
ALAT PERAGA KENDALI PEMANAS UDARA BERBASIS ARDUINO UNO SEBAGAI PENUNJANG PRAKTIKUM LABORATORIUM KENDALI POLITEKNIK NEGERI SEMARANG

[A DEMONSTRATION TOOL OF ARDUINO BASED AIR HEATER CONTROL TO SUPPORT CONTROL LABORATORY OF POLITEKNIK NEGERI SEMARANG]

Bambang Supriyo^{1*}, Samuel BETA Kuntardjo², dan Sihono³
^{1,3}Program Studi Teknik Elektronika, Politeknik Negeri Semarang,
Jl. Prof. H. Soedarto, S.H. Tembalang Semarang 50275

*Korespondensi : bsupriyo7763@yahoo.com

ABSTRACT

The objective of this research is to develop and test the demonstration tool of Arduino based air heater controller with Proportional Integral Derivative (PID) based control methods for the temperature range between 50°C and 70°C. Arduino Uno was programmed using C-language to do control tasks and to transfer control data serially to computer via USB port. The selection of the initial PID parameters are determined using the combination of Relay Feedback method and Ziegler-Nichols formula. The control methods were focused on proportional (P), proportional derivative (PD) dan Proportional Integral Derivative (PID) with the performance criteria based on overshoot dan steady state error. The experimental results show that the PID controllers give best output responses in terms of zero steady state errors, while P and PD controllers still result in steady state errors of about 3°C. In addition, P controllers still give about 2,5°C-4°C fluctuative output values below set points, while the PD and PID have very small fluctuative values and even very close to zero. This laboratory demonstration tool has shown very good PID based controller performance, so it is feasible to be used as a supporting demonstration tool for control system laboratory in Politeknik Negeri Semarang or even other universities.

Keywords : Arduino UNO, Air Heater, PID, Relay Feedback, Ziegler-Nichols

ABSTRAK

Penelitian ini bertujuan untuk membuat dan menguji kinerja alat peraga pengendali pemanas udara menggunakan Arduino UNO dengan metode kendali berbasis *Proportional Integral Derivative* (PID) untuk rentang temperatur antara 50°C hingga 70°C. Arduino Uno diprogram dengan bahasa C untuk tugas pengendalian dan mentransfer data pengendalian secara serial ke komputer melalui port USB. Pemilihan parameter awal PID menggunakan gabungan antara metode *Relay Feedback* dan formula *Ziegler-Nichols*. Hasil pengendalian difokuskan pada metode: *proportional* (P), *proportional derivative* (PD) dan *Proportional Integral Derivative* (PID) dengan kriteria kinerja pengendalian berdasarkan pada : *overshoot* dan *steady state error*. Hasil pengujian menunjukkan bahwa kendali PID memberikan tanggapan output paling bagus dengan *steady state error* mendekati 0°C, sedangkan kendali P dan PD masih menghasilkan *steady state error* sekitar 3°C. Kendali P juga masih memberikan output fluktuatif sekitar 2,5°C-4°C di bawah nilai *set point*, sedangkan kendali PD dan PID mempunyai output fluktuatif yang

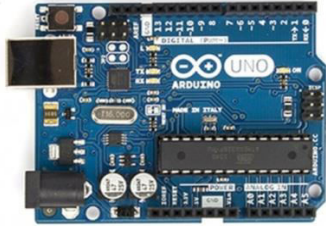
jauh lebih kecil atau bahkan mendekati nol. Alat peraga ini telah menunjukkan kinerja pengendalian berbasis PID yang bagus, sehingga layak untuk digunakan sebagai alat penunjang praktikum laboratorium sistem kendali di Politeknik Negeri Semarang atau bahkan di perguruan tinggi lainnya.

Kata kunci : Arduino UNO, PID, pemanas udara, Relay Feedback, Ziegler-Nichols

PENDAHULUAN

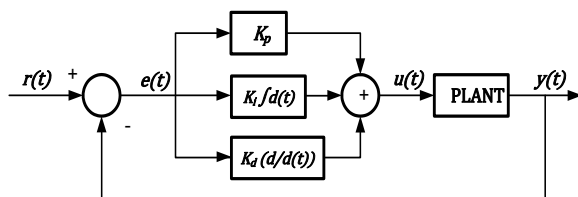
Ketersediaan alat peraga praktikum laboratorium sistem kendali merupakan salah satu faktor utama untuk peningkatan kemampuan ketrampilan mahasiswa dalam hal penguasaan ilmu elektronika kendali. Namun, harga alat peraga yang ada di pasaran umumnya sangat mahal. Untuk itu perlu dibuat sendiri alat peraga yang lebih murah, berkinerja bagus dan layak dipakai sebagai sarana praktik laboratorium, sehingga memudahkan mahasiswa untuk memahami konsep pengendalian, khususnya kendali PID. Sistem pemanas udara merupakan sistem *plant* orde satu yang sederhana, dan banyak dijumpai baik pada aplikasi rumah tangga maupun industri. Dalam skala industri, pemanas udara banyak dijumpai pada aplikasi pemanasan, ventilasi dan penyejuk ruangan, *Heating Ventilation and Air conditioning (HVAC)* (Guo *et al.*, 1998; Wang *et al.*, 2008; Jun, 2011). Salah satu pemanas udara yang sering dipakai untuk demonstrasi praktikum di laboratorium adalah *hair dryer* (Edwards, 2005; Recktenwald *et al.*, 2009).

