Mata Kuliah Sistem Kontrol, 25 Juni (2024)

Analisis Penggunaan PID untuk Mengatur Suhu Ruangan

Juhen Fashikha Wildan[1] (163221047), Gravano Alfa[1] (163221013), Ronald Silvester Sanger[1] (163221044), Daniel Pardamean H.S.[1] (1632210473)

[1]*Teknik Robotika dan Kecerdasan Buatan, Universitas Airlangga, Indonesia*

***Abstract*—Kontrol suhu ruangan yang optimal sangat penting untuk kenyamanan dan efisiensi energi. Tujuan dari penelitian ini adalah untuk menilai dampak penerapan kontrol PID dalam sistem yang dirancang untuk mengatur suhu ruangan. Teknik penelitian terdiri dari pengujian sistem menggunakan tiga jenis penalaan PID yang berbeda: overdamped, critically damped, dan underdamped. Hasilnya menunjukkan bahwa penyetelan yang sangat teredam menghasilkan respons sistem yang paling baik, ditandai dengan waktu respons terpendek dan overshoot yang dapat diabaikan. Di sisi lain, penyetelan overdamped menawarkan peningkatan stabilitas, namun dengan respons yang lebih lambat dan tidak adanya osilasi. Penyetelan underdamped mencapai setpoint dengan cepat, namun disertai dengan osilasi dan overshooting yang signifikan. Studi ini menunjukkan bahwa pemanfaatan kontrol PID yang tepat dapat meningkatkan efisiensi dan stabilitas sistem pengaturan suhu ruangan. Temuan ini memberikan kontribusi yang signifikan terhadap pengembangan sistem kontrol suhu dan menawarkan saran untuk penerapannya secara lebih luas.**

***Keywords*—Kontrol PID, Pengaturan suhu ruangan, Efisiensi energi, Penalaan PID.**

I. INTRODUCTION

Sistem kontrol adalah sebuah sistem yang mengatur satu variabel sehingga berada pada range tertentu [1]. Sistem kontrol mempelajari tingkah laku manusia dalam bekerja, dimana sistem akan mengamati kualitas dari apa yang mereka kerjakan sehingga memiliki karakteristik yang sesuai dengan keadaan yang diharapkan. Karena dengan definisinya yang dimana sistem kontrol dapat mengatur variabel tertentu sehingga akan menghasilkan data pada range tertentu, Sistem kontrol ini sangat memungkinkan untuk digunakan dalam pengaturan suhu ruangan.

Suhu merupakan satuan dari intensitas panas. Suhu dapat mempengaruhi kenyamanan manusia untuk melakukan suatu kegiatan [2]. Pada normalnya suhu ruangan yang nyaman untuk manusia adalah 18°C hingga 24°C. Namun, perbedaan suhu pada berbagai wilayah dan atau berbagai negara yang memiliki ciri khas geografis masing-masing membuat suhu ruangan tidak pada suhu normal tiap harinya. Willis Haviland Carrier menemukan teknologi Air Conditioner [3] dengan memaksa udara melewati filter kompresor yang digerakkan piston. Udara tersebut akan dipompa di atas koil yang didinginkan menggunakan pendingin lalu dialirkan ke ruangan tertutup menggunakan kipas angin. Meskipun terkesan sederhana, sistem pendingin udara tersebut terbukti efektif mengatasi permasalahan di percetakan Brooklyn.

Sistem kontrol dapat mengambil peran sangat penting untuk pengaturan suhu pada air conditioner. pada prototipe air conditioner sederhana, kami menggunakan motor DC sebagai aktuator untuk dijadikan seperti kipas dan sensor suhu sebagai observer untuk membaca suhu sekitar. Sistem diberikan input set point untuk menentukan suhu yang diinginkan kemudian sistem akan mengakumulasi dan memberikan sinyal kepada motor untuk berputar hingga suhu ruangan sekitar bersuhu sesuai dengan apa yang telah diinputkan.

Untuk membuat pengaturan suhu ruangan menggunakan sistem kontrol perlu membutuhkan beberapa variabel sistem seperti tegangan, laju kalor untuk input dan suhu untuk output.

II. TINJAUAN PUSTAKA

A. Sistem Kontrol

Sistem Kontrol (Control System) merupakan sebuah sistem yang saling terhubung satu sama lain dan akan membentuk suatu konfigurasi sistem yang kemudian memberikan respon (keluaran) seperti apa yang diharapkan [4]. Sistem kontrol dapat mengontrol diri sendiri, memberikan perintah-perintah, dan atau sistem yang lainnya sehingga kita mendapatkan respon yang kita inginkan.

B. Ziegler-Nichols

Teori Ziegler-Nichols adalah salah satu cara untuk menemukan nilai Kp, Ki, dan Kd. Nilai PID diperoleh dari hasil percobaan dengan memasukkan unit langkah. Hasilnya akan membentuk kurva berbentuk S, tetapi jika kurva ini tidak terbentuk, metode ini tidak dapat digunakan. Kurva bentuk S memiliki dua konstanta: waktu tunda L dan konstanta waktu T. Kedua parameter ini diperoleh dengan menggambar garis tangensial pada titik infleksi kurva S [5].

C. Suhu

Suhu menurut KBBI merupakan ukuran kuantitatif terhadap temperatur [6]. Menurut Fennani Arpan, Dewi Galuh, dan Sudjarwadi, Suhu bukan merupakan kualitas panas, melainkan satuan dari intensitas panas [7]. Suhu ruangan yang nyaman dan aman untuk kesehatan manusia adalah berkisar antara 18°C hingga 24°C menurut World Health Organization (WHO) [2].

D. Motor DC

Motor DC merupakan jenis motor yang dapat bergerak bila diberikan sumber tegangan DC [8]. Kecepatan motor DC ditentukan oleh perubahan tegangan yang diinputkan ke motor tersebut. Sedangkan arah putar motor DC ditentukan oleh arus masuk positif dan arus keluar negatif.

E. Sensor Suhu

Sensor suhu adalah salah satu jenis sensor yang membuat

suatu circuit atau objek robotika dapat mengenali variabel suhu. Sensor suhu bekerja dengan mengubah suhu sekitar menjadi bentuk tegangan input [9]. Sensor suhu dapat dikombinasikan dengan microcontroller untuk melakukan pembatasan range temperature dengan cara menggunakan pemrograman pada microcontroller [10]. Selain dengan melakukan pemrograman, Sensor suhu dapat mendeteksi suhu dengan keinginan kita sesuai dengan spesifikasi sensor yang digunakan [11]. Beberapa sensor suhu yang pada umumnya digunakan adalah DHT 11, DHT 22, MAX 6675, dan DS18B20.

F. PID (Proportional Integral Derivative)

PID, singkatan dari Proportional-Integral-Derivative, merupakan pengontrol yang memanfaatkan mekanisme umpan balik dalam loop kontrol [12]. Ini biasanya digunakan dalam sistem kontrol industri dan aplikasi lain yang memerlukan kontrol termodulasi berkelanjutan. Pengontrol ini beroperasi menggunakan tiga parameter utama:

F.1 Proportional (P):

Memonitoring reaksi variabel sebagai konstanta waktu dalam loop tertutup. Parameter ini memastikan tidak ada perubahan dalam tatanan sistem dengan melihat keluarannya sebanding dengan masukannya [13].

F.2 Integral (I):

Memeriksa variabel proses dari waktu ke waktu dan mengoreksi keluaran dengan mengurangi offset dari variabel proses [14].

F.3 Derivative (D):

Memantau laju perubahan variabel proses, oleh karena itu parameter ini dapat mengubah output ketika terdapat variasi yang tidak biasa [14].

G. Implementasi PID dalam Pengaturan Suhu Ruangan

Pengontrol PID digunakan dalam pengelolaan suhu ruangan untuk memastikan bahwa elemen pemanas atau pendingin dikontrol sedemikian rupa sehingga suhu ruangan tetap stabil, sebagaimana ditentukan oleh titik setel yang diinginkan. Prosedur ini memerlukan beberapa tahap. Awalnya, sensor suhu mengukur suhu ruangan secara tepat dan mengirimkan data ini ke mikrokontroler. Selanjutnya, mikrokontroler menghitung kesalahan dengan mengurangkan suhu saat ini dari titik setel target. Algoritma PID memanfaatkan kesalahan ini untuk menentukan penyesuaian tepat yang diperlukan untuk elemen pemanas atau pendingin [15]. Pada akhirnya, koreksi PID diterapkan untuk mengontrol kecepatan motor DC atau komponen lain yang terlibat dalam memodifikasi suhu ruangan.

H. Respon Sistem

Melakukan analisis terhadap reaksi sistem dalam konteks dinamika dan kontrol memungkinkan pemahanam komprehensif tentang bagaimana sistem berperilaku dalam menanggapi sebuah input yang diberikan.

H.1 Overdamped

Overdamped adalah respon dari sistem ketika sinyal sistem berhenti bergerak saat mencapai titik kesetimbangannya. Hal ini dapat terjadi karena adanya suatu redaman yang melewati sistem [16].

H.2 Critically damped

Critically Damped merupakan sistem respon yang mirip dengan sistem respon Overdamped. Perbedaannya adalah rise time pada Critically Damped lebih cepat daripada overdamped [17].

H.3 Underdamped

Underdamped adalah respon dari sistem ketika sinyal sistem menuju keadaan yang diinginkan dengan berosilasi, dimana amplitudo pada sinyal terhadap waktu akan perlahan mengecil, dan akhirnya menuju nol [18].

