

SISTEM KENDALI DAN MONITORING KECEPATAN MOTOR DC BERBASIS ESP32 DENGAN KENDALI *PROPORTIONAL-INTEGRAL* (PI) BERDASARKAN SUHU

Lovenzius Febrica Fajar Pradana ¹, Lusia Rakhmawati ², Endryansyah ³, Nurhayati ⁴

Program Studi Teknik Elektro, Universitas Negeri Surabaya

lovenzius.22099@mhs.unesa.ac.id¹

Abstract (English)

DC motors are widely used in automation and cooling systems due to their ease of speed regulation and fast response characteristics. However, variations in ambient temperature may affect the required motor speed, making an automatic control system necessary to adjust the motor speed accordingly. This study aims to design and implement an ESP32-based DC motor speed control and monitoring system using the Proportional-Integral (PI) control method based on temperature. The system consists of a DHT11 sensor for temperature measurement, an FC-03 optocoupler sensor for motor speed measurement, a MOSFET as the motor driver, an OLED as the local display, and a Telegram Bot for real-time monitoring. The Proportional-Integral (PI) control method is employed to regulate the Pulse Width Modulation (PWM) signal so that the motor speed follows the setpoint determined by the ambient temperature. The experimental results show that the DHT11 sensor achieved an accuracy of 99.48% with an average error of 0.52%. The PI control system produced an average stability of 96.37%, an average response time of 4.09 seconds, and an average delay of 0.46 seconds. At the 250 RPM setpoint, the implementation of PI control reduced the average error from 21.92% to 3.25% and improved the system stability from 94.06% to 96.68% compared with the system without PI control. Based on these results, the proposed ESP32-based DC motor speed control and monitoring system using the Proportional-Integral (PI) control method is capable of automatically regulating the motor speed according to temperature variations while providing real-time system monitoring.

Article History

Submitted: 17 Juni 2026

Accepted: 20 Juli 2026

Published: 21 Juli 2026

Key Words

DC motor, DHT11, ESP32, monitoring, Proportional-Integral (PI), temperature.

Abstrak (Indonesia)

Motor DC merupakan aktuator yang banyak digunakan pada sistem otomasi dan pendinginan karena memiliki karakteristik pengaturan kecepatan yang mudah serta respons yang cepat. Namun, perubahan temperatur lingkungan dapat memengaruhi kebutuhan kecepatan motor sehingga diperlukan suatu sistem kendali yang dapat menyesuaikan kecepatan motor secara otomatis. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem kendali dan *monitoring* kecepatan motor DC berbasis ESP32 menggunakan metode *Proportional-Integral* (PI) berdasarkan suhu. Sistem terdiri atas sensor DHT11 sebagai pembaca suhu, sensor *optocoupler* FC-03 sebagai pembaca kecepatan motor, MOSFET sebagai *driver* motor, OLED sebagai media tampilan lokal, serta *Telegram Bot* sebagai media *monitoring* secara *real-time*. Metode *Proportional-Integral* (PI) digunakan untuk mengatur nilai *Pulse Width Modulation* (PWM) sehingga kecepatan motor mengikuti *setpoint* yang ditentukan berdasarkan suhu lingkungan. Hasil pengujian menunjukkan bahwa sensor DHT11 memiliki akurasi sebesar 99,48% dengan *error* rata-rata 0,52%. Sistem kendali PI menghasilkan tingkat kestabilan rata-rata sebesar 96,37%, waktu respons rata-rata 4,09 detik, dan *delay* rata-rata 0,46 detik. Pada pengujian *setpoint* 250 RPM, penerapan kendali PI menurunkan *error* rata-rata dari 21,92% menjadi 3,25% serta meningkatkan kestabilan sistem dari 94,06% menjadi 96,68% dibandingkan sistem tanpa kendali PI. Berdasarkan hasil tersebut, sistem kendali dan *monitoring* kecepatan motor DC berbasis ESP32 menggunakan metode *Proportional-Integral* (PI) dapat mengatur kecepatan motor secara otomatis sesuai perubahan suhu serta menyediakan *monitoring* kondisi sistem secara *real-time*.

Sejarah Artikel

Submitted: 17 Juni 2026

Accepted: 20 Juli 2026

Published: 21 Juli 2026

Kata Kunci

Motor DC, DHT11, ESP32, *monitoring*, *Proportional-Integral* (PI), suhu

Pendahuluan

Motor DC merupakan salah satu aktuator yang banyak digunakan pada sistem otomasi, industri, sistem pendinginan, dan perangkat elektronik karena memiliki respon cepat, torsi awal yang besar, serta kemudahan dalam pengaturan kecepatannya. Penggunaan motor DC banyak dipilih karena memiliki kemampuan pengaturan kecepatan yang baik sehingga sesuai untuk aplikasi yang membutuhkan kontrol kecepatan secara presisi. Namun, motor DC memiliki karakteristik yang sensitif terhadap perubahan beban dan kondisi lingkungan sehingga kecepatan putaran motor dapat mengalami fluktuasi apabila tidak dikendalikan dengan baik. Oleh karena itu, diperlukan suatu sistem kendali yang mampu menjaga kestabilan kecepatan motor DC agar sistem dapat bekerja secara optimal (Hammoodi et al., 2020).

Berbagai metode pengendalian kecepatan motor telah banyak dikembangkan, seperti kontrol *Proportional* (P), *Integral* (I), *Proportional Integral* (PI), hingga *Proportional Integral Derivative* (PID). Di antara metode tersebut, kontrol PI menjadi salah satu metode yang paling banyak digunakan karena memiliki struktur yang sederhana, mudah diimplementasikan, serta mampu memperbaiki *steady state error* dan kestabilan sistem. Penelitian terkait kontrol PI pada motor DC menunjukkan bahwa metode PI mampu menghasilkan respon sistem yang lebih stabil, cepat, dan *robust* terhadap gangguan maupun perubahan beban. Selain itu, sekitar 95% sistem *closed loop control* di industri menggunakan keluarga PID terutama PI karena kemampuannya dalam menjaga kestabilan sistem kontrol (Alejandro-Sanjines et al., 2023).

Selain faktor beban, temperatur juga menjadi salah satu parameter penting dalam sistem pendinginan otomatis. Pada sistem pendinginan, peningkatan suhu lingkungan membutuhkan peningkatan sirkulasi udara agar proses pendinginan dapat berjalan lebih optimal. Penelitian mengenai *temperature controlled DC fan* menunjukkan bahwa kipas DC yang dikendalikan berdasarkan suhu mampu meningkatkan efisiensi sirkulasi udara serta membantu menjaga kestabilan temperatur sistem secara otomatis. Oleh karena itu, suhu dapat digunakan sebagai acuan dalam menentukan kecepatan putaran motor DC sehingga sistem mampu bekerja secara adaptif terhadap perubahan kondisi lingkungan (Zhu et al., 2022).

Perkembangan teknologi *Internet of Things* (IoT) memberikan kemudahan dalam proses *monitoring* sistem secara *real-time*. *Monitoring* diperlukan agar pengguna dapat mengetahui kondisi sistem seperti nilai suhu, kecepatan motor, dan respon pengendali selama sistem bekerja. Dengan adanya *monitoring* berbasis IoT, proses pengamatan performa sistem menjadi lebih mudah dilakukan karena data dapat ditampilkan secara langsung melalui media *monitoring* berbasis *web* maupun perangkat *mobile*. Penelitian sebelumnya menunjukkan bahwa ESP32 memiliki kemampuan yang baik dalam membaca dan memproses data sensor serta mengirimkan data melalui koneksi *Wi-Fi* untuk kebutuhan *monitoring* sistem secara *real-time* (Abu Sneineh & Shabaneh, 2023).

