

Pengaturan Level Ketinggian Air Menggunakan Kontrol PID

Thiang, Yohanes TDS, Andre Mulya

Fakultas Teknologi Industri, Jurusan Teknik Elektro, Universitas Kristen Petra
email: thiang@petra.ac.id

Abstrak

Makalah ini menjelaskan tentang aplikasi sistem kontrol pada pengaturan level ketinggian air. Metode kontrol yang digunakan dalam sistem ini adalah kontrol proporsional integral derivatif (PID). Semua proses kontrol dan akuisisi data dilakukan dengan menggunakan komputer (PC). Pengujian sistem telah dilakukan untuk mengontrol level air dengan ketinggian antara 0 – 30 cm. Pengujian dilakukan dengan beberapa variasi parameter kontrol PID dengan input berupa fungsi step. Dari hasil pengujian didapatkan bahwa parameter sistem kontrol level ketinggian air adalah $K_p = 1 - 10$, $K_i = 0,1 - 5$ dan $K_d = 0,1 - 10$.

Kata kunci: sistem kontrol, kontrol proporsional integral derivatif (PID), kontrol level air.

Abstract

This paper describes about the application of control system on liquid level control system. The control method used in this system is proportional integral derivative (PID) controller. All control processes and data acquisition is run by using a personal computer (PC). Experiment was done by controlling liquid level in level range 0 – 30 cm. Experiment was also done by using various PID control parameter in which the input signal is step function. Experiment results show that the parameter of liquid level control system are $K_p = 1 - 10$, $K_i = 0,1 - 5$ and $K_d = 0,1 - 10$.

Keywords : control system, proportional integral derivative controller (PID), liquid level control.

1. Pendahuluan

Dalam era globalisasi, sektor industri memegang peranan yang sangat penting khususnya di Indonesia. Banyak industri-industri yang berkembang dengan pesat. Di dalam industri, sangat dibutuhkan sistem kontrol yang baik untuk dapat menunjang proses berjalannya industri tersebut dan untuk meningkatkan efisiensi dalam proses produksi. Karena itu dalam makalah ini dipaparkan suatu eksperimen sistem kontrol yang diharapkan dapat memberikan kontribusi dalam dunia industri.

Dalam makalah ini akan dipaparkan tentang sistem kontrol ketinggian air dengan menggunakan metode kontrol proporsional integral derivatif (PID). Sistem kontrol PID itu sendiri akan dibahas pada bagian kedua dari makalah ini. Sedangkan bagian ketiga dari makalah ini akan memaparkan penjelasan tentang sistem yang telah dirancang baik perangkat lunak dan perangkat keras. Bagian keempat dari makalah ini akan memaparkan hasil pengujian yang telah

dilakukan dan makalah ini akan ditutup dengan beberapa kesimpulan yang telah didapatkan.

2. Kontrol PID

Kontroler Proporsional Integral Derivatif (PID) adalah kontroler yang menggabungkan kontroler proporsional, integral dan derivatif. Kontroler ini direpresentasikan dengan persamaan sebagai berikut [1]:

$$m(t) = K_p \cdot e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (1)$$

dimana K_p adalah konstanta proporsional, T_i menyatakan waktu integral dan T_d menyatakan waktu derivatif.

Persamaan 1 adalah persamaan dalam domain waktu. Untuk memudahkan penulisan dalam program, maka persamaan 1 dikonversikan ke dalam bentuk diskrit, dengan menggunakan *finite differential* orde pertama yang direpresentasikan dalam persamaan berikut:

$$\left. \frac{df}{dt} \right|_k = \frac{(f_k - f_{k-1})}{\Delta t} \quad (2)$$

Catatan: Diskusi untuk makalah ini diterima sebelum tanggal 1 Desember 2004. Diskusi yang layak muat akan diterbitkan pada Jurnal Teknik Elektro volume 5, nomor 1, Maret 2005.

$$\int e(t)dt = \sum_{k=0}^n e_k \cdot \Delta t \quad (3)$$

Sehingga persamaan 1 menjadi:

$$m_n = K_p \left[T_d \frac{(e_n - e_{n-1})}{\Delta t} + e_n + \frac{1}{T_i} \sum_{k=0}^n e_k \cdot \Delta t \right] \quad (4)$$