III. MODEL MATEMATIKA SISTEM

A. Plant Temperature Control

$$q' - w' = \frac{dU}{dt}, \text{persamaan konservasi energi} \quad (1)$$

$$\oint \frac{dq}{dt} \leq 0, \text{Teorema Clausius} \quad (2)$$

$$-q' + \frac{kA(T_{amb}-T)}{x} = \frac{d}{dt}\iiint_{CV} \rho u d\forall - m'Cp(T_{amb} - T) \quad (3)$$

Asumsikan bahwa T_{amb} adalah time-invariant, sehingga persamaan menjadi:

$$-q' = M.C\frac{d(T_{amb}-T)}{dt} + \left(m'Cp + \frac{kA}{x}\right).(T - T_{amb}) \quad (4)$$

M adalah massa termal (J/m^{3}K), C adalah total kapasitas panas spesifik (J/kg.K), m' adalah laju aliran udara yang dipertukarkan (m^3/h), Cp adalah kapasitas panas produk (J/kg.K), k adalah koefisien konduktivitas termal (W/m.K), A adalah luas area ruangan (m^2), x adalah tebal dinding (m), T_{amb} adalah suhu luar ruangan (Kelvin), T adalah suhu dalam ruangan (Kelvin).

Transfer Function:

$$-q'(s) = M.C.s + \left(m'Cp + \frac{kA}{x}\right).T(s) \quad (5)$$

$$G(s) = \frac{T(s)}{-q'(s)} = \frac{1}{M.C.s+\left(m'Cp+\frac{kA}{x}\right)} \quad (6)$$

B. Plant Sensor Suhu

$$m_{sen}C_{sen}\frac{dT_{sen}}{dt} = h_{conv}A_{sen}(T - T_{sen}) \quad (7)$$

M_{sen} adalah massa sensor (kg), C_{sen} adalah kapasitas panas sensor (J/kg.K), h_{conv} adalah koefisien konveksi (W/m^2.K), Asen adalah luas area sensor (m^2).

Transfer Function:

$$(m_{sen}.C_{sen}.s + h_{conv}.A_{sen}).T_{sen}(s) = h_{conv}.A_{sen}.T(s) \quad (8)$$

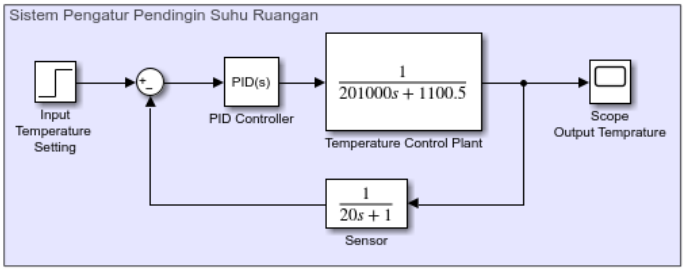
$$H(s) = \frac{T_{sen}(s)}{T(s)} = \frac{h_{conv}A_{sen}}{(m_{sen}C_{sen}.s+h_{conv}A_{sen})} \quad (9)$$

$$H(s) = \frac{T_{sen}(s)}{T(s)} = \frac{1}{\frac{m_{sen}C_{sen}.s}{h_{conv}A_{sen}}+1} \quad (10)$$

IV. DATA ANALYSIS

Inisialisasi parameter kritis sangat penting untuk konstruksi closed-loop transfer function yang diinginkan untuk sistem kontrol suhu di bagian ini. Kriteria seperti massa termal (M)

didefinisikan sebagai 200 kg. Kapasitas panas spesifik udara (C) ditetapkan pada 1005 J/kg·K. Kecepatan aliran massa udara (m) adalah 0,1 kg/s. Kapasitas spesifik udara (Cp) adalah 1005 J/kg·K. Koefisien transfer panas (k) adalah 10 Watt/m^2·K. Area permukaan pertukaran panas (A) adalah 30 m^2. Ketebalan dinding (x) adalah 0,3 m. Selain itu, parameter sensor juga diatur ke nilai asli, massa sensor (m_{sen}) adalah 0,01 kg, kapasitas panas spesifik sensor (C_{sen}) adalah 500 J/kg·K, koefisien transfer panas spesifik konvektif (h_{conv}) adalah 25 Watt/m^2·K, dan area permukaan sensor (A_{sen}) sebesar 0,01 m^2.



Gambar 4.1 Diagram Blok Sistem Pengatur Pendingin Suhu Ruangan

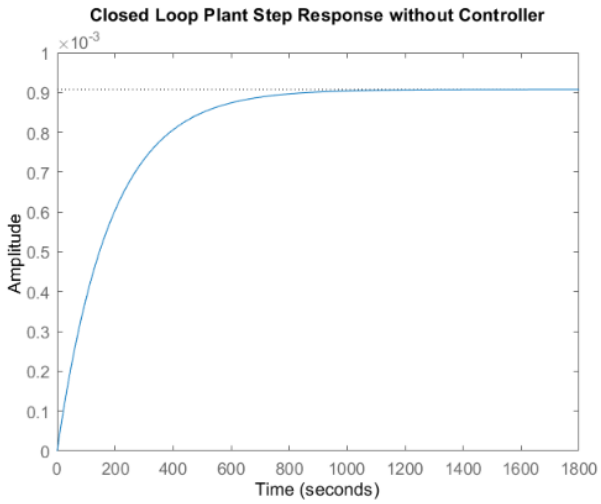
Berdasarkan inisialisasi parameter tersebut, respon dari system diprogram dengan menggunakan MATLAB karena menyediakan berbagai fungsi yang berguna untuk klasifikasi berbagai macam respon sistem. Bagian ini akan mengeksplorasi proses langkah demi langkah dalam mengevaluasi respons sistem menggunakan MATLAB, menyoroti dampak controller PID terhadap kinerja sistem.

A. Respon sistem sebelum PID

Sebelum diberikan pembaruan respon sistem berdasarkan PID, didapatkan transfer function untuk sistem sebagai berikut:

$$\text{Closed Loop System} = \frac{20s+1}{4020000s^2+223010s+1102} \qquad (11)$$

Berdasarkan transfer function dari closed loop system tersebut, dibentuklah suatu grafik respon sistem sebagai berikut:



Gambar 4.2 Respons Closed-Loop tak terkontrol

Didapatkan informasi system sebelum PID sebagai berikut:

- Rise time : 400 seconds
- Settling time : 713 seconds
- Settling min : 0.0008219
- Settling max : 0.00090783
- Peak : 0.00090783
- Peak time : 1924.2 seconds

Rise time, yang merupakan durasi yang diperlukan agar respons sistem berubah dari 10% menjadi 90% dari nilai akhir, adalah 400 detik. Hal ini menunjukkan reaksi yang relatif lambat terhadap perubahan masukan. Kemudian settling time yakni periode yang diperlukan agar respons tetap berada dalam persentase tertentu dari nilai akhir (umumnya 2% atau 5%), adalah 713 detik, yang menandakan system tidak begitu cepat menapai nilai akhir. Nilai settling max, yaitu amplitudo akhir maksimum yang dicapai oleh sistem, adalah 0,00090783. Selain itu, nilai peak, yang juga sesuai dengan nilai kondisi steady state pada 0,00090783 menandakan tidak adanya overshoot, yang menunjukkan bahwa sistem mengalami overdamped. Terakhir peak time, atau waktu yang diperlukan untuk mencapai respons puncak, adalah 1.924,2 detik, yang menunjukkan respons lambat.

B. Respon sistem setelah PID

Untuk mengoptimalkan kinerja sistem, control Proportional-Integral-Derivative (PID) dapat diimplemen-tasikan. Bagian ini akan menunjukkan tuning tersebut untuk berbagai kategori respon sistem yang berbeda seperti underdamped, overdamped, dan critically damped. Selain itu, metode Ziegler-Nichols akan digunakan untuk penyetelan PID yakni sebuah pendekatan heuristik yang banyak digunakan dan dirancang untuk mencapai keseimbangan yang diinginkan antara respon dan stabilitas. Berikut transfer function dari sistem closed-loop yang menggunakan parameter PID.

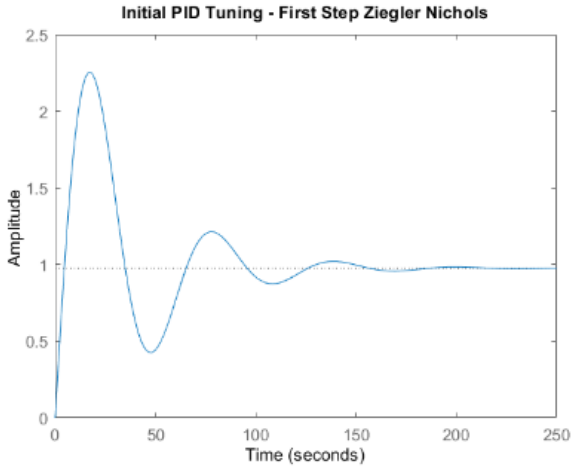
$$\frac{K_i+s(K_ds+K_p)}{K_i+s(20s+1)(201000s+1100.5)+s(K_ds+K_p)} \qquad (12)$$

B.1 Respon system tuning PID Ziegler-Nichols

Didapatkan transfer function untuk respon sistem dengan tuning Ziegler-Nichols sebagai berikut:

$$Tf = \frac{4.31\times10^6s^3+7.555\times10^5s^2+4.391\times10^4s+845.7}{4020000s^3+438500s^2+28100s+845.7} \qquad (19)$$

Metode Ziegler-Nichols diimplementasikan pertama kali dengan melakukan inisialiasi pada nilai proportional (Kp) dan mengaplikasian pergantian sinyal step pada setpoint yang diuji. Hal ini bertujuan untuk menaikkan nilai dari proportional gain (Kc) hingga mencapai osilasi yang stabil.



