

Rancang Bangun Sistem Kendali Kecepatan Putaran Motor DC Menggunakan PID Controller

Ramadhan Fajar Dwi Kuncahyo

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya
email : ramadhan.19065@mhs.unesa.ac.id

Puput Wanarti Rusimamto, Endryansyah, Nur Kholis

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya
email : puputwanarti@unesa.ac.id, endryansyah@unesa.ac.id, nurkholis@unesa.ac.id

Abstrak

Motor DC adalah alat yang dapat digunakan mengubah energi listrik arus searah menjadi energi mekanik. Pengaplikasian pada motor ini disesuaikan dengan kebutuhannya, salah satu fungsinya digunakan untuk mensupply beban DC seperti aplikasi yang berdaya rendah sehingga cocok untuk diaplikasikan pada kecepatan daya rendah sampai sedang. Kecepatan motor ini harus dikendalikan kegunaannya untuk mendapat kecepatan yang dibutuhkan agar memudahkan penelitian, salah satunya dengan mengontrol nilai PWM (*Pulse Width Modulation*) ketika mengoperasikan alat tersebut. Untuk mendapatkan performa yang konstan penelitian ini menggunakan PID Controller dan Arduino sebagai sistem kendalinya. Setelah melakukan lima kali percobaan pada dua metode didapat hasil yang signifikan dengan melihat respon dinamik yang terdapat pada *software* Matlab. Berdasarkan kinerja dari pengujian menggunakan PID variable *overshoot*, *steady state error*, dan *settling time* mendapat hasil yang lebih kecil dibandingkan dengan tidak menggunakan PID. Kecepatan putaran Motor DC lebih konstan jika menggunakan kontrol PID. Metode *Ziegler-Nichols* untuk menentukan parameter kendali PID dimulai dengan memberikan nilai input setpoint 1000 sebagai langkah awal mendapatkan parameter Kc dan Lc untuk menentukan nilai Kp, Ki, dan Kd. Diperoleh nilai Kp = 0,0048; Ki = 0,0181; Kd = 0,000318.

Kata Kunci: Motor DC, Kendali, PID Controller.

Abstract

A DC motor is a device that can be used to convert direct current electrical energy into mechanical energy. The application of this motor is tailored to its needs, one of its functions is to supply DC loads such as low power applications so it is suitable for application at low to medium power speeds. The speed of this motor must be controlled to obtain the required speed to facilitate research, one of which is by controlling the PWM (*Pulse Width Modulation*) value when operating the tool. To obtain constant performance, this research uses a PID Controller and Arduino as the control system. After carrying out five experiments on two methods, significant results were obtained by looking at the dynamic response in the Matlab software. Based on the performance of testing using PID variable *overshoot*, *steady state error*, and *settling time*, the results were smaller compared to not using PID. The rotation speed of a DC motor is more constant when using PID control. The *Ziegler-Nichols* method for determining PID control parameters begins by providing an input setpoint value of 1000 as the initial step to obtain the Kc and Lc parameters to determine the Kp, Ki and Kd values. Obtained value Kp = 0.0048; Ki = 0.0181; Kd = 0.000318.

Keywords: Motor DC, Control, PID Controller.

Universitas Negeri Surabaya

PENDAHULUAN

Jenis motor listrik yang biasa disebut dengan motor DC memiliki fungsinya masing-masing (Sadi dan Putra, 2018). Motor DC yang digunakan untuk mensupply beban-beban DC adalah salah satu alat listrik yang paling sering digunakan di industri (Prabowo dkk., 2020). Motor listrik arus searah dapat digunakan untuk menghasilkan daya mekanik. Motor DC ini biasanya digunakan untuk aplikasi daya rendah karena cocok untuk kecepatan rendah hingga daya sedang. Motor listrik sering digunakan dalam dunia industri seperti mesin bubut, dan lainnya. Selain di

industri kegunaan motor ini juga sering digunakan sebagai objek penelitian universitas. Menurut Esario dan Yuhendri (2020), kecepatan motor DC ini harus dikendalikan untuk memenuhi berbagai kebutuhan. Pramanda dan Aswardi (2020) menyatakan bahwa motor DC ini membutuhkan sistem untuk beroperasi. Motor DC sendiri bertujuan sebagai pengatur kecepatan dari sebuah motor dan mengontrol nilai *Pulse Width Modulation* (PWM) ketika motor tersebut beroperasi.

Motor DC yang biasanya dipakai dalam mendukung kegiatan di industri menjadikan motor tersebut harus memiliki kestabilan dari kecepatannya (Suryatini dan Firasanti, 2018). Pada saat ini, teknik pengendalian PID

adalah yang paling banyak digunakan di bidang industri. Kontrol PID dapat mengukur sistem yang dikendalikan dengan tepat melalui hasil feedback keluaran dengan memunculkan sinyal kesalahan yang terjadi maupun nilai (Firdaus dkk., 2022).