Dimana:

$$K_i = K_p \frac{T_s}{T_i} \text{ dan } K_d = K_p \frac{T_d}{T_s} \text{ dengan } \Delta t = T_s$$

Apabila

$$S_n = S_{n-1} + e_n \quad (5)$$

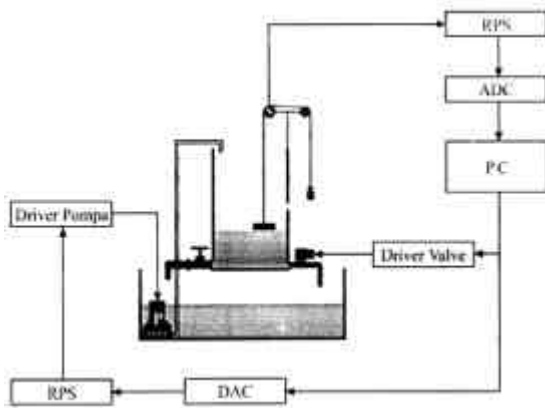
Maka persamaan kontroler PID dalam bentuk diskrit adalah sebagai berikut[1]:

$$m_n = K_p \cdot e_n + K_i \cdot S_n + K_d \cdot (e_n - e_{n-1}) \quad (6)$$

dimana S_n = Jumlah error, S_{n-1} = jumlah error sebelumnya, e_n = error sekarang, e_{n-1} = error sebelumnya, m_n = output sekarang.

3. Deskripsi Sistem

3.1 Perangkat Keras

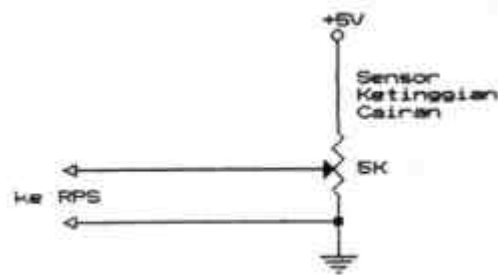


Gambar 1. Blok Diagram Perangkat Keras Sistem

Gambar 1 menunjukkan blok diagram sistem yang telah dibuat. *Plant* terdiri dari dua buah bejana atau tangki yang terbuat dari kaca. Tangki yang bawah merupakan tempat penampungan air berukuran 39 x 34 x 35 cm. Tangki yang di atas berfungsi sebagai *plant* yang diatur level ketinggian airnya. Tangki ini berukuran 19 x 14 x 40,5 cm. Sebuah pompa digunakan untuk menaikkan air dari tangki penampungan ke dalam tangki *plant*. Spesifikasi pompa yang digunakan adalah 12 VDC, 3,5 A. Untuk

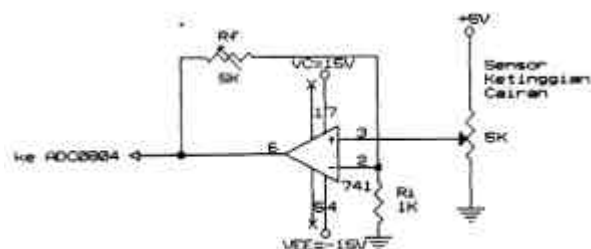
mengatur pembuangan air digunakan sebuah stop kran dan sebuah *solenoid valve*. *Stop* kran digunakan untuk pembuangan air secara manual dan *solenoid valve* untuk pembuangan air yang dikontrol melalui *personal computer* (PC).

Sensor yang digunakan untuk mengukur ketinggian air adalah potensiometer. Untuk mengkonversi gerakan naik-turun ketinggian air menjadi gerakan rotasi yang akan memutar potensiometer, digunakan sebuah pelampung dan pemberat yang dirangkai dalam sebuah sistem katrol (lihat gambar 1). Besar potensiometer yang digunakan adalah 5 k Ω . Berikut gambar 2 memperlihatkan rangkaian potensiometer yang mengubah ketinggian air menjadi tegangan. Rangkaian ini mengeluarkan tegangan output sebesar 0 – 1,194 volt untuk *range* pengukuran ketinggian air dari 0 – 30 cm.