Gambar 4.6 Respons Langkah Awal Tuning Ziegler-Nichols

Setelah menggapai osilasi yang stabil, didapatkan nilai jarak antara peak sinyal (Pu), serta nilai ultimate gain (Ku) yakni nilai Kp yang didapat saat sistem berosilasi secara stabil. Sehingga dengan nilai tersebut, dapat digunakan untuk perhitungan nilai PID pada sistem. Pada sistem, didapatkan nilai tersebut sebesar 45000 (Ku), dan 63.852 (Pu). Berikut merupakan tabel PID Ziegler-Nichols:

Table 1. PID Tuning Ziegler-Nichols [19]

Tipe Controller	Kp	τ_i	τ_d
P	0.5 Ku	-	-
PI	0.45 Ku	Pu/1.2	-
PID	0.6 Ku	Pu/2	Pu/8

Berdasarkan nilai pada tabel tersebut, didapatkan perhitungan untuk nilai PID sebagai berikut:

- Pencarian nilai Kp:

$$Kp = 0.6 \times Ku = 0.6 \times 45000 = 27000 \quad (20)$$

- Pencarian nilai Ki:

$$\tau i = \frac{Pu}{2} = \frac{63.852}{2} = 31.926 \quad (21)$$

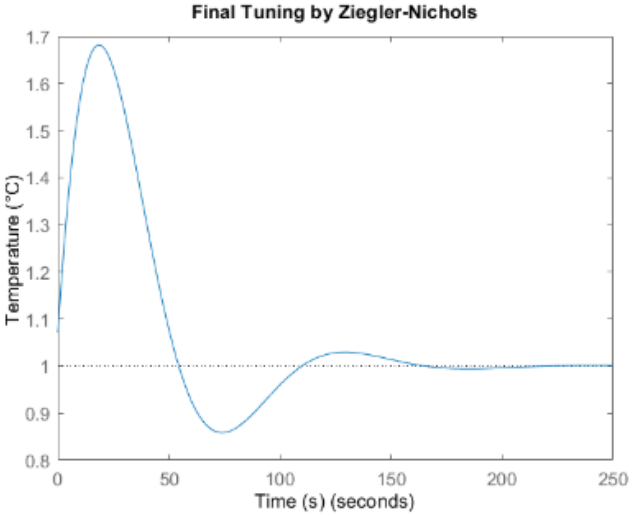
$$Ki = \frac{Kp}{\tau i} = \frac{27000}{31.926} = 845.70 \quad (22)$$

- Pencarian Nilai Kd:

$$di = \frac{Pu}{8} = \frac{63.852}{8} = 7.981 \quad (23)$$

$$Kd = Kp \times \tau d = 27000 \times 7.981 = 215500 \quad (24)$$

Nilai PID kemudian akan diinput ke dalam sistem yang menghasilkan output sistem sebagai berikut:



Gambar 4.7 Respons Final Tuning Ziegler-Nichols

Didapatkan informasi sistem menggunakan tuning Zieger-Nichols sebagai berikut:

- Rise time : 3.3139 seconds
- Settling time : 150.2672 seconds
- Settling min : 0.8587
- Settling max : 1.0294
- Overshoot : 68.15%
- Peak : 1.6815
- Peak time : 17.8512 seconds

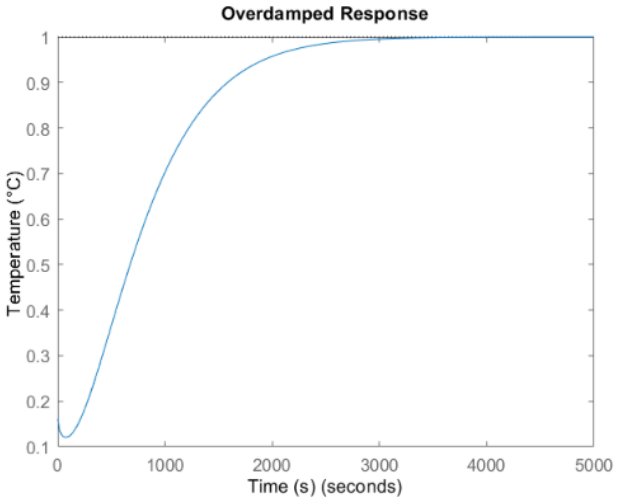
Setelah penerapan tuning PID menggunakan metode Ziegler-Nichols, waktu naik berkurang menjadi hanya 3,3139 detik, dan waktu menetap menurun menjadi 150,2672 detik, yang menunjukkan respons yang jauh lebih cepat dan stabilisasi lebih cepat. Namun, metode penyetelan ini menghasilkan overshoot substansial sebesar 68,15%, dengan nilai peak mencapai 1,6815. Peak time juga menurun secara signifikan menjadi 17,8512 detik, yang menunjukkan respons sistem yang lebih cepat.

B.2 Respon system overdamped

Didapatkan transfer function untuk respon sistem dalam kategori overdamped sebagai berikut:

$$Tf = \frac{6.469\times10^5 s^3 + 3.322\times10^4 s^2 + 71.51s + 1.377}{4020000s^3 + 255400s^2 + 1144s + 1.377} \quad (13)$$

Transfer function dari closed loop system tersebut, didapatkan dari persamaan (12) dengan rincian nilai Kp sebesar $27000 \times 0{,}7^{18}$, Ki sebesar $845{,}7057 \times 0{,}7^{18}$, dan Kd sebesar $215500 \times 0{,}9^{18}$. Transfer function tersebut menghasilkan output grafik respon sistem sebagai berikut:



Gambar 4.3 Respons Overdamped

Didapatkan informasi sistem menggunakan PID sehingga membuat sistem menjadi overdamped sebagai berikut:

- Rise time : 1326.7 seconds
- Settling time : 2406.9 seconds
- Settling min : 0.9191
- Settling max : 1
- Peak : 1
- Peak time : 5416.2 seconds

Nilai Rise time, yang menjadi 1326,7 detik, jauh lebih lama dibandingkan sebelumnya 400 detik, yang menunjukkan respons yang jauh lebih lambat terhadap perubahan input. Settling time juga meningkat secara signifikan menjadi 2406,9 detik dari sebelumnya 713 detik, yang menunjukkan perpanjangan durasi yang diperlukan agar sistem stabil dalam rentang yang dapat diterima di sekitar nilai akhir. Settling minimum sistem sebesar 0,9191 dan settling maksimum 1 menunjukkan bahwa respon tetap mendekati nilai akhir yang

diinginkan selama periode settling, tanpa adanya overshoot. Hal ini semakin diperkuat dengan nilai peak 1, yang menunjukkan bahwa sistem mencapai nilai akhir yang diinginkan tanpa overshoot. Untuk peak time, yakni 5416,2 detik, jauh lebih lama dari waktu aslinya yang 1924,2 detik menunjukkan respon lambat. Secara keseluruhan, setting PID yang telah ditetapkan berhasil mengkonfigurasi sistem menjadi overdamped, memastikan stabilitas dan tidak adanya overshoot. Nilai steady state error untuk sistem overdamped didapatkan melalui perhitungan sebagai berikut:

$$SSE\ Overdamped = \left(\frac{|24.999-25|}{25}\right) \times 100\% = 0.004\% \quad (14)$$

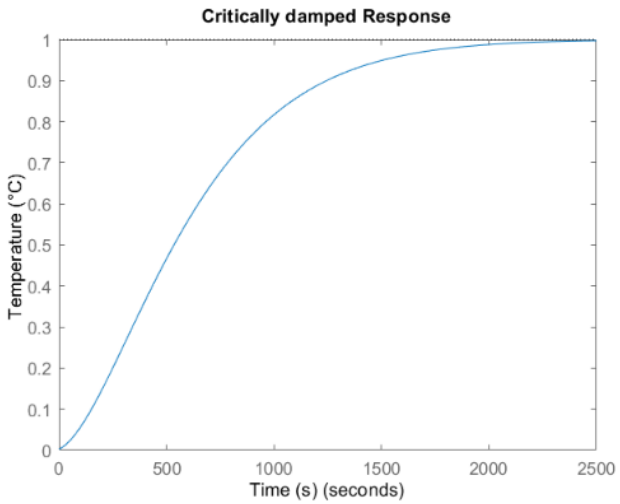
Nilai steady state error untuk sistem overdamped tersebut menunjukkan bahwa sistem tidak sepenuhnya berada pada kondisi steady state. Hal ini disebabkan respon pada overdamped sedikit tidak tepat pada steady state sekitar 0.004% dari nilai akhir.

B.3 Respon system critically damped

Didapatkan transfer function untuk respon sistem dalam kategori critically damped sebagai berikut:

$$\text{Tf} = \frac{8606s^3+1508s^2+87.68s+1.689}{4020000s^3+223400s^2+1154s+1.689} \quad (15)$$

Transfer function dari closed loop system tersebut, didapatkan dari persamaan (12) dengan rincian nilai Kp sebesar $27000 \times 0{,}9^{59}$, Ki sebesar $845{,}7057 \times 0{,}9^{59}$, dan Kd sebesar $215500 \times 0{,}9^{59}$. Transfer function tersebut menghasilkan output grafik respon sistem sebagai berikut:



Gambar 4.3 Respons Critically damped

Didapatkan informasi sistem menggunakan PID sehingga membuat sistem menjadi critically damped sebagai berikut:

- Rise time : 1091.6 seconds
- Settling time : 1829.9 seconds
- Settling min : 0.9030
- Settling max : 0.9996
- Peak : 0.9996
- Peak time : 2957.5 seconds

Setelah penerapan control PID untuk mencapai kondisi sistem critically damped, Rise time meningkat menjadi 1091,6 detik, menunjukkan respons awal yang lebih lambat. Settling time juga diperpanjang hingga 1829,9 detik, yang menggambarkan durasi yang lama bagi sistem untuk stabil di sekitar titik yang dikehendaki. Selain itu, rise time telah bergeser ke 2957,5 detik, yang menunjukkan respon lebih lama.

