pada penelitian ini yang mempunyai hasil terbaik adalah pengontrol PID karena mempunyai nilai *rise time* yang cepat, tidak memiliki *overshoot*, *settling time* dan kesalahan *steady-state* yang kecil. Dengan menggunakan kontrol PID, dapat mempertahankan level ketinggian air agar mencapai efisiensi dan produktivitas yang diperlukan.

Kata kunci *:* Ketinggian air, tangki, PID

Abstract *- One of the problems in the world industry is the water level control system. Water level control is carried out at the holding tank, which is a common problem when the water level in the holding tank is not known, and lack of monitoring of the holding tank can lead to storage space or tank emptying. This study aims to control the water level system so that it reaches the set water level level point. The methodology used is by utilizing the fluid system that enters the tank and can vary the speed of the water pump with a DC motor, this research is simulated using the MATLAB application. 4 types of control have been carried out, namely P, PI, PD, and PID controllers, this is to find out which controller is best applied to a water level control system. In this study, the PID controller has the best results because it has a fast rise time, no overshoot, settling time and small steady-state error. Using PID control, it is possible to maintain water level levels to achieve the required efficiency and productivity.*

Keywords *: Water level, tank, PID*

I. PENDAHULUAN

Air merupakan sumber daya alam yang sangat penting tidak hanya menopang kehidupan sehari-hari tetapi juga digunakan di banyak industri, kimia, komersial dan pertanian. Penggunaan dan pengolahan air berlebihan dapat mempengaruhi kelestarian lingkungan. Antara lain, disektor pertanian, pompa air harus dioperasikan untuk memasok air yang diatur ke perkebunan atau lahan pertanian. Air yang terkontrol dapat dikendalikan dengan merancang sistem kontrol elektronik otomatis [1], [2]. Salah satu masalah dalam dunia industri adalah sistem pengontrolan ketinggian air. Ketinggian air merupakan hal yang perlu diperhatikan untuk kelancaran proses produksi dan kualitas produk yang lebih baik [3]. Kontrol level air dilakukan pada tangki penampung air, yang merupakan masalah umum ketika level air pada tangki penampungan tidak diketahui, dan kurangnya pemantauan tangki penampungan dapat menyebabkan luapan atau pengosongan tangki [4]. Pengontrol *Proportional Integral Derivative* (PID) yaitu pengontrol yang terdiri dari konfigurasi standar Kp, Ki, dan Kd yang nilai nya tetap atau

disesuaikan untuk mencapai hasil atau kecepatan yang diinginkan, yaitu kecepatan dengan stabilitas yang baik dan tingkat kesalahan yang rendah [5]. Pengontrol PID, telah banyak digunakan dalam praktik industri selama 60 tahun terakhir. Penemuan kontrol PID pada tahun 1910 (terutama karena autopilot kapal Elmer Sperry) dan aturan penyetelan langsung Ziegler Nichols (ZN) 1942 [6]. Dalam pengaplikasian kontroler PID pada sistem kontrol level ketinggian air ini dapat diimplementasikan dengan menggunakan beberapa aplikasi pemrograman seperti MATLAB dan *Simulink*. MATLAB adalah sistem komputasi numerik interaktif yang banyak digunakan dalam pengajaran dan penelitian di indsutri dan akademisi. MATLAB menyediakan bahasa pemrograman modern dan lingkungan *debugging* dengan struktur data yang kuat, grafik yang dapat disesuaikan, dan alat pengeditan serta *debugging* yang mudah digunakan [7].

P. Berk* dkk. (2011) telah merancang sistem kontrol level air sintesis dengan logika *fuzzy* menggunakan simulasi MATLAB/*Simulink* 2008b. Hasil penelitian menunjukkan bahwa pengontrol logika *fuzzy* berguna pada aplikasi statistik non-linier, dimana kontroler PID klasik tidak memberikan hasil yang memuaskan. Pengontrol PID *fuzzy* dalam aplikasi proses kontrol level cairan terbukti menjadi pilihan tepat karena proses perencanaan pengontrol *fuzzy* relatif sederhana dan cocok untuk praktik rekayasa. Namun, dari respon langkah kontroler *fuzzy* dengan metode klasik untuk proses orde pertama menunjukkan bahwa nilai sebenarnya dari variable yang dikontrol mengambil nilai satu. Pengontrol PID *fuzzy* dengan metode klasik tidak memungkinkan penyimpangan dari regulasi. Tetapi juga tidak cocok untuk siklus kontrol *fuzzy* metode klasik dengan pengontrol PID *fuzzy* karena terdapat gangguan pada proses orde kedua [8].

