

Rancang Bangun Miniatur Hvac Berbasis Pid Cascade Menggunakan Mikroprosesor

Labib Arga Makarim¹, Febi Ariefka Septian Putra¹, Giga Verian Pratama¹

¹Jurusan Teknik Elektro, Politeknik Negeri Bandung, Bandung 40012

E-mail : labib.arga.toi21@polban.ac.id

E-mail : febi.ariefka@polban.ac.id

E-mail : giga.verian@polban.ac.id

ABSTRAK

Sistem HVAC (*Heating, Ventilation, and Air Conditioning*) digunakan untuk mengatur suhu dan kelembapan demi menciptakan kenyamanan termal dalam ruangan. Pada penelitian ini, dirancang dibangun sebuah Miniatur HVAC skala kecil menggunakan algoritma kendali PID *Cascade* yang dijalankan pada Raspberry pi 4. PID *Cascade* dipilih untuk meningkatkan performa sistem dalam merespons gangguan dan perubahan beban termal, dibandingkan kontrol PID tunggal. Pengujian dilakukan untuk membandingkan dua metode tuning PID, yaitu Ziegler-Nichols 1 dan *trial and error*. Hasil menunjukkan bahwa metode *trial and error* memberikan respon sistem yang lebih cepat, stabil, dan akurat, dengan *error steady-state* 0,83%, tanpa *overshoot*, serta waktu *rise* dan *settling* yang memenuhi target performa. Miniatur ini menunjukkan potensi sebagai solusi efisien untuk pengendalian HVAC skala menengah dengan dua variabel fisik yang saling terkait.

Kata Kunci

HVAC, PID *Cascade*, Raspberry Pi, Suhu, Kelembapan

HVAC (Heating, Ventilation, and Air Conditioning) systems are used to regulate temperature and humidity to create indoor thermal comfort. In this research, a small-scale HVAC prototype is designed using a PID Cascade control algorithm running on a Raspberry pi 4. PID Cascade is chosen to improve system performance in responding to disturbances and thermal load changes, compared to a single PID control. Tests were conducted to compared two PID tuning methods, namely Ziegler-Nichols 1 and Trial and Error method provides a faster, more stable and accurate system response, with a steady-state error of 0,83%, no Overshoot, and rise and settling times that meet performance targets. This prototype shows potential as an efficient solution for medium-scale HVAC control with two interrelated physical variables.

Keywords

HVAC, PID Cascade, Raspberry Pi, Temperature, Humidity

1. PENDAHULUAN

Heating, Ventilation, Air Conditioning (HVAC) adalah sistem yang berfungsi untuk mengelola suhu, kelembapan dan kualitas udara dalam suatu ruangan atau bangunan. Umumnya sistem pendingin udara dikendalikan oleh pengontrol PID *loop* tunggal. [1]. Untuk mengatasi kegagalan kontrol *PI* tradisional, kami akan menggunakan kontrol *Cascade*. Kontrol *Cascade* sangat berguna dalam mengurangi efek gangguan beban yang bergerak melalui sistem kontrol secara perlahan. *Loop* dalam memiliki efek mengurangi lag di *loop* luar, menghasilkan sistem *Cascade*. [2], [3]Telah dilakukannya penelitian bertujuan untuk mengembangkan sistem kendali HVAC berbasis PID *Cascade* menggunakan Raspberry Pi guna meningkatkan kenyamanan termal ruangan [4], [5], [6]. Sistem ini menggunakan sensor lingkungan yang terhubung secara nirkabel untuk mengukur suhu, kelembapan, dan kualitas udara [7], [8], [9]. Kendali PID *Cascade* diterapkan dengan *loop* dalam yang mengatur laju aliran udara atau kecepatan kipas, serta *loop* luar yang mengendalikan suhu ruang berdasarkan indeks kenyamanan termal (PMV) [10][11]. Parameter PID akan di tuning menggunakan algoritma genetika untuk meningkatkan performa sistem. Dengan

sistem ini, HVAC dapat beroperasi lebih efisien dan adaptif terhadap kondisi lingkungan, sehingga meningkatkan kenyamanan pengguna [12], [13]. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sebuah sistem HVAC (*Heating, Ventilation, and Air Conditioning*) dalam bentuk Miniatur yang dilengkapi dengan teknologi pengendalian otomatis berbasis PID *Cascade* dan mikroprosesor. Sistem HVAC yang dirancang memungkinkan untuk menjaga suhu pada rentang 24°C dan kelembapan antara 70% RH secara bersamaan. Rentang ini dipilih agar kondisi ruang kerja tetap ideal dan kualitas udara dan kelembapan tetap terjaga.

