

Rancang Bangun *Cooling pad* Otomatis Berbasis Pid Dengan Sistem Pendingin Aktif *Peltier* Untuk Laptop

1st Muhammad Hisbi Maulana
Direktorat Kampus Purwokerto
Universitas Telkom

Purwokerto, Indonesia
hisbimaaulana@students.telkomuniversit
y.ac.id

2nd Yulian Zetta Maulana
Direktorat Kampus Purwokerto
Universitas Telkom

Purwokerto, Indonesia
yulianm@telkomuniversity.ac.id

3rd Gunawan Wibisono
Direktorat Kampus Purwokerto
Universitas Telkom

Purwokerto, Indonesia
gwibisono@telkomuniversity.ac.id

Abstrak — *Overheating* pada laptop sering terjadi saat aktivitas berat seperti bermain *game* atau rendering. Penelitian ini mengembangkan *cooling pad* otomatis dengan sistem pendingin aktif berbasis modul *Peltier* yang dikendalikan oleh algoritma PID. Sistem terdiri dari sensor suhu DS18B20, Arduino Uno sebagai mikrokontroler, serta aktuator berupa kipas dan modul *Peltier*. Parameter PID diperoleh melalui dua metode: *autotuning* dan *trial error*. Sistem diuji pada dua jenis laptop, yaitu kasual dan *gaming*, dalam kondisi *idle* dan *stress test*. Hasilnya menunjukkan bahwa sistem mampu mempertahankan suhu sekitar 43 °C pada sensor, yang berkorespondensi dengan suhu internal sekitar 75 °C. Performa kontrol PID terbaik menghasilkan *overshoot* kurang dari 1.02%, *steady-state error* kurang dari 1%, dan waktu *settling* 143 detik. Sistem *closed-loop* lebih responsif dibandingkan *open-loop*. Sistem ini efektif meningkatkan efisiensi pendinginan laptop secara otomatis.

Kata kunci— PID, *Peltier*, *Cooling pad*, Arduino, DS18B20, suhu laptop

I. PENDAHULUAN

Perangkat laptop modern sering kali digunakan untuk aktivitas komputasi berat seperti pengeditan video, rendering 3D, atau bermain *game*, yang menyebabkan peningkatan signifikan pada suhu perangkat [1]. Kondisi ini memicu masalah *overheating* yang dapat mengganggu performa, mempercepat kerusakan komponen, serta menurunkan umur pakai laptop. Sistem pendinginan internal bawaan pada laptop, terutama pada laptop kelas menengah ke bawah, kerap kali tidak mampu menangani lonjakan suhu secara optimal [2].

Salah satu solusi tambahan yang umum digunakan adalah *cooling pad* eksternal. Namun, *cooling pad* konvensional umumnya menggunakan kipas yang beroperasi secara konstan tanpa mempertimbangkan suhu aktual perangkat [3]. Hal ini menyebabkan konsumsi daya yang tidak efisien dan pendinginan yang tidak adaptif. Oleh karena itu, diperlukan sistem pendingin yang lebih cerdas dan mampu menyesuaikan daya kerjanya secara otomatis.

Penelitian ini mengusulkan sistem *cooling pad* otomatis yang menggunakan modul *Peltier* sebagai sistem pendingin

aktif. Modul ini bekerja berdasarkan prinsip efek termoelektrik, yang memungkinkan transfer panas dari satu sisi ke sisi lainnya saat dialiri arus listrik [4]. Namun, karakteristik termal modul *Peltier* bersifat non-linier, sehingga membutuhkan pengendali suhu yang presisi untuk mengoptimalkan kinerjanya.

Algoritma *Proportional-Integral-Derivative* (PID) dipilih karena kemampuannya dalam mengatur sistem dinamis secara stabil dan responsif. Penelitian ini menerapkan dua pendekatan *tuning* PID, yaitu metode *autotuning* menggunakan MATLAB dan metode *trial error* berbasis pengamatan langsung [1]. Mikrokontroler Arduino Uno digunakan sebagai pusat kendali, sedangkan sensor DS18B20 digunakan untuk mendeteksi suhu dari ventilasi bawah laptop [2].

Ruang lingkup penelitian dibatasi pada laptop dengan ventilasi pembuangan udara di bagian bawah atau samping, serta menggunakan satu titik pengukuran suhu. Sistem diuji dalam kondisi laboratorium pada dua kategori laptop (kasual dan *gaming*) dan dibandingkan dalam skenario *open-loop* dan *closed-loop*. Aspek efisiensi energi dan integrasi IoT tidak menjadi fokus utama karena dianggap tidak krusial dalam studi ini.

Tujuan dari penelitian ini adalah untuk merancang sistem pendingin otomatis berbasis *Peltier* dan PID, mengimplementasikan kontrol suhu yang adaptif, serta mengevaluasi performanya melalui pengujian eksperimental. Penelitian ini diharapkan dapat memberikan kontribusi dalam mengembangkan *cooling pad* yang cerdas dan efisien untuk mendukung stabilitas suhu laptop saat bekerja dalam beban tinggi.

II. KAJIAN TEORI

A. Modul *Peltier*

Modul *Peltier* adalah perangkat termoelektrik yang dapat menciptakan perbedaan suhu antara dua sisi ketika dialiri arus listrik. Efek ini dikenal sebagai efek *Seebeck*. Salah satu sisi akan menjadi panas dan sisi lainnya dingin, sehingga dapat dimanfaatkan untuk keperluan pendinginan. Agar bekerja optimal, sisi panas harus didinginkan secara aktif agar panas tidak kembali ke sisi dingin [5].