ESP32 dipilih karena memiliki konektivitas *Wi-Fi* bawaan, kemampuan pemrosesan data yang baik, serta kompatibel dengan berbagai sensor dan sistem *monitoring*. Selain digunakan sebagai pengendali utama, ESP32 juga mampu menjalankan proses *monitoring* dan pengendalian secara bersamaan dalam satu perangkat. *Monitoring* sistem pada penelitian ini dilakukan menggunakan *Telegram Bot* sehingga pengguna dapat memantau nilai suhu dan kecepatan motor secara *real-time* melalui *smartphone*. *Telegram Bot* dipilih karena mampu mengirimkan data sensor, notifikasi otomatis ketika terjadi perubahan kondisi sistem, serta mendukung komunikasi dua arah antara pengguna dan perangkat IoT melalui jaringan internet. Selain itu, penggunaan *Telegram Bot* mempermudah pengguna dalam melakukan *monitoring* sistem berbasis IoT melalui

antarmuka yang sederhana, mudah diakses, dan tidak memerlukan aplikasi khusus selain Telegram (Maheswaran et al., 2024).

Jika dibandingkan dengan beberapa penelitian sebelumnya, sistem kendali motor DC berbasis suhu masih memiliki beberapa keterbatasan, seperti penggunaan PWM konvensional tanpa metode kontrol PI untuk menjaga kestabilan putaran motor, serta belum terintegrasi dengan sistem *monitoring real-time* berbasis *Internet of Things* (IoT). Oleh karena itu, pada penelitian ini motor DC digunakan sebagai simulasi *smart cooling fan* yang mampu menyesuaikan kecepatan putaran berdasarkan perubahan suhu menggunakan metode kontrol PI dan *monitoring real-time* melalui OLED dan *Telegram Bot*. Berdasarkan hal tersebut, penelitian ini dilakukan dengan tujuan merancang sistem kendali dan *monitoring* kecepatan motor DC berbasis ESP32 menggunakan metode *Proportional-Integral* (PI) berdasarkan suhu, serta menganalisis kinerja metode tersebut dalam mengatur kecepatan motor DC ditinjau dari respon sistem, kestabilan, dan *delay* sehingga dihasilkan sistem kendali motor DC yang lebih stabil, adaptif, dan mudah dipantau secara *real-time*.

Metode Penelitian

Penelitian ini menggunakan pendekatan eksperimen rekayasa (*engineering experiment*), yaitu proses perancangan, pembuatan, serta pengujian prototipe sistem kendali dan *monitoring* kecepatan motor DC berbasis ESP32. Pendekatan ini dipilih karena penelitian berfokus pada pengembangan perangkat yang dapat diuji secara langsung untuk mengetahui kinerja kontrol PI dengan acuan temperatur. Objek penelitian berupa satu unit prototipe *smart cooling fan* skala laboratorium yang diuji pada media simulasi ruang berukuran 30 cm × 20 cm × 20 cm dengan kondisi beban motor tetap (*no-load*), sedangkan sampel data diperoleh melalui teknik pengambilan data secara berulang pada setiap skenario pengujian, yaitu 30 kali pembacaan untuk pengujian sensor suhu, 30 titik data pada masing-masing variasi *setpoint* kestabilan motor, 5 kali pengulangan untuk pengujian respons sistem, dan 30 data perubahan *setpoint* untuk pengujian *delay*. Instrumen yang digunakan meliputi mikrokontroler ESP32 DevKit V1 sebagai pusat kendali, sensor DHT11 sebagai pembaca suhu lingkungan dengan termometer digital sebagai alat pembanding, sensor *encoder optocoupler* inframerah FC-03 sebagai pembaca kecepatan putaran motor (RPM), MOSFET IRLB3034 sebagai *driver* motor DC, serta OLED 0,96 inci dan *Telegram Bot* sebagai media *monitoring*.

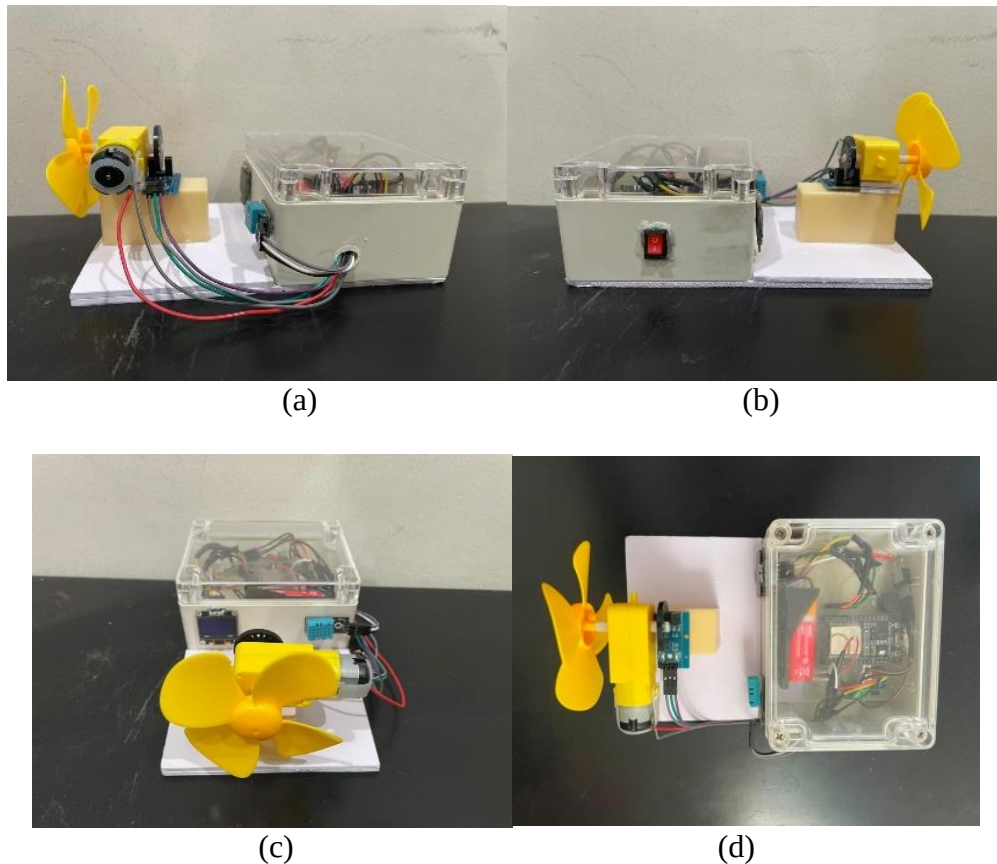
Prosedur penelitian diawali dengan identifikasi masalah dan studi literatur, dilanjutkan dengan perancangan perangkat keras berupa penyusunan blok diagram dan *wiring diagram* seluruh komponen, serta perancangan perangkat lunak berupa pemrograman ESP32 pada Arduino IDE untuk pembacaan sensor, perhitungan kendali PI, pembangkitan sinyal PWM, dan pengiriman data *monitoring*. Sistem bekerja dengan membaca suhu lingkungan melalui DHT11, kemudian menetapkan *setpoint* kecepatan secara otomatis, yaitu 150 RPM pada suhu di bawah 30°C, 200 RPM pada rentang 30–32,9°C, 250 RPM pada rentang 33–35,9°C, dan 300 RPM pada suhu $\geq 36^\circ\text{C}$. Setelah tahap implementasi dan verifikasi fungsi setiap sensor serta aktuator, dilakukan pengujian yang meliputi pengujian akurasi sensor suhu, kestabilan kecepatan motor pada empat variasi *setpoint*, respons sistem, *delay* sistem, kinerja kendali PI dengan parameter yang ditentukan melalui metode *trial and error*, serta pengujian sistem secara keseluruhan. Data hasil pengujian dianalisis secara deskriptif kuantitatif menggunakan perangkat lunak *Microsoft Excel* dengan menghitung nilai rata-rata, persentase *error* melalui rumus $\text{error} = |\text{nilai referensi} - \text{nilai aktual}| / \text{nilai referensi} \times 100\%$, tingkat kestabilan sistem melalui rumus $\text{kestabilan} = (100 - \text{error rata-rata})\%$, waktu respons melalui selisih waktu tercapainya kondisi tunak terhadap waktu perubahan

