

Rancang Bangun Kontrol Temperatur Minyak Goreng Berbasis Arduino Dengan Metode PID kontrol

Iwandoyo^a, Balok Hariadi^b, Lutfi Agung Swarga^c, Ahmad Ridhoⁱ^d

^{a,b,c,d}Department of Electrical Engineering, Universitas 17 Agustus 1945 Surabaya, Indonesia

ARTICLE INFO

Article history:

Received 2 April 2024
Received in revised form
10 May 2024
Accepted 20 May 2024
Available online 31 June 2024

Keywords:

Arduino
Max6675
PID kontrol
Thermocouple Type K
Servo

ABSTRACT

Arduino merupakan modul yang bersifat terbuka (open source). Perangkat ini bisa dikembangkan dengan bermacam-macam sensor dan aktuator. Contoh pengembangan yang dilakukan oleh penulis yaitu membuat kontrol temperatur minyak goreng berbasis Arduino dengan menggunakan metode PID kontrol. Penelitian ini menginvestigasi penggunaan PID (Proportional-Integral-Derivative) controller untuk mengatur suhu dalam sistem kontrol Arduino Uno dan modul MAX6675 Thermocouple. Penelitian ini menggunakan kompor gas 1 tungku yang di desain pada bagian knob untuk di modifikasi agar dapat dikendalikan oleh servo yang berputar oleh perintah PID. Thermocouple type K akan mengirimkan data suhu, sistem akan menghitung error, proporsional, integral, dan derivative untuk menghasilkan respon closed loop. Tuning pada penelitian ini menggunakan metode Ziegler-Nichols open loop dan menghasilkan parameter $K_P = 3$, $K_I = 118$, $K_D = 4$. Dan didapatkan hasil respon sistem terhadap pengaturan temperatur dengan nilai overshoot 5,5174, Error steady state 2% dengan waktu Settling time sebesar 576 detik, dan waktu rise time sebesar 105 detik, dan waktu delay time sebesar 23 detik.

1 Pendahuluan

Banyaknya industri rumahan yang sangat pesat membuat dorongan yang besar dalam implementasi teknologi yang baru untuk menunjang produktivitas dan menambah nilai efisiensi dalam melakukan pekerjaan. Saat ini UMKM khususnya pengolah keripik masih banyak yang menggunakan cara lama dalam melakukan produksi yang mana masih bergantung pada alat seadanya dan tenaga manusia. Kelemahan dari model kerja ini yaitu tidak dapat diketahui berapa nilai temperatur penggorengan yang digunakan. Semua bergantung pada kebiasaan dan skill yang dimiliki oleh manusia, adapun kekurangan yang lain yaitu, borosnya penggunaan bahan bakar gas dan hitamnya minyak apabila seorang pekerja tidak memiliki kemampuan yang baik dalam melakukan pekerjaan tersebut [1].

Di Indonesia UMKM mampu menyerap tenaga kerja hingga 97%, dan menyumbang Produk domestik bruto (PDB) hingga 61%. Ini menandakan bahwa UMKM memiliki peran yang besar dalam membantu perekonomian nasional. Produk-produk yang dihasilkan sudah banyak yang di ekspor ke luar negeri dan memiliki permintaan jumlah yang sangat besar. Ini mendorong UMKM untuk dapat bersaing di era digital saat ini, implementasi peralatan yang memadai dan efisien juga dapat membantu para pelaku usaha mikro menengah untuk memenuhi permintaan pasar global. Kualitas produk yang baik merupakan kunci dalam penjualan produk dalam pasar global. Permintaan yang banyak akan menuntut UMKM untuk melakukan produksi secara cepat dan tepat. [2]

Saat ini era digitalisasi sudah sangat maju. Implementasi mikrokontrol dapat digunakan pada banyak model peralatan. Termasuk dalam UMKM, mikrokontrol dapat digunakan untuk membuat alat yang memudahkan pelaku UMKM dalam melakukan produksi. Biaya yang murah dan peralatan yang dijual mudah untuk didapatkan. Model dan pengaturan dalam implementasi dapat dibuat sedemikian rupa untuk memenuhi kebutuhan UMKM. Seperti halnya pengontrolan temperatur minyak dapat dilakukan menggunakan alat yang dimodel untuk

mengendalikan putaran knob kompor. Besar kecilnya api dapat di kendalikan oleh mikrokontrol untuk menghindari temperatur yang terlalu tinggi, dan menjaga kestabilan suhu saat kompor digunakan.