B. Sensor Suhu

Sensor suhu merupakan perangkat penting dalam sistem pengendalian suhu karena mampu mendeteksi perubahan temperatur dan mengubahnya menjadi sinyal listrik. Prinsip kerjanya bergantung pada perubahan resistansi atau tegangan akibat variasi suhu. Sensor ini dapat mengukur suhu baik di udara maupun pada permukaan benda dan efektif merespons perubahan suhu yang cukup signifikan [6].

C. PID (*Proportional Integral Derivative*)

Metode *Proportional-Integral-Derivative* (PID) adalah salah satu metode pengendalian umpan balik tertutup yang umum digunakan. Pengontrol PID bekerja berdasarkan tiga parameter utama, yaitu:

1. K_p (*Proportional*): untuk mengurangi *error* saat ini
2. K_i (*Integral*): untuk menghilangkan *error* jangka panjang,
3. K_d (*Derivative*): untuk mengantisipasi *error* di masa mendatang.

Persamaan umum PID kontinu dituliskan sebagai:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (1)$$

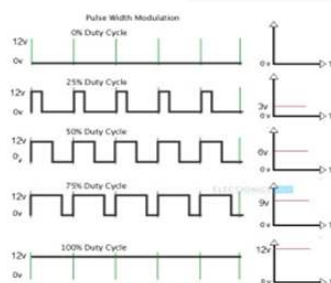
di mana $u(t)$ adalah sinyal keluaran dan $e(t)$ adalah *error* antara suhu yang diinginkan dan suhu aktual [7].

D. Auto-Tuning PID

Autotuning dilakukan dengan cara mengumpulkan data suhu dari sistem *open-loop*, kemudian dimasukkan ke MATLAB untuk dilakukan identifikasi sistem. Dengan metode pemodelan orde dua, sistem diperoleh fungsi transfer yang kemudian digunakan dalam simulasi PID *Tuner* untuk memperoleh parameter PID secara otomatis berdasarkan karakteristik sistem nyata [8].

E. PWM (*Pulse Width Modulation*)

PWM adalah teknik modulasi sinyal digital yang digunakan untuk mengatur daya rata-rata yang dikirim ke beban, seperti motor atau elemen pemanas. Dalam sistem ini, PWM dimanfaatkan untuk mengatur kecepatan kipas dan intensitas kerja modul *Peltier*. Variasi lebar pulsa mempengaruhi jumlah daya yang diterima oleh aktuator, memungkinkan pengendalian yang lebih halus dan efisien [9].



GAMBAR 1
(DUTY CYCLE YANG BERBEDA PADA PWM [10])

F. Mikrokontroler

Mikrokontroler adalah sebuah sistem komputer mini yang terintegrasi dalam satu chip dan berfungsi untuk mengontrol perangkat elektronik secara otomatis. Mikrokontroler memiliki unit pemrosesan (CPU), memori, dan antarmuka *input/output* dalam satu paket. Dalam konteks sistem pengendalian, mikrokontroler berperan membaca data dari

sensor, memproses logika kendali, dan memberikan sinyal keluaran ke aktuator. Mikrokontroler digunakan secara luas dalam sistem tertanam karena konsumsi dayanya yang rendah dan kemampuannya untuk bekerja secara *real-time* [11].

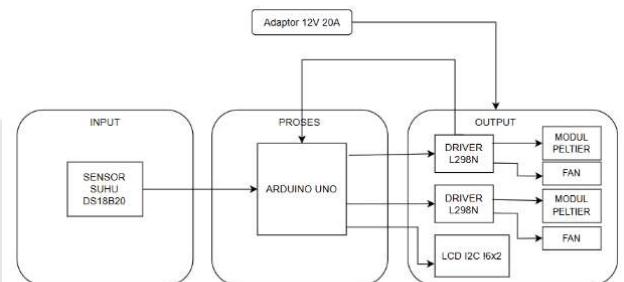
G. Pengkondisi Sinyal

Pengkondisi sinyal merupakan rangkaian atau komponen yang digunakan untuk menyesuaikan *output* sensor agar kompatibel dengan *star input* perangkat elektronik seperti mikrokontroler. Dalam banyak kasus, sinyal dari sensor memiliki karakteristik tegangan atau format data yang tidak langsung dapat diproses. Pengkondisi sinyal berperan dalam memperkuat sinyal lemah, menghilangkan *noise* melalui filtrasi, atau menyesuaikan level tegangan agar tetap berada dalam batas yang dapat diterima. Fungsinya sangat penting dalam sistem monitoring karena membantu menjaga akurasi dan keandalan data yang masuk ke unit pengolahan utama [12].

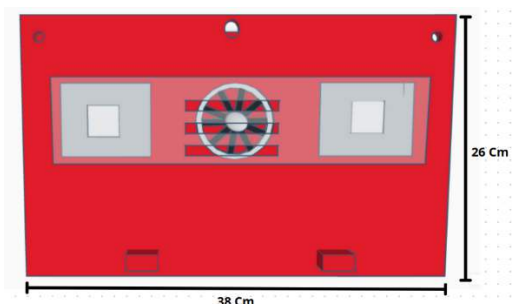
III. METODE

A. Perancangan *Hardware*

Perancangan *hardware* dilakukan dengan merakit komponen utama sistem ke dalam rangkaian pendingin aktif. Sistem terdiri dari sensor suhu DS18B20, mikrokontroler Arduino Uno, dua buah kipas DC 12V, dan modul *Peltier* tipe TEC1-12706 yang dipasang pada *heatsink* berbahan aluminium. Driver motor L298N digunakan untuk mengatur PWM ke kipas dan *Peltier*. Keseluruhan rangkaian dirakit dalam sebuah casing berbahan akrilik dengan dimensi panjang 38 cm, lebar 26 cm, dan tinggi 4 sampai 7 cm. Sistem dilengkapi dengan lubang ventilasi pada sisi kiri dan kanan untuk sirkulasi udara serta sebuah LCD I2C 16x2 untuk menampilkan suhu *real-time*. Sensor suhu ditempatkan tepat di bawah lubang ventilasi laptop agar suhu udara buangan dapat terbaca secara langsung dan akurat.