Putra dkk. (2021) yang merancang prototipe kendali kecepatan motor DC dengan metode *Fuzzy Integral Controller* (FIC) menunjukkan jika kontrol ini dibuat dengan menggabungkan pengendali integral dan pengendali logika *fuzzy*. Tujuan penambahan komponen integral adalah untuk menghilangkan *error steady state*. Sementara itu, dalam penelitian yang dilakukan oleh Firdaus dkk. (2022) yang menggunakan mikrokontroler Arduino Uno sebagai *port serial* dalam berkomunikasi dengan data. Di samping itu, pendekatan tuning Ziegler-Nichols telah digunakan pada kontroler PID guna menetapkan nilai setpoint untuk parameter P, I, dan D. Hasil dari kontroler ini mencakup keluaran berupa grafik yang menunjukkan data dari kecepatan motor DC.

Pada penelitian ini penulis tertarik dengan membuat dan menambahkan beberapa metode pada sistem kendalinya untuk mendapatkan performa putaran yang konstan yaitu dengan menggunakan *PID Controller* sebagai pengatur kecepatan Motor DC dan Arduino sebagai sistem kontrol pada rangkaian. *Output* dari rancangan alat ini yaitu berupa grafik yang terdapat pada *software* MATLAB untuk mengetahui respon sistem dan kecepatan putaran motor.

Dari latar belakang yang sudah dijelaskan dapat dirumuskan permasalahannya yaitu bagaimana rancangan dan kinerja dari sistem kendali kecepatan putaran Motor DC menggunakan *PID Controller*. Tujuannya yaitu untuk merancang dan membuat, serta mengetahui kinerja sistem kendali kecepatan putaran Motor DC menggunakan *PID Controller*. Pemanfaatan menggunakan kontroler ini yaitu untuk mengurangi *overshoot* serta mempercepat *steady state*.

KAJIAN PUSTAKA

Motor DC

Alat elektromagnetis yang berperan dalam menghasilkan energi mekanik dari energi listrik disebut dengan motor listrik. Energi mekanik digunakan untuk berbagai fungsi, seperti menggerakkan kompresor, memutar impeller pompa, mengangkat bahan, dan fan atau blower. Motor elektrik juga bisa dimanfaatkan dalam rumah tangga, seperti pada bor listrik, mixer, dan kipas angin. Diperkirakan bahwa sekitar 70% dari konsumsi daya listrik industri berasal dari penggunaan motor elektrik, sehingga motor ini sering disebut sebagai "mesin kerja" dalam industri. Motor DC bisa didefinisikan sebagai perangkat yang mampu mengubah energi listrik arus searah menjadi energi gerak, yang dihasilkan melalui rotasi rotor (Bagia dan Parsa,

2018). Gambar 1 dan tabel 1 menjelaskan mengenai alat yang digunakan dan spesifikasi dari alat tersebut.



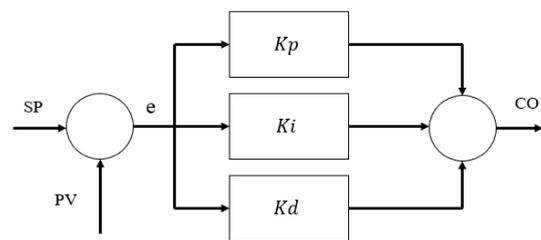
Gambar 1. Motor DC
(Sumber Bagia dan Parsa, 2018)

Tabel 1. Spesifikasi Motor DC

Spesifikasi	Keterangan
Tegangan input	12 V
Kecepatan	3000 RPM
Arus tanpa beban	1 A

PID Controller

PID Controller adalah sistem kontrol yang digunakan untuk mengukur presisi dalam sistem instrumentasi dengan menggunakan umpan balik sebagai karakteristik utama, gambar 2 menjelaskan mengenai blok dari sistem kontrol tersebut. Melibatkan pengaturan komponen fisik yang dihubungkan secara terstruktur untuk mengendalikan suatu kondisi menuju kondisi yang diinginkan. Input, proses, dan output adalah tiga komponen utama dari sistem ini. Sistem kontrol dapat dibagi menjadi dua jenis, yaitu loop terbuka (*open loop*) dan loop tertutup (*closed loop*). Pada dasarnya, sistem kontrol merupakan proses pengaturan atau pengendalian satu atau lebih variabel atau parameter sehingga nilainya berada pada nilai tertentu atau dalam kisaran nilai tertentu pada gambar 2. (Odinanto dkk., 2015).



Gambar 2. Diagram Blok Kontrol PID

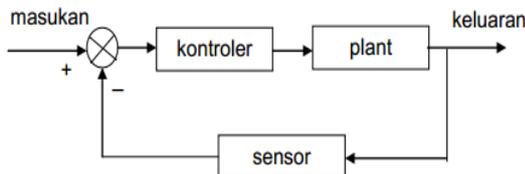
- K_p : Konstanta Proporsional
- K_i : Konstanta Integral
- K_d : Konstanta Detivatif
- SP : Setpoint
- PV : Process Variable
- e : Error
- CO : Control Output

Dikarenakan strukturnya yang sederhana, ketahanannya terhadap gangguan eksternal, dan

kemampuannya beradaptasi dengan berbagai kondisi operasional, pengontrol PID merupakan jenis pengontrol yang banyak digunakan dalam proses industri. Untuk memaksimalkan kinerja sistem, pengontrol PID harus disesuaikan dengan tepat. Hal ini melibatkan penentuan nilai-nilai konstanta pengontrol proporsional, integral, dan derivatif yang sesuai. Kontroler proporsional (P) dapat mengurangi waktu naik, meningkatkan *overshoot*, dan mengurangi kesalahan *steady state*. Kontroler integral (I) dapat mengurangi waktu naik, meningkatkan *overshoot*, dan menghilangkan kesalahan *steady state*. Sedangkan kontroler derivatif (D) dapat mengurangi *overshoot* dan mengurangi waktu penyelesaian.