Kompor rumahan digunakan pada penelitian ini sebagai alat pengontrol temperatur minyak, dimana difungsikan untuk menjaga kestabilan suhu. Knob kompor yang biasanya diputar manual oleh manusia, akan di modifikasi atau diubah agar bisa di putar oleh motor servo. Terdapat sensor Thermocouple max6675 untuk mendeteksi suhu. Arduino digunakan untuk memproses data dan memberikan nilai pada output. Keypad 4x4 akan difungsikan untuk memasukan data, dan LCD 16x2 digunakan untuk menampilkan informasi.

2 Studi Literatur

2.1 PID Sistem

Dalam dunia teknik kontrol, sistem kontrol otomatis diperlukan untuk mengatur dan mempertahankan variabel-variabel tertentu dalam suatu sistem. Pada banyak kasus, ini melibatkan pengaturan seperti suhu, kecepatan, atau posisi ke nilai yang diinginkan. Salah satu pendekatan digunakan adalah dengan metode PID (Proporsional, Integral, Derivatif). Kontrol PID merupakan metode kontrol yang menggabungkan tiga elemen utama: Proporsional (P), Integral (I), Derivatif (D). Dalam metode ini, nilai kesalahan antara nilai yang diinginkan (SetPoint) dan nilai yang di ukur (ValuePoint) digunakan untuk menghitung tiga komponen PID yang kemudian disatukan untuk menghasilkan sinyal kontrol. Pendekatan ini memungkinkan sistem untuk merespons secara cepat terhadap perubahan, mengurangi overshoot, dan meminimalkan kesalahan jangka panjang.

Dalam pengontrolan PID, pengaturan proporsional (P) adalah salah satu komponen utama yang menentukan seberapa besar sinyal kontrol yang diberikan sebagai respon antara nilai yang diinginkan (setpoint) dan nilai yang diukur (proses variabel). Komponen ini bekerja dengan cara menghitung besar perbedaan antara set point dan value point. Komponen proporsional merespons secara langsung terhadap kesalahan saat ini (error).

Semakin besar kesalahan, semakin besar sinyal kontrol yang dihasilkan.

Rumus: Output dari komponen proporsional $P(t)$ pada waktu (t) dihitung dengan rumus :

$$P(t) = K_p x e(t) \quad (1)$$

Komponen integral (I) dalam PID memberikan respons terhadap akumulasi kesalahan seiring Waktu. Ini memungkinkan PID untuk menanggapi kesalahan jangka panjang yang mungkin tidak diatasi oleh komponen proporsional saja.

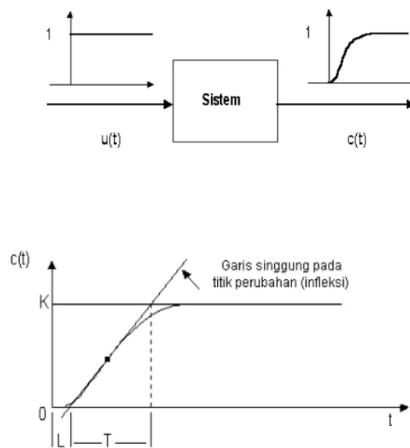
$$I(t) = K_i x \int_0^t e(\tau) d\tau \quad (2)$$

Komponen Derivatif / diferensial (D) dalam kontrol PID memberikan respons terhadap laju perubahan kesalahan. Dengan mengukur seberapa cepat kesalahan berubah seiring waktu, komponen derivatif membantu mengurangi overshoot dan meredam respon sistem terhadap perubahan cepat.

$$D(\dot{t}) = K_d x \frac{de(t)}{dt} \quad (3)$$

Pada perancangan sistem kendali PID diperlukan untuk mengatur parameter Proporsional, integral, dan derivatif agar suatu tanggapan atau respon sistem terhadap masukan tertentu sebagaimana yang diharapkan[3]