GAMBAR 2
(BLOK DIAGRAM SISTEM)



GAMBAR 3
(DESAIN PROTOTIPE 3D ALAT TAMPAK ATAS)

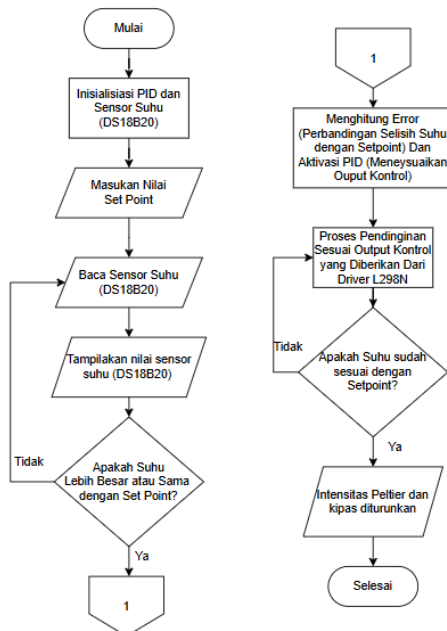


GAMBAR 4
(DESAIN PROTOTIPE 3D ALAT TAMPAK SAMPIING)

B. Perancangan Software

Perangkat *software* dikembangkan menggunakan Arduino IDE dengan bahasa C++. Program dikembangkan secara modular yang mencakup pembacaan sensor suhu DS18B20, perhitungan PID, dan pengendalian aktuator melalui sinyal PWM. Nilai referensi suhu disetel pada 43°C. Perhitungan PID dilakukan secara *real-time* berdasarkan parameter K_p, K_i, dan K_d yang diperoleh dari proses *autotuning* atau *trial error*. *Output* PID dikonversi ke sinyal PWM yang mengatur kecepatan kipas dan daya kerja *Peltier*. Program juga mengatur tampilan suhu pada LCD dan mencetak data suhu serta sinyal PWM ke serial monitor untuk dokumentasi dan analisis pengujian.

Flowchart sistem pengendalian suhu pada perangkat lunak dapat dilihat pada Gambar berikut.



GAMBAR 5
(FLOWCHART SISTEM)

C. Pengujian Sensor

Pengujian sensor dilakukan dengan membandingkan hasil pembacaan suhu dari DS18B20 terhadap data suhu internal yang dibaca dari *software* monitoring HWMonitor. Pengujian dilakukan pada dua laptop dengan spesifikasi berbeda untuk mengetahui korelasi antara suhu udara dari ventilasi bawah dengan suhu internal prosesor. Hasil pembacaan kemudian dianalisis untuk mendapatkan persamaan linier koreksi suhu

D. Pengujian Sistem Tanpa PID (*Open Loop*)

Sistem dijalankan tanpa kendali PID untuk mengetahui bagaimana respons suhu tanpa pengaturan otomatis. Data suhu dicatat untuk mengetahui performa dasar sistem tanpa pengendalian umpan balik. Pengujian dilakukan dalam kondisi *idle* dan beban tinggi pada dua jenis laptop.

E. Pengujian *Tuning* PID pada MATLAB (*Autotuning* dan *Trial Error*)

Data suhu dari sistem *Open Loop* diekspor ke Excel dan dianalisis di MATLAB untuk membentuk model matematis sistem. Proses *autotuning* dilakukan untuk memperoleh parameter PID optimal melalui simulasi MATLAB Simulink. Selain itu, dilakukan juga *tuning* manual (*trial error*) secara langsung pada sistem fisik hingga diperoleh respons suhu yang stabil.

F. Pengujian Sistem PID pada *Cooling pad* (*Autotuning* dan *Trial Error*)

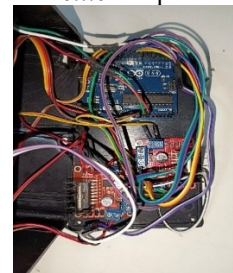
Sistem dijalankan dengan pengendali PID hasil dari kedua metode *tuning*. Evaluasi dilakukan dengan membandingkan performa suhu terhadap waktu, *overshoot*, waktu stabil, dan *error steady-state*. Pengujian ini membuktikan efektivitas kontrol PID dalam menjaga suhu laptop secara adaptif dan respon

IV. HASIL DAN PEMBAHASAN

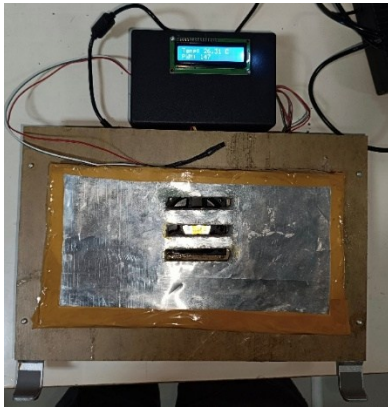
A. Implementasi Rancangan Sistem

Sistem *cooling pad* otomatis telah berhasil diimplementasikan berdasarkan hasil rancangan pada tahap sebelumnya. Perangkat terdiri dari beberapa komponen utama yang dirakit menjadi satu kesatuan sistem yang kompak dan portabel. Komponen utama meliputi mikrokontroler, sensor suhu, dua buah kipas DC, serta modul *Peltier* yang dipasang pada *heatsink* aluminium. Seluruh komponen ini ditempatkan dalam casing berbahan akrilik yang didesain memiliki ventilasi samping untuk memperlancar aliran udara panas keluar.