Gambar 2. Rangkaian Sensor Potensiometer

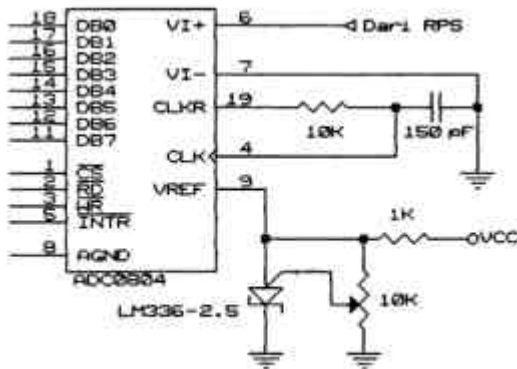
Karena tegangan input ADC dirancang sebesar 0 – 5 volt maka diperlukan sebuah rangkaian *amplifier* sebagai penguat tegangan dengan faktor penguatan sebesar 4,187. Rangkaian yang digunakan adalah rangkaian *non inverting amplifier* dengan rangkaian seperti yang ditunjukkan pada gambar 3.



Gambar 3. Rangkaian Non Inverting Amplifier

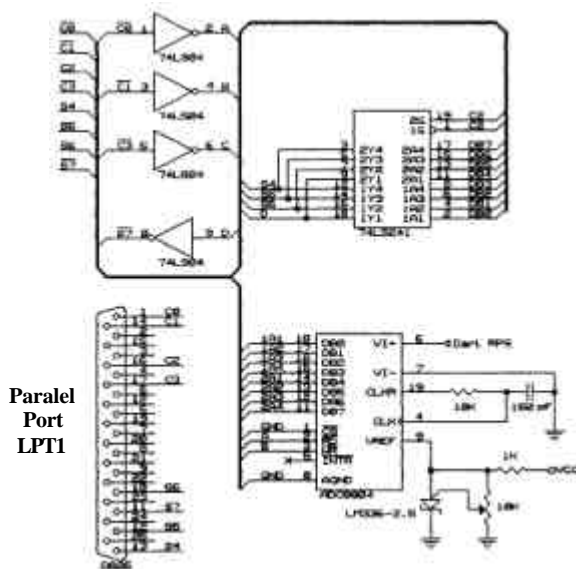
Rangkaian *analog to digital converter* (ADC) diperlukan untuk mengubah data ketinggian air yang sinyal *analog* menjadi *digital* untuk dapat dibaca oleh PC. Dalam aplikasi ini, ADC yang digunakan adalah ADC 0804 yang mempunyai satu input analog dan output digital 8 bit.

Rangkaian ADC ini membutuhkan tegangan referensi 2,5 volt. Tegangan referensi ini dihasilkan dari LM336-2,5. Secara lengkap rangkaian ADC dapat dilihat pada gambar 4.



Gambar 4. Rangkaian ADC 0804

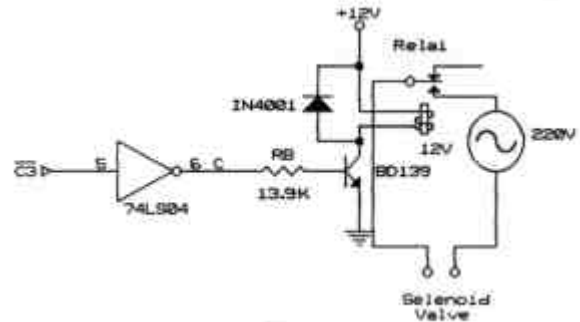
Output ADC dihubungkan ke PC melalui *parallel port* LPT1. LPT1 hanya mempunyai 5 bit input sedangkan ADC mempunyai output 8 bit. Karena itu untuk dapat melakukan pembacaan 8 bit, maka digunakan sebuah multiplexer yaitu IC 74LS241. Pin input yang digunakan adalah pin S4, S5, S6 dan S7. Pin-pin ini dapat diakses oleh program melalui port dengan alamat 379h. Sinyal kontrol ADC (WR dan RD) dan multiplexer (1G dan 2G) dihubungkan melalui pin-pin C0, C1 dan C2. Rangkaian lengkap koneksi ini dapat dilihat pada gambar 5.