Seiring dengan berkembangnya teknologi mikrokontroller saat ini, Arduino Uno termasuk modul mikrokontroller dengan harga murah dan mempunyai kinerja pengendalian yang cukup bagus. Arduino Uno berbasis sistem ATmega328, seperti terlihat pada Gambar 1. Arduino Uno mempunyai 14 pin input/output digital yang dapat dikonfigurasi, di mana 6 pin di antaranya dapat dipakai sebagai output *pulse width modulation* (PWM) dan 6 pin lainnya sebagai input analog. Komputer terhubung dengan Arduino Uno melalui port USB untuk proses pengunggahan program dan data transfer (Manoiu-Olaru dan Nitulescu, 2011). Beberapa aplikasi Arduino yang telah dipakai dalam kendali temperatur sebagai alat bantu praktikum maupun simulator skala laboratorium adalah: eksperimen pendidikan jarak jauh untuk bidang sistem kendali (Neto *et al.*, 2013), pengendalian pemanas sistem *thermostat* (Aradi, 2016), simulator pendingin ruangan dan pemanas otomatis (Malipatil, 2016) dan kendali pemanas tanki *boiler* skala laboratorium (Shirke *et al.*, 2017).



Gambar 1. Mikrokontroler Arduino UNO

Saat ini kendali PID masih banyak digunakan di industri walaupun teknik kendali modern telah banyak dikembangkan dengan berbagai algoritma baik teori maupun teknologi kendali. Berdasarkan survey oleh Honeywell (Desborough *et al.*, 2000), terhadap 11.600 pengendalian yang dipakai pada 18 industri, lebih dari 97% dari pengendalian praktis dilakukan dengan menggunakan algoritma kendali PID. Kendali PID mempunyai struktur sederhana, kinerja handal dan menjadi dasar kendali bagi algoritma kendali modern saat ini (Jianghua dan Huihe, 2003). Kendali PID juga mempunyai aturan kendali sederhana dan pengaturan parameter lebih sedikit (Zalm, 2004) serta handal untuk mendapatkan parameter PID yang tepat (Valério dan Costa, 2005). Struktur dasar kendali PID dapat dilihat pada Gambar 2.



Gambar 2. Struktur dasar kendali PID

Kendali PID dalam domain waktu dapat direpresentasikan dalam bentuk berikut ini:

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

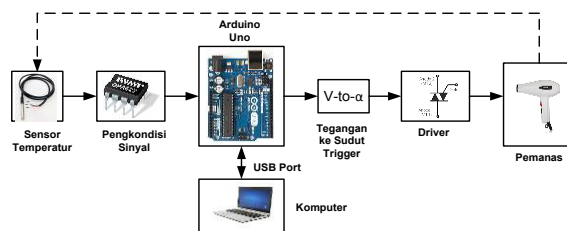
di mana $u(t)$ adalah variable yang dimanipulasi, $e(t)$ nilai kesalahan, K_p penguatan proporsional, K_i penguat integral dan K_d penguat derivatif. Beberapa aplikasi kendali temperatur berbasis PID menggunakan Arduino adalah: inkubator bayi (Theopaga *et al.*, 2014), percobaan laboratorium untuk kendali otomatis dan robotika (Candelas *et al.*, 2015), kendali sistem pemanas (Bayram *et al.*, 2016), inkubator telur burung otomatis (Tchimmoue *et al.*, 2016) dan simulator kendali temperatur (Asraf *et al.*, 2017).

Alat Peraga Kendali Pemanas Udara Berbasis Arduino Uno ini mempunyai rentang temperatur *set point* antara 50°C dan 70°C. Hasil pengendalian ini difokuskan pada metode kendali P, PD dan PID. Pemilihan parameter awal dari PID berdasarkan pada metode *Relay Feedback* dan formula *Ziegler-Nichols* (Åstrom dan Hägglund, 1984). Kriteria kinerja pengendalian menggunakan ketiga metode tersebut akan dibandingkan dan dianalisis berdasarkan pada nilai: *overshoot* dan *steady state error*.

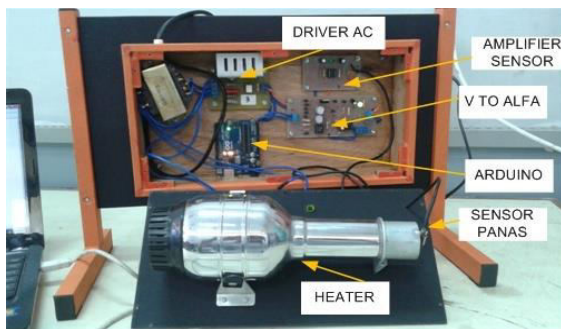
BAHAN DAN METODE

Bahan dan Alat

Diagram blok alat peraga kendali pemanas udara dapat dilihat pada Gambar 3, dan foto alat ditampilkan pada Gambar 4. Bahan yang dipakai pada penelitian ini adalah : sensor temperatur, pengkondisi sinyal, Arduino Uno, pengubah tegangan ke sudut *trigger* dan driver pemanas.