Bagian luar sistem menampilkan layar LCD I2C yang berfungsi untuk menunjukkan suhu secara *real-time*, serta indikator sistem aktif. Bagian dalam menunjukkan susunan komponen yang diatur agar aliran udara dari kipas langsung mengenai area pembuangan panas laptop, serta efisiensi pendinginan dari modul *Peltier* dapat tercapai maksimal.



GAMBAR 6
(RANGKAIAN SISTEM TAMPILAN DALAM)



GAMBAR 7
(RANCANGAN SISTEM TAMPILAN LUAR)

B. Hasil Pengujian Sensor DS18B20

Pengujian sensor DS18B20 dilakukan untuk mengetahui tingkat keakuratan sensor dalam membaca suhu udara dari ventilasi bawah laptop. Pengujian dilakukan dengan membandingkan hasil pembacaan dari DS18B20 terhadap dua referensi, yaitu suhu internal laptop yang dibaca menggunakan *software* HWMonitor dan pengukuran suhu menggunakan termometer digital.

Pengukuran dilakukan pada dua jenis laptop (kasual dan *gaming*) dalam kondisi *idle* dan *stress test*. Data suhu dari DS18B20 dibandingkan dengan nilai suhu yang ditampilkan HWMonitor dan termometer pada waktu yang bersamaan untuk memperoleh model hubungan suhu yang lebih akurat. Berdasarkan hasil pengujian, diperoleh korelasi linier antara suhu sensor dan suhu internal laptop, yang diformulasikan sebagai berikut:

1. Laptop casual:

$$T_{soft} = 1,236 \cdot T_{sensor} + 21,85 \quad (2)$$

2. Laptop *gaming*:

$$T_{soft} = 1,176 \cdot T_{sensor} + 24,6 \quad (3)$$

Model koreksi ini digunakan untuk memperkirakan suhu internal laptop berdasarkan pembacaan dari sensor DS18B20 selama pengujian sistem kontrol. Sebagai contoh, pada laptop casual, jika suhu sensor terbaca 43°C, maka berdasarkan persamaan koreksi ($T_{internal} = 1.236 \times 43 + 21.85$), diperoleh suhu internal sekitar 75°C. Oleh karena itu, nilai suhu 43°C dipilih sebagai *setpoint* pada sistem kendali PID, agar suhu internal laptop tetap berada pada ambang aman.

TABEL 1
(PENGUJIAN AWAL SENSOR SUHU DS18B20 PADA LAPTOP KASUAL)

No	Hasil Pengukuran Suhu Awal Laptop Tipe Kasual			Mean
	Software	Termometer	Sensor DS18B20	
T1	54	43,3	43,69	43,495
T2	56,5	44,3	43,19	43,745
T3	56,4	44,8	44,25	44,525
T4	56,9	43,8	43,56	43,680
T5	56,1	44,4	43,13	43,765
T6	56,8	44,6	43,85	44,225
T7	56,4	44,6	43,56	44,080
T8	56,5	44,4	43,81	44,105
T9	57,1	44,7	44,06	44,380
T10	57,3	44,5	43,56	44,030

TABEL 2
(PENGUJIAN AKHIR SENSOR SUHU DS18B20 PADA LAPTOP KASUAL)

No	Hasil Pengukuran Suhu Akhir Laptop Tipe Kasual			Mean
	Software	Termometer	Sensor DS18B20	
T1	92,3	56,5	56,25	56,375
T2	92,3	56,6	56,1	56,350
T3	92,3	58,1	57	57,550
T4	92,3	59,1	57,56	58,330
T5	92,3	58,3	57,31	57,805
T6	92,3	57,9	57,69	57,795
T7	92,3	58,5	57,31	57,905
T8	92,3	58,5	57,38	57,940
T9	92,3	58,5	57,5	58,000
T10	92,3	58,8	57,44	58,120

TABEL 3
(PENGUJIAN AWAL SENSOR SUHU DS18B20 PADA LAPTOP GAMING)

No	Hasil Pengukuran Suhu Awal Laptop Tipe Gaming			Mean
	Software	Termometer	Sensor DS18B20	
T1	48	47,8	47,75	47,775
T2	48	47,3	47,5	47,400
T3	48	47,8	47,88	47,840
T4	48	47,6	47,52	47,560
T5	47	47,8	47,63	47,715
T6	47	47,1	47,61	47,355
T7	47	47,9	47,56	47,730
T8	48	47,7	47,63	47,665
T9	48	47,5	47,44	47,470
T10	48	47,9	47,19	47,545

TABEL 4
(PENGUJIAN AKHIR SENSOR SUHU DS18B20 PADA LAPTOP GAMING)