Tuning konstanta proporsional, integral, dan derivatif bermacam-macam, umumnya menggunakan metode ziegler-nicols tipe 2. Penelitian ini menggunakan metode ziegler nicols untuk tuning parameternya. Tujuannya adalah untuk mencapai respons yang cepat dan stabil dari sistem kontrol dengan mengatur parameter PID yang sesuai [4]



Gambar 1. Kurva Penalaan PID Ziegler-Nicols

Tabel 1 Parameter Ziegler-Nicols

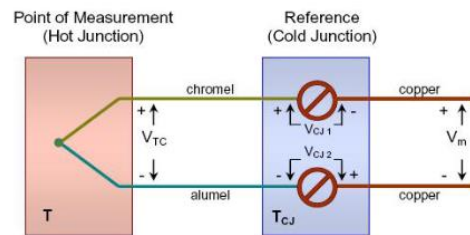
Tipe pengendali	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0,9 \frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2 \frac{T}{L}$	$2L$	$0,5L$

2.2 Arduino Uno

Arduino merupakan modul mikrokontroler. Modul ini memiliki banyak tipe dan model, salah satu yang digunakan pada penelitian ini adalah arduino uno yang menggunakan IC Atmega 325. Menggunakan input external 6-20 Volt DC, tegangan operasi 5v, memiliki pin digital sebanyak 14 buah (6 diantaranya PWM) dengan arus maksimal 40mA. Pada penelitian ini menggunakan pin digital arduino.

2.3 Thermocouple TypeK Max6675

Thermocouple bekerja dengan cara bertemunya 2 logam dengan jenis yang berbeda dan bertemu di satu titik yang menghasilkan tegangan dalam satuan milivolt [5]. Pengukuran dilakukan pada ujung-ujung logam, menghasilkan "efek seebeck", Tegangan yang diukur diujung logamberbanding lurus dengan tinggi suhu yang dirasakan di junction, lebih tinggi nilai mV yang dapat diukur di ujung lain nya. [6]



Gambar 2 Prinsip Kerja Thermocouple

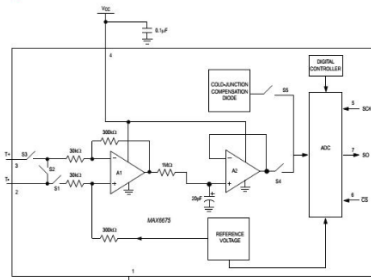
$$V_m = V_{TC} - (V_{CJ1} - V_{CJ2}) = V_{TC} - V_{CJ} \quad (4)$$

Thermocouple TypeK bekerja berdasarkan perubahan tegangan yang sangat kecil dengan skala milivolt.[7] Tegangan ini keluar berdasarkan efek seebeck dimana tegangan akan terus naik berbanding lurus dengan suhu. Karena tegangan yang keluar sangat kecil maka dibutuhkan ic max6675 sebagai penguat arus dan berfungsi sebagai konverter analog to digital [8]. Dimana data yang keluar dan masuk ke mikrokontrol berupa nilai adc sebesar 12 bit. Modul MAX6675 mengubah sinyal analog menjadi sinyal digital (ADC), nilai yang ditampilkan di arduino merupakan nilai digital yang nantinya akan di konversi menjadi satuan suhu. Konversi nilai digital menjadi suhu dengan menggunakan rumus :

$$X = \frac{\text{Nilai ADC}}{Z}$$

Dimana :

- X = Suhu Yang di Inginkan
- Nilai ADC = Read ADC pada mikrokontroler
- Z = Pembagi atau Nilai yang akan di cari



Gambar 3. Skematik Max6675

Tabel 2 Spesifikasi Max6675

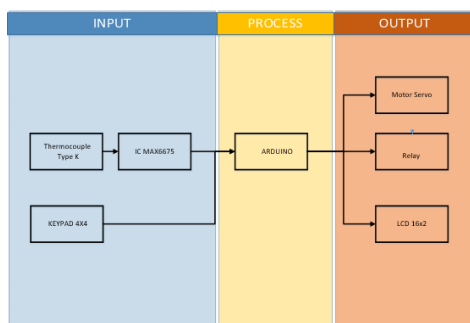
Tegangan Operasi	3VDC-5,5VDC
Antarmuka	High Speed Spi
Konsumsi Arus	700µA (1,5mA maks)
Range Temperatur	0°C-1024 °C
Akurasi	±3 °C.
Resolusi	12-Bit (0.25 °C)
Waktu Konversi	170mS