setpoint, serta *delay* melalui selisih waktu awal perubahan RPM terhadap waktu perubahan *setpoint*, yang selanjutnya dibandingkan antara sistem dengan dan tanpa kendali PI untuk menarik kesimpulan.

Hasil dan Pembahasan

Hasil Implementasi Sistem

Implementasi perangkat keras dan perangkat lunak dilakukan berdasarkan rancangan yang telah dibuat, meliputi proses perakitan rangkaian, pemrograman ESP32, integrasi sensor, serta pengujian awal sistem kendali dan *monitoring* kecepatan motor DC berbasis kontrol PI. Hasil dari perancangan tersebut ditunjukkan pada Gambar 1.



Gambar 1. Sistem Kendali dan *Monitoring* Kecepatan Motor DC, (a) Tampak Samping Kiri, (b) Tampak Samping Kanan, (c) Tampak Depan, (d) Tampak Atas

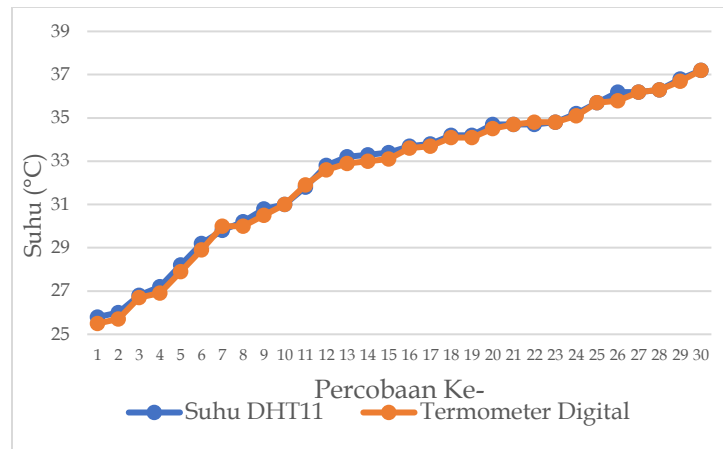
Hasil Pengujian Sensor Suhu DHT11

Pengujian sensor suhu DHT11 dilakukan untuk mengetahui kemampuan sensor dalam membaca temperatur lingkungan serta membandingkan hasil pembacaannya dengan alat ukur referensi berupa termometer digital. Pengujian dilakukan sebanyak 30 kali pada kondisi lingkungan yang relatif stabil dengan interval waktu yang sama pada setiap pengambilan data. Selisih antara kedua nilai tersebut digunakan untuk menghitung tingkat akurasi sensor dalam sistem yang dirancang. Hasil pengujian sensor suhu ditunjukkan pada Tabel 1.

Tabel 1. Hasil Pengujian Sensor Suhu

No	Suhu DHT11 (°C)	Termometer Digital (°C)	Error (%)	No	Suhu DHT11 (°C)	Termometer Digital (°C)	Error (%)
1	25.8	25.5	1.18	16	33.7	33.6	0.3
2	26	25.7	1.17	17	33.8	33.7	0.3
3	26.8	26.7	0.37	18	34.2	34.1	0.29
4	27.2	26.9	1.12	19	34.2	34.1	0.29
5	28.2	27.9	1.08	20	34.7	34.5	0.58
6	29.2	28.9	1.04	21	34.7	34.7	0
7	29.8	30	0.67	22	34.7	34.8	0.29
8	30.2	30	0.67	23	34.8	34.8	0
9	30.8	30.5	0.98	24	35.2	35.1	0.28
10	31	31	0	25	35.7	35.7	0
11	31.8	31.9	0.31	26	36.2	35.8	1.12
12	32.8	32.6	0.61	27	36.2	36.2	0
13	33.2	32.9	0.91	28	36.3	36.3	0
14	33.3	33	0.91	29	36.8	36.7	0.27
15	33.4	33.1	0.91	30	37.2	37.2	0

Berdasarkan hasil pengujian pada Tabel 1, sensor DHT11 menghasilkan rata-rata persentase *error* sebesar 0,52% terhadap termometer digital sebagai alat referensi, sehingga pembacaan sensor memiliki tingkat akurasi sebesar 99,48%. Nilai *error* terbesar terjadi pada rentang suhu rendah, yaitu 1,18% pada pembacaan 25,8°C dan 1,17% pada pembacaan 26°C, sedangkan pada rentang suhu menengah hingga tinggi selisih pembacaan semakin mengecil bahkan mencapai 0% pada beberapa titik pengukuran seperti 31°C, 34,7°C, 34,8°C, 35,7°C, 36,2°C, 36,3°C, dan 37,2°C. Kecenderungan ini menunjukkan bahwa sensor DHT11 justru semakin konsisten ketika bekerja pada rentang suhu yang menjadi acuan penentuan *setpoint* pada sistem, yaitu di atas 30°C. Perbedaan hasil pengukuran antara sensor DHT11 dan termometer digital secara keseluruhan relatif kecil sehingga data suhu yang diperoleh dapat digunakan sebagai masukan pada sistem kendali yang dirancang. Hasil ini menunjukkan tingkat akurasi yang lebih baik apabila dibandingkan dengan penelitian yang dilakukan oleh Suyanto Hadi (2022) mengenai perbandingan akurasi sensor suhu LM35 dan DHT11, yang menghasilkan tingkat akurasi sebesar 97,21% pada pengujian suhu ruangan, sehingga sensor DHT11 yang digunakan layak dijadikan acuan dalam penentuan *setpoint* kecepatan motor pada sistem kendali PI yang dirancang.



Gambar 2. Grafik Perbandingan Suhu

Berdasarkan Gambar 2, kurva pembacaan sensor DHT11 dan termometer digital terlihat hampir berimpit pada seluruh rentang pengukuran, mulai dari 25,5°C hingga 37,2°C. Kedua kurva menunjukkan pola kenaikan yang identik tanpa adanya penyimpangan yang menonjol pada titik tertentu, sehingga sensor DHT11 terbukti mampu mengikuti perubahan suhu lingkungan secara linier dan konsisten. Kondisi ini memperkuat hasil perhitungan numerik pada Tabel 1 bahwa sensor DHT11 memiliki karakteristik pembacaan yang stabil dan dapat diandalkan sebagai masukan utama sistem kendali.

