

APLIKASI ALGORITMA SARSA DALAM PENGENDALIAN MOTOR DC

Dion Dwi Wijaya¹, Joni Fat^{2*}, Viny Christanti Mawardi³

^{1,2} Program Studi Teknik Elektro, Universitas Tarumanagara; Jl. Letjen S. Parman No. 1, Jakarta Barat, 11440; Telp: 021 - 5671747

³Program Studi Teknik Informatika, Universitas Tarumanagara; Jl. Letjen S. Parman No. 1, Jakarta Barat, 11440; Telp: 021 - 5671747

Received: 18 Desember 2024

Accepted: 14 Januari 2025

Published: 20 Januari 2025

Keywords:

Motor DC;
Kontrol Adaptif;
Algoritma SARSA;
Kontroler PID;

Correspondent Email:

jonif@ft.untar.ac.id

Abstrak. Motor DC telah menjadi komponen penting dalam berbagai aplikasi industri dan perangkat elektronik. Namun, kendala dalam kestabilan kecepatan akibat variasi tegangan dan beban membutuhkan solusi kontrol yang adaptif. Penelitian ini mengimplementasikan algoritma SARSA berbasis *Reinforcement Learning* untuk mengontrol parameter *Proportional Integral Derivative* secara dinamis. Algoritma ini dirancang untuk meningkatkan respons sistem dalam menghadapi perubahan kondisi operasional. *Dataset* diperoleh melalui simulasi MATLAB, kemudian digunakan untuk melatih tabel Q yang memandu keputusan algoritma SARSA. Hasil pengujian menunjukkan bahwa sistem mampu menyesuaikan parameter PID dengan adaptif, menjaga kecepatan motor mendekati nilai target meskipun terdapat fluktuasi pada fase awal dan transien. Rata-rata kesalahan sistem sebesar 12,96%, mengindikasikan ruang untuk optimasi lebih lanjut. Penelitian ini membuktikan efektivitas SARSA dalam meningkatkan kestabilan motor DC dan berpotensi diterapkan pada sistem kontrol lainnya yang memerlukan adaptasi *real-time*.

Abstract. DC motors have become an essential component in various industrial applications and electronic devices. However, constraints in speed stability due to voltage and load variations require adaptive control solutions. This research implements the Reinforcement Learning-based SARSA algorithm to dynamically control the Proportional Integral Derivative parameters. The algorithm is designed to improve system response in the face of changing operational conditions. Datasets are obtained through MATLAB simulation, then used to train the Q table that guides the SARSA algorithm's decisions. The test results show that the system is able to adaptively adjust the PID parameters, keeping the motor speed close to the target value despite fluctuations in the initial and transient phases. The average system error is 12.96%, indicating room for further optimization. This research proves the effectiveness of SARSA in improving the stability of DC motors and has the potential to be applied to other control systems that require real-time adaptation.

1. PENDAHULUAN

Selama beberapa tahun terakhir, motor *Direct Current* (DC) telah mengalami peningkatan popularitas yang signifikan dalam berbagai aplikasi. Seiring dengan

perkembangan teknologi, penggunaan motor DC dalam berbagai bidang semakin meningkat. Dimulai dari industri berat, seperti pabrik dan otomotif, hingga penggunaan pada perangkat elektronik berukuran kecil dan presisi seperti

pada bidang medis. Penggunaan motor DC pada berbagai bidang memainkan peran penting menjadi penggerak atau aktuator pada sistem [1].

Namun, kecepatan motor DC dapat mengalami perubahan yang tidak menentu. Perubahan ini sering kali terjadi yang disebabkan oleh berbagai faktor. Salah satu faktor utama adalah perbedaan tegangan pada sistem, yang diketahui dapat mempengaruhi kestabilan rotasi motor dan performanya dalam berbagai aplikasi [2], [3]. Selain itu, penambahan beban pada motor menjadi salah satu penyebab utama penurunan kecepatan rotasi, karena torsi yang dibutuhkan meningkat seiring bertambahnya beban yang dikenakan pada motor. Dalam situasi seperti ini, kecepatan motor cenderung menurun secara signifikan, sehingga dibutuhkan sistem kontrol yang mampu beradaptasi dengan kondisi perubahan beban tersebut [3], [4].