Gambar 3. Diagram blok peraga kendali pemanas udara



Gambar 4. Peraga kendali pemanas udara

Sensor temperatur menggunakan IC LM35 dengan spesifikasi output linear $10\text{mV}/^\circ\text{C}$ dan rentang temperatur hingga 150°C . Sensor ini diletakkan pada ujung pemanas udara dan mendeteksi temperatur

udara yang keluar dari pemanas. Output dari sensor LM35 perlu dikondisikan lebih dulu menggunakan rangkaian pengkondisi sinyal, sebelum dibaca oleh input analog Arduino Uno.

Pengkondisi sinyal berfungsi untuk menguatkan tegangan output dari sensor LM35 agar pada saat temperatur mencapai 100°C , pengkondisi sinyal menghasilkan tegangan output sebesar 5 Volt. Dengan demikian, spesifikasi output pengkondisi sinyal ini adalah $20^\circ\text{C}/\text{V}$. Tegangan output ini diumpankan ke pin input analog (A_0) Arduino Uno untuk pemantauan temperatur udara pemanas. Pengkondisi sinyal ini merupakan rangkaian penguat positif 5 kali yang dibuat menggunakan OpAmp tipe CA3140.

Mikrokontroler Arduino Uno ini difungsikan untuk membaca tegangan sensor yang masuk pada pin input analog (A_0) yang mewakili temperatur nyata dari sistem pemanas, membandingkan dengan nilai temperatur referensi yang diinginkan (*set point*), menghitung nilai kesalahan antara temperatur nyata dengan *set point*, dan mempergunakan nilai kesalahan ini sebagai input pengendalian berdasarkan metode P, PD dan PID. Output hasil pengendalian ini diubah dalam bentuk tegangan PWM pada pin keluaran analog PWM (Pin3). Output

Analog ini difilter menggunakan rangkaian *Low Pass Filter* agar didapatkan tegangan output DC yang selanjutnya diumpankan ke rangkaian pengubah tegangan ke sudut trigger untuk mengendalikan rangkaian pemanas udara.

Rangkaian pengubah tegangan ke sudut trigger berfungsi untuk mengubah tegangan pengendalian (0 – 5 V) yang berasal dari Arduino Uno Pin3 ke sudut trigger (0°–80°) yang selanjutnya diumpankan ke Driver Pemanas untuk mengendalikan level panas pada pemanas listrik. Driver Pemanas berfungsi untuk mengaktifkan rangkaian pemanas listrik sesuai dengan sudut trigger yang diberikan oleh output rangkaian pengubah tegangan ke sudut trigger. Pemanas akan off penuh bila sudut trigger yang diberikan adalah 180°, dan on penuh bila sudut trigger yang diberikan adalah 0°. Pemanasan sebagian akan terjadi bila nilai sudut trigger antara 0°-180°.

Alat yang dipakai pada penelitian ini meliputi komputer dan osiloskop digital. Komputer berfungsi sebagai pengisi program untuk pengendalian Arduino Uno melalui Editor Arduino. Selain itu komputer juga berfungsi sebagai penerima data parameter pengendalian selama percobaan dan memroses data tersebut menggunakan

perangkat lunak *Excel* untuk ditampilkan dalam bentuk tabel maupun grafik. Sedangkan osiloskop digital dipakai untuk melihat hasil pengesetan sudut trigger apakah sudah sesuai dengan input tegangannya (0 – 5V).

Metode Penelitian

Metode penelitian ini mengacu pada urutan kerja seperti yang ditampilkan pada Gambar 5. Adapun urutan kerjanya dapat diterangkan sebagai berikut:

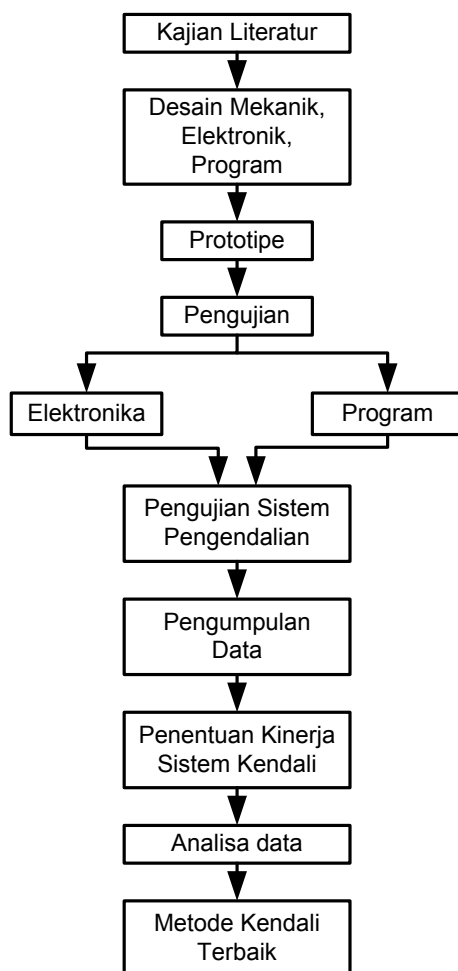
- a) Tahap awal mengadakan kajian literatur yang berhubungan dengan pengendalian pemanas udara, terutamanya yang menggunakan metode PID.
- b) Pembuatan desain mekanik, elektronik dan program Arduino.
- c) Pembuatan prototipe alat peraga kendali pemanas udara baik secara perangkat keras maupun perangkat lunak. Sensor temperatur yang dibuat perlu dikalibrasi dengan alat ukur standar untuk rentang temperatur antara 30° - 90°C.
- d) Pengujian rangkaian elektronika yang terdiri dari sensor temperatur, pengkondisi sinyal, pengubah tegangan ke sudut trigger dan driver pemanas.
- e) Pengujian perangkat lunak melibatkan perangkat keras mikrokontroler Arduino Uno, serta perangkat lunak Editor