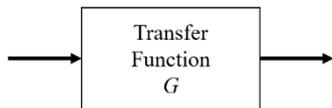
Close Loop

Close loop merupakan sistem kontrol umpan balik atau bisa dilihat pada gambar 3 jika sistem menutup. Sinyal kesalahan penggerak dalam sistem kendali *close loop* merupakan perbedaan antara sinyal umpan balik dan sinyal masukan dikirimkan ke pengontrol untuk membawa nilai *output* yang diinginkan dan mengurangi kesalahan sistem (Ogata, 2010).



Gambar 3. Blok Diagram *Close Loop*
(Sumber : Pramanda dan Aswardi, 2020)

Pada gambar 3 yaitu diagram blok suatu sistem menunjukkan fungsi yang dilakukan oleh setiap komponen dan aliran sinyal. Diagram ini menunjukkan hubungan yang ada di antara komponen.



Gambar 4. Elemen Diagram Blok
(Sumber : Ogata, 2010)

Gambar 4 elemen diagram blok menunjukkan panah yang mengarah ke blok adalah *input*, dan *output* ditunjukkan dengan panah yang menjauh dari blok. Panah diartikan sebagai sinyal. Sementara fungsi transfer $G(s)$ mewakili plant yang menjadi target kontrol.

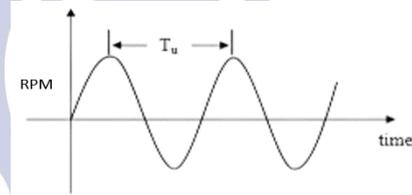
Metode Tuning Ziegler Nichols

Metode *tuning Ziegler-Nichols* adalah pendekatan penyetelan PID yang dapat dilakukan secara otomatis tanpa memerlukan pemodelan sistem (Allu dan Toding, 2020). Metode ini didasarkan pada pengujian *close loop* dan *open loop*. Untuk mencapai tujuan ini, parameter sistem harus ditemukan secara eksperimental. Hasil dari proses ini dapat ditafsirkan menjadi data yang tepat untuk persamaan kontroler PID *Ziegler-Nichols*.

$$G_c = K_p \left(1 + \frac{1}{T_i} + T_d \right) \tag{1}$$

- G_c : Gain Controller
- K_p : Konstanta Proporsional
- T_i : Periode integral
- T_d : Periode turunan

Dalam metode *Ziegler-Nichols*, pada persamaan (1) penguatan integral (K_i) dan keuntungan derivatif (K_d) diganti dengan waktu integral (T_i) dan waktu turunan (T_d) (Hui Hui Chin, 2006). K_p , T_i , dan T_d tersebut menentukan nilai gain controller (G_c) yang didapat.



Gambar 5. Pengukuran *close loop Ziegler-Nichols*

- T_u : Periode *ultimate*

Pedoman *Ziegler-Nichols* untuk Menyetel Kontroler PID yaitu metode ini memanfaatkan pedoman untuk mengidentifikasi nilai gain proporsional (K_p), waktu integral (T_i), dan waktu derivatif (T_d) berdasarkan karakteristik respon transien plant yang spesifik. Terdapat dua pendekatan yang dikenal sebagai metode pertama dan kedua dari aturan penyetelan *Ziegler-Nichols* (Ogata, 2010). Pendekatan pertama yang terdapat pada tabel 2 bergantung pada *respons plant* terhadap input tangga dalam lingkungan loop terbuka. Kurva *respons step input* dihasilkan dari plant yang tidak memiliki integrator. Kurva plant ini kemudian digunakan untuk mengidentifikasi waktu tunda (L) dan konstanta waktu (T). Parameter-parameter yang diperoleh dari kurva ini digunakan untuk menentukan parameter kontrol PID berdasarkan aturan empiris *Ziegler-Nichols* (Allu dan Toding, 2020).

Tabel 2. Metode pertama aturan penyetelan Ziegler-Nichols berdasarkan respon plant

Tipe pengontrol	Kp	Ti	Td
P	$\frac{T}{L}$	∞	0
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{T}{L}$	$2 L$	$0.5 L$

(Sumber : Ogata, 2010)

T : Waktu respon sistem
 L : Delay reaksi sistem

Pada pendekatan kedua yang terdapat pada tabel 3, penyetelan dilakukan dalam lingkungan loop tertutup dengan menggunakan input referensi berupa fungsi step. Metode ini hanya menggunakan kontrol proporsional. Nilai K_p dinaikkan dari 0 hingga mencapai nilai kritis K_p di mana keluaran sistem mulai berayun secara kontinu dengan amplitudo yang tetap. Nilai kritis K_p ini dikenal sebagai gain critical (K_{cr}). Sistem akan berayun stabil ketika K_p mencapai nilai K_{cr} tersebut. Nilai perioda critical (T_{cr}) diperoleh ketika keluaran sistem mencapai kondisi ayunan yang stabil. Nilai perioda critical (T_{cr}) dan gain critical (K_{cr}) kemudian digunakan untuk menentukan konstanta kontrol; K_p , K_i , dan K_d (Patel, 2020).