Mengingat bertambahnya banyak motor DC yang terus digunakan, sistem kontrol yang canggih semakin dibutuhkan. Tidak hanya motor DC, tetapi sistem kontrol yang sama juga dapat mengontrol *plant* yang semakin kompleks. Salah satu contohnya berada pada industri otomotif, di mana dibutuhkan sistem kontrol yang dapat mengatur mesin yang tidak berfungsi secara linear [5]. Salah satu sistem kontrol yang dapat digunakan adalah *Proportional Integral Derivative* (PID), yang saat ini banyak digunakan. Namun, tuning parameter PID secara manual pada sistem nonlinier sering kali menjadi tantangan, karena memerlukan eksperimen yang berulang dan tidak selalu menghasilkan parameter yang optimal [6]. Penelitian yang dilakukan oleh Nassim Messadi dan Abdelkader Amroun menunjukkan bahwa metode *Fuzzy Logic* dapat digunakan untuk mengatur parameter PID secara otomatis dengan memanfaatkan aturan *fuzzy* yang menerima error kecepatan dan perubahan *error* [7].

Pengaturan mesin yang tidak berfungsi secara linear atau non-linear tidak dapat dilakukan secara konvensional [8]. Dibutuhkan pengaturan yang adaptif untuk mengatur pada situasi yang tidak konstan. Pengaturan yang adaptif ini dapat dilakukan dengan menggunakan algoritma *Reinforcement Learning* (RL). Penggunaan RL pada sistem,

memungkinkan pada identifikasi pola dalam data dan membuat keputusan terhadap pola tersebut. Sebagai contoh, penelitian yang ditulis oleh Ulbio Alejandro-Sanjines et al. menunjukkan bahwa algoritma RL dapat digunakan untuk mengatur parameter PI dengan pelatihan berbasis iterasi, menghasilkan parameter yang lebih adaptif [9], [10].

State-Action-Reward-State-Action (SARSA) merupakan suatu algoritma RL berbasis *on-policy* yang dapat digunakan untuk mempelajari kebijakan yang sesuai dalam sistem dinamis. Algoritma ini bekerja dengan menggunakan pendekatan pembelajaran berbasis nilai Q untuk menyesuaikan kebijakan pada suatu kondisi tertentu. Proses pembelajaran SARSA melibatkan iterasi berulang antara kondisi (*state*) dan tindakan (*action*), di mana setiap pasangan kondisi dan tindakan dihubungkan dengan penghargaan (*reward*) yang diberikan oleh lingkungan [11].

Penelitian ini bertujuan untuk mengaplikasikan SARSA dalam mengatur parameter PID dan mengganti metode tradisional dengan cara yang lebih stabil dan adaptif. Sistem ini diaplikasikan agar mampu menyesuaikan parameter control secara dinamis berdasarkan kondisi perubahan pada sistem. Perubahan tersebut mencakup seperti perubahan tegangan atau variasi beban, sehingga motor DC dapat beradaptasi pada kondisi-kondisi tersebut. Menggunakan kemampuan ini, sistem diharapkan mampu menjaga stabilitas rotasi motor DC dan memungkinkan motor untuk beradaptasi pada kondisi operasi yang berbeda-beda.

Algoritma SARSA yang diimplementasikan dilatih dengan menggunakan *dataset*. *Dataset* tersebut didapatkan melalui simulasi yang dilakukan dengan MATLAB. Simulasi yang dilakukan merupakan simulasi dengan pemrograman dengan bahasa MATLAB.

2. TINJAUAN PUSTAKA

Penelitian ini didasari oleh teori-teori yang relevan untuk memberikan dasar konseptual dan metodologis. Teori-teori ini digunakan untuk mendukung pengembangan sistem kontrol yang diusulkan. Berikut merupakan teori-teori yang mendasari penelitian ini.

2.1. *Proportional Integral Derivative*

PID merupakan suatu prosedur matematis yang dilakukan untuk menghitung nilai dari keluaran suatu sistem agar mencapai nilai target. PID bekerja berdasarkan 3 komponen utama yaitu proporsional (P) yang berperan untuk menghasilkan keluaran yang sebanding dengan kesalahan yang juga digunakan untuk mengurangi *error* dengan cepat. Terdapat komponen integral (I) yang berperan untuk menghitung akumulasi kesalahan sepanjang waktu dan berfungsi dalam menghilangkan kesalahan yang terjadi setelah sistem dalam kondisi stabil. Komponen terakhir yaitu derivatif (D) berperan untuk memprediksi kesalahan berdasarkan laju perubahannya untuk mengurangi *overshoot* atau *undershoot* yang terjadi pada sistem [12]. Perhitungan PID untuk mengontrol suatu sistem dapat dilakukan dengan persamaan berikut.