Motor Servo

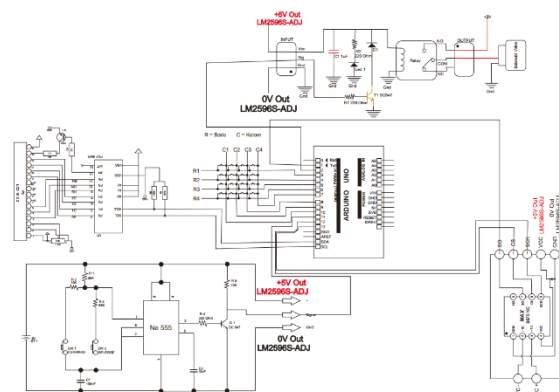
Servo bekerja dengan kendali dari kondrol yang memberikan pulse with modulation (PWM) pada signal servo [9]. Durasi yang diberikan akan menentukan arah putaran servo. Pada contoh apabila servo diberikan denyut dengan waktu 1,5ms akan menghasilkan putaran servo sebesar 90 derajat. Apabila kurang dari nilai tersebut maka akan berputar pada posisi 0 derajat atau kekiri berlawanan arah jarum jam.

3 Metodologi

Perancangan perangkat keras meliputi unit masukan dan unit keluaran pada mikrokontroler. Blok diagram dibawah ini akan menunjukan gambaran mengenai rancangan perangkat keras yang digunakan, mulai dari sensor yang digunakan, mikrokontrol, akuator, penampil informasi, dan perangkat penyesuaian nilai. Lihat blok diagram pada gambar 4



Gambar 4 Blok Diagram

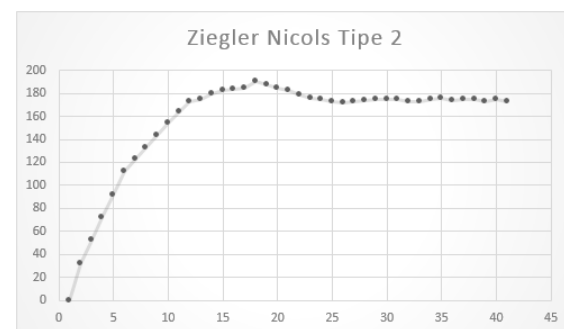


Gambar 5 Rangkaian keseluruhan total

Pada implementasi alat, input menggunakan sensor thermocouple typeK dengan ic max6675, proses menggunakan modul arduino, dan output menggunakan motor servo dan ditambahkan juga relay untuk mengendalikan solenoid gas. Sensor akan mendeteksi nilai suhu. Error antara setpoint dan pembacaan suhu akan diproses oleh sistem PID yang tertanam di dalam modul arduino. nilai dari perhitungan pid akan memutar servo. Apabila keluaran pid lebih dari 90 maka servo akan tetap berpada pada posisi 90 derajat.

3.1 Menentukan Nilai Kp, Ki, Kd

Pada penelitian yang dilakukan untuk menentukan nilai konstanta akan digunakan metode ZN tipe 2. Dihitung dari respon yang dihasilkan oleh slider gain yang menyebabkan respon beresilasi. Hanya kontrol proportional yang digunakan untuk mencari tuning konstanta. Dengan cara menaikkan nilai KP dari 0 sampai tercapainya nilai kp yang didapat menghasilkan respon sistem yang beresilasi [10].