Nilai steady state error untuk sistem critically damped didapatkan melalui perhitungan sebagai berikut:

$$SSE\ CD = \left(\frac{|25.00007-25|}{25}\right) \times 100\% = 0.00028\% \quad (16)$$

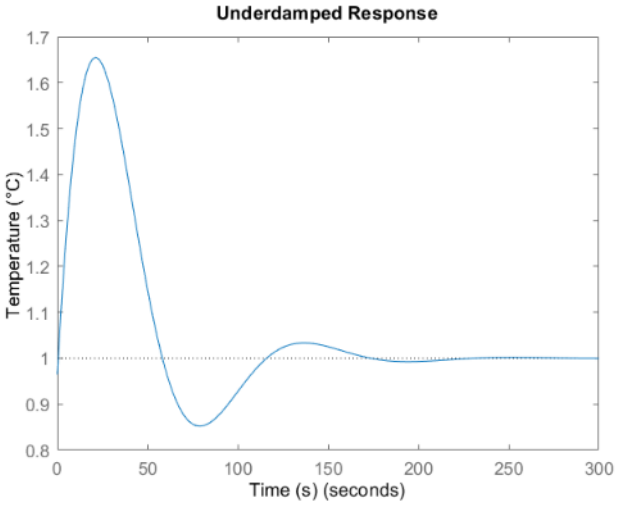
Nilai steady state error untuk sistem critically damped tersebut menunjukkan bahwa sistem tidak sepenuhnya berada pada kondisi steady state. Hal ini disebabkan respon pada overdamped sedikit tidak tepat pada steady state sekitar 0.0028% dari nilai akhir.

B.4 Respon system underdamped

Didapatkan transfer function untuk respon sistem dalam kategori underdamped sebagai berikut:

$$Tf = \frac{3.879\times10^6 s^3+679950s^2+39520s+761.1}{4020000s^3+416960s^2+25400s+761.1} \quad (17)$$

Transfer function dari closed loop system tersebut, didapatkan dari persamaan (12) dengan rincian nilai Kp sebesar $27000 \times 0{,}9$, Ki sebesar $845{,}7057 \times 0{,}9$, dan Kd sebesar $215500 \times 0{,}9$. Transfer function tersebut menghasilkan output grafik respon sistem sebagai berikut:



Gambar 4.5 Respons Underdamped

Didapatkan informasi sistem menggunakan PID sehingga membuat sistem menjadi critically damped sebagai berikut:

- Rise time : 0.4254 seconds
- Settling time : 160.51 seconds
- Settling min : 0.8521
- Settling max : 1.6550
- Peak : 1.6550
- Peak time : 21.445 seconds

Berdasarkan data tersebut, rise time yang sebesar 0,4254 detik menunjukkan seberapa cepat sistem mencapai nilai akhirnya dari awal, namun settlinig time yang sebesar 160,51 detik menunjukan bahwa sistem membutuhkan waktu yang cukup lama untuk mencapai steady state. Settling range dari 0,8521 hingga 1,6550, menunjukkan seberapa jauh osilasi respons sistem. Peak yang sebesar 1,6550 dan peak time pada 21,44 detik menunjukkan puncak maksimum selama respons,

yang merupakan fitur umum dari sistem underdamped di mana output melebihi nilai akhir sebelum mencapai steady state.

Nilai steady state error untuk sistem underdamped didapatkan melalui perhitungan sebagai berikut:

$$SSE\ CD = \left(\frac{|25-25|}{25}\right) \times 100\% = 0\% \quad (18)$$

Nilai steady state error untuk sistem critically damped tersebut menunjukkan bahwa sistem sepenuhnya berada pada kondisi steady state. Hal ini disebabkan respon pada overdamped memiliki nilai 0% dari nilai akhir.

C. *Time constant (τ)*

Time constant merupakan waktu yang diperlukan sistem untuk merespons perubahan pada setiap input. Ini adalah indikator kecepatan respons suatu sistem mendekati keadaan keseimbangan barunya setelah terjadi gangguan. Time constant menunjukkan seberapa cepat suatu sistem dapat mencapai 37% dari kondisi awal. Perhitungan time constant dapat diperoleh melalui persamaan sebagai berikut:

$$\tau = \frac{1}{|a|} \quad (25)$$

Variable meliputi:
- τ = time constant
- a = dominant pole

Nilai dari dominant pole dapat dilihat dalam program matlab. Berikut merupakan time constant dari berbagai macam respon dari sistem berdasarkan nilai dominant pole program:

- Tuning Ziegler-Nichols:

$$\tau = \frac{1}{|-0.0284|} = 35.21\ second \quad (26)$$

- PID overdamped:

$$\tau = \frac{1}{|-0.0284|} = 35.21\ second \quad (27)$$

- PID critically damped:

$$\tau = \frac{1}{|-0.0284|} = 35.21\ second \quad (28)$$

- PID underdamped:

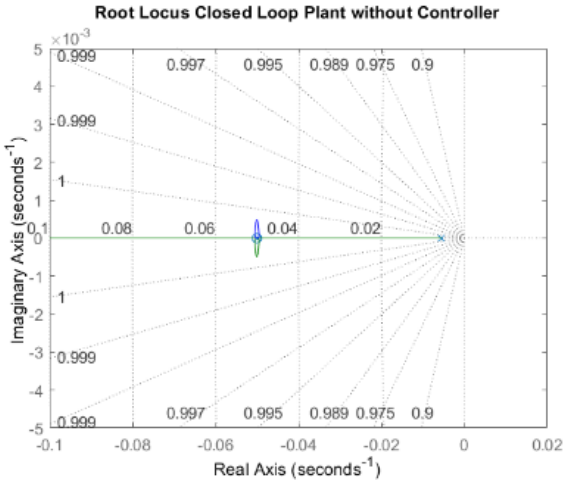
$$\tau = \frac{1}{|-0.0284|} = 35.21\ second \quad (29)$$

D. *Root Locus*

Root Locus digunakan untuk mempelajari bagaimana root dari persamaan karakteristik (pole) bergerak saat suatu parameter sistem seperti gain divariasikan. Hal ini berguna untuk memvisualisasikan bagaimana perubahan dalam parameter sistem mempengaruhi stabilitas. Plot Root Locus meliputi pole (X) dan zero (O) pada plot. Sifat pole saat gain divariasikan dari nol hingga tak terhingga menunjukkan bagaimana sistem bereaksi terhadap perubahan gain. Pole di setengah bagian kiri bidang kompleks menunjukkan stabilitas, sedangkan pole di setengah bagian kanan menunjukkan ketidakstabilan.

Penggunaan fungsi "rlocusplot" MATLAB digunakan untuk menganalisis stabilitas dan karakteristik berbagai respon sistem. Fungsi rlocusplot di MATLAB memungkinkan analisa tentang perilaku sistem dalam berbagai perolehan feedback. Dengan rlocusplot, fungsi ini membantu memvisualisasikan bagaimana perubahan parameter gain mempengaruhi stabilitas dan respons sistem.

D.1 Root locus sebelum PID



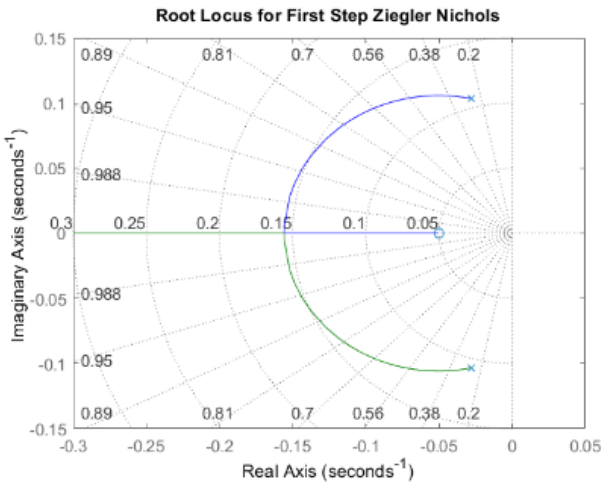
Gambar 4.8 Root Locus Kontrol Temperatur Tidak Terkontrol

Pada root locus untuk sistem sebelum PID, karena semua pole dan zero berada di sisi kiri s-plane, sistem ini dapat dikatergorikan sebagai sistem yang stabil. Stabilitas tersebut memastikan bahwa respons sistem tidak menyimpang seiring waktu.

Untuk pole di s = −0.0054831 dan -0.05 tanpa adanya sumbu imaginer dan berada bada sumbu real menunjukkan bahwa sistem dalam kategori critically damped. Hal ini menunjukkan respons terhadap perubahan input kembali ke keadaan seimbang secepat mungkin tanpa berosilasi.

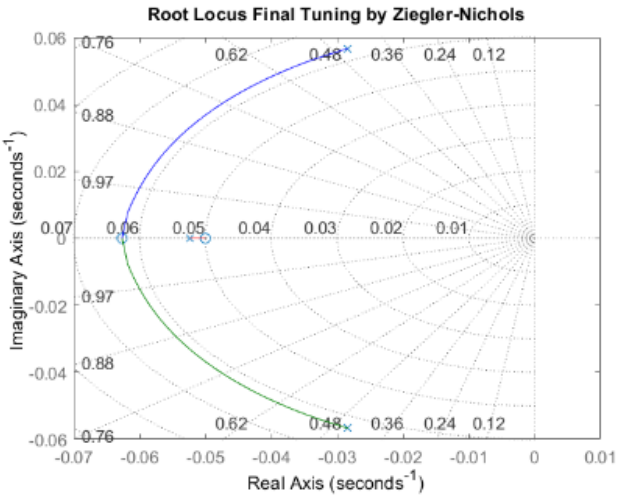
D.2 Root locus PID Ziegler-Nichols

Initial Root locus:



Gambar 4.9 Root Locus Langkah Awal Tuning Ziegler-Nichols

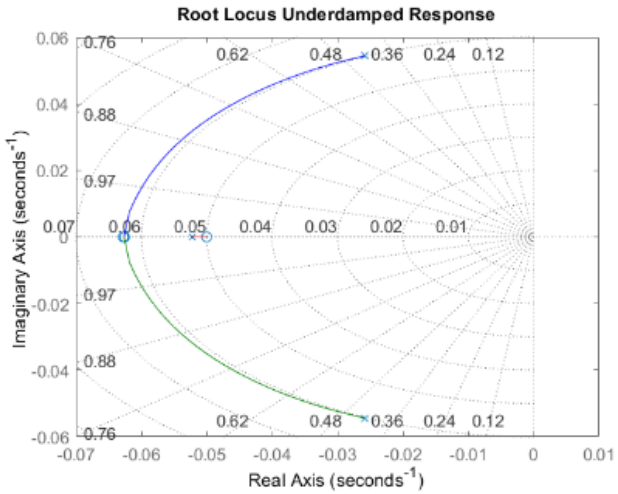
Final tuning root locus:



Gambar 4.10 Root Locus Kontrol Temperatur Final Tuning Ziegler-Nichols

Root locus tersebut menunjukkan loci dari closed-loop pole sistem saat Gain (K) bervariasi. Setiap pole pada sistem terletak pada sisi kiri (negatif) dari s-plane, yang menandakan bahwa sistem dalam keadaan stabil. Disaat nilai K bertambah, pole bergerak lebih ke kiri, yang menunjukkan bahwa peningkatan nilai K akan meningkatkan damping sehingga membuat sistem semakin stabil. Terdapat pole yang berada dekat pada sumbu vertikal (sumbu imaginer), hal ini menunjukkan bahwa komponen imaginer dari pole lebih signifikan dibandingkan dengan komponen realnya, sehingga sistem cenderung berosilasi. Ini menandakan bahwa respon sistem termasuk dalam kategori underdamped.