Vineet Kumar dkk. (2008) melakukan studi banding mengevaluasi kinerja *real-time* dari *proportional – integral fuzzy* dengan kontroler *fuzzy proportional – derivative* (*Fuzzy* PI + *Fuzzy* PD) menggunakan *software* LabVIEW. Implementasi *real-time* dari kontroler *Fuzzy* PI + *Fuzzy* PD dilakukan dalam dua konfigurasi yaitu umpan balik dan kaskade. Percobaan ini menunjukkan perilaku yang sangat non-linier karena persentase pneumatik yang sama. Tetapi pada penelitian ini tidak ada perbandingan lanjut dari pengontrol *fuzzy* PID [9].

Apip Pudin mempelajari kontrol level cairan dalam tangki menggunakan simulasi *fuzzy* berbasis MATLAB, sistem terdiri dari katup (*valve*), tangki, sensor level, dan kontroler (FLC). Tetapi dalam penelitian ini menemukan bahwa respon kontroler PID masih meghasilkan *overshoot* dan *setting time* yang relatif lama [10]. Maka pada penelitian ini akan mengimplementasikan sistem kontrol level ketinggian air ini dengan pengaplikasian kontroler PID menggunakan MATLAB.

Berdasarkan latar belakang yang telah dikemukakan, peneliti memiliki tujuan dalam penelitian yaitu untuk mencapai kontrol proses yang sesuai dengan titik level ketinggian air yang ditetapkan, dan sebagai hasilnya meningkatkan efisiensi dan produktivitas. Serta dapat memahami perbandingan antara pengontrolan *proportional* (P), pengontrolan *proportional – integral* (PI), pengontrolan *proportional – derivative* (PD), dan pengontrolan *proportional – integral –derivative* (PID).

II. METODOLOGI

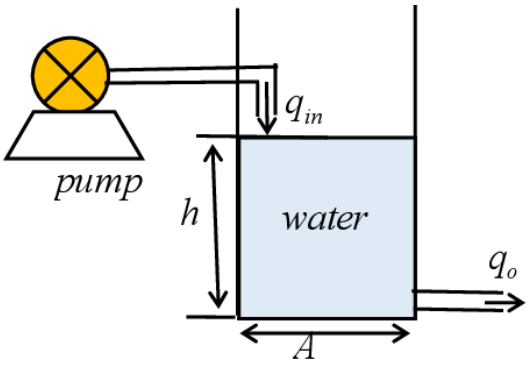
A. Model Perancangan Sistem Fluida

Sistem yang diusulkan terdiri dari pompa air dengan motor DC yang memompa air ke dalam wadah, pengontrol PID, dan tangki air atau sistem penyimpanan lainnya. Dapat dilihat pada **Gambar 1**, dimana motor memompa air ke dalam tangki dengan q_{in} [m^3/s] melalui pipa saluran masuk, tangki dengan luas penampang A [m^2] diisi hingga ketinggian air (h), dan air keluar dari wadah dengan kecepatan q_o [m^3/s] melalui pipa keluaran. Menggunakan keseimbangan arus masuk dan keluar dari tangki, ketinggian (h) berhubungan dengan q_{in} dan q_o seperti pada persamaan 1. Aliran keluar dari tangki juga dapat digabungkan dengan mengasumsikan hambatan linier terhadap aliran untuk menyederhanakan analisis, pada persamaan (2) [11].

$$q_{in} - q_o = A\frac{dh}{dt} \quad (1)$$

$$q_o = \frac{h}{R} \quad (2)$$

Aliran air di tangki dan *output* bisa stabil atau tidak stabil. Aliran dikatakan *steady-state* jika kecepatan di setiap titik konstan terhadap waktu. Ini tidak berarti bahwa kecepatannya sama di setiap titik, tapi tidak berubah setiap saat.



Gambar 1 Wadah Tangki Tunggal [12]

Kondisi *steady-state* dapat dicapai pada kecepatan fluida rendah. Kondisi aliran stabil diasumsikan jika area *output* terlalu besar, aliran melalui tangki penampung air tidak cukup lambat untuk menciptakan kondisi *steady-state* yang di asumsikan, dan model sistem tidak akan memprediksi aliran fluida secara akurat. Keseimbangan material di sekitar tangki level cairan dirumuskan seperti persamaan (3). Sedangkan untuk kondisi *steady-state* menggunakan persamaan (7) sampai dengan persamaan (11). Dari persamaan-persamaan di bawah dapat diketahui bahwa q_{in} dan q_o adalah laju aliran fluida, R adalah resistansi termal isolasi, $H_{(s)}$ merupakan variabel transformasi ketinggian air, dan $Q_{(s)}$ adalah transformasi laju aliran fluida [12].