Gambar 6 Proses design metode ziegler nicols tipe 2

Gambar di atas merupakan percobaan tuning menggunakan metode ZN tipe 2 yang menghasilkan nilai kcr sebesar 53. Setelah di dapat respon dari Closed Loop. Setelah didapatkan nilai, maka dilanjutkan dengan mencari nilai per. Nilai Pcr diperoleh dengan mengurangkan waktu pada kedua tinggi gelombang tersebut pada saat sistem mencapai setpoint

$$Pcr = x1 - x0$$

$$Pcr = 15,56 - 14,48$$

$$Pcr = 1,08$$

Ketika nilai Kcr dan Pcr diperoleh, masukan rumus-rumus PID ZN tipe 2, maka akan di dapat sebagai berikut :

$$Kp = 0,6 Kcr = 0,6 \times 53 = 3,18$$

$$Ti = 0,5 Pcr = 0,5 \times 1,08 = 0,54$$

$$Td = 0,125 Pcr = 0,125 \times 1,08 = 0,135$$

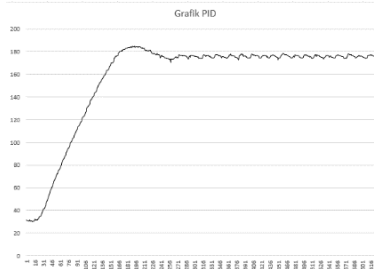
Selanjutnya cari nilai Ki dan Kd.

$$Ki = 2 \frac{Kp}{Ti} = 2 \frac{3,18}{0,54} = 117,8$$

$$Kd = Kp \times Td = 31,8 \times 0,135 = 4,3$$

4 Pembahasan

Pengujian ini dilakukan dengan memberikan nilai $Kp = 3$, $Ki = 118$, $Kd = 4$. Pengujian dilakukan dengan mengambil sample data pembacaan suhu dan di masukan ke dalam tabel. Percobaan dilakukan dengan menggunakan minyak sebanyak 2 liter.



Gambar 7 Hasil Percobaan

Dari Hasil Pengukuran di dapatkan nilai overshoot 5,5174 % , steady state 0 dengan waktu Settling time sebesar 576 detik, dan waktu rise time sebesar 105 detik, dan waktu delay time sebesar 23 detik

4.1 Rise time

Rise time merupakan waktu yang dibutuhkan oleh respons sistem untuk naik dari 10% hingga 90% dari nilai akhir (steady-state value) dalam sistem kontrol. Untuk menghitung rise time, pertama-tama tentukan nilai awal (T_0) dan nilai akhir (T_{ss}) dari respons sistem. Kemudian, hitung 10% dan 90% dari total kenaikan nilai dengan rumus:

$$T_{10\%} = T_0 + 0,1 \times (T_{ss} - T_0)$$

$$T_{90\%} = T_0 + 0,9 \times (T_{ss} - T_0)$$

Selanjutnya, identifikasi waktu ketika respons pertama kali mencapai atau melebihi 10% (t_1) dan 90% (t_2) dari nilai akhir. Rise time (T_r) kemudian dihitung sebagai selisih antara t_2 dan t_1

$$T_r = t_2 - t_1$$

Perhitungan waktu t_1 :

$$T_{10\%} = 31,75 + 0,1 \times (175 - 31,75)$$

$$T_{10\%} = 46,75$$

Didapatkan temperatur pada t_1 yaitu 46,75°C di waktu 36 detik

Perhitungan waktu t_2 :

$$T_{90\%} = 31,75 + 0,9 \times (175 - 31,75)$$

$$T_{90\%} = 160,75$$

Didapatkan temperatur pada t_2 yaitu 160,75°C di waktu 141 detik

Rise time = $t_2 - t_1 = 141 - 36 = 105$ detik

Rise time yaitu 105 detik

4.2 Error Steady State

Steady State merupakan selisih antara masukan dan keluaran suatu sistem dalam batas seiring berjalannya waktu hingga tak terhingga (ketika respon sudah mencapai keadaan tunak). Cara mencari nilai steady state error yaitu dengan cara mengidentifikasi waktu dimana respon telah berada pada keadaan Steady-state, dimana nilai keluaran sudah tidak berubah lagi secara signifikan. Ini dilakukan dengan cara melihat grafik dan menentukan interval waktu dimana nilai respons stabil.

Diketahui pada tabel (**Lampiran 2**) nilai sudah memasuki stabil pada waktu 4 menit 26 detik sampai 10 menit 23 detik

(waktu selesai pengambilan data) dalam periode waktu padawaktu 4 menit 26 detik sampai 10 menit 23 detik dapat kita hitung rata-rata nilai respons.