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

2.2. Mikrokontroler

Penelitian ini menggunakan mikrokontroler ATmega328P sebagai pengontrol serta pemroses, dengan bahasa pemrograman berbasis C++ [10]. ATmega328P adalah mikrokontroler 8-bit berbasis arsitektur AVR RISC yang memiliki 28 pin *input/output* (I/O), terdiri dari port B, port C, dan port D. Selain itu, mikrokontroler ini dilengkapi dengan 6 kanal konverter analog ke digital (ADC) dengan resolusi 10-bit, tiga timer/penghitung (dua 8-bit dan satu 16-bit), serta unit pemrosesan pusat (CPU) yang mampu mencapai kecepatan hingga 20 MHz.

2.3. Motor DC

Motor DC merupakan perangkat yang bergerak dengan mengonversi energi listrik menjadi energi mekanik dalam bentuk rotasi. Perputaran pada motor terjadi diakibatkan dengan menggunakan prinsip gaya Lorentz, di mana motor berfungsi dengan listrik yang mengalir melalui kumparan yang berada dalam medan magnet [13]. Dalam studi ini, digunakan motor DC yang bekerja dengan aliran listrik searah. Motor yang bekerja dengan aliran listrik searah dapat berputar selaras dengan arah jarum jam atau melawan arah jarum jam dengan mengubah polaritas pada kaki motor. Fungsi

alih dari motor DC dapat dirumuskan berikut ini.

$$\frac{C(s)}{R(s)} = \frac{K_t}{JL_a s^2 + (BL_a + JR_a)s + (R_a B + K_t K_e)} \quad (2)$$

Dalam persamaan diatas, K_t merupakan konstanta torsi motor, K_e adalah konstanta gaya gerak listrik baik, R_a merepresentasikan resistansi jangkar, L_a menunjukkan induktansi jangkar, J melambangkan momen inersia, dan B menyatakan koefisiensi gesekan viskositas.

2.4. Sensor

Sensor adalah sebuah komponen penting yang digunakan untuk mendeteksi lingkungan dan menjadi seperti indra pada manusia pada suatu perangkat elektronik [14]. Pada penelitian ini, digunakan sensor optocoupler, yang berfungsi mendeteksi rotasi motor menggunakan prinsip transmisi cahaya. Optocoupler terdiri dari pemancar cahaya inframerah dan fotodioda sebagai penerima. Sensor optocoupler bekerja dengan mendeteksi cahaya yang terputus akibat dari pergerakan bilah yang dipasang pada poros motor. Sinyal listrik yang dihasilkan digunakan untuk menghitung kecepatan rotasi motor.

2.5. Algoritma State-Action-Reward-State-Action

SARSA adalah salah satu algoritma yang dalam RL. Algoritma ini memiliki sifat *on-policy* dimana agen memperbaharui nilai Q berdasarkan kebijakan yang digunakan pada saat itu. SARSA digunakan untuk memecahkan masalah, dimulai dari bagaimana suatu agen dapat belajar untuk mengambil sebuah keputusan terhadap tindakan yang telah diambil di lingkungan untuk memaksimalkan total penghargaan yang diterima [15]. Pembaharuan nilai Q dapat dirumuskan menjadi persamaan berikut [16].

$$Q(S, A) \leftarrow Q(S, A) + \alpha [R + \gamma Q(S', A') - Q(S, A)] \quad (3)$$

Dalam persamaan diatas, $Q(S, A)$ merupakan nilai Q untuk pasangan *state-action*, R adalah penghargaan yang diterima setelah melakukan keputusan, dan S' , A' menjadi *state* dan *action* berikutnya sesuai dengan *policy*. Terdapat 2 konstanta yaitu α sebagai nilai tingkat pembelajaran algoritma, γ menjadi faktor diskon untuk mengatur seberapa penting sebuah penghargaan dalam jangka panjang. Studi ini

mengimplementasikan kendali motor dengan menggunakan algoritma SARSA sebagai pengatur dari parameter PID sesuai dengan kondisi yang didapatkan dari sensor.