Gambar 5. Koneksi ADC ke *Parallel Port* LPT1

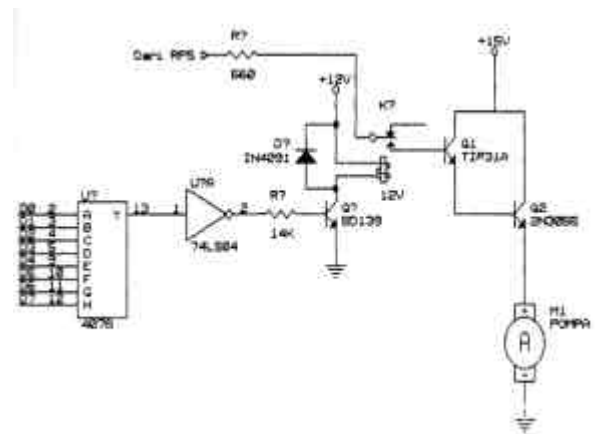
Proses kontrol PID dilakukan oleh PC dengan menggunakan persamaan 6. Output dari kontroler digunakan untuk menggerakkan pompa air

atau *solenoid valve* untuk pembuangan air. Rangkaian *driver solenoid valve* dibentuk dari sebuah *relay* dan transistor BD139. Tegangan basis dari transistor BD139 di-drive dari sebuah inverter IC 74LS04 yang terkoneksi dengan komputer melalui *parallel port* LPT1 Pin C3. Rangkaian lengkap *driver solenoid valve* dapat dilihat pada gambar 6.



Gambar 6. Rangkaian *Driver Solenoid Valve*

Rangkaian *driver pompa* digunakan dua buah transistor NPN yaitu 2N3005 dan TIP31A. Kedua transistor ini dihubungkan membentuk rangkaian *darlington* agar mendapatkan penguatan yang besar, sehingga arus I_b transistor yang digunakan kecil. Sebuah transistor BD139 dan *relay* digunakan untuk mematikan dan menyalakan pompa. Gambar lengkap rangkaian *driver pompa* dapat dilihat pada gambar 7.

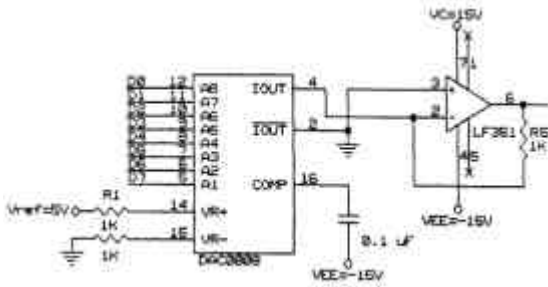


Gambar 7. rangkaian *Driver Pompa*

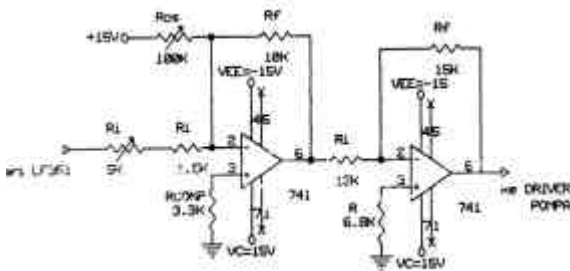
Dari hasil percobaan pompa, diketahui bahwa pompa bekerja (mengeluarkan air lewat pipa PVC) pada tegangan 4 sampai 12 volt. Bila tegangan input pompa kurang dari 4 volt, pompa belum dapat menaikkan air ke atas. Dengan demikian, tegangan input *driver motor* dapat diketahui (dengan menggunakan persamaan 7) yaitu 5,8 sampai 13,8 volt dengan asumsi tegangan jatuh pada R_b sebesar 0,4 volt.

$$V_{in} = V_{Rb} + 2 \cdot V_{BE} + V_{pompa} \quad (7)$$

Rangkaian *driver* pompa mendapat inputan dari output rangkaian *digital to analog converter* (DAC) yang dilewatkan sebuah rangkaian pengkondisi sinyal yaitu rangkaian *zero and span*. Rangkaian DAC dan rangkaian *zero and span* dapat dilihat pada gambar 8 dan 9.