$$q_{in} - \frac{h}{R} = A\frac{dh}{dt} \quad (3)$$

Terapkan transformasi *laplace*,

$$Q_{(s)} = \frac{1}{R.H_S + A_S.H_{(s)}} \quad (4)$$

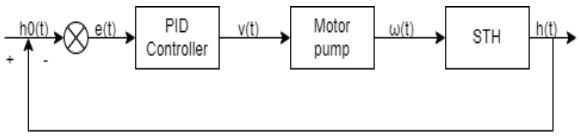
Fungsi alih sistem dari persamaan (4) adalah,

$$\frac{H_{(s)}}{Qi(s)} = \frac{R}{RA_S+1} \quad (5)$$

$$\frac{R.Q_o(s)}{Q_o(s)} = \frac{R}{\tau s+1} \quad (6)$$

$$G_{(s)} = \frac{1}{\tau s+1} \quad (7)$$

Seluruh sistem adalah sistem kontrol umpan balik seperti yang ditunjukkan pada diagram blok, yang ditunjukkan pada **Gambar 2**. Pertama, level umpan $h_0(t)$ diatur, yang menunjukkan level yang diinginkan dimana tangki harus diisi. Selanjutnya, pengontrol PID yang mengontrol kecepatan motor DC. Kecepatan motor berbanding lurus dengan laju aliran air q_{in} yang masuk ke tangki. Pada *output* dari keseluruhan sistem, memiliki level air h(t) dan informasi ini diumpankan kembali ke *input* dan dibandingkan dengan level referensi yang diinginkan. Sinyal kesalahan antara *output* aktual dan referensi e(t) adalah sinyal *input* dari pengontrol PID, dan kecepatan motor [ω(t) rad/sekon] disesuaikan (naik atau turun) untuk menyesuaikan q_{in} dengan laju aliran. Sampai ketinggian air yang dibutuhkan tercapai. Data putaran mesin diasumsikan diperoleh dari alat pengukur kecepatan seperti *tachometer*. Kecepatan diubah menjadi *Speed to Height Transformation* (STH) oleh blok transformasi yang menghubungkan kecepatan dengan kecepatan aliran dan kemudian dengan ketinggian air h(t). Untuk menyederhanakan studi, diasumsikan hubungan linier sederhana antara kecepatan dan laju aliran tangki penampung air di persamaan (8).

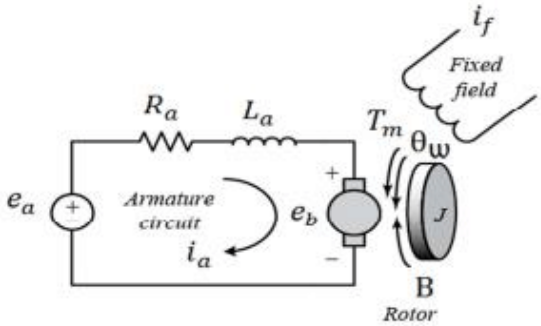


Gambar 2 Diagram Blok Sistem Pengontrol Ketinggian Air [12]

$$\omega(t) = K_f.q_{in}(t) \quad (8)$$

B. Model Perancangan Pompa Motor dan Kontroler PID

Pompa air adalah motor DC dengan rangkaian ekuivalen elektrik dari armatur dan diagram benda bebas dari rotor seperti yang ditunjukkan pada **Gambar 3**. Diasumsikan bahwa torsi yang dikembangkan oleh motor berbanding lurus dengan ϕ(t) dan arus pada armatur. Dalam transformasi *laplace* $i_a = I_a$ adalah arus konstan pada armatur dan K_m didefinisikan sebagai konstanta motor. Arus keluaran sebanding dengan tegangan medan seperti pada persamaan (9) sampai persamaan (14).



Gambar 3 Rangkaian Ekivalen Motor DC [12]

$$T_m(s) = (K_1K_fI_f)I_a(s) = K_mI_a(s) \quad (9)$$

$$V_a(s) = (R_a + L_a s)I_a(s) + V_b(s) \quad (10)$$

$$V_b(s) = K_b\omega(s) \quad (11)$$

$$I_a(s) = \frac{V_a(s) - K_b\omega(s)}{R_a + L_a s} \quad (12)$$

$$T_L(s) = Js^2\theta(s) + bs\theta(s) \quad (13)$$

$$G(s) = \frac{\theta(s)}{V_a(s)} = \frac{K_m}{s[(R_a+L_a s)(Js+b)+K_b K_m]} \quad (14)$$