2. METODOLOGI

Miniatur HVAC dikembangkan dengan Raspberry Pi 4 sebagai unit kontrol, sensor DHT22 untuk membaca suhu dan kelembapan, serta actuator berupa peltier, PTC Heater, Fan DC, dan Fog Maker. Pengujian dilakukan dalam dua tahap : *Open Loop* untuk memperoleh parameter sistem (L dan T) dan *Close Loop* untuk mengevaluasi performa PID *Cascade* dengan dua metode *tuning* (Ziegler–Nichlos 1 dan *Trial and Error*). Parameter performa sistem yang diuji meliputi *rise time* (Tr), *settling time* (Ts), *error steady-state* (Ess), dan *overshoot* (Os).

2.1 PID Controller

Pengendali umpan balik seperti *proportional-integral-derivative* (PID) banyak digunakan dalam berbagai aplikasi kendali. Karena tujuan utamanya adalah menjaga proses tetap sedekat mungkin dengan titik setel (*set point*) yang diinginkan sambil meminimalkan kesalahan, pengendali PID menganalisis selisih antara variabel proses masukannya dan nilai target tersebut sebelum mengambil tindakan yang sesuai untuk mengurangi kesalahan proses tersebut sekecil mungkin. Tindakan diambil berdasarkan kesalahan saat ini, penjumlahan kesalahan, dan laju perubahan kesalahan. Komponen Proporsional merespons berdasarkan kesalahan saat ini.

Dengan menggunakan ketiga parameter ini secara bersamaan, proses dapat disesuaikan secara tepat untuk mencapai hasil yang diinginkan. Persamaan PID disajikan di bawah ini [14], [15].

$$u(t) = K_p \cdot e(t) + K_I \cdot \int_{t_0}^t e(\tau) \cdot d\tau + K_D \frac{de(t)}{dt} \quad (1)$$

Pengendali PID menghasilkan *Process Variable (PV)*, yaitu hasil penjumlahan dari ketiga komponen korektifnya, yaitu Proporsional, Integral, dan Turunan.

$$PV(t) = P_{out} + I_{out} + D_{out} \quad (2)$$

Dimana,

$u(t)$ = output kontroler

$e(t)$ = selisih antara *set point* dan nilai *error*

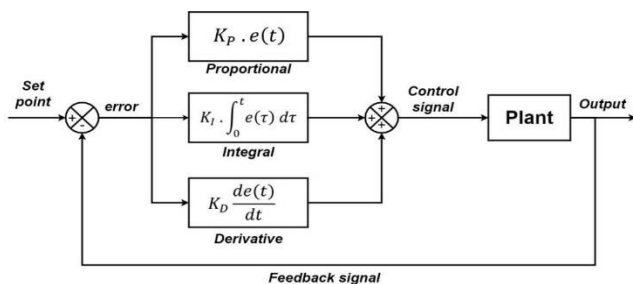
$\tau \int_0^t e(\tau) d\tau$ = integral *error* dari waktu 0 hingga t

$\frac{de(t)}{dt}$ = turunan *error* terhadap waktu

K_p = gain proporsional

K_I = gain integral (K_p/T_i)

K_D = gain *derivative* ($K_p \cdot T_d$)



Gambar 1 Diagram Blok Kendali PID

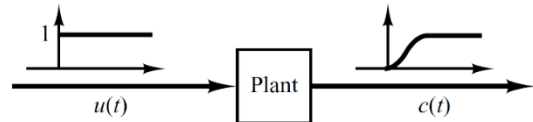
Terdapat berbagai metode untuk menentukan parameter-parameter yang optimal untuk kontroler PID. Metode-metode yang paling umum digunakan diantaranya yaitu: pemodelan matematika, *trial and error*, pendekatan respon frekuensi, otomatisasi (*auto-tuning*), Coehen-Coon, dan Ziegler-Nichols 1.

Ziegler dan Nichols mengusulkan suatu metode tuning parameter PID (gain proporsional K_p , waktu integral T_i ,

dan waktu derivatif T_d) karakteristik respon transien dari suatu plant. Karakteristik respon ini dapat diperoleh melalui uji coba langsung plant yang akan dikontrol. Metode *tuning* PID metode Ziegler-Nichols 1.

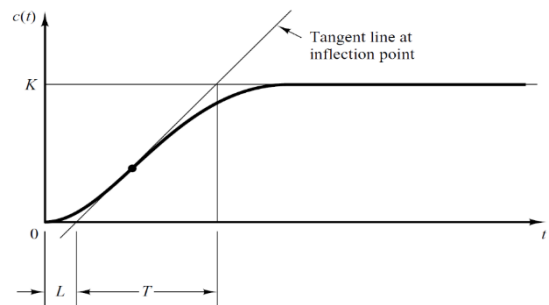
1. Metode Ziegler-Nichols 1

Pada metode ini, respon plant terhadap *input unit-step* diperoleh melalui eksperimen. Metode ini berlaku jika respon terhadap input step menunjukkan kurva berbentuk S, seperti pada Gambar 2.2 di bawah. Kurva respon *step* tersebut dapat dibuat secara eksperimen atau dari simulasi dinamis plant.