Hasil Pengujian Kestabilan Kecepatan Motor

Pengujian kestabilan kecepatan motor dilakukan untuk mengetahui kemampuan sistem kendali PI dalam mempertahankan putaran motor DC pada nilai *setpoint* yang telah ditentukan. *Setpoint* RPM pada sistem ditentukan berdasarkan suhu yang terbaca oleh sensor DHT11. Pengujian dilakukan pada empat variasi *setpoint*, yaitu 150 RPM, 200 RPM, 250 RPM, dan 300 RPM. Pada setiap *setpoint*, data RPM aktual motor dicatat secara berkala menggunakan sensor *optocoupler* yang terhubung dengan ESP32. Semakin kecil selisih antara RPM aktual dengan *setpoint*, maka semakin baik performa sistem kendali yang diterapkan. Hasil pengujian kestabilan motor untuk masing-masing *setpoint* ditunjukkan pada Tabel 2 sampai dengan Tabel 5.

Tabel 2. Pengujian Kestabilan Motor pada *Setpoint* 150 RPM

No	Suhu (°C)	Setpoint	RPM Aktual	Error (%)	No	Suhu (°C)	Setpoint	RPM Aktual	Error (%)
1	26.2	150	150	0	16	26.7	150	156	4
2	26.2	150	156	4	17	26.7	150	150	0
3	26.2	150	156	4	18	27.1	150	150	0
4	26.2	150	156	4	19	27.1	150	156	4
5	26.2	150	150	0	20	27.1	150	144	4
6	26.3	150	144	4	21	27.1	150	138	8
7	26.3	150	150	0	22	27.1	150	144	4
8	26.7	150	150	0	23	27.1	150	150	0
9	26.7	150	138	8	24	27.1	150	156	4
10	26.7	150	138	8	25	27.1	150	150	0
11	26.7	150	144	4	26	27.2	150	156	4

No	Suhu (°C)	Setpoint <i>t</i>	RPM Aktual	Error (%)	No	Suhu (°C)	Setpoint <i>t</i>	RPM Aktual	Error (%)
12	26.7	150	144	4	27	27.2	150	150	0
13	26.7	150	144	4	28	27.6	150	144	4
14	26.7	150	150	0	29	27.6	150	144	4
15	26.7	150	156	4	30	27.6	150	150	0

Berdasarkan hasil pengujian pada Tabel 2, motor bekerja pada rentang suhu 26,2–27,6°C sehingga sistem menetapkan *setpoint* sebesar 150 RPM. Nilai RPM aktual yang terbaca berada pada kisaran 138–156 RPM dengan rata-rata sebesar 148,80 RPM dan *error* rata-rata sebesar 2,93%. Sebanyak sebelas titik pengukuran menghasilkan *error* 0% karena RPM aktual tepat berada pada nilai *setpoint*, sedangkan penyimpangan terbesar hanya sebesar 8% yang terjadi pada tiga titik pengukuran dengan RPM 138. Pola fluktuasi tersebut menunjukkan bahwa penyimpangan yang terjadi bersifat acak di sekitar nilai referensi, bukan berupa penyimpangan searah yang menandakan *steady state error*. Berdasarkan perhitungan kestabilan sistem, diperoleh nilai kestabilan sebesar 97,07%, sehingga sistem memiliki performa pengendalian yang sangat baik dan mampu menjaga putaran motor tetap stabil pada *setpoint* 150 RPM.

Tabel 3. Pengujian Kestabilan Motor pada *Setpoint* 200 RPM

No	Suhu (°C)	Setpoint <i>t</i>	RPM Aktual	Error (%)	No	Suhu (°C)	Setpoint <i>t</i>	RPM Aktual	Error (%)
1	30.2	200	210	5	16	32.3	200	204	2
2	30.8	200	216	8	17	32.3	200	186	7
3	30.8	200	186	7	18	32.8	200	198	1
4	30.8	200	198	1	19	32.8	200	210	5
5	30.8	200	204	2	20	32.8	200	210	5
6	31.3	200	198	1	21	32.8	200	204	2
7	31.3	200	192	4	22	32.8	200	192	4
8	31.3	200	198	1	23	32.8	200	186	7
9	31.3	200	192	4	24	32.3	200	204	2
10	31.8	200	198	1	25	32.3	200	192	4
11	31.8	200	186	7	26	32.3	200	222	11
12	31.8	200	216	8	27	32.3	200	204	2
13	31.8	200	222	11	28	32.3	200	186	7
14	32.3	200	204	2	29	32.3	200	192	4
15	32.3	200	210	5	30	32.3	200	210	5

Berdasarkan Tabel 3, pengujian dilakukan pada rentang suhu 30,2–32,8°C sehingga sistem menetapkan *setpoint* sebesar 200 RPM. Nilai RPM aktual motor berada di sekitar nilai referensi dengan kisaran 186–222 RPM, rata-rata RPM sebesar 201,40 RPM, dan *error* rata-rata sebesar 4,80%. Sebagian besar data pengukuran, yaitu sekitar dua puluh titik, memiliki *error* di bawah 5%, sementara penyimpangan terbesar sebesar 11% hanya muncul pada dua titik pengukuran dengan RPM 222. Meskipun terdapat beberapa data yang menunjukkan penyimpangan RPM yang cukup besar, sebagian besar hasil pengukuran masih berada di sekitar *setpoint* 200 RPM sehingga nilai rata-rata RPM tetap sangat dekat dengan nilai referensi. Berdasarkan perhitungan kestabilan sistem, diperoleh nilai kestabilan sebesar 95,20%. Dengan demikian, sistem kendali PI yang

diterapkan pada ESP32 tetap mampu menjaga kestabilan putaran motor dengan baik pada kondisi *setpoint* 200 RPM.

Tabel 4. Pengujian Kestabilan Motor pada *Setpoint* 250 RPM

No	Suhu (°C)	Setpoint <i>t</i>	RPM Aktual	Error (%)	No	Suhu (°C)	Setpoint <i>t</i>	RPM Aktual	Error (%)
1	33.3	250	246	1.6	16	34.7	250	258	3.2
2	33.3	250	252	0.8	17	34.7	250	258	3.2
3	33.3	250	234	6.4	18	34.7	250	234	6.4
4	33.3	250	252	0.8	19	34.7	250	246	1.6
5	33.8	250	258	3.2	20	34.7	250	252	0.8
6	33.8	250	258	3.2	21	35.2	250	258	3.2
7	33.8	250	252	0.8	22	35.2	250	258	3.2
8	33.8	250	240	4	23	35.2	250	234	6.4
9	33.8	250	258	3.2	24	35.2	250	252	0.8
10	33.8	250	240	4	25	35.2	250	240	4
11	34.2	250	234	6.4	26	35.2	250	240	4
12	34.2	250	264	5.6	27	35.6	250	234	6.4
13	34.2	250	252	0.8	28	35.6	250	264	5.6
14	34.2	250	246	1.6	29	35.6	250	252	0.8
15	34.7	250	240	4	30	35.6	250	246	1.6

Berdasarkan hasil pengujian pada Tabel 4, pengujian dilakukan pada rentang suhu 33,3–35,6°C sehingga sistem menetapkan *setpoint* sebesar 250 RPM. Nilai RPM aktual berada pada kisaran 234–264 RPM dengan rata-rata sebesar 248,40 RPM dan *error* rata-rata sebesar 3,25%. Sebagian besar data pengukuran menghasilkan *error* di bawah 4%, dengan penyimpangan terbesar sebesar 6,4% yang terjadi pada empat titik pengukuran dengan RPM 234. Distribusi data menunjukkan bahwa penyimpangan terjadi secara seimbang di atas dan di bawah nilai *setpoint*, sehingga rata-rata RPM aktual hanya berselisih 1,60 RPM dari nilai referensi. Berdasarkan perhitungan kestabilan sistem, diperoleh nilai kestabilan sebesar 96,75%, sehingga sistem memiliki performa pengendalian yang baik dan mampu menjaga putaran motor tetap stabil pada *setpoint* 250 RPM.