Tabel 3. Metode kedua aturan penyetelan Ziegler-Nichols berdasarkan perolehan gain kritis K dan periode kritis P

Tipe pengontrol	Kp	Ti	Td
P	$0.5 K_{cr}$	∞	0
PI	$0.45 K_{cr}$	$\frac{1}{1.2} P_{cr}$	0
PID	$0.6 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$

(Sumber : Ogata, 2010)

K_{cr} : Gain critical
 P_{cr}/T_{cr} : Periode critical

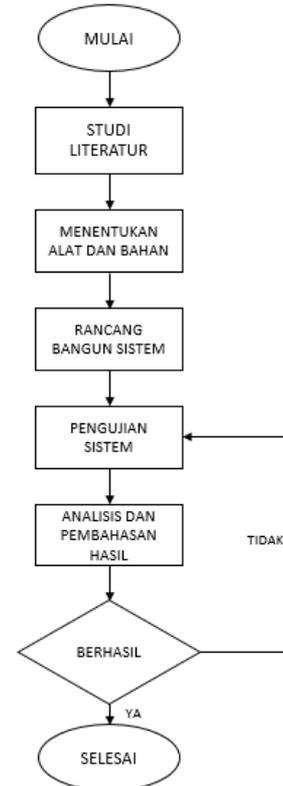
METODE

Pendekatan Penelitian

Pendekatan penelitian kuantitatif digunakan untuk penelitian ini dengan tujuan dari pendekatan ini adalah untuk menciptakan model matematis. Penelitian ini tidak hanya bergantung pada teori yang ada dalam literatur atau studi teoritis, tetapi juga mengedepankan pembangunan hipotesis yang terkait dengan fenomena alam yang akan diselidiki.

Tahapan Penelitian

Untuk merancang *prototype* sistem kendali kecepatan putaran Motor DC menggunakan PID berbasis Arduino ada beberapa tahapan penelitian yaitu tahap awal sampai akhir seperti yang terdapat pada gambar 6.



Gambar 6. Tahapan penelitian

Studi Literatur

Pada tahap awal penelitian ini dimulai dengan studi literatur yaitu dengan mencari, membaca, mengumpulkan, dan mencatat berbagai macam sumber dan referensi seperti buku, jurnal, artikel, dll yang relevan dengan penelitian yang dilakukan.

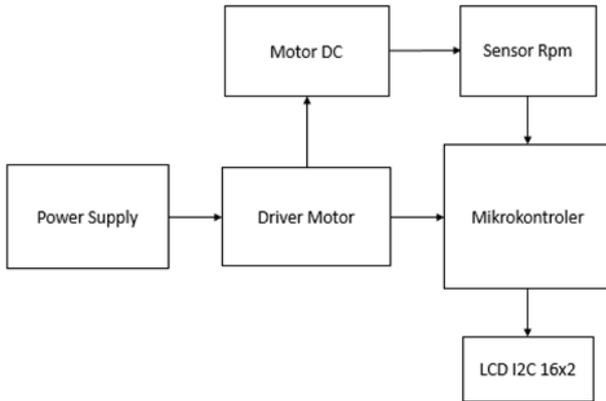
Menentukan alat dan bahan

Mendata dan mencatat kebutuhan alat dan bahan serta spesifikasi yang dibutuhkan agar penelitian seperti motor DC, catu daya, Arduino uno, driver motor, lcd 16x2, rotary encoder, pcb, kabel, dll agar sesuai dan maksimal seperti yang diharapkan oleh penulis.

Rancang bangun sistem

Dapat dijelaskan dari gambar 7 blok diagram sistem yaitu Power Supply sebagai sumber tegangan 12 volt untuk inpuatan ke driver motor, kemudian driver menerima sinyal PWM dari Arduino dan memberikan tegangan ke motor berdasarkan PWM yang diterima. Setelah itu motor bekerja dan menghasilkan putaran pada baling dari tegangan yang diberikan driver. Sensor RPM membaca kecepatan putar baling dan mengirim hasil bacaan ke Arduino. Kemudian Arduino menerima hasil pembacaan kecepatan yang telah diperoleh sensor dan menghitung

PWM optimal dengan memanfaatkan algoritma PID kemudian Arduino mengirim data RPM ke LCD untuk ditampilkan. Yang terakhir yaitu LCD menampilkan data RPM yang telah dikirim oleh Arduino.