Gambar 8. Rangkaian DAC



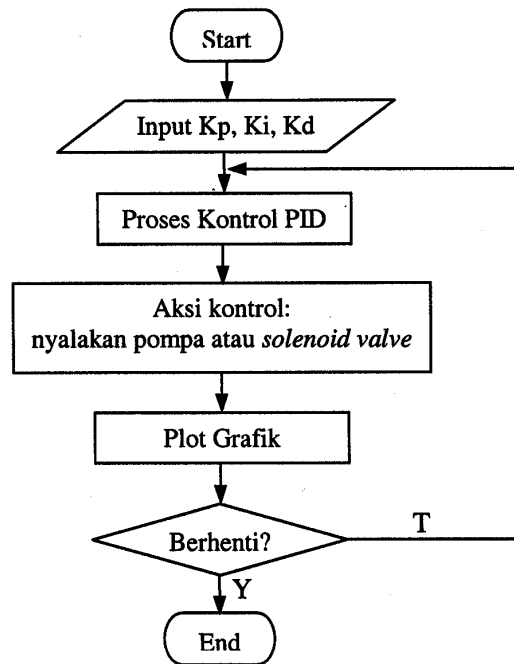
Gambar 9. Rangkaian *Zero and Span*

Dalam aplikasi ini, DAC yang digunakan adalah DAC 0808 yang merupakan DAC 8 bit dengan output dari 0 sampai 5 volt. Karena *range* tegangan input *driver* pompa sebesar 5,8 sampai 13,8 volt maka digunakan rangkaian *zero and span* untuk menyesuaikan tegangan output DAC dengan tegangan input *driver* pompa. Rangkaian DAC mendapat input dari PC melalui paralel port LPT1. Jalur data DAC D0 samapai D7 dihubungkan dengan jalur data paralel port LPT1 D0 samapai D7 pada pin 2 sampai 9. Dengan demikian untuk mengubah kecepatan motor pompa dapat dilakukan dengan mudah yaitu hanya dengan mengubah data input DAC.

3.2 Perangkat Lunak

Pada umumnya program yang dibuat bertugas untuk memonitor ketinggian air, melakukan proses kontrol dan menggerakkan pompa dan *solenoid valve*. Proses ini terus dilakukan sampai ketinggian air dalam tangki sesuai dengan yang diinginkan. Proses perhitungan kontrolerID dilakukan dengan menggunakan persamaan yang

telah dijabarkan terdahulu yaitu persamaan 5 dan 6. Karena DAC yang digunakan adalah DAC 8 bit (*range* data digital dari 0 – 255) maka bila hasil perhitungan (persamaan 6) dari kontroler melebihi 255 maka data DAC di-*set* 255 yang berarti pompa bekerja maksial. Tetapi, bila hasil perhitungan dari kontroler kurang dari 0 maka data DAC di-*set* 0 yang berarti pompa tidak bekerja sama sekali dan *solenoid valve* dibuka. Program telah dirancang dengan menggunakan bahasa pemrograman borland delphi. Berikut gambar 10 menunjukkan diagram alir dari program yang telah dibuat.



Gambar 10. Diagram Alir Program

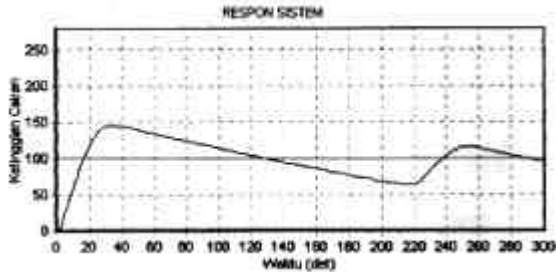
4. Pengujian Sistem

Pengujian sistem pengaturan level ketinggian air dilakukan dengan memberikan input fungsi step. Pengujian juga dilakukan dengan berbagai konstanta Kp, Ki dan Kd serta dua kondisi sistem yaitu sistem tanpa gangguan dan sistem dengan gangguan. *Range* pengukuran ketinggian air adalah 0 – 30 cm yang dikonversikan oleh ADC menjadi data 0 – 255. Dalam pengujian ini data ketinggian air yang digunakan adalah dalam skala yang ditunjukkan oleh ADC 0 – 255.