Diketahui nilai rata-rata = 175

Untuk menghitung steady state, dilakukan dengan cara melihat perbedaan antara rata-rata dengan setpoint.

Diketahui Setpoint = 175

Nilai Rata-rata = 173

Maka, Steady state error = nilai setpoint – nilai rata-rata

Hasilnya : $175 - 173 = 2\%$.

Maka dapat disimpulkan bahwa nilai Error steady state = 2%.

4.3 Overshoot

Overshoot adalah kondisi di mana respons suatu sistem kontrol melebihi nilai yang diinginkan atau nilai steady-state setelah diberi input perubahan, seperti input langkah. Dalam istilah sederhana, ini adalah seberapa jauh sistem "melampaui" target sebelum akhirnya menetap pada nilai yang benar.

Rumus Menghitung overshoot :

Dimana

$$M_p = \frac{y_{max} - y_{ss}}{y_{ss}} \times 100\%$$

$M_p =$ Overshoot

$y_{max} =$ nilai puncak pertama sebelum steady state

$y_{ss} =$ nilai rata-rata steady

Diketahui :

$M_p = ??$

$y_{max} = 184,75$

$y_{ss} = 175$

Maka,

$$M_p = \frac{184,75 - 175}{175} \times 100\%$$

Hasilnya $M_p = 5,51743$

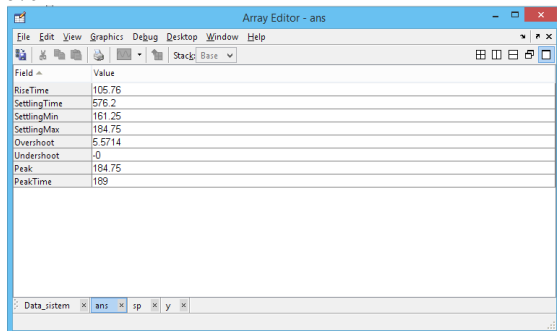
4.4 Delay Time

Delay time (waktu tunda) adalah interval waktu antara pemberian input atau perubahan yang diberikan kepada suatu sistem dan saat respons sistem pertama kali mulai merespons atau bereaksi secara signifikan terhadap input tersebut. Dalam sistem kontrol, delay time mencerminkan waktu yang diperlukan bagi sistem untuk merespons perubahan input setelah pemberian input tersebut, atau waktu yang diperlukan bagi suatu sinyal untuk mengalir melalui sistem dan mencapai hasilnya. Delay time dapat disebabkan oleh berbagai faktor, seperti waktu yang diperlukan bagi sinyal untuk melakukan perjalanan melalui sistem fisik atau waktu pemrosesan dalam sistem elektronik. Pentingnya delay time tergantung pada aplikasi sistem tersebut, tetapi dalam banyak kasus, delay time yang terlalu besar dapat mengakibatkan kinerja sistem yang buruk atau bahkan ketidakstabilan, terutama dalam sistem kontrol waktu nyata atau sistem yang memerlukan respons cepat.

Jika kita lihat pada data menyebutkan perubahan suhu secara signifikan terjadi pada detik ke 23 yang berarti sistem ini memiliki delay time sebesar 23 detik

4.5 Settling time

Settling time (waktu pemantapan) adalah waktu yang diperlukan oleh respons sistem untuk mencapai dan tetap berada dalam batas toleransi tertentu dari nilai steady-state setelah terjadi perubahan input. Dalam istilah sederhana, settling time menunjukkan berapa lama sistem membutuhkan waktu untuk "diam" atau "pemantapan" setelah terjadi perubahan atau gangguan. Ini penting karena menunjukkan seberapa cepat sistem bisa kembali stabil setelah mengalami perubahan. Pada penelitian ini penulis menggunakan aplikasi matlab untuk menghitung settling time dan di simpulkan bahwa nilai settling time sebesar 576 detik