Gambar 7. Blok diagram sistem

Pengujian sistem

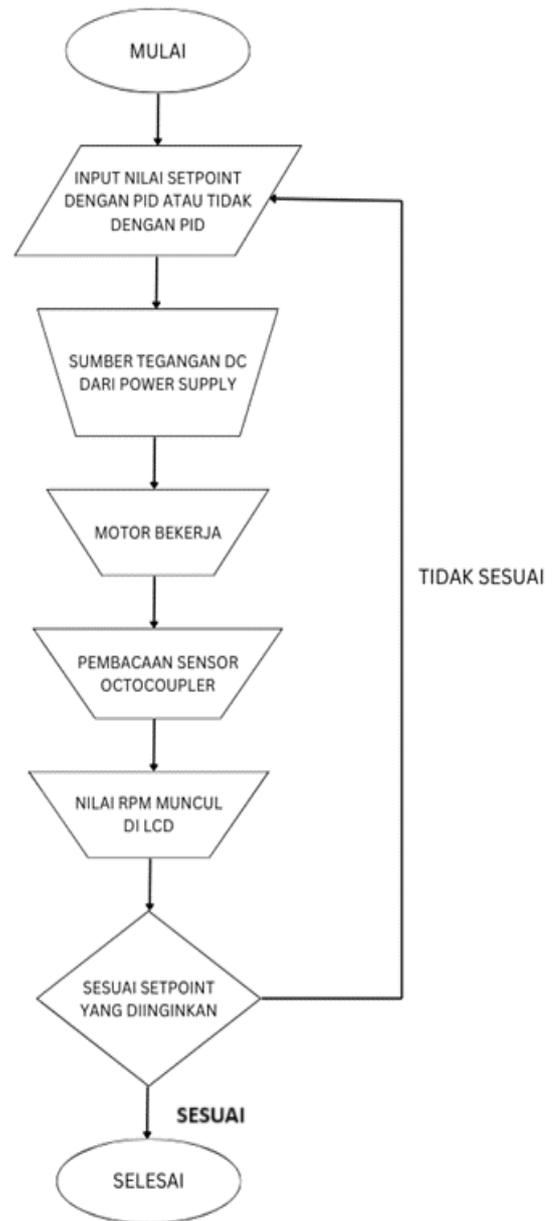
Tahapan selanjutnya setelah melakukan rancang bangun system yaitu pengujian system baik perangkat keras maupun lunak tujuannya untuk mengetahui apakah system bekerja sesuai perintah atau tidak.

Analisis dan pembahasan hasil

Tahap akhir pada penelitian ini yaitu analisis dan pembahasan setelah tahap pengujian system didapat data putaran motor melalui tampilan LCD, jika setpoint tidak sesuai maka dilakukan pengujian system ulang untuk mendapatkan setpoint yang diinginkan, jika setpoint sesuai selanjutnya yaitu membuat kesimpulan analisis dan pembahasan hasil dari prototype system kendali kecepatan putaran pada motor DC menggunakan PID controller berbasis Arduino.

Flowchart kerja sistem

Dapat dijelaskan *flowchart* dari gambar 8 maka kerja dari sistem yaitu, langkah awal memasukan nilai input setpoint pada software MATLAB dan Arduino IDE kemudian power supply 12 volt sebagai sumber tegangan input untuk menggerakkan Motor DC sehingga baling dapat berputar dan sensor RPM dapat membaca putaran dari baling tersebut. Kemudian nilai RPM muncul pada board LCD, jika dirasa nilai RPM tidak sesuai dengan setpoint maka dapat dilakukan pengujian ulang hingga nilai tersebut mendekati nilai setpoint yang diinginkan dan jika nilai RPM tersebut mendekati nilai setpoint yang diinginkan maka pengujian sistem dapat dicukupkan.



Gambar 8. Flowchart kerja sistem

HASIL DAN PEMBAHASAN

Perancangan Sistem Kendali Kecepatan Putaran Pada Motor DC Menggunakan PID Controller

Perancangan sistem ini merupakan hasil dari rancangan sistem kendali kecepatan putaran pada Motor DC menggunakan PID controller, ada beberapa komponen yang digunakan seperti Motor DC, Power Supply, Arduino Wemos D1 82966, Driver Motor, Sensor Octocoupler, dan LCD 2x16 i2c. *hardware* tersebut dapat dilihat pada gambar 9.



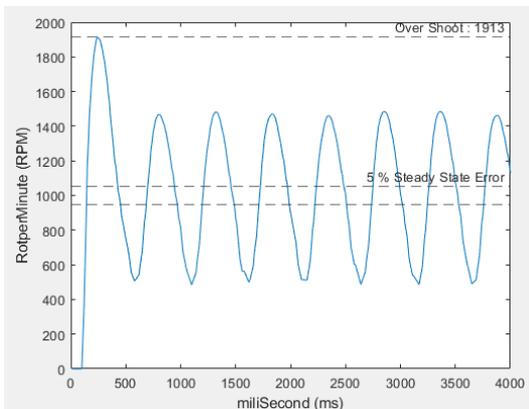
Gambar 9. Gambar sistem

Pengujian sistem

Pada pengujian kali ini dibedakan menjadi dua metode yaitu menggunakan PID dan tidak menggunakan PID kemudian terdapat lima kali percobaan pada tiap metode dengan memasukkan setpoint sebagai inputan.

Menggunakan PID

Pengambilan data dengan PID menggunakan metode tuning *Ziegler-Nichols* memanfaatkan prinsip kerja loop tertutup guna memaksimalkan fungsi *controller* pada parameter PID dan penalaan sistem. Terdapat tiga variable yang diambil yaitu *overshoot*, *steady state error*, dan *settling time*. Hasil percobaan dari metode ini dapat dilihat secara singkat pada tabel 4.