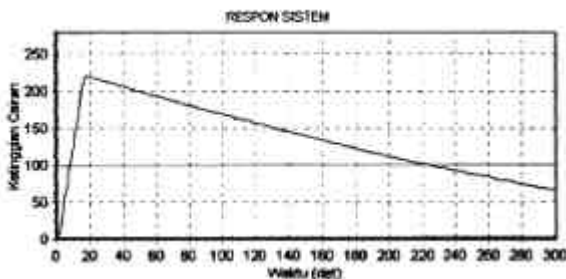
4.1 Pengujian Sistem Tanpa Gangguan

Dalam pengujian ini, dilakukan percobaan dengan nilai salah satu konstanta kontroler diubah sedangkan nilai konstanta kontroler yang lain tetap. Pengujian sistem ini dilakukan dengan

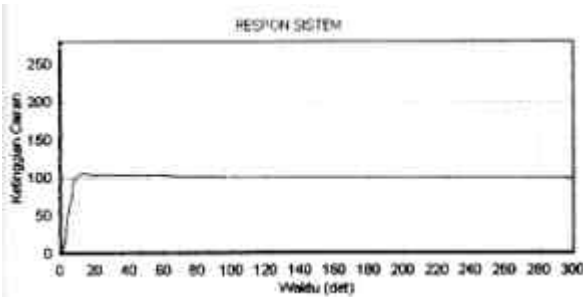
nilai *setting point* ketinggian air sebesar 100 yang menunjukkan ketinggian air 11,76 cm. Gambar 11 sampai gambar 14 menunjukkan respon sistem dari hasil percobaan yang telah dilakukan.



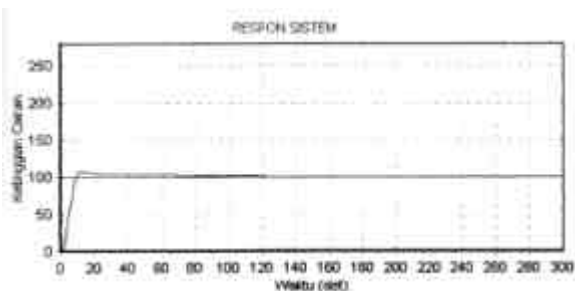
Gambar 11. Respon Sistem Kontroler PID dengan $K_p=1$, $K_i=0,1$ dan $K_d=0,1$



Gambar 12. Respon Sistem Kontroler PID dengan $K_p=1$, $K_i=5$ dan $K_d=0,1$



Gambar 13. Respon Sistem Kontroler PID dengan $K_p=10$, $K_i=0,1$ dan $K_d=5$

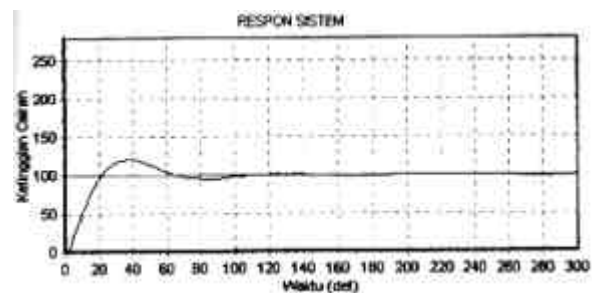


Gambar 14. Respon Sistem Kontroler PID dengan $K_p=10$, $K_i=0,1$ dan $K_d=0,1$

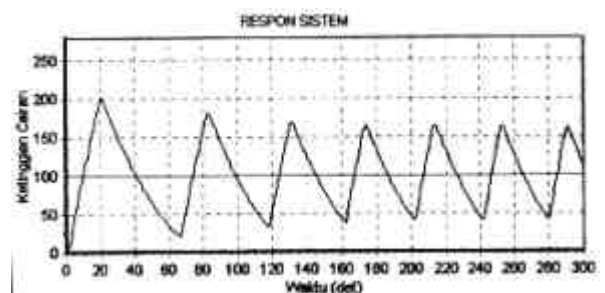
Dari hasil percobaan yang ditunjukkan oleh gambar 11 dan gambar 12 terlihat dengan harga K_i yang besar (K_p dan K_d tetap) akan memperbesar *overshoot* dan *undershoot*. Dan juga dengan K_i yang besar pencapaian keadaan *steady state* jauh lebih lama dan sistem berosilasi. Sedangkan pada gambar 13 dan 14 terlihat dengan harga K_d yang besar (K_p dan K_i tetap) akan memperkecil *overshoot* dan juga mempercepat pencapaian keadaan *steady state*. Dan dari gambar 11 dan gambar 14 untuk harga K_p yang besar (K_i dan K_d tetap) akan memperkecil *overshoot* dan mempercepat pencapaian *steady state*.