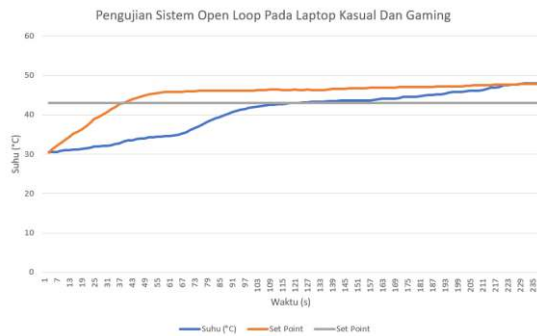
No	Hasil Pengukuran Suhu Akhir Laptop Tipe Gaming			Mean
	Software	Termometer	Sensor DS18B20	
T1	92	57,7	57,3	57,500
T2	92	59,2	59	59,100
T3	92	58,2	57,9	58,050
T4	92	58,3	58,44	58,370
T5	92	58,6	58,31	58,455
T6	92	59	58,53	58,765
T7	92	58,8	58,55	58,675
T8	92	58,9	58,63	58,765
T9	92	59	58,24	58,620
T10	92	59,9	59,25	59,575

C. Hasil Pengujian Sistem Tanpa PID (*Open Loop*)

Pada pengujian ini, sistem dijalankan tanpa kendali PID. Sinyal PWM untuk kipas dan modul *Peltier* diset tetap pada 50%, sesuai dengan konfigurasi dasar sistem. Pengaturan ini dilakukan agar daya kerja tetap konstan dan dapat dijadikan

acuan untuk membandingkan performa sistem ketika beralih ke mode *closed-loop*.

Pada pengujian ini, sistem dijalankan tanpa kendali PID. Saat suhu meningkat karena beban aplikasi berat, kipas dan *Peltier* aktif terus-menerus tanpa penyesuaian intensitas. Hal ini menyebabkan respons sistem yang kurang efisien dan suhu tidak terkontrol secara optimal. Suhu sensor menunjukkan fluktuasi dan waktu stabil yang lambat.



GAMBAR 8
(HASIL PENGUJIAN SISTEM *OPEN LOOP* PADA LAPTOP KASUAL DAN *GAMING*)

D. Hasil Pengujian *Tuning* PID melalui MATLAB

Pengujian *tuning* PID diawali dengan pengambilan data sistem dalam kondisi *open-loop* dari dua jenis laptop, yaitu laptop kasual dan laptop *gaming*. Data suhu yang diperoleh dari sensor DS18B20 dicatat menggunakan Serial Monitor Arduino dan disimpan dalam format Excel. Data tersebut kemudian dimasukkan ke MATLAB untuk dilakukan proses identifikasi sistem.

Pemodelan dilakukan dengan pendekatan orde dua untuk masing-masing jenis laptop. Berdasarkan hasil identifikasi sistem menggunakan *toolBox System Identification* pada MATLAB, diperoleh dua model fungsi transfer sebagai berikut:

1. Laptop kasual:

$$G(s) = \frac{0.1513s + 0.0003894}{s^2 + 0.9063s + 0.002266} \quad (4)$$

2. Laptop *gaming*:

$$G(s) = \frac{1.855s + 0.07643}{s^2 + 10.36s + 0.4392} \quad (5)$$

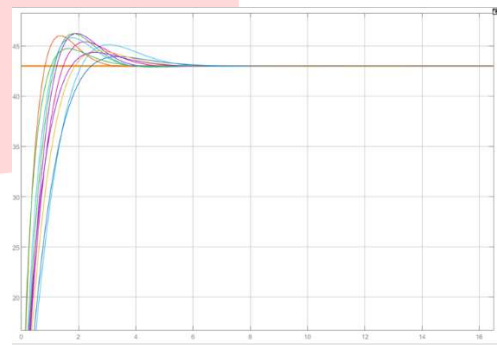
Model ini menggambarkan dinamika sistem pendingin laptop berbasis *Peltier*. Fungsi transfer ini kemudian digunakan dalam PID *Tuner* MATLAB untuk melakukan *autotuning*, yaitu perhitungan otomatis parameter PID berdasarkan respons sistem terhadap perubahan suhu.

Selain *autotuning*, dilakukan pula metode *tuning* manual atau *trial error* secara langsung pada sistem fisik. Proses ini dimulai dengan pengaturan nilai K_p , kemudian dilanjutkan dengan penyetelan K_i dan K_d berdasarkan grafik respons suhu terhadap waktu. Tujuannya adalah mencapai performa terbaik berupa kestabilan suhu tanpa *overshoot* besar serta waktu tunda yang minim. Nilai K_p , K_i , dan K_d disesuaikan secara bertahap hingga sistem menunjukkan performa respons suhu yang optimal tanpa *overshoot* besar dan waktu stabil yang terlalu lama. Rangkaian penyesuaian ini dilakukan berdasarkan grafik respons suhu terhadap waktu dan disesuaikan dengan karakteristik sistem fisik yang diuji.

Tabel dan gambar berikut memperlihatkan parameter PID yang diperoleh dari masing-masing metode:

TABEL 5
(NILAI PARAMETER PID METODE *AUTOTUNING* PADA TIPE LAPTOP KASUAL)

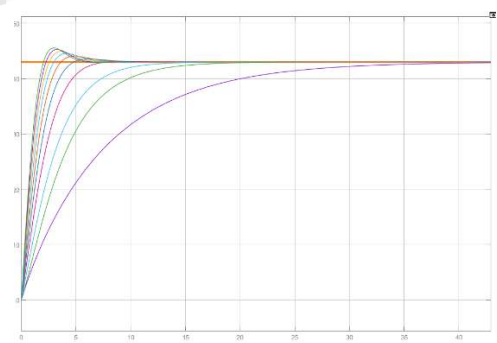
PID	Nilai Parameter			
	K_p	K_i	K_d	N
PID 1	9,24	16,58	-0,045	203,21
PID 2	10,73	19,21	0,366	10,63
PID 3	5,33	8,85	0	100
PID 4	8,89	14,06	0,136	185,33
PID 5	6,62	8,48	0	100
PID 6	17,5	31,61	0,73	100
PID 7	12,14	20,84	0,59	4,54
PID 8	8,05	11,06	0,15	161,42
PID 9	10,01	13,21	0,67	3,55
PID 10	18,3	26,36	1,23	5,18