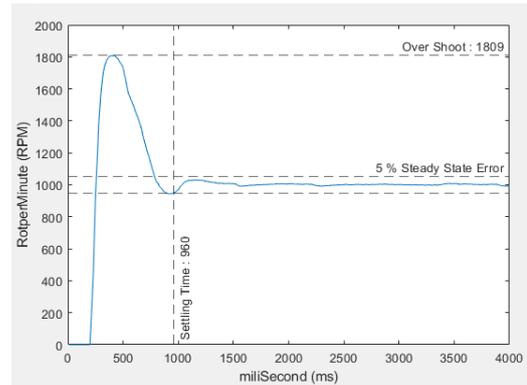


Gambar 10. Osilasi tuning *Ziegler-Nichols*

Langkah awal untuk mendapatkan nilai K_c yaitu melakukan osilasi terlebih dahulu seperti pada gambar 10 kemudian memberikan nilai setpoint 1000 RPM hingga gelombang pada grafik Matlab stabil, jika gelombang dari grafik Matlab belum stabil maka dilakukan osilasi berulang kali hingga gelombang tersebut cukup stabil, pada

penelitian kali ini didapat nilai K_c dari percobaan osilasi *tuning Ziegler-Nichols* yaitu 0,008. Nilai K_c 0,008 adalah respon awal gelombang stabil dari setpoint 1000 kemudian L_c adalah periode dari K_c , nilai puncak ke puncak 530ms dibulatkan menjadi 0,53s. Perhitungan K_p yaitu 0,6 dikali K_c 0,6 dikali 0,008 mendapatkan hasil 0,0048. Perhitungan T_i yaitu 0,5 dikali L_c 0,5 dikali 0,53 mendapatkan hasil 0,265. Perhitungan T_d yaitu 0,125 dikali L_c 0,125 dikali 0,53 mendapatkan hasil 0,06625. Perhitungan K_i yaitu K_p dibagi T_i 0,0048 dibagi 0,265 mendapatkan hasil 0,0181. Perhitungan K_d yaitu K_p dikali T_d 0,0048 dikali 0,06625 mendapatkan hasil 0,000318.

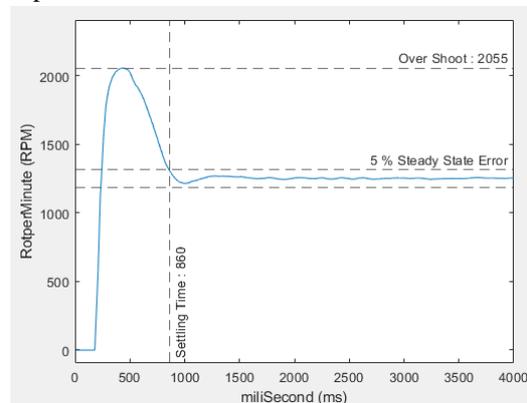
1. Setpoint 1000 RPM



Gambar 11. Setpoint 1000 RPM PID Controller

Percobaan pertama yang terdapat pada gambar 11 menggunakan PID Controller dengan setpoint 1000 mendapatkan nilai *overshoot* 1809 RPM, *settling time* 0,96 detik, dan *steady state error* 5%

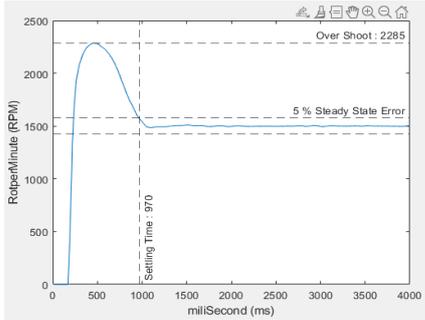
2. Setpoint 1250 RPM



Gambar 12. Setpoint 1250 RPM PID Controller

Percobaan kedua yang terdapat pada gambar 12 menggunakan PID Controller dengan setpoint 1250 mendapatkan nilai *overshoot* 2055 RPM, *settling time* 0,86 detik, dan *steady state error* 5%

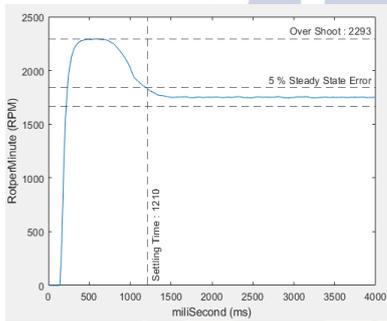
3. Setpoint 1500 RPM



Gambar 13. Setpoint 1500 RPM PID Controller

Percobaan ketiga yang terdapat pada gambar 13 menggunakan PID Controller dengan setpoint 1500 mendapatkan nilai overshoot 2285 RPM, settling time 0,97 detik, dan steady state error 5%

4. Setpoint 1750 RPM

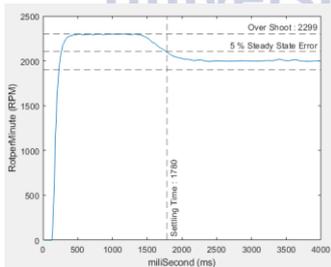


Gambar 14. Setpoint 1750 RPM PID Controller

Percobaan keempat yang terdapat pada gambar 14 menggunakan PID Controller dengan setpoint 1750 mendapatkan nilai overshoot 2293 RPM, settling time 1,21 detik, dan steady state error 5%.