Gambar 2 Respon Plant dengan Input Unit-Step

Kurva berbentuk S memiliki ciri dua konstanta utama, yaitu waktu tunda atau *delay* (L) dan konstanta waktu (T). Penentuan dua konstanta ini dilakukan dengan menggambar garis singgung di titik infleksi kurva S dan mengamati titik di mana garis singgung tersebut memotong sumbu waktu dan garis $c(t) = K$, seperti yang diilustrasikan pada Gambar 2.3 dibawah ini.



Gambar 3 Respon Kurva S

Ziegler dan Nichols mengusulkan bahwa untuk mengatur nilai K_p , T_i , T_d , berdasarkan formula pada Tabel 2.2 di bawah.

Tabel 1. Aturan Tuning PID Ziegler-Nichols Metode Pertama

Jenis Pengendali	K_p	T_i	T_d
P	T/L	∞	0
PI	$0,9T/L$	$0,3L$	0
PID	$1,2T/L$	$2L$	$0,5L$

Berikut ini cara aturan pertama Ziegler-Nichols 1 digunakan untuk melakukan tuning parameter PID.

$$G_C(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (3)$$

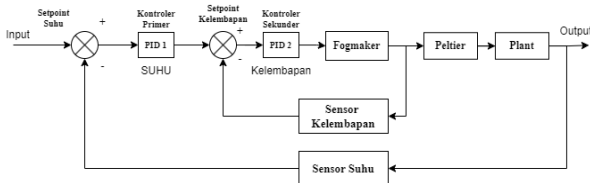
$$G_C(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right)$$

$$G_C(s) = K_P \left(1 + \frac{1}{T_I s} + T_D s \right)$$

$$K_i = \frac{K_p}{T_i}$$

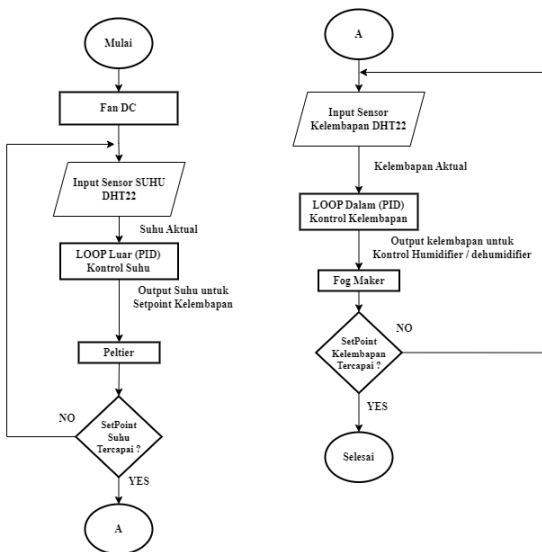
$$K_d = K_p \cdot T_d$$

2.2 Perancangan Model Sistem



Gambar 4 Diagram Blok Sistem

Diagram blok yang ditampilkan menggambarkan sistem kontrol PID dengan konfigurasi kendali *Cascade* yang berfungsi untuk mengatur suhu dan kelembapan. Sistem ini melibatkan dua pengendali PID, di mana PID 1 bertindak sebagai kontroler utama yang menerima *setpoint* suhu dan menghasilkan *setpoint* kelembapan, sedangkan PID 2 sebagai kontroler sekunder mengendalikan *humidifier* sesuai dengan *setpoint* kelembapan yang diberikan agar membantu untuk mendinginkan dari ruangan atas dasar input yang diberikan oleh PID 1. Sensor suhu dan kelembapan menyediakan umpan balik ke masing-masing kontroler untuk memastikan kestabilan sistem. Keluaran dari pendingin (*peltier*) selanjutnya dikontrol oleh unit *humidifier* agar mencapai kondisi yang diinginkan. Dengan konfigurasi ini, sistem menjadi lebih responsif dan lebih tahan terhadap gangguan eksternal yang memengaruhi suhu dan kelembapan.

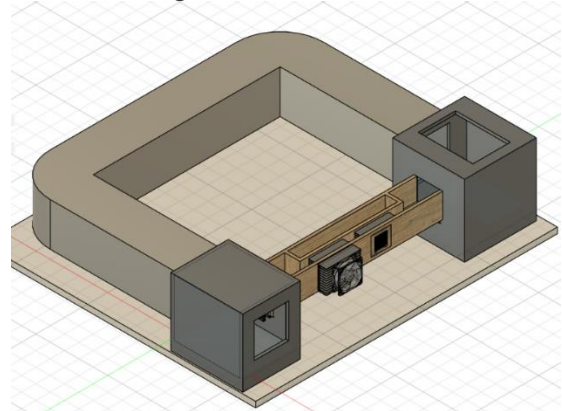


Gambar 5 Diagram Alir Sistem

Diagram alir sistem menggambarkan kendali otomatis berbasis PID *Cascade* untuk mengatur suhu dan kelembapan menggunakan sensor DHT22. Proses dimulai

dengan pengaturan suhu melalui kipas DC hingga mendekati *setpoint*, lalu sistem menghitung error suhu menggunakan PID luar. Setelah suhu stabil, sistem membaca data kelembapan dan mengaktifkan *loop* PID dalam untuk mengontrol *humidifier* jika nilai kelembapan belum sesuai. Kedua *loop* bekerja secara berurutan agar tidak saling mengganggu dan memastikan kestabilan lingkungan secara bertahap. Sistem berhenti ketika suhu dan kelembapan telah mencapai *setpoint*. Pendekatan ini cocok diterapkan pada ruang kerja atau gedung perkantoran yang membutuhkan kontrol iklim presisi dalam skala miniatur.