D.3 Root locus PID underdamped



Gambar 4.11 Root Locus Respons Underdamped

Berdasarkan root locus plot tersebut, didapatkan nilai untuk pole dan zero sebagai berikut:

Pole:

- -0.0258 + 0.0544i
- -0.0258 - 0.0544i
- -0.0522 + 0.0000i

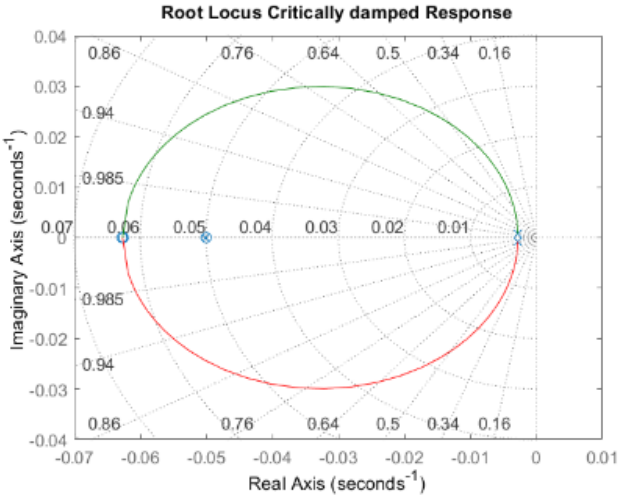
Zero:

- -0.0642
- -0.0609
- -0.0501

Adanya pole dengan nilai dalam sumbu imaginer −0.0258±0.0544i menunjukkan bahwa sistem berosilasi. Pole dengan nilai negatif menunjukkan bahwa osilasi akan meredam seiring waktu, Zero dari sistem ini semuanya real dan negative. Zero tersebut mempengaruhi respons sistem dengan membentuk bagaimana input diproses melalui fungsi transfer, meskipun tidak langsung mempengaruhi stabilitas.

Sehingga sistem ini diklasifikasikan sebagai underdamped. Hal ini disebabkan oleh sifat osilasi yang ditunjukkan oleh pole-pole ini, yang akan menyebabkan sistem berosilasi sebelum mencapai nilai akhirnya. Bagian real yang relatif kecil dari pole menunjukkan bahwa osilasi akan meredam secara perlahan. Pole real negatif tambahan pada −0.0522 lebih lanjut memastikan bahwa sistem ini stabil dan akhirnya akan menetap tanpa osilasi yang berkelanjutan.

D.4 Root locus PID critically damped



Gambar 4.12 Root Locus Respons Critically damped

Berdasarkan root locus plot tersebut, didapatkan nilai untuk pole dan zero sebagai berikut:

Pole:

- -0.0500 + 0.0000i
- -0.0028 + 0.0008i
- -0.0028 - 0.0008i

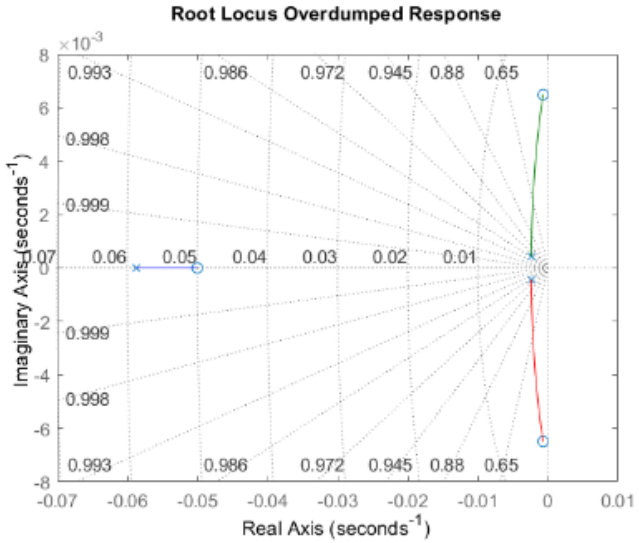
Zero:

- -0.0629 + 0.0036i
- -0.0629 - 0.0036i
- -0.0495 + 0.0000i

Pole pada −0.0500 adalah pole real yang negatif yang menunjukkan perilaku stabil tanpa adanya osilasi. Pole lainnya memiliki nilai imaginer dengan bagian imaginer yang sangat kecil dan bagian real yang kecil. Hal ini menunjukkan potensi untuk perilaku osilasi minim yang meredam sangat lambat. Zero dari sistem mempengaruhi respons tetapi tidak secara langsung mempengaruhi stabilitas atau klasifikasi redaman sistem.

Sistem critically damped ditandai oleh pole real yang berulang yang memastikan sistem kembali ke keseimbangan secepat mungkin tanpa berosilasi. Meskipun kehadiran pole terdapat bagian imaginer yang sangat kecil mungkin awalnya menunjukkan redaman rendah, perilaku dominan ditentukan oleh pole real terbesar pada −0.0500. Konfigurasi ini berarti bahwa sistem akan kembali ke keseimbangan tanpa osilasi signifikan, yang sesuai dengan karakteristik sistem yang critically damped.

D.5 Root locus PID overdamped



Gambar 4.13 Root Locus Respons Overdamped

Berdasarkan root locus plot tersebut, didapatkan nilai untuk pole dan zero sebagai berikut:

Poles:

- -0.0588 + 0.0000i
- -0.0024 + 0.0005i
- -0.0024 - 0.0005i

Zeros:

- -0.0500 + 0.0000i
- -0.0007 + 0.0065i
- -0.0007 - 0.0065i

Pole dengan nilai −0.0588 merupakan pole real negatif yang menunjukkan perilaku stabil tanpa adanya osilasi. Sementara pole lainnya menunjukkan adanya komponen osilasi. Bagian real yang sangat kecil menunjukkan bahwa osilasi akan meredam sangat perlahan, sementara bagian imajiner menunjukkan frekuensi rendah dari osilasi tersebut. Nilai zero menunjukkan bahwa input ke sistem akan memiliki beberapa karakteristik osilasi dalam respon sistem yang mempengaruhi bagaimana sistem merespons input sebelum mencapai keadaan stabil.

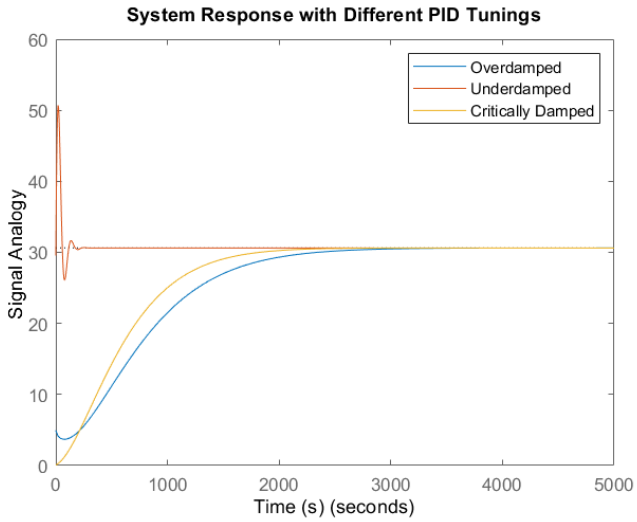
Sistem yang overdamped ditandai oleh pole-pole real negatif yang memastikan sistem kembali ke keseimbangan secara lambat tanpa adanay osilasi. Dalam sistem ini, pole dominan berada pada −0.0588, yang menunjukkan bahwa sistem akan kembali ke keseimbangan tanpa osilasi. Pole imaginer dengan bagian real yang sangat kecil sehingga tampaknya menunjukkan bahwa sistem memiliki karakteristik redaman yang tinggi, yang berarti sistem ini overdamped.

V. DISKUSI

Fungsi transfer kontrol temperatur digunakan untuk menganalisis respon sistem terhadap perubahan suhu yang diinginkan. Dari hasil analisis ini, kita dapat melihat bagaimana sistem merespons input set point dan bagaimana parameter-parameter tersebut mempengaruhi kestabilan dan kecepatan sistem dalam mencapai suhu yang diinginkan. Penerapan langkah-langkah koreksi PID pada fungsi transfer ini memungkinkan kita untuk mengatur suhu ruangan dengan presisi yang tinggi.

Fungsi transfer sensor suhu digunakan untuk menganalisis bagaimana sensor merespons perubahan suhu dalam sistem. Sensor mengukur suhu aktual sistem pendingin dan memberikan feedback ke sistem. Analisis respons sensor ini penting untuk memastikan bahwa data suhu yang diukur adalah tepat dan dapat digunakan sebagai dasar untuk perhitungan koreksi PID.