Kontroler PID merupakan bagian lain dari sistem seperti pada **Gambar 2**. Tugas kontroler PID adalah mengatur *output*, dalam hal ini level ketinggian air pada titik yang diinginkan, agar tidak terjadi kesalahan antara *output* yang diamati, h(t), dan tingkat referensi yag diinginkan. Dalam desain PID secara umum, kesalahan e(t) sesuai dengan *output* v(t) pada persamaan (15) [13].

$$v(t) = K_p e(t) + K_i \int e(t) + K_d \frac{de(t)}{dt} \quad (15)$$

Kp, Ki, dan Kd masing-masing disebut sebagai nilai parameter *proportional*, nilai parameter *integral*, dan nilai parameter *derivative*. Dengan menggunakan tiga nilai parameter, pengontrol PID dapat memenuhi persyaratan proses tertentu. Menggunakan transformasi *laplace*, hubungan antara *input-output* dari kontroler seperti pada persamaan (16).

$$C(s) = \frac{V(s)}{e(s)} = K_p + \frac{K_i}{s} + sK_d \quad (16)$$

III. HASIL DAN PEMBAHASAN

Pada penelitian ini dilakukan 4 jenis pengontrolan, yaitu pengontrolan P, pengontrolan PI, pengontrolan PD, dan pengontrolan PID. Masing-masing memiliki parameter konstanta tertentu yang harus dtentukan supaya dapat beroperasi dengan baik. Parameter-parameter yang ditentukan tidak bersifat independen. sehingga pada saat salah satu nilainya diubah, memungkinkan sistem tidak memberikan reaksi seperti yang diinginkan. Percobaan dan pengujian pada penelitian ini menggunakan aplikasi MATLAB. Aplikasi MATLAB mampu menganalisa suatu sistem apabila diberi masukan berupa *transfer function* yang ditulis dalam bentuk tranformasi *laplace* (dalam s-dominan) atau matriks. *Transfer function* yang digunakan pada penelitian ini dapat dilihat pada persamaan (17).

$$H(s) = \frac{1}{s^2+10s+20} \quad (17)$$

Nilai-nilai pada persamaan (17) adalah persamaan yang sering digunakan pada penelitian-penelitian lain yang sudah dilakukan dan terbukti berhasil. *Transfer function* tersebut masih bersifat *Open-Loop* karena belum digabungkan dengan pengontrol sehingga masih memiliki *steady state error* yang tinggi dan *rise time* yang lambat. *Transfer function* tersebut perlu digabungkan dengan pengontrol supaya dapat bersifat *Closed-Loop* dan akan memenuhi kriteria perancangan menghasilkan *rise time* yang cepat, *overshoot* yang kecil, dan tidak memiliki *steady state error*. *Transfer function* persamaan (17) akan diterapkan pada pengontrolan P, pengontrolan PI, pengontrolan PD, dan pengontrolan PID.

A. Pengontrolan *Proportional*

Pengontrol *proportional* mempunyai keluaran yang proporsional dengan besarnya sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya). Meningkatkan nilai Kp mengakibatkan efek meningkatkan sinyal kontrol secara proporsional untuk tingkat kesalahan yang sama.

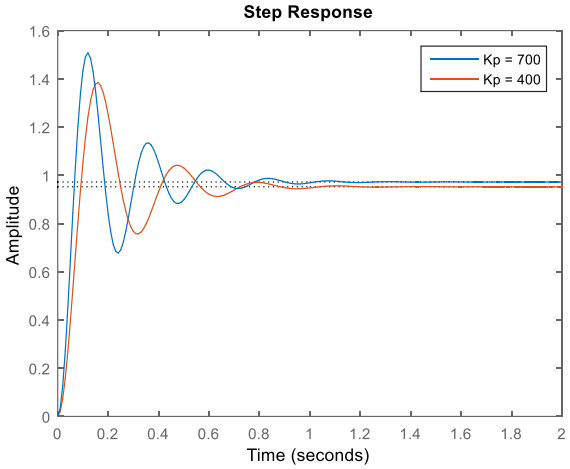
Pada persamaan (17) akan diterapkan untuk perhitungan mencari nilai T, yang nantinya akan menghasilkan grafik pengontrol *proportional*. Pada saat mensimulasikan menggunakan program MatLab akan ditentukan parameter nilai Kp. Program MatLab untuk mensimulasikan pengontrolan P dapat dilihat pada **Gambar 4**.

```
s = tf('s');
P = 1/(s^2 + 10*s + 20);
step(P);

Kp = 400;
C = pid(Kp);
T = feedback(C*P,1);

t = 0:0.01:2;
step(T,t);
```

Gambar 4 Simulasi MATLAB Pengontrol P

Pada perancangan pengontrolan P menggunakan MATLAB, menentapkan nilai Kp sebesar 400. Untuk mencari nilai Kp digunakan metode *trial and error*, yaitu dengan mencari nilai parameter dengan cara mencoba-coba suatu nilai tertentu sebagai parameter sampai mendapatkan sebuah performansi kontrol P yang baik. Sehingga nilai Kp sebesar 400 pada pengontrol P didapatkan

setelah dilakukan *trial and error*. Pada pengontrolan P dengan menggunakan program pada **Gambar 4** telah menghasilkan grafik yang telah ditampilkan pada **Gambar 5**.