2.3 Perancangan Miniatur



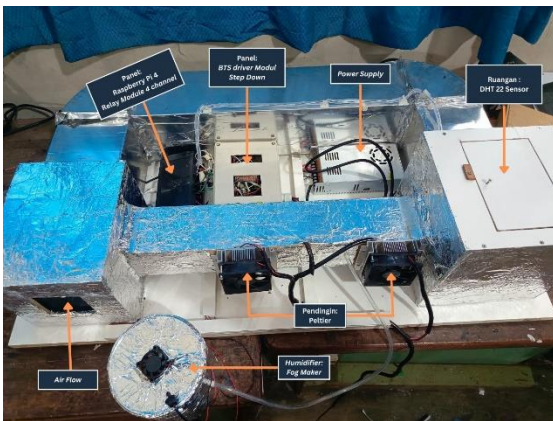
Gambar 6 Perancangan Miniatur 3D

Perancangan model 3D Miniatur HVAC dilakukan secara komprehensif dengan mempertimbangkan aspek fungsionalitas, ergonomi, dan efisiensi termal. Proses desain diawali dengan pemodelan ruang utama berukuran 50×50×50cm menggunakan perangkat lunak Fusion 360, yang dirancang khusus untuk mensimulasikan kondisi ruang kerja. Dinding ruang dibuat dari Kayu setebal 5 mm untuk memungkinkan observasi visual sekaligus memberikan insulasi termal dasar. Pada bagian atas ruang, dirancang saluran udara berbentuk *diffuser* dengan sudut 45° untuk mendistribusikan aliran udara dari kipas DC.

Sensor DHT22 diposisikan di titik tengah ruang dengan *holder* khusus yang meminimalkan gangguan aliran udara, menjamin akurasi pengukuran suhu ($\pm 0.5^\circ\text{C}$). Sistem pendingin terdiri dari modul Peltier berdaya 60W yang terintegrasi dengan *heat sink* aluminium dan kipas pendingin 12V, dipasang pada dinding samping dengan orientasi tertentu untuk optimasi pertukaran panas. Di sisi berlawanan, PTC *heater* 120°C dipasang dengan pelat distribusi panas untuk memastikan pemanasan merata.

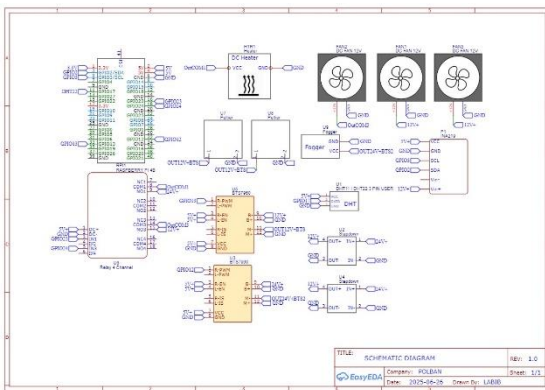
Untuk kontrol kelembapan, *mist maker* ultrasonik 24V ditempatkan dalam chamber terpisah yang terhubung dengan ruang utama melalui saluran, mengurangi risiko kondensasi berlebihan. Seluruh komponen elektronik termasuk Raspberry Pi 4 dan modul relay 4-channel diintegrasikan dalam panel kontrol eksternal dengan sistem kabel management yang terorganisir. Desain ini telah melalui simulasi CFD (*Computational Fluid Dynamics*) untuk memvalidasi distribusi suhu dan kelembapan, menunjukkan unjuk kerja yang stabil dengan variasi maksimal $\pm 0.8^\circ\text{C}$ di seluruh volume ruang. Miniatur akhir

dicetak 3D menggunakan material PLA dengan ketebalan dinding 8mm untuk keseimbangan antara kekuatan struktural dan bobot keseluruhan.