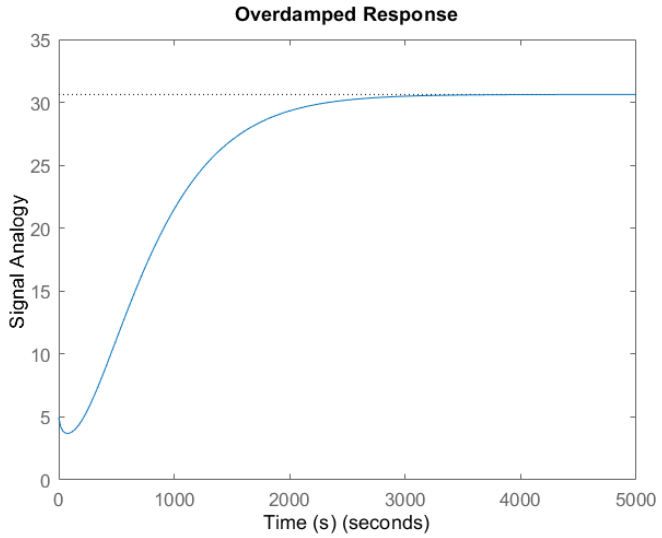
A. Respon Sistem



Gambar 5.1 Berbagai Respons Sistem dengan Pengaturan PID Berbeda

Pada grafik diatas menunjukan grafik dari ketiga jenis sistem respons yang telah dibuat. Dapat terlihat bahwa grafik warna jingga untuk underdamped mengalami osilasi, dan grafik critically damped lebih cepat mencapai set point dibandingkan dengan grafik overdamped. Hal ini dijabarkan dalam point-point di bawah berikut:

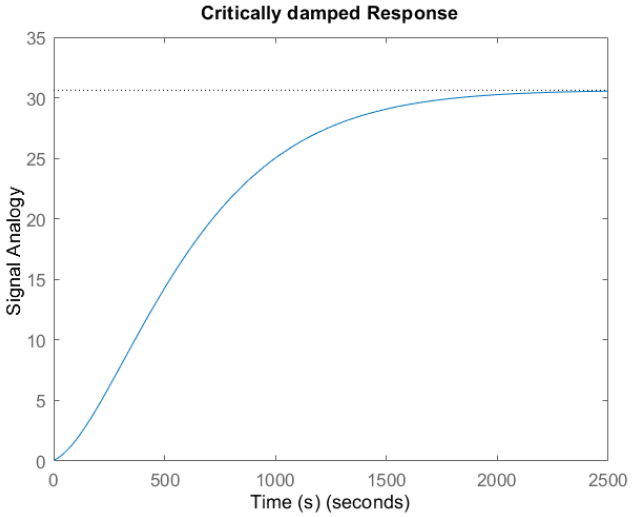
A.1 Overdamped



Gambar 5.2 Respons Sistem dengan Sinyal Overdamped

Grafik diatas menunjukan respon motor kompresor pada kondisi respon sistem overdamped. Pada respon sistem overdamped, motor kompresor akan beroperasi dengan kecepatan yang lebih lambat dibandingkan respon sistem critically damped, hal ini bertujuan untuk menghindari osilasi yang besar. Hal ini memiliki keuntungan untuk mengurangi keausan pada komponen motor namun mengkonsumsi energi yang lebih tinggi dikarenakan kompresor beroperasi lebih lama untuk mencapai kondisi yang diinginkan.

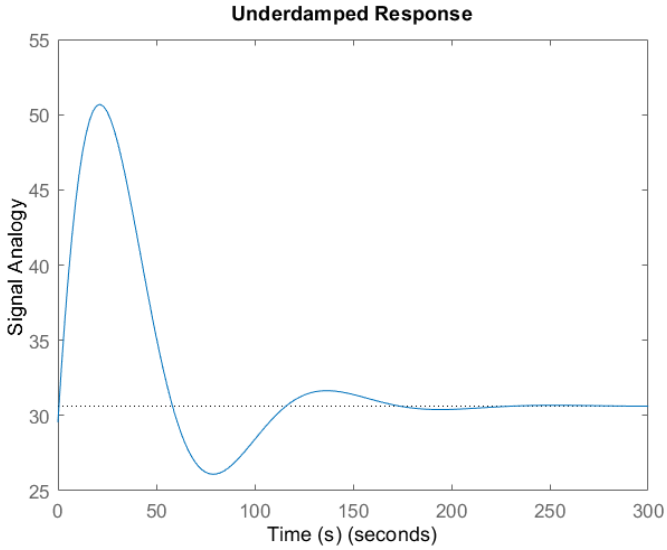
A.2 Critically Damped



Gambar 5.3 Respons Sistem dengan Sinyal Critically Damped

Grafik diatas menunjukan respon motor kompresor pada kondisi respon sistem critically damped. Pada respon sistem Critically Damped, motor kompresor akan beroperasi dengan kecepatan tertentu untuk mencapai suhu target dengan cepat tanpa osilasi. Selain itu, critically damped menyentuh titik set point lebih cepat dibandingkan overdamped. Hal ini dapat meningkatkan efisiensi energi dan mengurangi keausan pada komponen motor, karena motor tidak perlu beroperasi pada kecepatan tinggi untuk waktu yang lebih lama. Efisiensi operasi ini adalah kondisi yang paling diinginkan untuk motor kompresor, karena menghasilkan stabilitas suhu yang optimal dengan penggunaan energi yang lebih minimal.

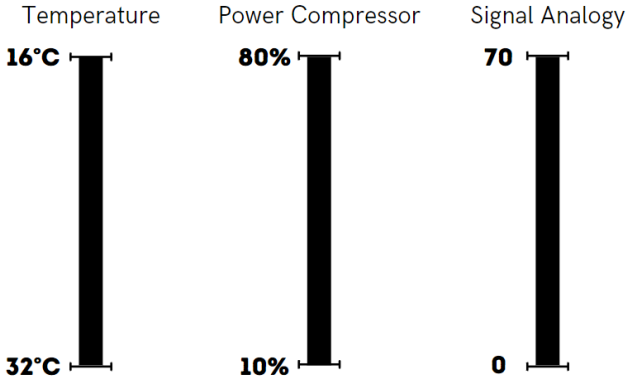
A.3 Underdamped



Gambar 5.4 Respons Sistem dengan Sinyal Underdamped

Grafik diatas menunjukan respon motor kompresor pada kondisi respon sistem underdamped. Pada respon sistem underdamped, motor kompresor akan beroperasi dengan kecepatan tinggi untuk segera menurunkan suhu hingga melebihi target (mengalami overshoot). Hal ini menyebabkan osilasi pada kecepatan motor, hal ini juga dapat meningkatkan keausan pada komponen dan mengurangi efisiensi energi pada motor kompresor. Penggunaan energi yang tidak efisien ini terjadi karena kompresor bekerja lebih keras untuk mengoreksi suhu berlebih.

B. Kinerja Sistem



Gambar 5.5 Deskripsi nilai Suhu, Tenaga Kompresor, dan Analogi Sinyal

Tabel 5.1 Rumus Konversi Analogi Terhadap Suhu dan Tenaga Kompresor

	TEMPERATURE	POWER COMPRESSOR	SIGNAL ANALOGY
TEMPERATURE	-	$\left(\frac{35}{8}(32-Temp)\right)+10$	$\frac{35}{8}(32-Temp)$
POWER COMPRESSOR	$32-\left(\frac{8}{35}(Compressor-10)\right)$	-	$(Compressor-10)$
SIGNAL ANALOGY	$32-\left(\frac{8}{35}(Signal)\right)$	$(Signal+10)$	-

Sistem pengatur suhu ruangan menghasilkan sinyal keluaran dengan respon langkah yang selalu dimulai dari 0. Berdasarkan rumus pada Tabel 5.1, dilakukan konversi untuk menentukan nilai daya dan suhu pada kompresor. 25^{o}C adalah nilai setpoint suhu yang diberikan berdasarkan reaksi sistem yang ditunjukkan pada Gambar 5.1. Misalnya, angka ini sama dengan 40,625% daya kompresor dengan nilai sinyal identik sebesar 30,625 jika diterjemahkan menggunakan rumus ini. Dengan mengetahui nilai setpoint suhu dan menerjemahkannya ke parameter yang diperlukan oleh sistem, konversi ini dapat digunakan untuk mengatur sistem kontrol suhu ruangan secara efektif.

VI. KESIMPULAN

Studi ini telah menilai dampak penggunaan kontrol PID dalam sistem yang dirancang untuk mengatur suhu ruangan. Berdasarkan hasil yang diperoleh, penggunaan kontrol PID menunjukkan peningkatan substansial dalam stabilitas dan waktu respons sistem dibandingkan dengan sistem yang tidak terkontrol. Proses manajemen suhu melalui penyetelan PID menghasilkan tiga bentuk respons yang berbeda: overdamped, critically damped, dan underdamped. Sistem overdamped menjaga stabilitas tanpa osilasi, namun memiliki waktu respons yang lebih lama untuk mencapai suhu yang diinginkan. Hal ini mengurangi keausan pada komponen motor namun mengakibatkan peningkatan konsumsi energi. Sistem yang dikalibrasi hingga critically damped mencapai suhu yang diinginkan dengan cepat dan tanpa osilasi apa pun, menghasilkan efisiensi energi yang sangat baik dan meminimalkan ketegangan pada komponen motor. Sistem yang disetel dengan underdamped mencapai suhu setpoint dengan cepat namun menunjukkan osilasi dan overshoot, yang menyebabkan peningkatan keausan komponen dan penurunan efisiensi energi.

Oleh karena itu, pemanfaatan kontrol PID dalam sistem pengaturan suhu ruangan telah menunjukkan kemanjurannya dalam meningkatkan kinerja sistem, dan kalibrasi yang akurat dapat menghasilkan hasil yang optimal berdasarkan persyaratan aplikasi tertentu. Penelitian ini memberikan kontribusi yang signifikan terhadap desain sistem kendali suhu dan menawarkan rekomendasi untuk penerapan lebih lanjut dalam skala yang lebih luas.