Percobaan ketiga yang terdapat pada gambar 15 menggunakan PID Controller dengan setpoint 2000 mendapatkan nilai overshoot 2299 RPM, settling time 1,78 detik, dan steady state error 5%

5. Setpoint 2000 RPM



Gambar 15. Setpoint 2000 RPM PID Controller

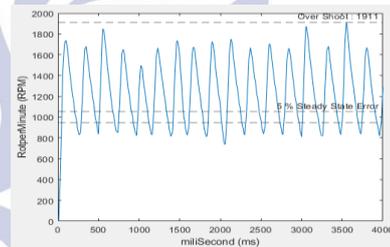
Tabel 4. Data hasil pengujian tiap setpoint menggunakan PID

Setpoint	Overshoot	Steady state error	Settling Time
1000 RPM	1809 RPM	5%	0,96 detik
1250 RPM	2055 RPM	5%	0,86 detik
1500 RPM	2285 RPM	5%	0,97 detik
1750 RPM	2293 RPM	5%	1,21 detik
2000 RPM	2299 RPM	5%	1,78 detik

Tidak menggunakan PID

Pada metode tidak menggunakan PID ini variabel yang diambil berbeda dengan metode sebelumnya yaitu overshoot dan steady state error saja, dikarenakan settling time ada ketika sistem sudah steady state. Hasil percobaan dari metode ini dapat dilihat secara singkat pada tabel 5.

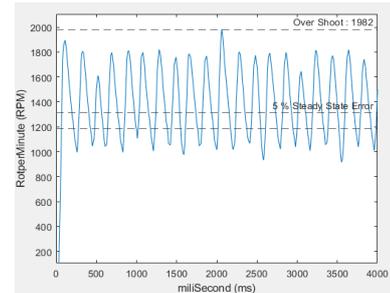
1. Setpoint 1000 RPM



Gambar 16. Setpoint 1000 RPM Non PID Controller

Percobaan pertama yang terdapat pada gambar 16 tidak menggunakan PID Controller dengan setpoint 1000 mendapatkan nilai overshoot 1911 RPM dan steady state error 5%

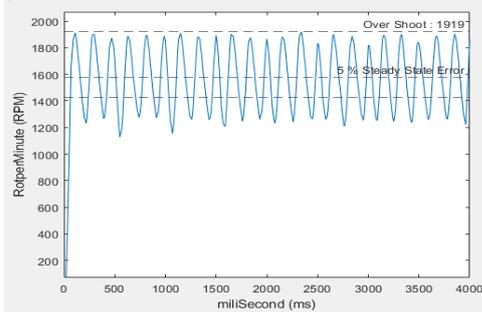
2. Setpoint 1250 RPM



Gambar 17. Setpoint 1250 RPM Non PID Controller

Percobaan kedua yang terdapat pada gambar 17 tidak menggunakan PID Controller dengan setpoint 1250 mendapatkan nilai *overshoot* 1982 RPM dan *steady state error* 5%

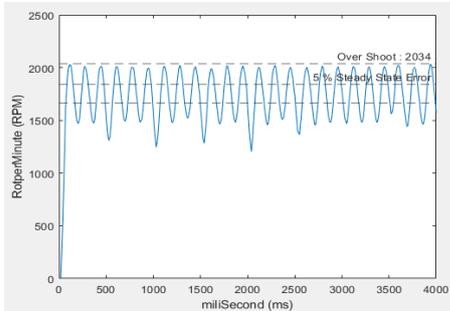
3. Setpoint 1500 RPM



Gambar 18. Setpoint 1500 RPM Non PID Controller

Percobaan ketiga yang terdapat pada gambar 18 tidak menggunakan PID Controller dengan setpoint 1500 mendapatkan nilai *overshoot* 1919 RPM dan *steady state error* 5%

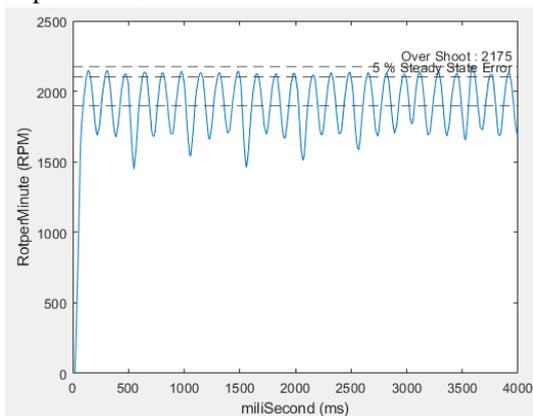
4. Setpoint 1750 RPM



Gambar 19. Setpoint 1750 RPM Non PID Controller

Percobaan keempat yang terdapat pada gambar 19 tidak menggunakan PID Controller dengan setpoint 1750 mendapatkan nilai *overshoot* 2034 RPM dan *steady state error* 5%

5. Setpoint 2000 RPM



Gambar 20. Setpoint 2000 RPM Non PID Controller

Percobaan kelima yang terdapat pada gambar 20 tidak menggunakan PID Controller dengan setpoint 1000 mendapatkan nilai *overshoot* 1911 RPM dan *steady state error* 5%

Tabel 5. Data hasil pengujian tiap setpoint tidak menggunakan PID

Setpoint	Overshoot	Steady State Error
1000 RPM	1911 RPM	5%
1250 RPM	1982 RPM	5%
1500 RPM	1919 RPM	5%
1750 RPM	2034 RPM	5%
2000 RPM	2175 RPM	5%