GAMBAR 9
(HASIL PENALAAN SISTEM DENGAN PENGENDALI PID PADA SIMULINK DI TIPE LAPTOP KASUAL (METODE *AUTOTUNING*))

TABEL 6
(NILAI PARAMETER PID METODE *TRIAL ERROR* PADA TIPE LAPTOP KASUAL)

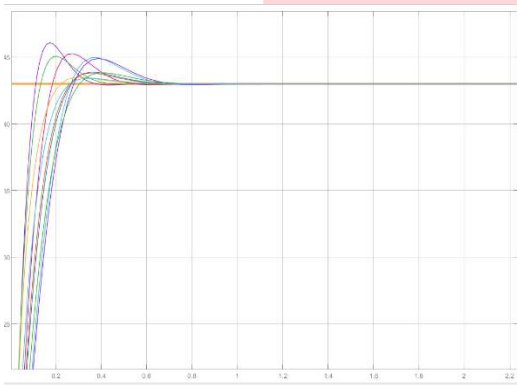
PID	Nilai Parameter			
	K_p	K_i	K_d	N
PID 1	1	0,8	0,3	0
PID 2	1	1,5	0,5	0
PID 3	1,5	2	1	0
PID 4	2	3	0,5	0
PID 5	2,5	4	1	0
PID 6	3	5	0,3	0
PID 7	3,5	6	0,5	0
PID 8	4	7	1	0
PID 9	4,5	8	0,5	0
PID 10	5	9	1	0



GAMBAR 10
(HASIL PENALAAN SISTEM DENGAN PENGENDALI PID PADA SIMULINK DI TIPE LAPTOP KASUAL (METODE *TRIAL ERROR*))

TABEL 7
(NILAI PARAMETER PID METODE *AUTOTUNING* PADA TIPE
LAPTOP *GAMING*)

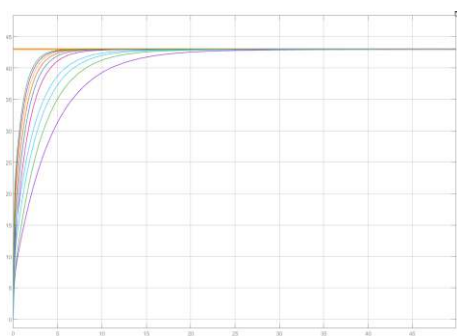
PID	Nilai Parameter			
	Kp	Ki	Kd	N
PID 1	8,44	168,82	- 0,0037	2265,04
PID 2	8,82	150,63	0,0331	69,03
PID 3	2,5	58,95	0	100
PID 4	6,5	83,17	0,0114	1496,5
PID 5	4,69	92,21	0	100
PID 6	4,36	68,48	0	100
PID 7	4,72	71,5	0	100
PID 8	8,85	109,55	0,0607	29,7
PID 9	2,31	55,71	0	100
PID 10	3,55	58,83	0	100



GAMBAR 11
(HASIL PENALAAAN SISTEM DENGAN PENGENDALI PID PADA
SIMULINK DI TIPE LAPTOP *GAMING* (METODE *AUTOTUNING*))

TABEL 8
(NILAI PARAMETER PID METODE *TRIAL ERROR* PADA TIPE
LAPTOP *GAMING*)

PID	Nilai Parameter			
	Kp	Ki	Kd	N
PID 1	1	1,5	0,5	0
PID 2	1	2	0,8	0
PID 3	1,5	2,5	1	0
PID 4	2	3	0,5	0
PID 5	1,5	4	0,5	0
PID 6	2	5	0,3	0
PID 7	2,5	6	0,5	0
PID 8	3	7	0,3	0
PID 9	3,5	8	0,5	0
PID 10	4	9	0,3	0



GAMBAR 12
(HASIL PENALAAAN SISTEM DENGAN PENGENDALI PID PADA
SIMULINK DI TIPE LAPTOP *GAMING* (METODE *TRIAL ERROR*))

E. Pengujian Sistem Closed Loop pada *Cooling pad*

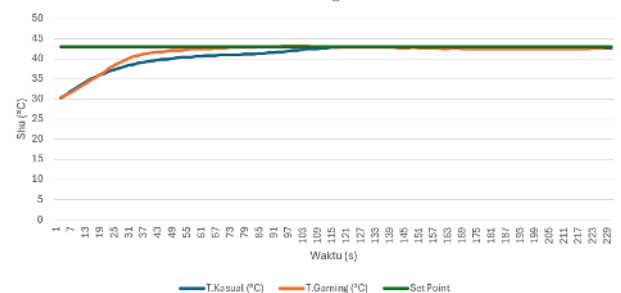
Sistem dijalankan dalam mode *closed-loop* menggunakan dua metode *tuning* PID, yaitu *autotuning* dan *trial error*, pada dua jenis laptop: kasual dan *gaming*. Evaluasi dilakukan berdasarkan performa respons suhu, meliputi waktu stabil (*settling time*), *overshoot*, dan *steady-state error*.

Pada laptop kasual, metode *trial error* menunjukkan hasil paling stabil. Waktu stabil yang dicapai sekitar 175 detik dengan *overshoot* <0.5% dan *error* akhir <1%. Meskipun metode *autotuning* pada laptop ini menghasilkan waktu stabil lebih cepat (143 detik), respons suhu mengalami fluktuasi lebih besar dan *overshoot* awal yang sedikit lebih tinggi.