4.2 Pengujian Sistem dengan Gangguan

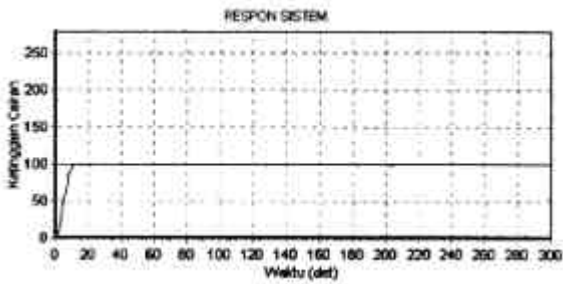
Dalam percobaan ini, parameter pengujian yang digunakan sama dengan pengujian yang dilakukan sebelumnya yaitu tanpa gangguan. Perbedaannya adalah dalam pengujian ini, sistem diberi gangguan berupa pembukaan kran air manual sebanyak dua putaran. Berikut gambar 15 sampai 18 menunjukkan hasil percobaan yang telah dilakukan.



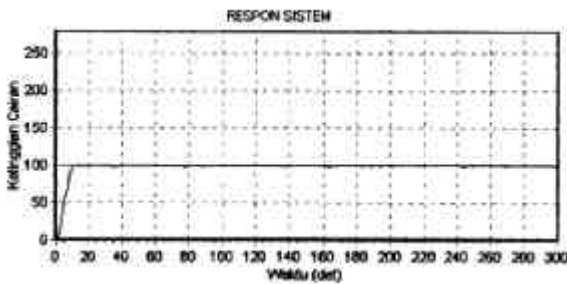
Gambar 15. Respon Sistem Kontroler PID dengan $K_p=1$, $K_i=0,1$ $K_d=0,1$ dan Gangguan 2 Putaran Kran Manual



Gambar 16. Respon Sistem Kontroler PID dengan $K_p=1$, $K_i=5$, $K_d=0,1$ dan Gangguan 2 Putaran Kran Manual



Gambar 17. Respon Sistem Kontroler PID dengan $K_p=10$, $K_i=0,1$ $K_d=5$ dan Gangguan 2 Putaran Kran Manual



Gambar 18. Respon Sistem Kontroler PID dengan $K_p=10$, $K_i=0,1$ $K_d=0,1$ dan 2 Gangguan Putaran Kran Manual

Dari hasil percobaan yang ditunjukkan oleh gambar 15 dan gambar 16 terlihat dengan harga K_i yang besar (K_p dan K_d tetap) akan memperbesar *overshoot* dan *undershoot*. Dan juga dengan K_i yang besar, pencapaian keadaan *steady state* jauh lebih lama dan sistem berosilasi. Sedangkan pada gambar 17 dan gambar 18 terlihat dengan harga K_d yang besar (K_p dan K_i tetap) akan memperkecil *overshoot* dan juga mempercepat pencapaian keadaan *steady state*. Dan dari gambar 15 dan gambar 18 untuk harga K_p yang besar (K_i dan K_d tetap) akan memperkecil *overshoot* dan mempercepat pencapaian *steady state*. Secara keseluruhan, untuk kontroler dengan parameter K_i yang tidak terlalu besar, sistem masih dapat dikontrol walaupun ada gangguan.

5. Kesimpulan

Dari hasil pengujian yang telah dilakukan, dapat diambil kesimpulan bahwa sistem kontrol PID yang dibuat mampu mengontrol level ketinggian air. Secara umum untuk aplikasi yang telah dibuat, semakin besar harga K_p maka sistem lebih cepat mencapai *steady state* dan juga *rise time* semakin kecil. Semakin besar harga K_i akan

membuat respon sistem berosilasi dan lebih lambat mencapai keadaan *steady state*. Semakin besar harga K_d dapat membuat *overshoot* semakin kecil.

Daftar Pustaka

- [1]. Stuart Bennet, *Real Time Computer Control An Introduction*. Hertfordshire: Prentice Hall International (UK) Ltd, 1988.
- [2]. Michael J. Jacob, *Industril Control Electronics*. New Jersey: Prentice Hall, 1988.
- [3]. Christopher T. Killian, *Modern Control Technology Components and Systems*, St. Paul Minneapolis: West Publishing Company, 1996.
- [4]. Dale R. Patrick, Stephen W. Fardo, *Industrial Procees Control Systems*, New York : International Thomson Publishing Company, 1997.
- [5]. Katsuhiko Ogata, *Modern Control Engineering*, New Jersey : Prentice Hall, 1997.