PENUTUP

Simpulan

Rancangan awal dan prinsip kerja sistem ini yaitu *Power Supply* sebagai sumber tegangan 12 Volt untuk inputan ke driver motor, kemudian driver menerima sinyal PWM dari Arduino dan memberikan tegangan ke motor berdasarkan PWM yang diterima. Setelah itu motor bekerja dan menghasilkan putaran pada baling dari tegangan yang diberikan driver. Sensor RPM membaca kecepatan putar baling dan mengirim hasil bacaan ke Arduino. Kemudian Arduino menerima hasil pembacaan kecepatan yang telah diperoleh sensor dan menghitung PWM optimal dengan memanfaatkan algoritma PID kemudian Arduino mengirim data RPM ke LCD untuk ditampilkan. Yang terakhir yaitu LCD menampilkan data RPM yang telah dikirim oleh Arduino.

Berdasarkan kinerja dari pengujian menggunakan PID variabel *overshoot*, *steady state error*, dan *settling time* mendapat hasil yang lebih kecil dibandingkan dengan tidak menggunakan PID. Kecepatan putaran Motor DC lebih konstan jika menggunakan kontrol PID. Penyetelan *Ziegler-Nichols* untuk menentukan parameter dalam kendali PID dimulai dengan memberikan nilai *setpoint* 1000 sebagai langkah awal untuk mendapatkan parameter K_c dan L_c yang digunakan dalam penentuan nilai K_p , K_i , dan K_d . Setelah mendapatkan nilai-nilai tersebut, respons dinamis dari berbagai percobaan dilakukan dengan menggunakan PID Controller pada berbagai setpoint. Dalam percobaan pertama dengan setpoint 1000, terjadi *overshoot* sebesar 1809 RPM, *settling time* mencapai 0,96 detik, dan *steady state error* mencapai 5%. Percobaan-percobaan selanjutnya dilakukan dengan penyesuaian *setpoint* yang berbeda, yaitu 1250, 1500, 1750, dan 2000, masing-masing menghasilkan nilai *overshoot*, *settling time*, dan *steady state error* yang berbeda pula. Hal ini

menggambarkan proses penyetelan dan evaluasi parameter kendali PID yang dilakukan dalam berbagai kondisi percobaan untuk mencapai kinerja yang optimal.

Saran

Penulis menyadari banyak kekurangan dari penelitian yang dilakukan dikarenakan minimnya biaya, maka diperlukan komponen hardware dengan kualitas yang baik agar memperoleh hasil penelitian yang maksimal. Seperti contoh pembacaan sensor RPM yang naik turun dikarenakan spesifikasi hardware yang biasa saja.

DAFTAR PUSTAKA

- Allu, N., dan Toding, A. 2020. *Tuning with Ziegler Nichols Method for Design PID Controller at Rotate Speed DC Motor*. IOP Conference Series: Materials Science and Engineering, 846(1).
- Bagia, dan Parsa. 2018. *Motor-Motor Listrik*. Kupang: CV. Rasi Terbit.
- Esario, M. I., dan Yuhendri, M. 2020. *Kendali Kecepatan Motor DC Menggunakan DC Chopper Satu Kuadran Berbasis Kontroller PI*. Jurnal Teknik Elektro Dan Vokasional, 6(1), 296–305.
- Firdaus. 2022. *Prototype Sistem Kendali Kecepatan Motor DC Dengan Proportional Integral Derivative (PID) Controller*. Journal Of Energy And Electrical Engineering, 4(1), 32–40.
- Odinanto. 2015. *Perancangan Pegendali Kecepatan Motor Arus Searah 1 Hp 220 Volt Dengan Metode PID Berbasis Mikrokontroler*. Jurnal IPTEK, 19(1), 55–61.
- Ogata, K. 2010. *Modern Control Engineering* (Fifth Edit). Prentice Hall International, Inc.
- Patel, V. V. 2020. *Ziegler-Nichols Tuning Method*. Resonance, 25(10), 1385–1397.
- Prabowo. 2020. *Pengenalan dan Penerapan Pembangkit Listrik Tenaga Surya di Desa Muara Kilis Kabupaten Tebo Jambi*. Abdimas: Jurnal Pengabdian Masyarakat Universitas Merdeka Malang, 5(1), 70–78.
- Pramanda, D., dan Aswardi. 2020. *Sistem Kendali Kecepatan Motor DC Berbasis Arduino Dengan Metode Open Loop*. Jurnal Teknik Elektro Dan Vokasional, 6(1), 187–198.
- Putra, dkk. 2021. *Pengendalian Kecepatan Motor DC Menggunakan Metode Fuzzy Integral Controller*. Prosiding Industrial Research Workshop and National Seminar, 12, 52–56.
- Sadi, S., dan Putra, I. S. 2018. *Rancang Bangun Monitoring Ketinggian Air Dan Sistem Kontrol Pada Pintu Air Berbasis Arduino Dan Sms Gateway*. Jurnal Teknik Universitas Muhammadiyah Tangerang, 7(1), 77–91.
- Suryatini, dan Firasanti. 2018. *Kendali P, PI, Dan PID Analog Pada Pengaturan Kecepatan Motor DC Dengan Penalaan Ziegler-Nichols*. Journal of Electrical and Electronics, 6(1), 65–80.