Gambar 8 Hasil Perhitungan Matlab

4.6 Tuning nilai konstanta

Pengambilan nilai konstanta yang dilakukan sebelumnya mendapatkan nilai $K_P = 3$, $K_I = 118$, $K_D = 4$ dari proses tuning menggunakan metode ziegler nicols tipe 2. Nilai ini merupakan nilai acuan yang digunakan, namun masih bisa disesuaikan dengan cara melakukan trial & error untuk melihat respon sistem yang memiliki nilai paling efisien. Berikut tabel 10 pengambilan data yang telah di lakukan

Tabel 4. 1 Hasil Pengambilan Data

INPUT			Output				
KP	KI	KD	Delay time	Resting time	Overshoot	Error Steady State	setling time
6	121	7	83	162	5,28%	2,25%	456
9	124	10	86	164	5,57%	1,00%	555
12	127	13	88	176	5,57%	1,00%	495
15	130	16	90	171	6,85%	1,25%	495
18	133	19	89	168	6,71%	3,00%	495
2	100	13	86	167	4,71%	-0,50%	495
2	97	10	91	170	5,14%	-0,25%	447
2	94	7	116	193	9,57%	3,75%	442
2	91	4	116	191	9,42%	2,50%	447
1,5	87	2	117	191	9,57%	6,50%	430

5 Kesimpulan

Pada pengujian sistem ke 7 didapatkan respon sistem terhadap pengaturan temperatur dengan nilai overshoot 5,14 , Error steady state -0.25% dengan waktu Settling time sebesar 447 detik, dan waktu rise time sebesar 170 detik, dan waktu delay time sebesar 91 detik. yang di dihasilkan dari nilai $K_p = 2$, $K_i = 97$, $K_d = 10$.

Referensi

- [1] B. A. Prastyo and H. Mahmudi, "Perancangan Sistem Penggorengan Pada Mesin Pembuat Keripik Serbaguna Dengan Metode Deep Frying," *Semin. Nas. Inov. Teknol. UN PGRI Kediri*, vol. 4, no. 1, pp. 1–6, 2020.
- [2] S. Gunawan *et al.*, "Pendampingan Produk UMKM di Sukolilo menuju Sertifikasi Halalan Thayyiban," *Sewagati*, vol. 4, no. 1, p. 14, 2020,
- [3] W. M. P. Wiratama, "Pengendali Pid (Proportional, Integral Dan Derivative) Dengan Flc (Fuzzy Logic Control)," *J. Simetris*, vol. 14, no. 1, pp. 77–88, 2023.
- [4] M. Putri, A. Ma'arif, and R. Puriyanto, "Pengendali Kecepatan Sudut Motor Dc Menggunakan ," *J. Techno*, vol. 23, no. 1, pp. 09–18, 2022.
- [5] R. P. Hati and A. Subari, "Rancang Bangun Sistem Pengontrol Batch Mixer Pada Industri Minuman Dengan Metode Pid Berbasis Arduino Uno R3," *Gema Teknol.*, vol. 20, no. 1, p. 10, 2018,
- [6] P. Wibowo and D. A. Prasetya, "Rancang Bangun Data Logger Multi Kanal Terhubung IoT (Internet Of Things) Sebagai Pengukur Temperatur dengan Sensor Thermocouple," *Emit. J. Tek. Elektro*, vol. 21, no. 2, pp. 87–94, 2021,
- [7] S. Rijal, "Rancang Bangun Sistem Pengendalian Temperature Proses Rekrystalisasi Pada Rancang Bangun Sistem Pengendalian Temperature Proses Rekrystalisasi," 2017.
- [8] M. Kharis Mubarak *et al.*, "Rancang-Bangun Sistem Monitoring Suhu Dan Getaran Motor Pompa 3 Fasa Mini Plant Dcs Berbasis Labview," *291 Pros. NCIET*, vol. 3, pp. 291–304, 2022.
- [9] W. Wahyudi, A. Rahman, and M. Nawawi, "Perbandingan Nilai Ukur Sensor Load Cell pada Alat Penyortir Buah Otomatis terhadap Timbangan Manual," *Elkomika J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 5, no. 2, p. 207, 2018,
- [10] P. Agastansa., "Kendali Temperatur Menggunakan PID untuk Sistem Pasteurisasi Susu," *Tugas Akhir*, 2018.

Halaman ini sengaja dikosongkan