REFERENCES

[1] R. Friadi, “Sistem Kontrol Intensitas Cahaya, Suhu dan Kelembaban Udara Pada Greenhouse Berbasis Raspberry PI,” 2019.
[2] WHO, “Heatwaves: How to stay cool.” Accessed: Jun. 24, 2024. [Online]. Available: https://www.who.int/news-room/questions-and-answers/item/heatwaves-how-to-stay-cool
[3] W. H. Carrier, “The Control of Humidity and Temperature as applied to Manufacturing Processes and Human Comfort,” Domestic Engineering, vol. 50, 1930,,ccessed: Jun. 24, 2024. [Online]. Available: https://www.cabidigitallibrary.org/doi/full/10.5555/19302702151
[4] E. Yudaningtyas, Belajar Sistem Kontrol: Soal dan Pembahasan. Universitas Brawijaya Press. 2017. Accessed: Jun. 24, 2024. [Online]. Available:https://books.google.co.id/books?hl=id&lr=&id=UjZTDwAAQBAJ&oi=fnd&pg=PR5&dq=E.+Yudaningtyas,+Belajar+Sistem+Kontrol:+Soal+dan+Pembahasan.+Universitas+Brawijaya+Press,+2017.&ots=ibS_RU92d9&sig=iRFehFMuhPJ2w9CoGa8d35J4Mdo&redir_esc=y#v=onepage&q=E.%20Yudaningtyas%2C%20Belajar%20Sistem%20Kontrol%3A%20Soal%20dan%20Pembahasan.%20Universitas%20Brawijaya%20Press%2C%202017.&f=false
[5] A. Riyanto and M. Syafrullah, “Pemantauan Suhu Pada Sistem Pemanas Air Menggunakan Temperatur Kontrol Dengan Metode Pid Ziegler Nichols Berbasis Web,” Proceeding Seminar Nasional Sistem Informasi dan Teknologi Informasi, 2018, Accessed: Jun. 24, 2024. [Online]. Available:https://sisfotenika.stmikpontianak.ac.id/index.php/sensitek/article/view/392
[6] KEMENDIKBUD, “Hasil Pencarian - KBBI VI Daring.” Accessed: Jun. 24, 2024. [Online]. Available: https://kbbi.kemdikbud.go.id/entri/Suhu
[7] F. Arpan, D. G. Kirono, and S. Sudjarwadi, “KAJIAN METEOROLOGIS HUBUNGAN ANTARA HUJAN HARLAN DAN UNSUR-UNSUR CUACA STUDI KASUS DI STASIUN METEOROLOGI ADISUCIPTO YOGYAKARTA,” Majalah Geografi Indonesia, 2016, Accessed: Jun. 24, 2024. [Online]. Available: https://jurnal.ugm.ac.id/mgi/article/view/13268
[8] D. Setiawan, “SISTEM KONTROL MOTOR DC MENGGUNAKAN PWM ARDUINO BERBASIS ANDROID SYSTEM,” 2017, Accessed: Jun. 24, 2024. [Online]. Available: https://ejournal.uin-suska.ac.id/index.php/sitekin/article/view/4131
[9] B. Nvs and P. L. Saranya, “Water pollutants monitoring based on Internet of Things,” in Inorganic Pollutants in Water, Elsevier, 2020, pp. 371–397. doi: 10.1016/B978-0-12-818965-8.00018-4.
[10] R. A. Pratama and I. Permana, “Simulasi Permodelan Menggunakan Sensor Suhu Berbasis Arduino,” 2021, Accessed: Jun. 24, 2024. [Online]. Available:https://journal.unnes.ac.id/sju/index.php/eduel/article/view/47112
[11] M. W. Hariyanto, A. H. Hendrawan, and R. Ritzkal, “Monitoring the Environmental Temperature of the Arduino Assistance Engineering Faculty Using Telegram,” Journal of Robotics and Control (JRC), vol. 1, no. 3, 2020, doi: 10.18196/jrc.1321.
[12] S. Zhuge, “PID Control Theory,” Crystal Instruments. Accessed: Jun. 24, 2024. [Online]. Available: https://www.crystalinstruments.com/blog/2020/8/23/pid-control-theory
[13] S. B. Joseph, E. G. Dada, A. Abidemi, D. O. Oyewola, and B. M. Khammas, “Metaheuristic algorithms for PID controller parameters tuning: review, approaches and open problems,” Heliyon, vol. 8, no. 5, p. e09399, May 2022, doi: 10.1016/j.heliyon.2022.e09399.
[14] R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, “A review of PID control, tuning methods and applications,” Int J Dyn Control, vol. 9, no. 2, pp. 818–827, Jun. 2021, doi: 10.1007/s40435-020-00665-4.
[15] O. A. Somefun, K. Akingbade, and F. Dahunsi, “The dilemma of PID tuning,” Annu Rev Control, vol. 52, pp. 65–74, 2021, doi: 10.1016/j.arcontrol.2021.05.002.
[16] J. Aszhar, I. Setiawan, and R. Medriati, “Method for Accelerating Equilibrium in Perfectly Damped Brownian Motion with Harmonic Potential,” 2024, Accessed: Jun. 24, 2024. [Online]. Available: https://journalfkipunipa.org/index.php/kpej/article/view/485
[17] I. A. Resen, R. I. K. Zaki, and H. K. Risan, “Graphical Comparison of Critically Damped Versus Overdamped Free Vibration of Single Degree of Freedom Structure,” Journal of Southwest Jiaotong University, vol. 54, no. 5, 2019, doi: 10.35741/issn.0258-2724.54.5.13.
[18] N. Elisa, I. Setiawan, and D. Hamdani, “ENERGI PENGGERAK UNTUK MEMPERCEPAT KESETIMBANGAN GERAK BROWN TEREDAM SEBAGIAN (UNDERDAMPED),” JURNAL INOVASI DAN PEMBELAJARAN FISIKA, 2023, Accessed: Jun. 24, 2024. [Online]. Available: https://jipf.ejournal.unsri.ac.id/index.php/jipf/article/view/244

[19] S. K. Pandey, K. Veeranna, B. Kumar, and K. U. Deshmukh, “A Robust Auto-tuning Scheme for PID Controllers,” in IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, IEEE, Oct. 2020, pp. 47–52. doi: 10.1109/IECON43393.2020.9254382.

LAMPIRAN

-- Start Matlab Code for The System --

```
% Define parameters
M   = 200;        % Thermal mass including air and occupants (kg)
C   = 1005;       % Specific heat capacity of air (J/kg·K)
m   = 0.1;        % Mass flow rate of air (kg/s)
Cp  = 1005;       % Specific heat capacity of air (J/kg·K)
k   = 10;         % Heat transfer coefficient (W/m²·K)
A   = 30;         % Surface area for heat exchange (m²)
X   = 0.3;        % Wall thickness (m)
% Sensor parameters
m_sen = 0.01;     % Mass of sensor (kg)
C_sen = 500;      % Specific heat capacity of sensor (J/kg·K)
h_conv = 25;      % Convection heat transfer coefficient (W/m²·K)
A_sen = 0.01;     % Surface area of sensor (m²)
% Transfer function for Temperature Control Plant
num_plant = 1;
den_plant = [(M * C), (m * Cp + (k * A) / X)];
plant_tf = tf(num_plant, den_plant);
% Transfer function for Sensor
num_sensor = 1;
den_sensor = [(m_sen * C_sen) / (h_conv * A_sen), 1];
sensor_tf = tf(num_sensor, den_sensor);
```

Open Loop Plant

```
% Open-loop transfer function (Plant only)
open_loop_tf = plant_tf
figure;
step(open_loop_tf);
title('Open Loop Plant Step Response');
hold off;
figure;
rlocusplot(open_loop_tf);
grid on;
title('Root Locus Open Loop Plant');
hold off;
```

Closed Loop System Without Controller

```
closed_loop_sys = feedback(open_loop_tf, sensor_tf)
figure;
step(closed_loop_sys);
title('Closed Loop Plant Step Response without Controller');
hold off;
stepinfo(closed_loop_sys)
figure;
rlocusplot(closed_loop_sys);
grid on;
title('Root Locus Closed Loop Plant without Controller');
hold off;
```

Close Loop System (First Step Tuning Method Ziegler Nichols)

```
Kp = 45000; % Signal Oscilated
Ki = 0;
Kd = 0;
pid_controller = pid(Kp, Ki, Kd);
% Closed-loop transfer function
closed_loop_sys = feedback(series(pid_controller, open_loop_tf), sensor_tf);
% Step response
figure;
step(closed_loop_sys);
title('Initial PID Tuning - First Step Ziegler Nichols');
hold off;
stepinfo(closed_loop_sys)
figure;
rlocusplot(closed_loop_sys);
grid on;
title('Root Locus for First Step Ziegler Nichols');
hold off;
```

*Dari hasil setting awal Kp, didapatkan Kcr = 45000 dan f = 15.661 mHz atau Pcr = 63.852 detik; Sehingga:

```
% Measure Ku and Pu from the oscillations
Kcr = 45000;
Pcr = 63.852; % second
% Calculate PID parameters using Ziegler-Nichols or Tyreus-Luyben
Kp = 0.6 * Kcr
Kp = 27000
Ti = 0.5 * Pcr;
Td = 0.125 * Pcr;
% Convert to PID gains
Ki = Kp / Ti
Ki = 845.7057
Kd = Kp * Td
Kd = 2.1550e+05
pid_controller = pid(Kp, Ki, Kd);
% Closed-loop transfer function
```

```
closed_loop_sys = feedback(series(pid_controller,
open_loop_tf), sensor_tf);
figure;
step(closed_loop_sys);
title('Final Tuning by Ziegler-Nichols');
xlabel('Time (s)');
ylabel('Temperature (°C)');
hold off;

% Step response information
step_info = stepinfo(closed_loop_sys);
disp('Step Response Information:');
disp(step_info);
figure;
rlocusplot(closed_loop_sys);
grid on;
title('Root Locus Final Tuning by Ziegler-
Nichols');
hold off;
```