Arduino yang diinstalasikan di komputer untuk memprogram Arduino Uno agar dapat melaksanakan tugas pengendalian metode P, PD maupun PID. Arduino Uno juga harus mampu mentransfer data parameter hasil pengendalian ke komputer, yang nantinya dapat diproses menggunakan perangkat lunak Excel untuk dibuat dalam bentuk tabel maupun grafik.

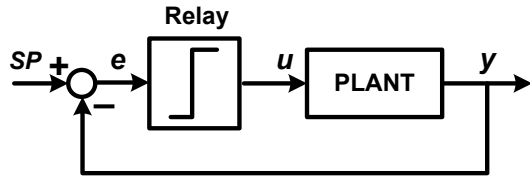
- f) Pengujian sistem keseluruhan untuk mengerjakan tugas pengendalian pemanas udara berdasarkan metode P, PD dan PID.
- g) Pengumpulan data-data hasil pengujian dan penentuan level kinerja dari masing-masing metode pengendalian (P, PD, PID) berdasarkan kriteria: *overshoot* dan *steady state error*.
- h) Berdasarkan pada data level kinerja dari masing-masing metode pengendalian, maka data-data tersebut dibandingkan dan dianalisis agar didapatkan metode pengendalian yang menghasilkan respon output yang paling baik.

Penentuan Parameter Awal PID

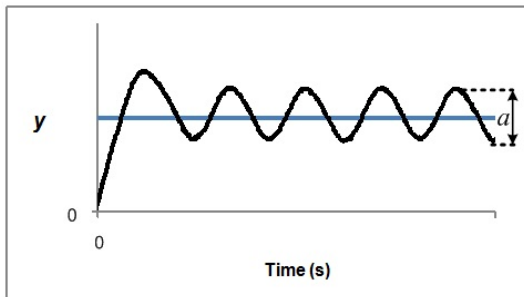
Metode *Relay Feedback* secara eksperimen telah sering digunakan dalam penalaan (*tuning*) kendali PID secara otomatis. Struktur *Relay Feedback* dapat dilihat pada Gambar 6. Metode ini digagas oleh Åström and Hägglund (1984) untuk menghasilkan output (y) yang berosilasi kontinu dengan amplitudo gelombang terbatas di mana estimasi nilai kritis tentang perioda osilasi (T_c), amplitudo osilasi (a) dan ketinggian level output relay (h) dapat diekstrak dari gelombang osilasi output, seperti terlihat pada Gambar 7 dan output relay (u), seperti terlihat pada Gambar 8.



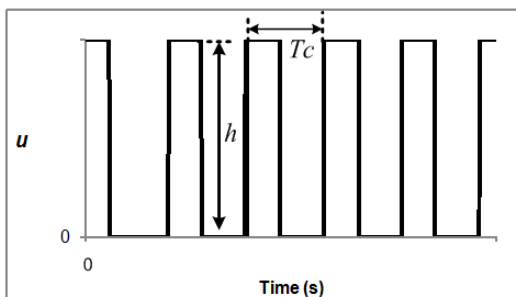
Gambar 5. Metode Penelitian



Gambar 6. Relay Feedback



Gambar 7. Output plant osilasi



Gambar 8. Output relay

Parameter penguatan kritis (K_c) dari kendali PID dapat dihitung dengan menggunakan Persamaan (2). Sedangkan nilai parameter penguatan proporsional (K_p), penguatan integral (K_i) dan penguatan derivatif (K_d) dapat ditentukan dengan menggunakan nilai perioda kritis, nilai penguatan kritis dan dengan menggunakan Persamaan (3), (4) dan formula Ziegler-Nichols yang diberikan pada Tabel 1.

$$K_c = 4h/\pi a \quad (2)$$

$$K_i = K_p/T_i \quad (3)$$

$$K_d = K_p T_d \quad (4)$$

di mana T_i waktu integral dan T_d waktu derivatif.