Gambar 7 Realisasi Plant

2.4 Schematic Diagram



Gambar 8 Schematic Diagram

Gambar di atas merupakan diagram skematik sistem pengendali suhu dan kelembapan berbasis Raspberry Pi 4, yang dirancang untuk mengontrol perangkat-perangkat seperti pemanas (*heater*), kipas DC, *fog maker*, dan Peltier. Sensor DHT22 digunakan untuk mengukur suhu dan kelembapan, yang kemudian datanya diolah oleh Raspberry Pi untuk mengaktifkan aktuator melalui modul relay 4 channel dan *driver* motor BTS7960. *Fan DC* 12V diaktifkan melalui relay untuk mengatur sirkulasi udara, sementara Peltier digunakan untuk pendinginan, dan heater untuk pemanasan ruangan. Sistem juga menyertakan *fog maker* untuk menambah kelembapan dan sensor INA219 untuk pemantauan arus dan tegangan. Pengaturan suplai daya dilakukan melalui beberapa modul stepdown yang menurunkan tegangan 24V menjadi 12V untuk kebutuhan komponen tertentu. Diagram ini mencerminkan sistem kontrol lingkungan otomatis yang terintegrasi untuk aplikasi seperti ruangan kerja.

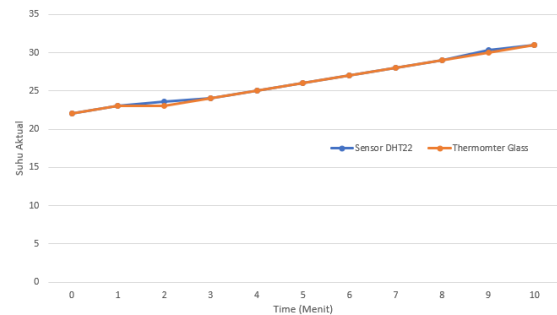
3. HASIL DAN ANALISA DATA

3.1 Pengujian Hardware

mengevaluasi kinerja dan akurasi sensor dalam mengukur suhu ruangan secara *real-time*. Pengujian ini dilakukan untuk memastikan bahwa sensor DHT 22 dapat

memberikan data yang stabil, konsisten, dan sesuai dengan kondisi lingkungan sebenarnya. Melalui proses pengujian ini, dapat diketahui *respons* sensor terhadap perubahan suhu dari waktu ke waktu, serta validitas hasil pembacaan yang akan digunakan dalam sistem atau penelitian lebih lanjut. Bab ini menjadi dasar penting dalam menilai kelayakan sensor untuk diaplikasikan dalam proyek *monitoring* suhu atau sistem kendali berbasis suhu.

3.1.1 Pengujian Respon Suhu

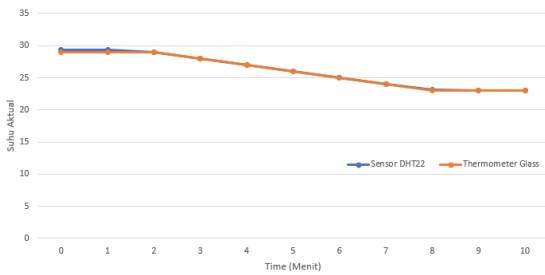


Gambar 9 Respon Sensor Menggunakan Heater

Tabel 2 Perbandingan Kenaikan Suhu pada Sensor dan Thermometer Glass

Waktu (menit)	Suhu Sensor	Suhu Thermometer
0	22°C	22°C
1	23°C	23°C
2	23°C	23°C
3	24°C	24°C
4	25°C	25°C
5	26°C	26°C
6	27°C	27°C
7	28°C	28°C
8	29°C	29°C
9	30°C	30°C
10	31°C	31°C

Grafik di atas menggambarkan perubahan suhu ruangan yang terdeteksi oleh sensor DHT22 dalam rentang waktu 0 hingga 10 Menit. Terlihat bahwa suhu mengalami kenaikan secara bertahap, dimulai dari 22°C pada awal pengamatan dan mencapai 31°C di akhir. Kenaikan suhu ini menunjukkan pola yang cukup linier dan stabil, yang mengindikasikan bahwa sensor berfungsi secara akurat dalam merekam perubahan suhu. Garis merah pada grafik menunjukkan tren suhu secara keseluruhan, sementara titik-titik biru merepresentasikan data hasil pembacaan sensor yang konsisten terhadap tren tersebut. Secara keseluruhan, grafik ini menunjukkan bahwa selama periode pengamatan, suhu ruangan mengalami peningkatan yang teratur.



Gambar 10 Respon Sensor Menggunakan Peltier

Tabel 3 Perbandingan Penurunan Suhu pada Sensor dan Thermomter Glass

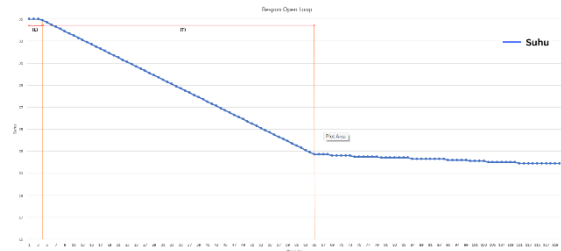
Waktu (menit)	Suhu Sensor	Suhu Thermometer
0	29,3°C	29°C
1	29,3°C	29°C
2	29°C	29°C
3	28°C	28°C
4	27°C	27°C
5	26°C	26°C
6	25°C	25°C
7	24°C	24°C
8	23,2°C	23°C
9	23°C	23°C
10	23°C	23°C