Berbeda dengan itu, pada laptop *gaming*, metode *autotuning* justru menunjukkan performa paling baik. Sistem mampu mencapai suhu stabil hanya dalam waktu 140 detik, dengan *overshoot* <1.02% dan *steady-state error* <1%. Sementara itu, metode *trial error* menghasilkan waktu stabil lebih lama, sekitar 165 detik, dan cenderung lambat beradaptasi terhadap lonjakan suhu akibat beban aplikasi berat.

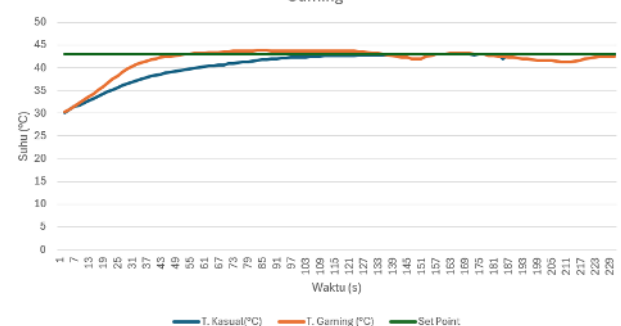
Berdasarkan hasil tersebut, dapat disimpulkan bahwa karakteristik sistem sangat memengaruhi kecocokan metode *tuning* PID. Laptop kasual lebih cocok dengan metode *trial error* karena beban yang tidak terlalu dinamis, sedangkan laptop *gaming* lebih cocok menggunakan *autotuning* karena membutuhkan respons yang lebih cepat dan adaptif.

Perbandingan Sistem Close Loop Auto-Tuning Laptop Kasual Dan Gaming



GAMBAR 13
(PERBANDINGAN SISTEM *CLOSE LOOP AUTO-TUNING* PADA
LAPTOP KASUAL DAN *GAMING*)

Perbandingan Sistem Close Loop Trial Error Laptop Kasual Dan Gaming



GAMBAR 14
(PERBANDINGAN SISTEM *CLOSE LOOP TRIAL ERROR* PADA
LAPTOP KASUAL DAN *GAMING*)

Dari pengamatan ini, dapat disimpulkan bahwa metode *autotuning* menghasilkan kendali yang lebih cepat dan akurat terhadap perubahan suhu dibandingkan metode *trial error*.

Namun, *trial error* tetap memberikan hasil yang cukup stabil dan layak digunakan dalam implementasi sistem sederhana.

F. Perbandingan Sistem Open Loop dan Closed Loop

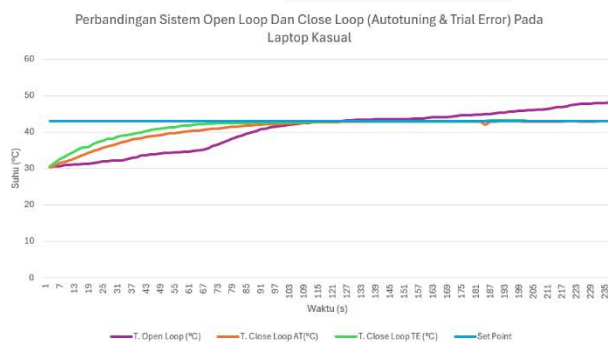
Pengujian ini bertujuan untuk membandingkan performa antara sistem *open-loop* (tanpa kendali PID) dan *closed-loop* (menggunakan kendali PID) dalam mengatur suhu udara buangan dari laptop. Pengujian dilakukan dengan kondisi *idle* dan beban (*stress*) pada dua jenis laptop: kasual dan *gaming*.

Pada sistem *open-loop*, kipas dan *Peltier* dijalankan dengan PWM konstan sebesar 50%. Tidak ada penyesuaian otomatis terhadap suhu yang terukur, sehingga sistem bekerja secara statis. Hasilnya menunjukkan bahwa suhu mengalami fluktuasi cukup besar, terutama saat beban meningkat. Waktu untuk mencapai kestabilan suhu sangat lambat, dan suhu sensor kerap melebihi ambang aman.

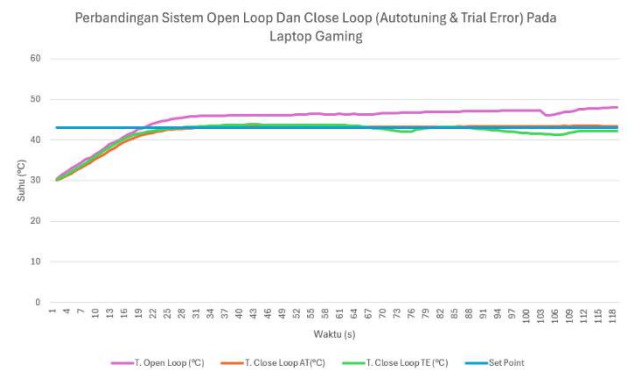
Sebaliknya, pada sistem *closed-loop*, kendali PID memungkinkan penyesuaian dinamis terhadap kondisi suhu yang berubah. Sistem mampu merespons kenaikan suhu dengan cepat dan menyesuaikan daya kerja kipas dan *Peltier* untuk menjaga suhu tetap mendekati *setpoint*. Berdasarkan hasil pengujian, *closed-loop* dengan *autotuning* menghasilkan waktu stabil tercepat dan *overshoot* paling kecil pada laptop *gaming*. Sementara itu, *trial error* menunjukkan performa lebih baik pada laptop kasual karena memberikan kestabilan suhu dengan respons yang lebih halus. Secara keseluruhan, sistem *closed-loop* unggul dalam hal:

1. Adaptivitas terhadap beban kerja
2. Pengendalian suhu presisi
3. Penghematan energi karena aktuasi variatif

Perbandingan ini membuktikan bahwa implementasi kontrol PID signifikan dalam meningkatkan kinerja sistem pendinginan otomatis berbasis *Peltier* dibanding sistem tanpa kendali.