Tabel 1. Formula Ziegler-Nichols untuk overshoot rendah

Kendali	K_p	T_i	T_d
PID	$0,33K_c$	$0,5T_c$	$0,33T_c$

Algoritma Pemrograman

Algoritma pemrograman kendali PID menggunakan bahasa C pada Editor Arduino dapat dilakukan dengan urutan langkah sebagai berikut:

- a. Inisialisasi beberapa variabel ke nilai nol; yaitu variabel waktu (t), Error awal (E0), akumulasi Error (Ea).
- b. Tentukan nilai Temperatur *set point* (Tsp). Contoh Tsp =50; Nilai $K_p=41.23$, $K_i=6.44$ dan $K_d=65.97$
- c. Buat program interrupt timer 1 detik. (Pewaktuan untuk proses dibuat tiap 1 detik).
- d. Bila terjadi interrupt maka:
 1. Tambahkan waktu (t) 1 detik;
 $// t = t + 1;$

-
- | | |
|--|---|
| 2. Baca tegangan sensor (Vs) yang masuk ke pin Analog A0;
//Vs = analogRead(A0) *
(5.0/1023.0); | //If (Okpid > 255) { Okpid =
255}
Else if (Okpid < 0) { Okpid = 0}; |
| 3. Ubah tegangan (Vs) ke nilai temperatur nyata (Tn);
// Tn = 20*Vs; | 12. Tentukan nilai output PWM;
// Opwm = 255-Okpid; (Opwm
= 0 Full ON; Opwm = 255 Full
OFF) |
| 4. Hitung Error antara Temperatur <i>set point</i> (Tsp) dengan Temperatur nyata (Tn);
//E = Tsp – Tn; | 13. Outputkan Nilai Opwm ke Pin 3 (Analog Output);
// analogWrite(3, Opwm); |
| 5. Hitung delta Error dan Simpan nilai Error baru ke Error yang lama (proses derivatif).
//dE = E – E0; E0 = E; | 14. Kirim dan tampilkan data (t, Tsp, Tn) ke serial monitor pada Editor Arduino. |
| 6. Hitung akumulasi Error (Ea) untuk proses integral.
//Ea = Ea + 1 | 15. Jika waktu proses telah selesai, maka salin dan simpan data tersebut pada program Excel yang nantinya data tersebut dapat diproses dalam bentuk grafik. |
| 7. Hitung Output Kendali Proporsional (Okp);
//Okp = Kp*E; | |
| 8. Hitung Output Kendali Integral (Oki);
//Oki = Ki*Ea; | |
| 9. Hitung Output Kendali Derivatif (Okd);
// Okd = Kd*dE; | |
| 10. Hitung Output Kendali PID (Okpid);
//Okpid = Okp + Oki + Okd; | |
| 11. Batasi nilai Okpid antara 0 – 255; | |

Percobaan 1 : Parameter Awal PID

Percobaan ini dilakukan untuk mendapatkan nilai awal dari parameter PID menggunakan metode *Relay Feedback*, di mana bagian pengendali sistem ini bekerja seperti *relay* (on / off). Bila temperatur *set point* (Tsp) lebih besar dari temperatur nyata (Tn), maka output relay memberikan nilai PWM maksimum, yaitu 255, sehingga pemanas akan on penuh untuk mempercepat pencapaian temperatur yang diinginkan. Tetapi bila temperatur *set point* (Tsp) lebih

kecil atau sama dengan temperatur nyata (T_n), maka temperatur nyata telah mencapai temperatur *set point*, output relay memberikan output PWM minimum yaitu 0, sehingga pemanas akan off penuh untuk mendinginkan pemanas udara. Temperatur *set point* yang digunakan adalah 50°C, 60°C dan 70°C.

Percobaan dilakukan dengan memberikan perubahan temperatur *set point* dari 30°C ke nilai temperatur *set point* yang dikehendaki (50°C, 60°C atau 70°C). Tiap percobaan untuk setiap *set point* akan menghasilkan 2 buah kurva, yaitu kurva tanggapan output temperatur nyata hasil pengendalian dan kurva output PWM dari pengendali relay on/off yang dipakai untuk mengendalikan pemanas udara. Dari kedua kurva tersebut, nilai parameter awal dari PID dapat ditentukan dengan menggunakan Persamaan (2), (3), (4) dan formula Ziegler-Nichols pada Tabel 1.

Percobaan 2 : Kendali P, PD dan PID dengan Set Point = 50°C

Percobaan ini dilakukan untuk mengetahui kinerja kendali P, PD dan PID untuk temperatur *set point* (SP) 50°C, dengan menggunakan parameter K_p , K_d dan K_i sesuai dengan yang dihasilkan dari

percobaan nilai awal PID saat temperatur *set point* (SP) 50°C.

Percobaan 3 : Kendali P, PD dan PID dengan Set Point = 60°C

Percobaan ini dilakukan untuk mengetahui kinerja kendali P, PD dan PID untuk temperatur *set point* (SP) 60°C, dengan menggunakan parameter K_p , K_d dan K_i sesuai dengan yang dihasilkan dari percobaan nilai awal PID saat temperatur *set point* (SP) 60°C

Percobaan 4 : Kendali P, PD dan PID dengan Set Point = 70°C

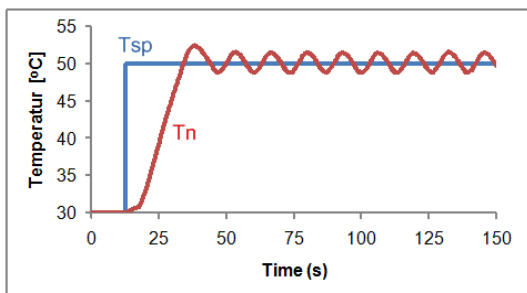
Percobaan ini dilakukan untuk mengetahui kinerja kendali P, PD dan PID untuk temperatur *set point* (SP) 70°C, dengan menggunakan parameter K_p , K_d dan K_i sesuai dengan yang dihasilkan dari percobaan nilai awal PID temperatur *set point* (SP) 70°C.