Grafik di atas menunjukkan penurunan suhu ruangan yang terukur menggunakan sensor DHT22 dalam rentang waktu dari 0 hingga 10 Menit. Awalnya, suhu berada di kisaran 29,3°C dan secara bertahap menurun hingga mencapai 23°C. Penurunan suhu ini berlangsung secara perlahan dan konsisten, mengindikasikan adanya proses pendinginan yang stabil di dalam ruangan. Titik-titik pada grafik mewakili hasil pembacaan sensor, yang menunjukkan akurasi dan konsistensi dalam merekam perubahan suhu. Secara keseluruhan, grafik ini menggambarkan kondisi di mana suhu ruangan mengalami penurunan terkontrol selama periode pengamatan.

3.2 Pengujian Software

3.2.1 Pengujian Open Loop

Pengujian open loop dilakukan untuk mengetahui karakteristik awal sistem sebelum diterapkan kendali PID Cascade. Tujuan utama dari pengujian ini adalah untuk memperoleh parameter dinamis sistem berupa waktu tunda (L) dan konstanta waktu (T) dengan cara memberikan input langsung ke sistem tanpa umpan balik, dimana nilai L dan T tersebut digunakan sebagai dasar dalam perhitungan parameter (Kp, Ki, dan Kd) menggunakan metode Ziegler-Nichols 1. Dengan demikian, pengujian open loop menjadi langkah awal yang penting dalam merancang kendali PID Cascade yang sesuai dengan karakteristik sistem sebenarnya.



Gambar 11 Respon Open Loop Suhu

Dari grafik respon open loop diatas menghasilkan parameter waktu tunda (L) dan konstanta waktu (T) sebesar $L = 40$ s dan $T = 610$ s, maka dari nilai tersebut kita bisa tentukan nilai Kp, Ti, dan Td menggunakan rumus penalaan PID Cascade dengan menggunakan Metode Ziegler-Nochols 1.

Setelah mengetahui nilai L dan T di atas, maka hasil tersebut harus diolah dengan menggunakan tabel Ziegler-Nichols 1 dibawah ini, sehingga menghasilkan nilai Kp, Ki, dan Kd. Sehingga mendapatkan nilai berikut :

Tabel 4 Hasil Penalaan Tuning Ziegler-Nichols 1

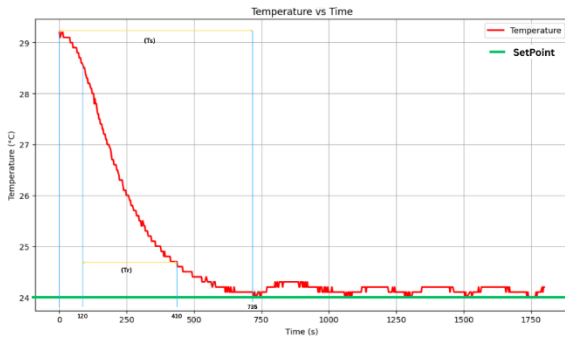
Jenis Pengendali	Kp	Ti	Td
P	T/L	∞	0
PI	0,9T/L	0,3L	0
PID	18,3	2,2	36,6

3.2.2 Pengujian Close Loop

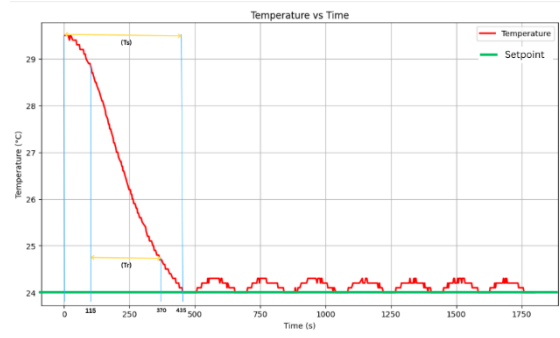
Pengujian close loop dilakukan untuk mengevaluasi kinerja sistem kendali PID Cascade dalam menjaga suhu dan kelembapan agar sesuai dengan nilai setpoint. Tujuan utama dari pengujian ini adalah untuk mengamati bagaimana sistem merespon terhadap perubahan setpoint ketika PID Cascade diaktifkan, serta untuk menilai karakteristik performa sistem berdasarkan parameter seperti overshoot, rise time, settling time, dan error steady-state. Hasil pengujian ini digunakan untuk memastikan bahwa sistem telah memenuhi kriteria performa yang diharapkan, serta dasar evaluasi apakah penyetelan parameter PID Cascade sudah optimal atau masih memerlukan peyesuaian lebih lanjut. Dalam pengujian close loop ini, digunakan dua metode tuning yaitu metode Ziegler-Nichols dan metode trial and error. Kedua metode tersebut diterapkan untuk membandingkan efektivitasnya dalam mengendalikan sistem, sehingga dapat diketahui metode mana yang memberikan kinerja terbaik pada sistem ini.