Gambar 5 Grafik Hasil Simulasi MATLAB Pengontrol P

Dari grafik yang telah dihasilkan, bahwa pengontrolan P dengan menerapkan nilai parameter Kp sebesar 400 dapat mengurangi *rise time*, mengurangi error *steady state* tetapi terlihat pada grafik bahwa mengakibatkan nilai *overshoot* yang besar dan *settling time* yang besar. Terlihat perbandingannya pada **Gambar 5** bahwa jika nilai Kp diperbesar maka nilai *overshoot* akan semakin besar dan memiliki *settling time* yang besar tetapi memiliki *rise time* yang kecil dan dapat mengurangi *error steady state*. Sebaliknya jika nilai Kp diperkecil maka menghasilkan nilai *overshoot* dan *settling time* yang lebih kecil, namun menghasilkan nilai *rise time* yang lebih besar dan dapat meningkatkan nilai kesalahan *steady-state*.

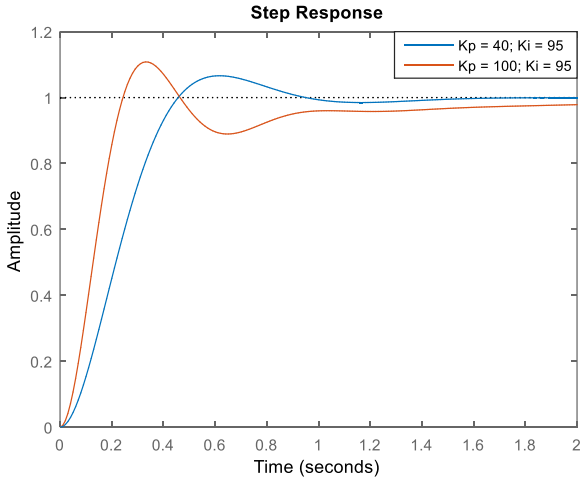
B. Pengontrol *Propotional – Integral*

Pengontrolan PI adalah pengontrolan *proportional* yang digabung dengan pengontrolan integral. Pengontrolan integral berfungsi menghasilkan respon sistem yang memiliki kesalahan mantap nol (*error steady state* bernilai nol). Persamaan (17) akan diterapkan pada program untuk mensimulasikan pengontrol PI di aplikasi MATLAB. Pada pengontrolan PI akan menggunakan parameter konstanta Ki. Pengontrol *proportional* dan intgeral memiliki karakteristik yang sama yaitu dalam hal *rise time* dan *overshoot*. Sehingga nilai Kp harus dikurangi untuk menghindari *overshoot* yang besar, nilai Ki akan dibuat lebih besar dari Kp. Hal ini karena akan menghilangkan kesalahan *steady-state*, jika nilai Kp lebih besar daripada nilai Ki maka kesalahan *steady-state* tidak dapat dihilangkan. Program MATLAB untuk mensimulasikan pengontrol PI dapat dilihat pada **Gambar 6**.

```
s = tf('s');
P = 1/(s^2 + 10*s + 20);
step(P)

Kp = 40;
Ki = 95;
C = pid(Kp,Ki);
T = feedback(C*P,1);

t = 0:0.01:2;
step(T,t)
```

Gambar 6 Simulasi MATLAB Pengontrol PI

Pada pengontrolan PI menetapkan nilai Kp sebesar 40 dan nilai Ki sebesar 95. Untuk menentukan nilai Kp dan Ki dilakukan dengan cara metode *trial and error*, dengan mencari dan mencoba nilai parameter sampai mendapatkan sebuah performansi pengontrol PI yang baik. Sehingga nilai Kp dan Ki yang ditetapkan pada pengontrol PI adalah nilai yang paling baik menurut peneliti setelah dilakukan *trial and error*. Nilai Kp dan nilai Ki akan sangat mempengaruhi grafik sistem yang dihasilkan. Untuk melihat Grafik sistem dapat dilihat pada **Gambar 7**.