Tabel 5. Pengujian Kestabilan Motor pada *Setpoint* 300 RPM

No	Suhu (°C)	Setpoint <i>t</i>	RPM Aktual	Error (%)	No	Suhu (°C)	Setpoint <i>t</i>	RPM Aktual	Error (%)
1	36.3	300	300	0	16	38	300	300	0
2	36.3	300	306	2	17	38	300	294	2
3	36.3	300	294	2	18	38	300	294	2
4	36.3	300	300	0	19	38	300	312	4
5	36.4	300	306	2	20	38	300	288	4
6	36.4	300	294	2	21	38.5	300	306	2
7	36.9	300	288	4	22	38.5	300	294	2
8	36.9	300	282	6	23	38.5	300	294	2
9	36.9	300	288	4	24	38.5	300	312	4
10	36.9	300	306	2	25	39	300	306	2

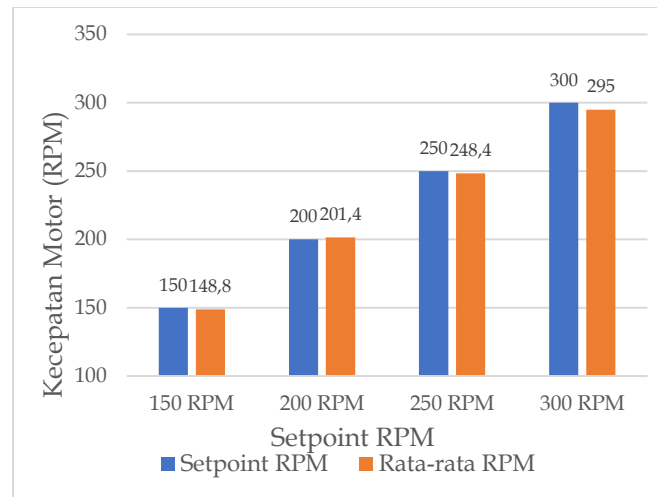
No	Suhu (°C)	Setpoint <i>t</i>	RPM Aktual	Error (%)	No	Suhu (°C)	Setpoint <i>t</i>	RPM Aktual	Error (%)
11	37.4	300	294	2	26	39	300	300	0
12	37.4	300	276	8	27	39	300	318	6
13	37.4	300	294	2	28	39	300	312	4
14	37.4	300	270	10	29	39.5	300	276	8
15	38	300	270	10	30	39.5	300	276	8

Berdasarkan hasil pengujian pada Tabel 5, pengujian dilakukan pada rentang suhu 36,3–39,5°C sehingga sistem menetapkan *setpoint* sebesar 300 RPM. Nilai RPM aktual berada pada kisaran 270–318 RPM dengan rata-rata sebesar 295,00 RPM dan *error* rata-rata sebesar 3,53%. Terdapat empat titik pengukuran yang menghasilkan *error* 0% karena RPM aktual tepat mencapai nilai *setpoint*, sedangkan penyimpangan terbesar sebesar 10% muncul pada dua titik pengukuran dengan RPM 270. Penyimpangan yang sedikit lebih besar pada *setpoint* ini disebabkan oleh kebutuhan keluaran PWM yang lebih tinggi sehingga motor bekerja mendekati batas kemampuannya. Meskipun demikian, nilai kestabilan sistem yang diperoleh tetap tinggi, yaitu sebesar 96,47%, sehingga sistem masih memiliki performa pengendalian yang baik dan mampu menjaga putaran motor tetap stabil pada *setpoint* 300 RPM.

Tabel 6. Ringkasan Hasil Pengujian Kestabilan Motor

No	Setpoint (RPM)	RPM Rata-rata (RPM)	Error Rata-rata (%)	Kestabilan (%)
1	150	148,80	2,93	97,07
2	200	201,40	4,80	95,20
3	250	248,40	3,25	96,75
4	300	295,00	3,53	96,47
Rata-rata	—	223,40	3,63	96,37

Berdasarkan Tabel 6, diperoleh nilai kestabilan sistem sebesar 97,07% pada *setpoint* 150 RPM, 95,20% pada *setpoint* 200 RPM, 96,75% pada *setpoint* 250 RPM, dan 96,47% pada *setpoint* 300 RPM. Secara keseluruhan, sistem menghasilkan nilai kestabilan rata-rata sebesar 96,37% dengan *error* rata-rata sebesar 3,63% dan RPM rata-rata sebesar 223,40 RPM. Nilai kestabilan tertinggi diperoleh pada *setpoint* 150 RPM, sedangkan yang terendah terjadi pada *setpoint* 200 RPM, namun selisih antara keduanya hanya sebesar 1,87% sehingga performa sistem dapat dikatakan konsisten pada seluruh rentang kerja. Hasil tersebut menunjukkan bahwa sistem kendali PI mampu mempertahankan kecepatan motor tetap berada di sekitar nilai *setpoint* yang ditentukan, dan meskipun terdapat fluktuasi RPM pada beberapa kondisi pengujian, penyimpangan yang terjadi relatif kecil sehingga motor tetap dapat beroperasi secara stabil pada setiap variasi suhu yang diberikan.



Gambar 3. Grafik Perbandingan *Setpoint* dan Rata-rata RPM Aktual

Berdasarkan Gambar 3, batang nilai RPM rata-rata aktual terlihat hampir sama tinggi dengan batang nilai *setpoint* pada keempat kondisi pengujian, yaitu 148,80 RPM terhadap 150 RPM, 201,40 RPM terhadap 200 RPM, 248,40 RPM terhadap 250 RPM, dan 295,00 RPM terhadap 300 RPM. Selisih terbesar hanya sebesar 5 RPM yang terjadi pada *setpoint* 300 RPM, sedangkan pada *setpoint* 200 RPM nilai RPM rata-rata justru sedikit melampaui nilai referensi. Pola ini menegaskan bahwa kendali PI mampu membawa kecepatan motor menuju nilai *setpoint* secara konsisten di seluruh rentang suhu kerja tanpa kecenderungan penyimpangan yang bertambah besar seiring naiknya nilai *setpoint*.

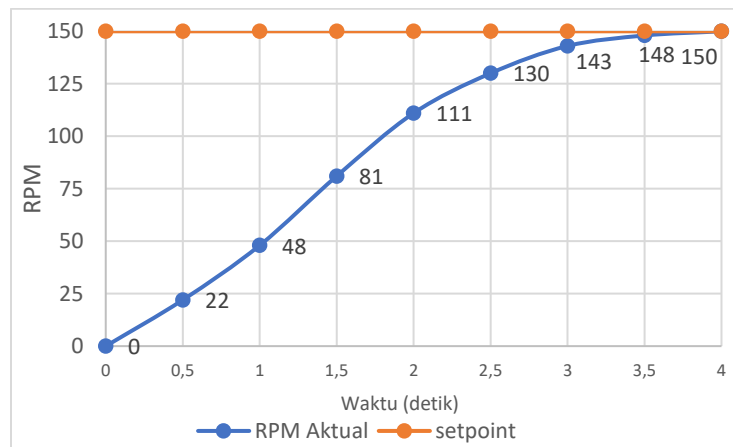
Hasil Pengujian Respons Sistem

Pengujian respons sistem dilakukan untuk mengetahui karakteristik respons kecepatan motor DC dari kondisi awal (0 RPM) hingga mencapai *setpoint* yang telah ditentukan. Pada pengujian ini digunakan *setpoint* sebesar 150 RPM, karena merupakan kondisi awal sistem ketika suhu berada pada rentang di bawah 30°C. Pengujian dilakukan sebanyak 5 kali pada kondisi yang sama, kemudian nilai RPM pada setiap interval waktu dirata-ratakan sehingga diperoleh kurva respons sistem. Hasil rata-rata pengujian respons sistem ditunjukkan pada Tabel 7.