Set Damping Ratio to 0 < zeta < 1 (Underdamped Response)

```
% Definisikan parameter sistem dan PID
Kp = 27000;
Ki = 845.7057;
Kd = 2.1550e+05;
% Transfer function dari PID controller
pid_controller = pid(Kp, Ki, Kd);
closed_loop_sys = feedback(series(pid_controller,
open_loop_tf), sensor_tf);
% Poles dari closed-loop system
poles = pole(closed_loop_sys);
% Hitung damping ratio dan natural frequency
[zeta, omega_n] = damp(closed_loop_sys);
% Tampilkan hasil
disp('Poles of the closed-loop system:');
disp(poles);
disp('Damping Ratio:');
disp(zeta);
disp('Natural Frequency:');
disp(omega_n);
% Identifikasi mode dominan (poles dengan bagian
real terkecil)
[~, idx] = min(abs(real(poles)));
dominant_pole = poles(idx);
dominant_zeta = zeta(idx);
dominant_omega_n = omega_n(idx);
% Tampilkan mode dominan
disp('Dominant Pole:');
Dominant Pole:
disp(dominant_pole);
disp('Dominant Damping Ratio:');
disp(dominant_zeta);
disp('Dominant Natural Frequency:');
Dominant Natural Frequency:
disp(dominant_omega_n);
% Analyze initial response
info = stepinfo(closed_loop_sys);
disp('Initial Step Response Information:');
disp(info);
% Iteratively adjust PID parameters to achieve
underdamped response
for i = 1:100
    % Analyze current response
    info = stepinfo(closed_loop_sys);
    damping_ratio = dominant_zeta;
    natural_frequency = dominant_omega_n;
    % Adjust Kp, Ki, and Kd to maintain
underdamped response
    if damping_ratio >= 1
        Kp = Kp * 1.1; % Increase Kp
        Ki = Ki * 1.1; % Increase Ki
        Kd = Kd * 1.1; % Increase Kd
    elseif damping_ratio < 0.1
        Kp = Kp * 0.9; % Decrease Kp
        Ki = Ki * 0.9; % Decrease Ki
        Kd = Kd * 0.9; % Decrease Kd
    end

    % Create updated PID controller
    pid_controller = pid(Kp, Ki, Kd);

    % Closed-loop transfer function with updated
PID
    closed_loop_sys =
feedback(series(pid_controller, plant_tf),
sensor_tf);

    % Step response with updated PID
    figure;
    step(closed_loop_sys);
    title(['Step Response with Updated PID
(Iteration ', num2str(i), ')']);
    xlabel('Time (s)');
    ylabel('Temperature (°C)');
    grid on;

    % Analysis of updated response
    info = stepinfo(closed_loop_sys);
    disp(['Updated Step Response Information
(Iteration ', num2str(i), '):']);
    disp(info);

    % Check if the system is underdamped with
desired characteristics
```

```
    if damping_ratio > 0 && damping_ratio < 1
        disp('System is underdamped.');
        break;
    end

    % Pause to observe the response before next
iteration
    pause(2);
end

Underdamped = closed_loop_sys;
```

Set Damping Ratio to 1 (Critically damped)

```
% Definisikan parameter sistem dan PID
Kp = 27000;
Ki = 845.7057;
Kd = 2.1550e+05;
% Transfer function dari PID controller
pid_controller = pid(Kp, Ki, Kd);
closed_loop_sys = feedback(series(pid_controller,
open_loop_tf), sensor_tf);
% Closed-loop transfer function
G_cl = closed_loop_sys;
% Poles dari closed-loop system
poles = pole(G_cl);
% Hitung damping ratio dan natural frequency
[zeta, omega_n] = damp(G_cl);
% Tampilkan hasil
disp('Poles of the closed-loop system:');
disp(poles);
disp('Damping Ratio:');
disp(zeta);
disp('Natural Frequency:');
disp(omega_n);
% Identifikasi mode dominan (poles dengan bagian
real terkecil)
[~, idx] = min(abs(real(poles)));
dominant_pole = poles(idx);
dominant_zeta = zeta(idx);
dominant_omega_n = omega_n(idx);
% Tampilkan mode dominan
disp('Dominant Pole:');
disp(dominant_pole);
disp('Dominant Damping Ratio:');
disp(dominant_zeta);
disp('Dominant Natural Frequency:');
disp(dominant_omega_n);
% Analyze initial response
info = stepinfo(closed_loop_sys);
disp('Initial Step Response Information:');
Initial Step Response Information:
disp(info);
% Iteratively adjust PID parameters
for i = 1:100
    % Analyze current response
    info = stepinfo(closed_loop_sys);

    % Adjust Kp, Ki, and Kd to increase damping
and reduce overshoot
    if info.Overshoot > 0
        Kp = Kp * 0.9; % Decrease Kp by 10%
        Ki = Ki * 0.9; % Decrease Ki by 10%
        Kd = Kd * 0.9; % Decrease Kd by 10%
    else
        break; % Break the loop if overshoot is
zero
    end

    % Create updated PID controller
    pid_controller = pid(Kp, Ki, Kd);

    % Closed-loop transfer function with updated
PID
    closed_loop_sys =
feedback(series(pid_controller, plant_tf),
sensor_tf);

    % Step response with updated PID
    figure;
    step(closed_loop_sys);
    title(['Step Response with Updated PID
(Iteration ', num2str(i), ')']);
    xlabel('Time (s)');
    ylabel('Output');
    grid on;

    % Analysis of updated response
    info = stepinfo(closed_loop_sys);
    disp(['Updated Step Response Information
(Iteration ', num2str(i), '):']);
    disp(info);

    % Check if the system is overdamped
    if info.Overshoot == 0 && info.PeakTime > 0
        disp('System is Critically damped.');
        break;
    end

    % Pause to observe the response before next
iteration
    pause(2);
end
Criticallydamped = closed_loop_sys;
```

Set Damping Ratio more than 1 (Overdamped)

```
% Definisikan parameter sistem dan PID
```

```
Kp = 27000;
Ki = 845.7057;
Kd = 2.1550e+05;
% Transfer function dari PID controller
pid_controller = pid(Kp, Ki, Kd);
closed_loop_sys = feedback(series(pid_controller,
open_loop_tf), sensor_tf);
% rlocusplot(closed_loop_sys)
% grid on;
% title('Root Locus Overdamped Response');
% Closed-loop transfer function
G_cl = closed_loop_sys;
% Poles dari closed-loop system
poles = pole(G_cl);
% Hitung damping ratio dan natural frequency
[zeta, omega_n] = damp(G_cl);
% Tampilkan hasil
disp('Poles of the closed-loop system:');
disp(poles);
disp('Damping Ratio:');
disp(zeta);
disp('Natural Frequency:');
disp(omega_n);
% Identifikasi mode dominan (poles dengan bagian
real terkecil)
[~, idx] = min(abs(real(poles)));
dominant_pole = poles(idx);
dominant_zeta = zeta(idx);
dominant_omega_n = omega_n(idx);
% Tampilkan mode dominan
disp('Dominant Pole:');
disp(dominant_pole);
disp('Dominant Damping Ratio:');
disp(dominant_zeta);
disp('Dominant Natural Frequency:');
disp(dominant_omega_n);
% Analyze initial response
info = stepinfo(closed_loop_sys);
disp('Initial Step Response Information:');
disp(info);
% Iteratively adjust PID parameters
for i = 1:100
    % Analyze current response
    info = stepinfo(closed_loop_sys);

    % Adjust Kp, Ki, and Kd to increase damping
and reduce overshoot
    if info.Overshoot > 0
        Kp = Kp * 0.7; % Decrease Kp by 30%
        Ki = Ki * 0.7; % Decrease Ki by 30%
        Kd = Kd * 0.9; % Decrease Kd by 10%
    else
        break; % Break the loop if overshoot is
zero
    end

    % Create updated PID controller
    pid_controller = pid(Kp, Ki, Kd);

    % Closed-loop transfer function with updated
PID
    closed_loop_sys =
feedback(series(pid_controller, plant_tf),
sensor_tf);

    % Step response with updated PID
    figure;
    step(closed_loop_sys);
    title(['Step Response with Updated PID
(Iteration ', num2str(i), ')']);
    xlabel('Time (s)');
    ylabel('Output');
    grid on;

    % Analysis of updated response
    info = stepinfo(closed_loop_sys);
    disp(['Updated Step Response Information
(Iteration ', num2str(i), '):']);
    disp(info);

    % Check if the system is overdamped
    if info.Overshoot == 0 && info.PeakTime > 0
        disp('System is overdamped.');
        break;
    end

    % Pause to observe the response before next
iteration
    pause(2);
end
Overdamped = closed_loop_sys;
```

Final Result

```
figure;
hold on;
% Overdamped Response
step(Overdamped);
% Underdamped Response
step(Underdamped);
% Critically Damped Response
step(Criticallydamped);
title('System Response with Different PID
Tunings');
xlabel('Time (s)');
ylabel('Temperature (°C)');
```

```
legend('Overdamped', 'Underdamped', 'Critically
Damped');
hold off;

% Overdamped Response
step(Overdamped);
title('Overdamped Response');
xlabel('Time (s)');
ylabel('Temperature (°C)');
hold off;

info = stepinfo(Overdamped);
disp(['Step Response Information:']);
disp(info);
rlocusplot(Overdamped)
grid on;
title('Root Locus Overdamped Response');

% Underdamped Response
step(Underdamped);
title('Underdamped Response');
xlabel('Time (s)');
ylabel('Temperature (°C)');
hold off;

info = stepinfo(Underdamped);
disp(['Step Response Information:']);
disp(info);
rlocusplot(Underdamped)
grid on;
title('Root Locus Underdamped Response');

% Critically damped Response
step(Criticallydamped);
title('Critically damped Response');
xlabel('Time (s)');
ylabel('Temperature (°C)');
hold off;

info = stepinfo(Criticallydamped);
disp(['Step Response Information:']);
disp(info);
rlocusplot(Criticallydamped)
grid on;
title('Root Locus Critically damped Response');
```