Gambar 7 Hasil Simulasi MATLAB Pengontrol PI

Dari grafik sistem pengontrolan PI telah dapat dianalisa bahwa nilai nilai *rise time* mengalami penurunan, memiliki *overshoot* yang jauh lebih kecil, dan memiliki kesalahan *steady-state* yang hampir dapat dihilangkan. Terlihat perbandingannya pada grafik sistem bahwa apabila nilai Kp diperbesar maka akan menghasilkan *overshoot* yang semakin besar tetapi menghasilkan rise time yang lebih cepat.

C. Pengontrol *Proportional - Derivative*

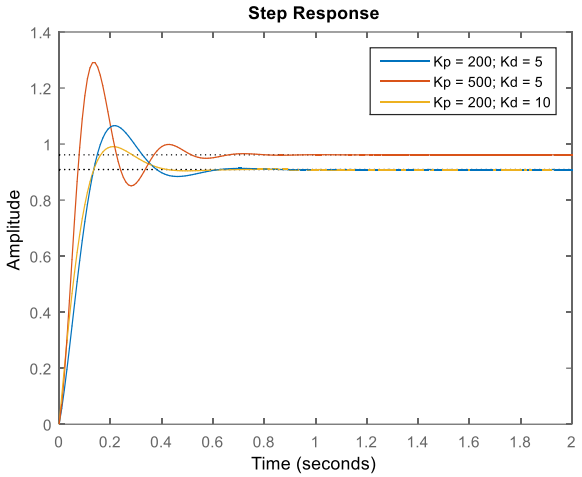
Pengontrolan PD merupakan pengontrolan gabungan dari pengontrol *proportional* dan pengontrol *derivative*. Pengontrol *derivative* adalah pengontrol laju karena *output* kontroler sebanding dengan laju perubahan sinyal error. Pada pengontrol *proportional-derivative* akan diubah parameter nilai Kp dan parameter nilai Kd.

Pengontrolan PD jika sinyal kesalahan berubah terhadap waktu, maka keluaran yang dihasilkan pengontrol tergantung pada nilai parameter konstanta Kd dan laju perubahan sinyal kesalahan. Dengan meningkatkan nilai Kd, dapat meningkatkan stabilitas sistem dan mengurangi *overshoot*. Untuk mensimulasikan pengontrol PD dapat dilakukan di MATLAB dengan memprogramnya seperti pada **Gambar 8**.

```
s = tf('s');
P = 1/(s^2 + 10*s + 20);
step(P)

Kp = 200;
Kd = 5;
C = pid(Kp,0,Kd);
T = feedback(C*P,1);

t = 0:0.01:2;
step(T,t)
```

Gambar 8 Simulasi MATLAB PD

Pada pengontrol PD menetapkan nilai Kp sebesar 200 dan nilai Kd sebesar 5. Untuk mencari nilai Kp dan Kd digunakan metode *trial and error*, yaitu dengan mencari nilai parameter dengan cara mencoba suatu nilai tertentu sebagai parameter sampai mendapatkan sebuah performansi kontrol PD yang baik. Sehingga nilai Kp sebesar 200 dan Kd sebesar 5 pada pengontrol PD menurut peneliti adalah nilai yang paling baik setelah dilakukan *trial and error*. Nilai Kp dan nilai Kd akan berpengaruh pada grafik sistem yang dihasilkan. Untuk melihat grafik sistem yang dihasilkan dapat dilihat pada **Gambar 9**.



Gambar 9 Hasil Simulasi MATLAB Pengontrol PD

Dari grafik sistem yang dihasilkan dapat dianalisa bahwa pengontrol PD dapat mengurangi *overshoot* dan mengurangi *settling time*, serta memiliki efek yang dapat diabaikan pada waktu naik dan kesalahan dengan keadaan tunak. Terlihat perbandingan pada grafik bahwa jika nilai Kp semakin besar maka akan menimbulkan *overshoot* yang semakin besar pula, namun jika nilai Kd yang diperbesar akan menghasilkan nilai *rise time* yang semakin cepat dan *overshoot* yang sangat kecil.

D. Pengontrolan *Proportional – Integral - Derivative*

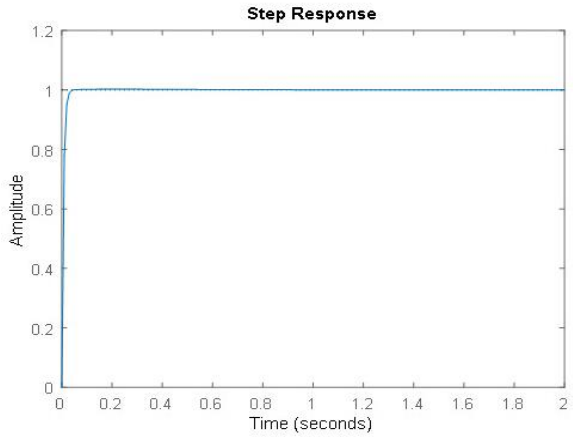
Pengontrol PID adalah gabungan dari pengontrolan *proportional, integral, dan derivative*. Setiap kekurangan dan kelebihan masing-masing pengontrol dapat menutupi dengan menggabungkan ketiganya secara paralel. Pengontrol P, I, dan D bertujuan mempercepat reaksi sebuah sistem mencapai *set point*, menghilangkan *offset*, dan menghasilkan perubahan awal yang besar dan dapat mengurangi *overshoot.*