Tabel 7. Hasil Pengujian Respons Sistem

Waktu (detik)	Uji 1 (RPM)	Uji 2 (RPM)	Uji 3 (RPM)	Uji 4 (RPM)	Uji 5 (RPM)	Rata-rata (RPM)
0,0	0	0	0	0	0	0
0,5	23	21	22	24	22	22
1,0	49	47	48	50	48	48
1,5	82	80	81	83	81	81
2,0	112	110	111	111	111	111
2,5	131	129	130	131	129	130
3,0	144	142	143	144	142	143
3,5	148	148	149	148	147	148
4,0	150	150	150	150	150	150

Berdasarkan Tabel 7, kelima pengujian menunjukkan pola kenaikan kecepatan yang hampir identik dengan selisih antar-pengujian pada setiap interval waktu tidak lebih dari 3 RPM, sehingga sistem terbukti memiliki tingkat keterulangan (*repeatability*) yang tinggi. Kecepatan motor meningkat secara bertahap dari kondisi awal 0 RPM, mencapai rata-rata 22 RPM pada detik ke-0,5, kemudian 48 RPM pada detik ke-1, 81 RPM pada detik ke-1,5, dan 111 RPM pada detik ke-2. Kenaikan tercepat terjadi pada rentang 1,0–2,0 detik dengan laju sekitar 63 RPM per detik, kemudian melambat menjadi 130 RPM pada detik ke-2,5 dan 143 RPM pada detik ke-3 seiring mengecilnya nilai *error*. Pada detik ke-3,5 motor telah mencapai 148 RPM, kemudian mencapai kondisi tunak (*steady state*) pada detik ke-4 dengan kecepatan tepat 150 RPM sesuai nilai *setpoint* tanpa disertai *overshoot* maupun osilasi.



Gambar 4. Grafik Respons Kecepatan Motor DC Menuju Setpoint 150 RPM

Berdasarkan Gambar 4, kurva respons sistem membentuk pola kenaikan menyerupai huruf S yang khas pada respons sistem orde satu dengan pengendali PI, yaitu kenaikan lambat pada awal, curam pada bagian tengah, kemudian melandai saat mendekati nilai *setpoint*. Kurva menyentuh garis *setpoint* 150 RPM pada detik ke-4 dan selanjutnya bertahan datar tanpa melewati garis referensi, yang menandakan bahwa kombinasi parameter K_p dan K_i yang digunakan tidak menimbulkan *overshoot*. Hasil ini sejalan dengan penelitian yang dilakukan oleh Aliyya et al. (2020) pada pengendalian kecepatan motor DC menggunakan metode PI dan *Fuzzy Logic Controller*, yang menunjukkan bahwa pengendali PI mampu menghasilkan respon sistem yang cepat dengan *settling time* sebesar 0,6809 detik serta mampu mencapai kecepatan referensi secara stabil meskipun masih terdapat *overshoot* pada beberapa kondisi operasi. Meskipun waktu respon pada penelitian ini lebih besar, perbedaan tersebut dapat disebabkan oleh perbedaan spesifikasi motor, metode pengukuran, parameter kendali, serta penggunaan sensor suhu sebagai pemicu perubahan *setpoint* secara bertingkat, sehingga dengan rata-rata waktu respon sebesar 4,09 detik sistem kendali PI yang diterapkan pada ESP32 tetap menunjukkan kemampuan yang baik dalam mengatur kecepatan motor DC berdasarkan perubahan suhu.

Hasil Pengujian Delay Sistem

Pengujian *delay* sistem dilakukan untuk mengetahui waktu tunda yang terjadi sejak perubahan suhu menyebabkan perubahan *setpoint* RPM hingga motor mulai memberikan respons terhadap perubahan tersebut. *Delay* sistem dihitung berdasarkan selisih waktu antara saat

perubahan *setpoint* terdeteksi (t_p) dan saat RPM motor mulai mengalami perubahan (t_r). Hasil pengujian *delay* sistem ditunjukkan pada Tabel 8.

Tabel 8. Hasil Pengujian *Delay* Sistem

Perubahan					Perubahan				
No	Setpoint (RPM)	t_p (s)	t_r (s)	Delay (s)	No	Setpoint (RPM)	t_p (s)	t_r (s)	Delay (s)
1	150 → 200	18.6	19.05	0.45	16	200 → 250	39.18	40.08	0.9
2	150 → 200	20.65	21.11	0.46	17	250 → 300	88.58	89.26	0.68
3	150 → 200	12.42	12.87	0.45	18	200 → 250	10.37	10.37	0.45
4	150 → 200	22.71	23.16	0.45	19	250 → 300	20.66	20.66	0.68
5	150 → 200	16.54	17	0.46	20	200 → 250	14.48	14.94	0.46
6	150 → 200	10.37	10.82	0.45	21	250 → 300	30.95	30.95	0.68
7	150 → 200	152.43	152.88	0.45	22	200 → 250	24.77	25.23	0.46
8	150 → 200	240.97	241.42	0.45	23	250 → 300	39.19	40.09	0.9
9	150 → 200	341.86	342.31	0.45	24	200 → 250	18.6	19.51	0.91
10	150 → 200	440.69	441.14	0.45	25	250 → 300	28.9	29.8	0.9
11	150 → 200	556	556.45	0.45	26	200 → 250	14.49	14.94	0.45
12	200 → 250	16.54	17.22	0.68	27	250 → 300	26.84	27.52	0.68
13	250 → 300	30.95	30.95	0.68	28	200 → 250	20.66	20.66	0.45
14	200 → 250	18.6	19.5	0.9	29	200 → 250	20.66	20.66	0.45
15	250 → 300	37.13	37.8	0.67	30	250 → 300	35.07	35.53	0.46

Berdasarkan hasil pengujian pada Tabel 8, diperoleh nilai *delay* rata-rata sistem sebesar 0,46 detik, yang menunjukkan bahwa sistem mampu memberikan respons awal terhadap perubahan suhu dan perubahan *setpoint* motor DC dalam waktu yang relatif singkat. Pada perubahan *setpoint* dari 150 RPM menjadi 200 RPM, nilai *delay* cenderung sangat stabil pada kisaran 0,45–0,46 detik meskipun pengujian dilakukan pada waktu t_p yang sangat bervariasi, mulai dari 10,37 detik hingga 556 detik, sehingga terbukti bahwa lamanya sistem beroperasi tidak memengaruhi kecepatan responsnya. Ketika *setpoint* dinaikkan dari 200 RPM menjadi 250 RPM, nilai *delay* mulai meningkat hingga berkisar antara 0,45–0,91 detik karena motor memerlukan waktu yang lebih lama untuk menyesuaikan keluaran PWM agar mampu mencapai kecepatan yang lebih tinggi. Kondisi serupa terlihat pada perubahan *setpoint* dari 250 RPM menjadi 300 RPM dengan nilai *delay* pada kisaran 0,46–0,90 detik. Meskipun terjadi peningkatan *delay* pada perubahan *setpoint* yang lebih besar, waktu tunda tersebut masih tergolong singkat sehingga sistem tetap mampu merespons perubahan suhu dengan baik. Hasil ini sejalan dengan penelitian yang dilakukan oleh Aliyya et al. (2020) yang menganalisis karakteristik respons transien motor DC, yang menunjukkan bahwa nilai *time delay* yang kecil tidak memberikan pengaruh signifikan terhadap kestabilan sistem dan kemampuan motor dalam mencapai kondisi tunak, sehingga nilai *delay* sebesar 0,46 detik yang diperoleh masih tergolong baik karena relatif kecil dibandingkan waktu respons sistem secara keseluruhan.