HASIL DAN PEMBAHASAN

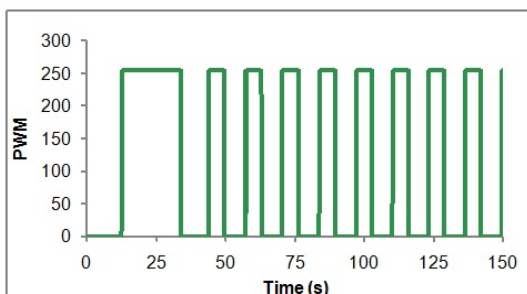
Parameter Awal PID

Hasil percobaan relay feedback untuk temperatur *set point* 50°C, 60°C dan 70°C dapat dilihat pada Gambar 9 - 14. Parameter PID yang didapatkan berdasarkan

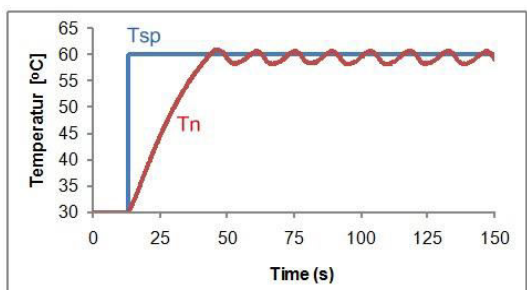
pada metode kombinasi Relay Feedback dan formula Ziegler-Nichols dapat dilihat pada Tabel 2.



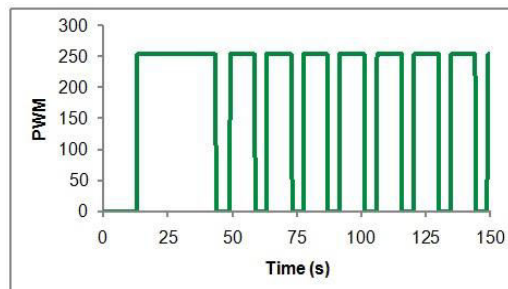
Gambar 9. Output temperatur nyata (Tsp=50°C)



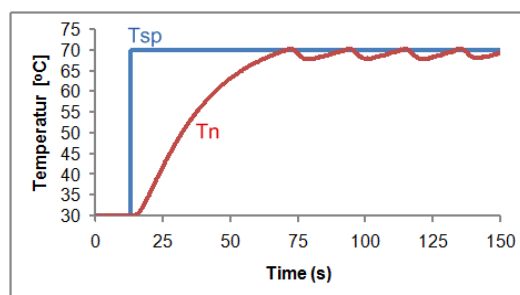
Gambar 10. Output relay (Tsp=50°C)



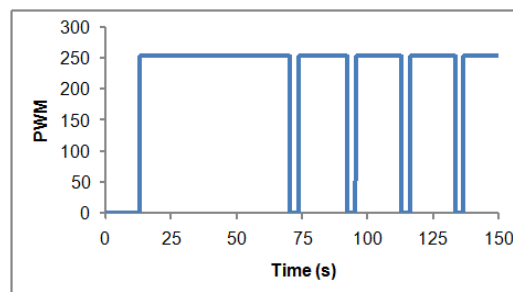
Gambar 11. Output temperatur nyata (Tsp=60°C)



Gambar 12. Output relay (Tsp=60°C)



Gambar 13. Output temperatur nyata (Tsp=70°C)



Gambar 14. Output relay (Tsp=70°C)

Tabel 2. Parameter PID

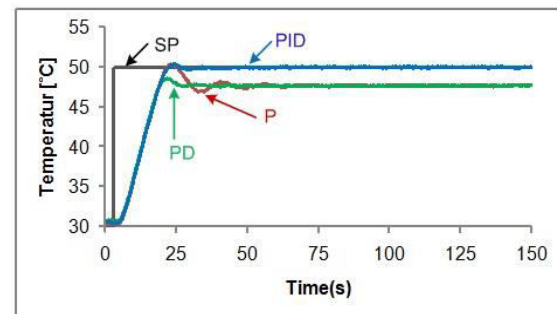
SP(°C)	K_p	K_d	K_i
50	41,23	65,97	6,44
60	39,70	71,46	5,51
70	44,67	113,90	4,38

Dari Gambar 9, 11 dan 13 dapat dilihat bahwa untuk semua temperatur *set point*, besarnya amplitudo output osilasi cenderung hampir sama yaitu sekitar $2,5^{\circ}\text{C}$ sehingga nilai penguatan proporsional (K_p) cenderung hampir sama, yaitu antara 41 hingga 45. Dari Gambar 10, 12 dan 14 dapat dilihat bahwa besarnya waktu perioda kritis (T_c) semakin naik seiring dengan naiknya nilai temperatur *set point* dari 50°C ke 70°C . Dengan adanya kenaikan nilai waktu perioda kritis, maka nilai penguatan derivatif (K_d) juga menaik secara signifikan, yaitu dari sekitar 66 (50°C) ke 114 (70°C), sedangkan nilai penguatan integral (K_i) mengalami penurunan yang sedikit, sekitar 1, yaitu dari 6,4 (50°C), 5,5 (60°C) dan 4,3 (70°C).