Pada pengontrol PID akan menambahkan nilai parameter Kp, Ki, dan Kd. Kemudian mencari nilai Kp, Ki, dan Kd yang tepat sehingga menghasilkan kriteria yang sesuai. Untuk mensimulasikan pengontrol PID dapat dilakukan menggunakan aplikasi MATLAB yang programnya dapat dilihat pada **Gambar 10**.

```
s = tf('s');
P = 1/(s^2 + 10*s + 20);
step(P)

Kp = 1511;
Ki = 3776;
Kd = 151.1;
C = pid(Kp,Ki,Kd);
T = feedback(C*P,1);

t = 0:0.01:2;
step(T,t)
```

Gambar 10 Simulasi MATLAB PID

Untuk mensimulasikan pengontrol PID mengubah nilai Kp sebesar 1511, nilai Ki sebesar 3776, dan nilai Kd sebesar 151.1. Untuk mencari nilai Kp, Ki, dan Kd digunakan metode *trial and error*, yaitu dengan mencari nilai parameter dengan cara mencoba suatu nilai tertentu sebagai parameter sampai mendapatkan sebuah performansi kontrol PID yang baik. Sehingga nilai Kp sebesar 1511, Ki sebesar 3776, dan Kd sebesar 151.1 pada pengontrol PID menurut peneliti adalah nilai yang paling baik setelah dilakukan *trial and error*. Ketiga nilai parameter tersebut akan sangat berpengaruh untuk grafik sistem yang dihasilkan. Untuk melihat grafik pengontrol PID dapat dilihat pada **Gambar 11**.



Gambar 11 Hasil Simulasi MATLAB Pengontrol PID

Berdasarkan grafik yang dihasilkan, dapat dianalisa bahwa grafik sudah mencapai kriteria sistem yang diinginkan. Pengontrol PID memiliki *rise time* yang cepat, tidak memiliki *overshoot*, dan tidak memiliki *error steady state*. Pengaturan nilai Kp, Ki, dan Kd akan mengakibatkan penonjolan sifat dari masing-masing elemen. Parameter konstanta yang menonjol itu yang akan memberikan kontribusi pengaruh kepada respon sistem secara keseluruhan.

Tabel 1 menunjukkan perbandingan kontroler yang berbeda dalam hal *rise time*, *settling time*, *overshoot* dan kondisi *steady-state*. Dari **Tabel 1** dapat dilihat bahwa pengontrol *proportional* cenderung mengurangi *rise time* dan *setting time*, tetapi tidak menghilangkan kesalahan pada *steady-state*, dan *overshoot* yang besar.

Tabel 1. Perbandingan Pengontrol P, PI, PD, dan PID

Tipe Pengontrol	Rise Time	Setling time	Over shoot	Error Steady State
P K_p =400	0,06	0,68	45,5	0,03
PI K_p =40; K_i = 95;	0,29	0,83	6,64	0,05
PD K_p =200; K_d =5;	0,1	0,48	15,2	0,02
PID K_p =1511; K_i =3776; K_d =151.5;	0,01	0,02	0	0,001

Pengontrol *proportional - integral* (PI) menghilangkan kesalahan saat keadaan *steady-state* ketika nilai akhir mencapai input referensi yang diinginkan, tetapi menurunkan respons transien dengan *overshoot* yang lebih besar dan juga *settling time* yang lama. Pengontrol *proportional – derivative* (PD) meningkatkan stabilitas sistem dengan mengurangi *overshoot*, *rise time*, *settling time*, dan secara umum meningkatkan respons transien. Namun, masih ada kesalahan *steady-state* yang besar. Pengontrol *proportional – integral – derivative* (PID) bekerja dengan baik dengan semua parameter. Oleh karena itu, dengan menyesuaikan dan memilih nilai Kp, Ki, dan Kd yang tepat, untuk memenuhi persyaratan desain dan mencapai kinerja optimal dengan akurasi yang baik.