3.2.2.1 Pengujian Sistem Kendali tuning PID Cascade Ziegler – Nichols 1

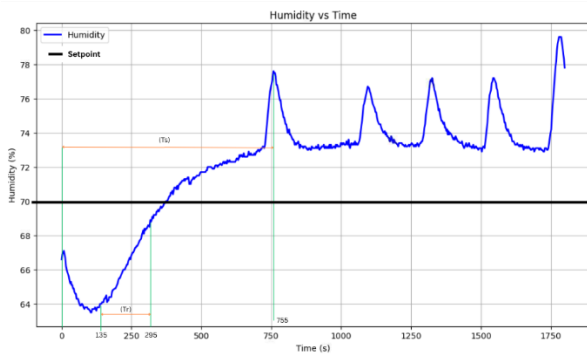
Percobaan kendali PID menggunakan tuning Ziegler-Nichols dengan parameter nilai Kp, Ki, dan Kd yang telah ditentukan diatas yaitu, $K_p = 18,3$, $K_i = 2,2$, dan $K_d = 36,6$ dengan setpoint suhu 24°C. Berikut grafik respon percobaan yang dihasilkan.



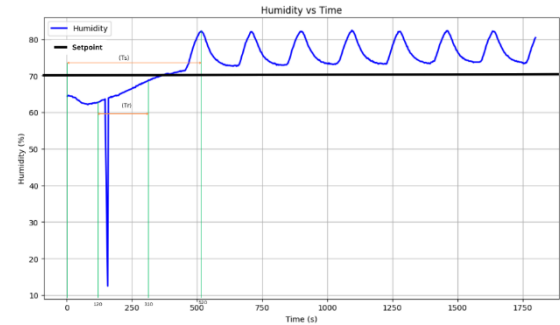
Gambar 12 Respon *Close Loop* PID Cascade Ziegler-Nichols 1 Suhu



Gambar 14 Respon *Close Loop* PID Cascade Trial and Error Suhu



Gambar 13 Respon *Close Loop* PID Cascade Ziegler-Nichols 1 Kelembapan



Gambar 15 Respon *Close Loop* PID Cascade Trial and Error Kelembapan

Berdasarkan grafik respon hasil pengujian diatas, dapat dilakukan pengukuran terhadap parameter-parameter performa sistem kontrol, yaitu *overshoot*, *rise time*, *settling time*, dan *error steady-state*. Hasil pengukuran dari masing-masing parameter tersebut disajikan pada tabel berikut.

Tabel 4 Hasil Respon Metode Ziegler-Nichols 1

Kp	Ki	Kd	Setpoint t	ESS (%)	Tr (s)	OS (%)	Ts (s)	Ket
18,3	2,	36,	24°C	0,6	29	0	73	Suhu
	2	6		2	0		5	
10	1	3	70%	5,7	16	11,1	75	Kelembapan
				1	0	4	5	n

3.2.2.2 Pengujian Sistem Kendali *tuning* PID Cascade Trial and Error

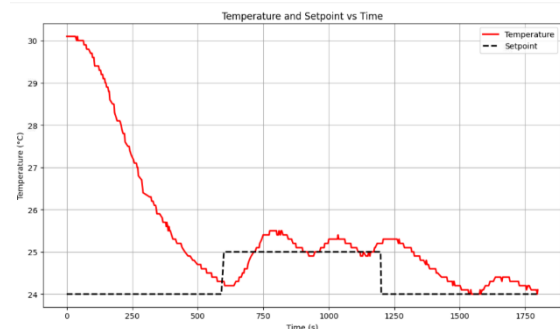
Percobaan kendali PID menggunakan *tuning trial and error* dengan parameter nilai Kp, Ki, dan Kd yang telah ditentukan diatas yaitu, Kp = 20, Ki = 10, dan Kd = 15 dengan *setpoint* suhu 24°C, berikut grafik respon percobaan yang dihasilkan.

Berdasarkan grafik respon hasil pengujian diatas, dapat dilakukan pengukuran terhadap parameter-parameter performa sistem kontrol, yaitu *overshoot*, *rise time*, *settling time*, dan *error steady-state*. Hasil pengukuran dari masing-masing parameter tersebut disajikan pada tabel berikut.