Kendali Temperatur Set Point 50°C

Tanggapan sistem kendali P, PD dan PID pemanas udara dengan *set point* 50°C dapat dilihat pada Gambar 15. Kendali P mampu mengendalikan output pemanas udara untuk mencapai *set point* walau hanya sesaat, karena setelah itu output turun secara perlahan menuju ke temperatur 48°C dan tetap di posisi tersebut. Kendali PD tidak mampu mengendalikan output pemanas udara untuk mencapai *set point*, karena hanya mampu mencapai temperatur

maksimum 49°C , kemudian turun perlahan dan bertahan pada posisi konstan sekitar 48°C . Kendali PID mampu mengendalikan output pemanas udara mencapai *set point* dan bertahan pada posisi tersebut, sehingga *zero steady state error* dapat dicapai.

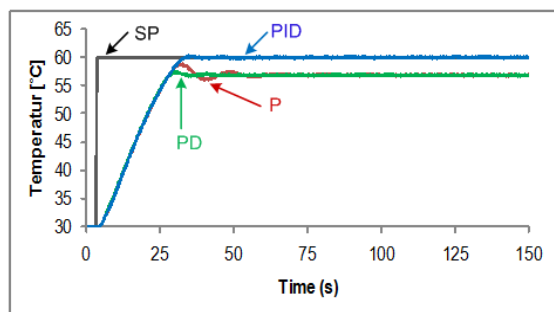


Gambar 15. Kendali pemanas dengan *set point* 50°C

Kendali Temperatur Set Point 60°C

Tanggapan sistem kendali P, PD dan PID pemanas udara dengan *set point* 60°C dapat dilihat pada Gambar 16. Kendali P tidak mampu mengendalikan output pemanas udara untuk mencapai *set point*, karena temperatur tertinggi yang dicapai adalah sekitar 59°C . Setelah itu output turun secara perlahan menuju ke temperatur konstan 57°C . Kendali PD tidak mampu mengendalikan output pemanas udara untuk mencapai *set point*, karena output yang dikendalikannya hanya mampu mencapai temperatur maksimum $57,5^{\circ}\text{C}$, kemudian turun secara perlahan dan bertahan pada posisi temperatur konstan sekitar 57°C .

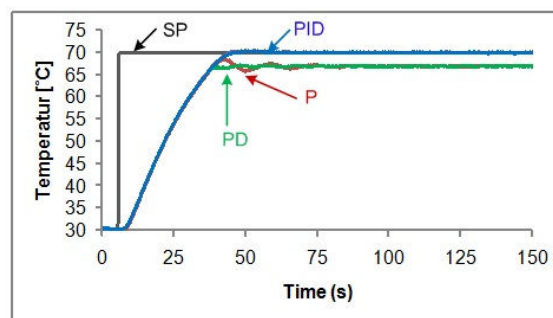
Kendali PID mampu mengendalikan output pemanas udara untuk mencapai *set point* 60°C dan bertahan pada posisi *set point* tersebut.



Gambar 16. Kendali pemanas dengan *set point* 60°C

Kendali Temperatur Set Point 70°C

Tanggapan sistem kendali P, PD dan PID pemanas udara dengan *set point* 70°C dapat dilihat pada Gambar 17. Kendali P tidak mampu mengendalikan output pemanas udara untuk mencapai *set point*, karena hanya mampu mencapai temperatur tertinggi sekitar 68,5°C. Setelah itu output turun secara perlahan menuju ke temperatur konstan 67°C. Kendali PD tidak mampu mengendalikan output pemanas udara untuk mencapai *set point*, karena output yang dikendalikannya hanya mampu mencapai temperatur maksimum 67 °C, dan seterusnya bertahan pada posisi tersebut. Kendali PID mampu mengendalikan output pemanas udara untuk mencapai *set point* 70°C dan bertahan pada posisi *set point* tersebut.



Gambar 17. Kendali pemanas dengan *set point* 70°C

KESIMPULAN

Penelitian Alat Peraga Kendali Pemanas Udara Berbasis Arduino Uno Sebagai Penunjang Praktikum Laboratorium Kendali Politeknik Negeri Semarang telah dilaksanakan dengan hasil yang memuaskan. Hasil yang didapat menunjukkan bahwa kendali PID memberikan tanggapan output yang paling bagus dalam hal pencapaian steady error mendekati 0°C untuk semua *set point* (50°C, 60°C dan 70°C). Kendali P masih memberikan output fluktuatif kecil di bawah *set point*, sedangkan kendali PD dan PID mempunyai output fluktuatif yang jauh lebih kecil atau bahkan mendekati nol. Selain itu, kendali P dan PD masih menyisakan steady state error sekitar 3°C. Di dalam percobaan ini tidak ditemukan adanya *overshoot* yang berarti. Hal ini menunjukkan bahwa pemilihan parameter PID berdasarkan metode *Relay Feedback* dan formula Ziegler-Nichols untuk *overshoot* yang rendah dapat diterima dengan baik

dalam aplikasi kendali pemanas udara berskala laboratorium ini.