Analisis Kinerja Kontrol PI

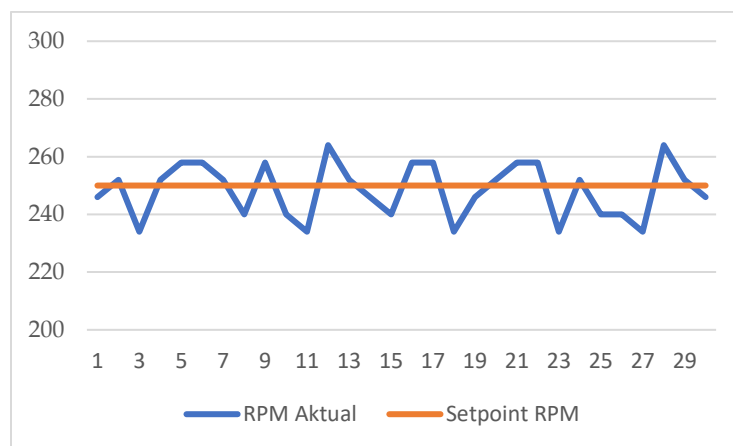
Penentuan parameter kendali PI dilakukan menggunakan metode *trial and error*, yaitu dengan memberikan beberapa nilai parameter *Proportional* (K_p) dan *Integral* (K_i) secara bertahap, kemudian mengamati respon sistem terhadap perubahan *setpoint*. Berdasarkan beberapa

percobaan yang dilakukan, diperoleh nilai parameter kendali yang memberikan performa terbaik yaitu $K_p = 0,4$ dan $K_i = 0,03$. Kombinasi parameter tersebut dipilih karena mampu menghasilkan putaran motor yang mendekati *setpoint* dengan *error* rata-rata yang rendah serta kestabilan sistem yang tinggi. Selanjutnya, pengujian kendali PI dilakukan dengan membandingkan sistem yang menggunakan kendali PI dengan sistem tanpa kendali PI pada *setpoint* 250 RPM. Parameter yang diamati meliputi RPM rata-rata, *error* rata-rata, dan kestabilan sistem. Hasil perbandingan pengujian ditunjukkan pada Tabel 9.

Tabel 9. Perbandingan Sistem Dengan dan Tanpa Kendali PI

No	Metode Pengendalian	Setpoint (RPM)	RPM Rata-rata (RPM)	Error Rata-rata (%)	Kestabilan (%)
1	Tanpa Kendali PI	250	195,20	21,92	94,06
2	Kendali PI ($K_p = 0,4$; $K_i = 0,03$)	250	248,40	3,25	96,68

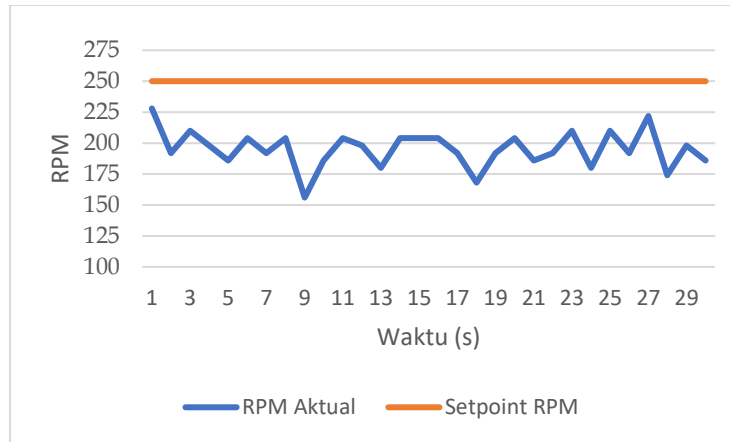
Berdasarkan Tabel 9, sistem tanpa kendali PI menghasilkan RPM rata-rata sebesar 195,20 RPM dengan *error* rata-rata 21,92% dan kestabilan sebesar 94,06%, yang menunjukkan bahwa motor belum mampu mengikuti *setpoint* 250 RPM dengan baik sehingga terjadi penyimpangan kecepatan sebesar 54,80 RPM dari nilai referensi. Setelah diterapkan kendali PI dengan parameter $K_p = 0,4$ dan $K_i = 0,03$, RPM rata-rata meningkat menjadi 248,40 RPM sehingga selisih terhadap *setpoint* menyusut menjadi hanya 1,60 RPM, *error* rata-rata menurun tajam menjadi 3,25% atau berkurang sekitar 85% dari kondisi sebelumnya, dan kestabilan meningkat menjadi 96,68%. Hasil tersebut menunjukkan bahwa kendali PI mampu memperbaiki performa sistem dengan mengurangi *error* serta menjaga putaran motor tetap berada di sekitar nilai *setpoint* yang diinginkan. Temuan ini sejalan dengan penelitian yang dilakukan oleh Qori Ariyansyah (2023) yang menerapkan kendali PID pada motor DC berbasis *encoder* dan Arduino, yang menunjukkan bahwa penerapan kendali mampu menghasilkan respon sistem yang stabil dan mempertahankan kecepatan motor agar tetap mengikuti *setpoint* yang ditentukan melalui proses *trial and error* parameter kendali.



Gambar 5. Grafik Respon Sistem Dengan Kendali PI pada *Setpoint* 250 RPM

Berdasarkan Gambar 5, kurva RPM aktual sistem dengan kendali PI terlihat berosilasi secara rapat di sekitar garis *setpoint* 250 RPM sepanjang durasi pengamatan, dengan amplitudo

penyimpangan yang kecil dan terdistribusi seimbang di atas maupun di bawah garis referensi. Tidak terdapat kecenderungan kurva menjauh dari garis *setpoint* pada arah tertentu, sehingga aksi *integral* pada pengendali terbukti berhasil menghilangkan *steady state error* yang bersifat menetap. Pola inilah yang menghasilkan nilai kestabilan sebesar 96,68% pada Tabel 9.



Gambar 6. Grafik Respon Sistem Tanpa Kendali PI pada *Setpoint* 250 RPM

Berdasarkan Gambar 6, kurva RPM aktual sistem tanpa kendali PI justru berada secara konsisten di bawah garis *setpoint* 250 RPM sepanjang pengamatan, tanpa adanya mekanisme koreksi yang membawa kecepatan motor kembali menuju nilai referensi. Penyimpangan yang terjadi bersifat searah dan menetap, sehingga terbentuk *steady state error* yang besar dan menghasilkan RPM rata-rata hanya sebesar 195,20 RPM. Perbandingan antara Gambar 5 dan Gambar 6 memperjelas bahwa kendali PI yang diterapkan pada ESP32 terbukti efektif dalam meningkatkan akurasi pengaturan kecepatan motor DC.