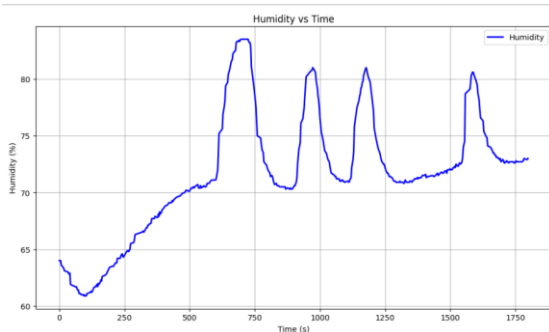
Tabel 5 Hasil Respon Metode Trial and Error

Kp	Ki	Kd	Setpoint	ESS (%)	Tr (s)	OS (%)	Ts (s)	Ket
20	10	15	24°C	0,83	245	0	435	Suhu
10	5	15	70 %	0	190	17,14	520	Kelembapan

3.2.2.3 Pengujian Setpoint Berubah



Gambar 16 Respon Setpoint Berubah Suhu



Gambar 17 Respon *Setpoint* Berubah Kelambapan

4. KESIMPULAN

Berdasarkan hasil pengujian dan analisis yang telah dilakukan terhadap sistem kendali PID *Cascade* untuk mengatur Suhu dan Kelembapan, maka dapat disimpulkan sebagai berikut:

1. Dari sistem Kendali PID *Cascade* mampu mempertahankan Suhu 24°C meskipun ada *Error Steady State* yang tercatat 0,83%, dan hal tersebut memenuhi parameter yang diinginkan.
2. Metode *Trial and Error* menghasilkan performa sistem yang lebih baik dibandingkan metode *Ziegler-Nichols 1* pada plant ini. Hal ini terlihat dari waktu *rise time* dan *settling time* yang lebih cepat serta *error steady-state* yang lebih kecil sehingga hal tersebut telah memenuhi parameter yang ditentukan.

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada Pembimbing Tugas Akhir Bapak Febi Ariefka Septian Putra, S.ST., M.T. dan Bapak Giga Verian Pratama, S.Si., M.T. atas dukungan dalam penyusunan makalah ini.

DAFTAR PUSTAKA

1. Ivanova D, Valov N, Deyanov M. Application of the genetic algorithm for cascade control of a HVAC system. MATEC Web Conf. 2019. <https://doi.org/10.1051/mateconf/2019>
2. Legweel KMB, Lazić DV, Ristanović MR, Lozanović Šajić JV. The performance of PIP-cascade controller in HVAC system. Therm Sci. 2014;18(Suppl 1):S213–20. <https://doi.org/10.2298/TSCI130812183L>
3. Bai J, Bai J. Development an Adaptive Incremental Fuzzy PI Controller for a HVAC System. 2013.
4. Tang A, Yang L, Zeng T, Yu Q. Cascade Control Method of Sliding Mode and PID for PEMFC Air Supply System. Energies. 2023;16(1). <https://doi.org/10.3390/en16010228>
5. Gulan M, Salaj M, Rohal'-Ilkiv B. Application of adaptive multivariable generalized predictive control to a HVAC system in real time. Arch Control Sci. 2014;24(1):67–84. <https://doi.org/10.2478/acsc-2014-0005>
6. Rohit RA, Padmasine KG. IoT-Based Centralized Governing System for Industrial Automation Using

- Raspberry Pi. Int J Res Publ Rev. 2023;4(3):4675–81. <https://doi.org/10.55248/gengpi.2023.4.3.33742>
7. Busu N, Isa NM, Hariri A, Hussein M. The Comparison of P, PI and PID Strategy Performance as Temperature Controller in Active Iris Damper for Centralized Air Conditioning System. J Adv Res Fluid Mech Therm Sci. 2023;102(2):143–54. <https://doi.org/10.37934/arfmts.102.2.143154>
8. Xue H, Lv H, Xiao Y, Wang X. Cascade PID Control Systems Based on DNA Strand Displacement With Application in Polarization of Tumor-Associated Macrophages. IEEE Access. 2023;11:38563–77. <https://doi.org/10.1109/ACCESS.2023.3267738>
9. Abood LH, Kadhim NN, Mohammed YA. Dual stage cascade controller for temperature control in greenhouse. Bull Electr Eng Inform. 2023;12(1):51–8. <https://doi.org/10.11591/eei.v12i1.4328>
10. Putri AR, Rahayu PN, Putri I. Food warmer system based on DHT-22. 2020.
11. Ivanova D, Valov N, Deyanov M. Application of the genetic algorithm for cascade control of a HVAC system. MATEC Web Conf. 2019. <https://doi.org/10.1051/mateconf/2019>
12. Espejel-Blanco DF, et al. HVAC Control System Using Predicted Mean Vote Index for Energy Savings in Buildings. Buildings. 2022;12(1). <https://doi.org/10.3390/buildings12010038>
13. Almabrok A, Psarakis M, Dounis A. Fast tuning of the PID controller in an HVAC system using the Big Bang-Big Crunch algorithm and FPGA technology. Algorithms. 2018;11(10). <https://doi.org/10.3390/a11100146>
14. Kumavat M, Thale S. Analysis of CSTR Temperature Control with PID, MPC & Hybrid MPC-PID Controller. ITM Web Conf. 2022;44:01001. <https://doi.org/10.1051/itmconf/20224401001>
15. Introduction to Industrial Automation. [Online]. Available from: <http://taylorandfrancis.com>