IV. KESIMPULAN

Penelitian ini menyajikan sistem pompa air yang dikendalikan oleh pengontrol PID, untuk mempertahankan ketinggian air yang diinginkan dalam tangki penampungan air. Kecepatan motor dan aliran air ke tangki dikontrol dengan menyesuaikan nilai-nilai parameter Kp, Ki, dan Kd dalam kontroler PID. Sistem kontrol level air dengan kontroler PID ini digunakan untuk meminimalkan *overshoot*, meningkatkan *settling*

time (respons transien yang ditingkatkan), dan kesalahan *steady-state*.

Dapat disimpulkan bahwa perbandingan dari 4 pengontrol adalah pada pengontrol P mampu mengurangi kesalahan *steady-state*, mempunyai *overshoot* yang besar dan *settling time* yang besar. Pada pengontrol PI mampu menurunkan *rise time*, memiliki *overshoot* yang kecil, dan memiliki kesalahan *steady-state* yang hampir dapat dihilangkan. Pada pengontrol PD masih terdapat *overshoot*, mengurangi *settling time*, serta memiliki efek yang dapat diabaikan pada waktu naik dan kesalahan dengan keadaan *steady-state*. Pada pengontrol PID dapat menghasilkan *rise time* yang cepat, tidak memiliki *overshoot*, dan tidak memiliki kesalahan *steady-state*. .Dari ke-4 jenis pengontrolan P, PI, PD, dan PID, yang memiliki hasil terbaik adalah pengontrol PID. Karena, mempunyai nilai *rise time* yang tepat, tidak memiliki *overshoot*, *settling time* dan kesalahan *steady-state* yang kecil. Sehingga sistem pengontrolan ini sangat diperlukan untuk mempertahankan level ketinggian air agar mencapai efisiensi dan produktivitas yang diperlukan.

DAFTAR PUSTAKA

[1] Beza Negash Getu dan Hussain A. Attia, "Automatic Control of Agricultural Pumps Based on Soil Moisture Sensing," *Proceedings of the IEEE AFRICON 2015 Conference*, pp. 667-671, 14-17 September 2015.

[2] Beza N. Getu, Nasser A. Hamad, dan Hussain A. Attia, "Remote Controlling of an Agricultural Pump System Based on the Dual Tone Multi-Frequency (DTMF) Technique," *Journal of Engineering Science and Technology (JESTECH)*, vol. 10, no.10, pp. 1261-1274, 2015.

[3] Laith Abed Sabri dan Hussein Ahmed Al-Mshat, "Implementation Fuzzy and PID Controller to Water Level System Using Labview," *International Journal of Computer Applications*, vol. 116, no. 11, 2015.

[4] Fahmi Fahroje Pane dkk, "Sistem Pengendalian Water Pump Untuk Mengatur Tinggi Level Air dengan Algoritma PID pada Plant Water Treatment," *Seminar Nasional Applicable Innovation of Engineering and Science Research (AvoER) XI*, 336-341, 2019.

[5] Alifa Restu Janwar Wiriawan dan Andry Wirawan, "Pengaturan Kecepatan Motor DC dengan Kontrol Proporsional Integral Derifatif (PID) Berbasis Labview," *Telekontran*, vol. 4, no. 2, Oktober 2016.

[6] J. Swder, G. Wszoek, W. Carvalho, "Programmable Controller Design Electropneumatic Systems," *Journal of Material Processing Technology*, 164-1655, 2005.

[7] Desmond J. Higham dan Nicholas J. Higham, "MATLAB Guide," *SIAM*, 2016.

[8] P. Berk* dkk, "Synthesis Water Level Control by Fuzzy Logic," *Journal of Achievements in Materials and Manufacturing Engineering*, vol. 45, issue 2, April 2011.

[9] Vineet Kumar dkk, "Real-Time Performance Evaluation of a Fuzzy PI + Fuzzy PD Controller for Liquid-Level Process," *International Journal of Intelligent Control and Systems*, vol. 13, no. 2, Juni 2008.

[10] Apip Pudin, "Pengendalian Level Cairan Tangki dengan Menggunakan Simulasi Fuzzy Berbasis Matlab," *Jurnal Teknik Energi*, vol. 3, no. 1, April 2013.

[11] Richard C. Dorf dan Robert H. Bishop, "Modern Control Systems," *Pearson*, 2017.

[12] Sony Cahya Pratama dkk, "Design and Implementation of Water Level Control Using Gain Scheduling PID Back Calculation Integrator Anti Windup," *International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, 2016.

[13] Beza Negash Getu, "Water Level Controlling System Using PID Controller," *International Journal of Applied Engineering Research*, vol. 11, no. 23, pp. 11223-11227, 2016.