GAMBAR 15
(PERBANDINGAN PENGUJIAN SISTEM OPEN LOOP DAN CLOSE LOOP PADA LAPTOP KASUAL)



GAMBAR 16
(PERBANDINGAN PENGUJIAN SISTEM OPEN LOOP DAN CLOSE LOOP PADA LAPTOP GAMING)

V. KESIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan sistem *cooling pad* otomatis berbasis modul *Peltier* dan kendali PID yang mampu menjaga suhu udara buangan laptop agar tetap stabil. Sistem menggunakan mikrokontroler untuk membaca suhu dari sensor DS18B20 dan mengatur kerja kipas serta modul *Peltier* melalui sinyal PWM secara otomatis.

Berdasarkan hasil pengujian, sistem *closed-loop* memberikan performa yang jauh lebih adaptif dibanding *open-loop*. Pada laptop kasual, metode *tuning* PID terbaik adalah *trial error*, dengan waktu stabil 175 detik, *overshoot* <0.5%, dan *steady-state error* <1%. Sedangkan pada laptop *gaming*, metode *autotuning* menunjukkan performa paling optimal dengan waktu stabil tercepat 140 detik, *overshoot* <1.02%, dan *error* akhir <1%.

Dengan demikian, penggunaan metode *tuning* PID perlu disesuaikan dengan karakteristik perangkat. Sistem yang dirancang menunjukkan efektivitas kendali PID dalam mengatur suhu pendinginan laptop secara adaptif dan efisien.

REFERENSI

- [1] T. Papanchev, "A fuzzy control of *Peltier* -based thermal chamber for reliability tests," 2020 21st Int. Symp. Electr. Appar. Technol. SIELA 2020 - Proc., pp. 19–22, 2020, doi: 10.1109/SIELA49118.2020.9167106.
- [2] A. S. Wahyudi, M. Rifa'i, T. S. Patma, "Sistem Pendinginan Suhu Menggunakan Metode Pid Pada *Aquascape* Berbasis Arduino," *J. Multidisiplin Saintek*, vol. 3, no. 12, 2024, doi: 10.8734/Kohesi.v1i2.365.
- [3] E. Yudiyanto *et al.*, "Pemanfaatan *Peltier* Sebagai Sistem Pendinginan Untuk Medicine Cooler Box Utilization of *Peltier* As a Cooling System for Medicine Cooler Box," *Snitt*, pp. 213–218, 2020.
- [4] S. Maulana Y. A. Suryo, "Sistem Rancang Bangun Pendingin Otomatis Ruang Pergudangan Menggunakan Pengendali PID dan SCADA," *G-Tech J. Teknol. Terap.*, vol. 8, no. 3, pp. 1879–1889,

- 2024, doi: 10.33379/gtech.v8i3.4607.
- [5] S. Triyono, H. Muchtar, W. Sudarwati, "Perancangan Pendingin Minuman Portable Menggunakan Efek *Peltier* Berbasis Raspberry-Pi," *Telekomun. Tenaga List.*, vol. 7, no. 1, pp. 2–5, 2024, doi: 10.24853/resistor.7.1.73-76.
- [6] Sutrisno, "Perancangan Dan Pembuatan Sensor Suhu," *J. Teknol. Inf.*, vol. 2, no. 1, pp. 40–48, 2021.
- [7] F. Firdaus, E. Priatna, N. Hiron, N. Busaeri, "Prototype Sistem Kendali Kecepatan Motor Dc Dengan *Proportional Integral Derivative* (Pid) Controller," *J. Energy Electr. Eng.*, vol. 32, no. 1, pp. 32–40, 2022, doi: 10.37058/jeee.v4i1.4426.
- [8] MathWorks, "How PID Autotuning Works," *MathWorks Documentation*. [Online]. Available: <https://www.mathworks.com/help/slcontrol/ug/how-pid-autotuning-works.html>
- [9] F. B. Lubis A. Yanie, "Implementasi *Pulse Width Modulation* (PWM) Pada Penyaluran Limbah Cair Pupuk Kelapa Sawit Berbasis Arduino," *JET* (*Journal Electr. Technol.*, vol. 7, no. 2, pp. 39–46, 2022, doi: 10.30743/jet.v7i2.5394.
- [10] H. Timothy Wijaya, D. Pangestu, M. Timothy Sutjipto, W. Surya Widjaja, K. Octavianus Bachri, T. Nur, "Simulasi Sistem Pengendalian Motor Stepper dengan Metode *Pulse Width Modulation*," *J. Elektro*, vol. 15, no. 1, pp. 21–30, 2024, doi: 10.25170/jurnalelektro.v15i1.5120.
- [11] B. S. Panulisan *et al.*, "Pengendali Robot dengan Mikrokontroler Arduino Berbasis Smartphone Android," *J. Kridatama Sains Dan Teknologi*, vol. 5, no. 02, pp. 421–432, 2023, doi: 10.53863/kst.v5i02.937.
- [12] K. Indriawati, "Alat Monitoring Temperatur, Salinitas, dan Oksigen Terlarut Berbasis IoT pada Budi Daya Tambak Bandeng di Desa Kemangi Kabupaten Gresik," *Sewagati*, vol. 6, no. 5, pp. 1–7, 2022, doi: 10.12962/j26139960.v6i5.279.