Pembahasan Keseluruhan Sistem

Sistem kendali dan *monitoring* kecepatan motor DC berbasis ESP32 dengan metode *Proportional-Integral* (PI) telah berhasil direalisasikan dan diuji secara keseluruhan. Sistem bekerja dengan memanfaatkan sensor DHT11 untuk membaca suhu lingkungan secara *real-time*, di mana nilai suhu yang diperoleh digunakan sebagai dasar penentuan *setpoint* kecepatan motor secara otomatis, yaitu 150 RPM pada suhu di bawah 30°C, 200 RPM pada suhu 30°C hingga kurang dari 33°C, 250 RPM pada suhu 33°C hingga kurang dari 36°C, dan 300 RPM pada suhu sama dengan atau lebih dari 36°C. Setelah *setpoint* ditentukan, pengendali *Proportional-Integral* (PI) mengatur sinyal PWM yang diberikan kepada motor DC berdasarkan selisih antara *setpoint* dan kecepatan aktual yang dibaca oleh sensor *optocoupler*. Berdasarkan hasil pengujian, sistem mampu mencapai dan mempertahankan kecepatan motor mendekati *setpoint* yang ditentukan, yaitu pada kisaran 144–156 RPM untuk *setpoint* 150 RPM, 192–210 RPM untuk *setpoint* 200 RPM, 240–258 RPM untuk *setpoint* 250 RPM, dan 294–312 RPM untuk *setpoint* 300 RPM, sehingga metode *Proportional-Integral* (PI) terbukti mampu menjaga kestabilan kecepatan motor dengan baik.

Selain fungsi kendali, sistem juga dilengkapi dengan fasilitas *monitoring* lokal menggunakan OLED yang menampilkan informasi suhu, *setpoint* RPM, RPM aktual, dan nilai PWM secara *real-time*. Informasi tersebut memudahkan pengguna untuk mengetahui kondisi sistem secara langsung tanpa memerlukan perangkat tambahan. *Monitoring* jarak jauh dilakukan

menggunakan *Telegram Bot* yang terhubung dengan ESP32 melalui jaringan *Wi-Fi*, sehingga pengguna dapat berinteraksi dengan sistem menggunakan perintah `/start` dan `/data` untuk memperoleh informasi kondisi sistem. Kemampuan tersebut sejalan dengan hasil penelitian yang menunjukkan bahwa ESP32 dapat digunakan sebagai pengendali sekaligus media pengiriman data sensor secara *real-time* melalui jaringan nirkabel (Abu Sneineh & Shabaneh, 2023). Tampilan *monitoring* melalui *Telegram Bot* ditunjukkan pada Gambar 7.



Gambar 7. Tampilan *Monitoring Sistem Menggunakan Telegram Bot*

Berdasarkan Gambar 7, *Telegram Bot* berhasil menerima dan memproses perintah dari pengguna serta menampilkan informasi suhu, kecepatan motor (RPM), dan nilai PWM secara *real-time*. Selain itu, sistem juga dilengkapi dengan LED merah dan *buzzer* sebagai indikator peringatan ketika suhu mencapai atau melebihi 36°C. Secara keseluruhan, seluruh komponen yang terdiri dari sensor DHT11, sensor *optocoupler*, ESP32, pengendali *Proportional-Integral* (PI), motor DC, OLED, *Telegram Bot*, LED indikator, dan *buzzer* telah bekerja secara terintegrasi sesuai dengan fungsi yang dirancang sehingga tujuan penelitian berhasil dicapai. Kemudahan pemantauan melalui antarmuka Telegram ini juga sejalan dengan temuan bahwa *Telegram Bot* mampu mendukung komunikasi dua arah antara pengguna dan perangkat IoT tanpa memerlukan aplikasi khusus (Maheswaran et al., 2024).

Kesimpulan

Berdasarkan hasil perancangan, implementasi, pengujian, dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa sistem kendali dan *monitoring* kecepatan motor DC berbasis ESP32 menggunakan metode *Proportional-Integral* (PI) berdasarkan suhu berhasil dirancang dan diimplementasikan dengan memanfaatkan sensor DHT11 sebagai pembaca suhu lingkungan yang

memiliki akurasi 99,48% dengan *error* rata-rata 0,52% untuk penentuan *setpoint* kecepatan motor secara otomatis, sensor *optocoupler* sebagai pembaca kecepatan motor, OLED sebagai media *monitoring* lokal, *Telegram Bot* sebagai media *monitoring* jarak jauh secara *real-time*, serta LED dan *buzzer* sebagai indikator peringatan sehingga seluruh komponen bekerja secara terintegrasi; metode *Proportional-Integral* (PI) terbukti mampu mengatur kecepatan motor DC sesuai perubahan suhu dengan tingkat kestabilan rata-rata sebesar 96,37%, waktu respons rata-rata 4,09 detik, dan *delay* rata-rata 0,46 detik, di mana pada pengujian *setpoint* 250 RPM penerapan kendali PI mampu menurunkan *error* rata-rata dari 21,92% menjadi 3,25% serta meningkatkan kestabilan sistem dari 94,06% menjadi 96,68% dibandingkan sistem tanpa kendali PI, sehingga sistem yang dirancang dapat menyesuaikan kecepatan motor secara otomatis terhadap perubahan suhu dan menyediakan *monitoring* kondisi sistem secara *real-time*.

Referensi

- Abu Sneineh, A., & Shabaneh, A. A. A. (2023). Design of a smart hydroponics monitoring system using an ESP32 microcontroller and the Internet of Things. *MethodsX*, 11. <https://doi.org/10.1016/j.mex.2023.102401>
- Alejandro-Sanjines, U., Maisincho-Jivaja, A., Asanza, V., Lorente-Leyva, L. L., & Peluffo-Ordóñez, D. H. (2023). Adaptive PI Controller Based on a Reinforcement Learning Algorithm for Speed Control of a DC Motor. *Biomimetics*, 8(5). <https://doi.org/10.3390/biomimetics8050434>
- Aliyya, J., Arif, Y. C., & Qudsi, O. A. (2020). Performance Comparison of PI Control and Fuzzy Logic Control for Speed Control of DC Shunt Motor Using Matlab/Simulink. *Journal on Advanced Research in Electrical Engineering*, 4(2).
- Hammoodi, S. J., Flayyih, K. S., & Hamad, A. R. (2020). Design and implementation speed control system of DC motor based on PID control and matlab simulink. *International Journal of Power Electronics and Drive Systems*, 11(1), 127–134. <https://doi.org/10.11591/ijpeds.v11.i1.pp127-134>
- Maheswaran, S., Gomathi, R. D., Sathesh, S., Poovizhi, S., Ridhish, R., Nanthakkumaran, S., & Chinnadurai, M. (2024). Intelligent Cold Chain Security: Nano Power Temperature Sensors, ESP32 and Telegram Bot Integration for Temperature Assurance and Environmental Harm Prevention. *Journal of Environmental Nanotechnology*, 13(1), 17–25. <https://doi.org/10.13074/jent.2024.03.241492>
- Qori Ariyansyah. (2023). *Penerapan Kendali PID pada Pengaturan Kecepatan Motor DC Berbasis Encoder dan Arduino*. Skripsi, tidak dipublikasikan.
- Suyanto Hadi. (2022). Perbandingan Akurasi Pengukuran Sensor LM35 dan Sensor DHT11 untuk Monitoring Suhu Berbasis Internet of Things. *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*.
- Zhu, Y., Zhang, T., Ma, Q., & Fukuda, H. (2022). Thermal Performance and Optimizing of Composite Trombe Wall with Temperature-Controlled DC Fan in Winter. *Sustainability (Switzerland)*, 14(5). <https://doi.org/10.3390/su14053080>