DAFTAR PUSTAKA

- Aradi, P. 2016. Offline and online thermostat experiment with LabVIEW and Arduino. International Symposium on Small-scale Intelligent Manufacturing Systems (SIMS), p. 1-6. Narvik, Norway.
- Asraf, H. M., Dalila, K.A. N., Hakim, A.W. M. and Faizzuan, R.H.M. 2017. Development of experimental simulator via Arduino-based PID temperature control system using LabVIEW. Journal of Telecommunication, Electronic and Computer Engineering 9 (1): 53-57.
- Åström, K. J. and Hägglund, T. 1984. Automatic tuning of simple regulators with specifications on phase and amplitude margins. Automata 20 (5) : 645-651.
- Bayram, A., Moammed, S. A., and Kara, F. 2016. Design of Heating system controlled by Arduino. Proceeding of the 4th International Symposium on Innovative Technologies in Engineering and Science (ISITES 2016), p. 1583-1591. Alanya/Antalya ,Turkey.
- Candelas, F. A., García, G. J., Puente, S., Pomares, J., Jara, C. A., Pérez, J. and Torres, F. 2015. Experiences on using Arduino for laboratory experiments of Automatic Control and Robotics. IFAC-Papers On Line 48 (29) : 105-110.
- Desborough, L., Miller, R. and Nordh, P. 2000. Regulatory Control Survey. Unpublished Manuscript, Honeywell.
- Edwards, R. C. 2005. A simple hairdryer experiment to demonstrate the first law of thermodynamics. Proceedings of the 2005 American Society for Engineering Education Annual Conference & Exposition, Session #1348, p. 1-9. Portland, Oregon, USA.
- Guo, P.Y., Guang, Z.H. dan Bien, Z. 1998. A simple fuzzy adaptive control method and application in HVAC. The 1998 IEEE International Conference on Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence, p. 1-7. Anchorage, Alaska,USA.
- Jianghua, X., and Huihe, S. 2003. A novel method of PID tuning for integrating processes. Proceedings 42nd IEEE Conference on Decision and Control, p. 139-142.
- Jun, Z. 2011. A particle swarm optimization approach for optimal design of PID controller for temperature control in HVAC. 2011 Third International Conference on Measuring Technology and Mechatronics Automation, p.230-233. Shanghai, China.
- Malipatil, S.S. 2016. Simulation of automatic air conditioning and heating control with RF Arduino-melody by using LM35 temperature sensor in Proteus software. International Journal of Emerging Research in Management and Technology 5 (5) :1-6.
- Manoiu-Olaru, S., and Nitulescu, M. 2011. Hexapod Robot. Virtual Models for Preliminary Studies. 15th International Conference on System Theory, Control and Computing (ICSTCC), p.1-6.

- Neto, J. M., Silva, C.S., Spacek, A. D., and Junior, O. H. A. 2013. Remote educational experiment applied to the discipline of control systems. ICBL2013 – International Conference on Interactive Computer aided Blended Learning, p. 369-374.
- Recktenwald, G., Edwards, R., Howe, D., Faulkner, J., and Hsieh, C. 2009. The Engineering of Everyday Things : Simple experiments for the thermal and fluid sciences. Paper AC 2009-704, ASEE Annual Conference and Exposition, p. 1-8. Austin, Texas, USA.
- Shirke, P.A., Thorkar, U., Kshirsagar, K.B., Pusdekar, S.G. and Welukar, R.J. 2017. Monitoring and controlling of different process parameters by using LabVIEW and Arduino. International Journal of Advance Research, Ideas and Innovation in Technology 3 (3) : 1617-1622.
- Tchimmoue, G.E. , Kamdem, J. , Sone, M.E. and Tchappa, C.T. 2016. Development of a cost-effective ARDUINO based automatic BIRD-EGG incubator. Sciences, Technologies et Développement, Edition spéciale 17 (1) : 54-58.
- Theopaga, A.K., Rizal, A., and Susanto, E. 2014. Design and implementation of PID control based Baby Incubator. Journal of Theoretical and Applied Information Technology 70 (1) : 19-24.
- Valério, D. and Costa, J.S.D. 2005. Ziegler-Nichols type tuning rules for fractional PID controllers. Proceedings of ASME 2005 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, p. 1-6. Long Beach, USA.
- Wang, J., Zhang, C. and Jing, Y. 2008. Application of an intelligent PID control in heating ventilating and airconditioning system. The 7th World Congress on Intelligent Control and Automation, 2008 (WCICA 2008), p. 1-5. Chongqing China.
- Zalm, G. M. V. D. 2004. Tuning of PID-type controllers: Literature overview. DCT-report.