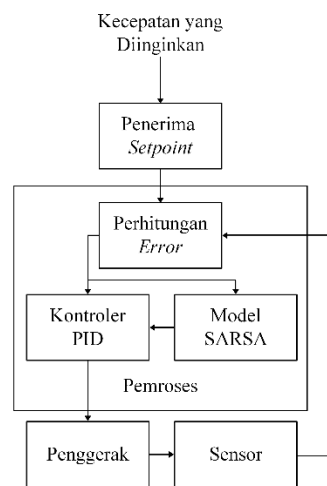
2.6. MATLAB

MATLAB adalah bahasa pemrograman tinggi, tertutup, case sensitive, dalam lingkungan komputasi numerik yang dikembangkan oleh MathWorks. MATLAB adalah singkatan dari *matrix laboratory* dikarenakan setiap data pada MATLAB menggunakan dasar *matrix* yang sering digunakan untuk menyelesaikan masalah-masalah yang melibatkan operasi matematika elemen, matrix, optimasi, aproksimasi, dan lain sebagainya [17], [18]. Pada studi ini, MATLAB digunakan untuk melakukan simulasi untuk mengakuisisi *dataset* yang digunakan untuk melatih algoritma SARSA.

3. METODE PENELITIAN

3.1. Desain Sistem

Pengaplikasian algoritma SARSA pada pengendalian motor DC menggunakan kontroler PID digambarkan dengan diagram blok berikut ini.



Gambar 1 Diagram Blok Sistem

Sementara itu, untuk setiap modul pada blok di atas dapat dijelaskan sebagai berikut.

3.1.1. Penerima Setpoint

Penerima *setpoint* berperan sebagai modul yang menerima nilai yang diinginkan pada sistem. *Input* yang diterima berupa data yang dilanjutkan ke dalam pemroses. Modul penerima menjadi antarmuka komunikasi sesuai dengan kebutuhan sistem.

3.1.2. Pemroses

Dalam sistem, ATmega328P berperan sebagai mikrokontroler utama dan pemroses. Mikrokontroler ini beroperasi dengan tegangan 5 Volt DC yang dialirkan pada pin VCC dan AVCC. Mikrokontroler berperan untuk menghitung *error* dan PID serta pemroses untuk menentukan kebijakan pada algoritma SARSA.

Pada sistem, digunakan beberapa port I/O untuk memproses instruksi kepada sistem kontrol. Port PB6 digunakan sebagai saluran keluaran pada salah satu kaki masukkan motor. Sementara itu, kaki motor yang tersisah dipasangkan kepada port PB7 pada mikrokontroler. Port PD0 berfungsi sebagai saluran masukkan dari sensor, bekerja untuk menerima dan mengolah data dari sensor.

3.1.3. Penggerak

Penggerak yang digunakan merupakan motor DC yang dikontrol sistem. Motor DC mendapatkan sinyal yang diberikan setelah perhitungan PID oleh kontroler di dalam mikrokontroler. Penggerak akan bergerak Bersama dengan bilah sesuai dengan kontrol yang sudah diberikan oleh pemroses.

3.1.4. Sensor

Modul Sensor berfungsi untuk mendeteksi bilah yang melewati diantara LED inframerah dan fotodiode. Data yang diterima kemudian diolah terhadap waktu untuk mendapatkan kecepatan actual dari motor dengan persamaan berikut.

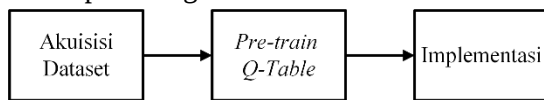
$$rpm = \frac{\text{Jumlah Pulsa} \times 60000}{\text{Waktu yang Berlalu}} \quad (4)$$

Perhitungan kecepatan actual dihitung dengan jumlah bilah melewati sensor pada suatu periode tertentu, yang dihitung dalam milidetik. Mengukur jumlah bilah dan waktu yang berlalu, kecepatan motor dapat dihitung dengan akurasi waktu yang sesuai.

3.2. Implementasi SARSA

Algoritma SARSA diimplementasikan untuk mengatur parameter PID sesuai dengan keadaan yang terjadi. SARSA diimplementasi di dalam mikrokontroler sebagai pemroses yang digunakan secara langsung dengan perhitungan *error* serta pengendalian menggunakan PID.

Proses pengembangan algoritma ini dapat dilihat pada diagram blok berikut.



Gambar 2 Diagram Blok Proses Pengembangan Algoritma SARSA

Pengembangan algoritma bermula dengan mengakuisisi *dataset*. *Dataset* didapatkan melalui simulasi menggunakan bahasa pemrograman MATLAB. *Dataset* disimulasi pada motor DC dengan sistem kontrol berbasis PID dengan konstanta *proporsional*, *integral*, dan *derivative* tanpa penggunaan beban dengan fungsi alih. Hasil data berupa data kecepatan, *setpoint*, *error*, *Kp*, *Ki*, dan *Kd*.

Dataset yang didapatkan pada simulasi kemudian digunakan untuk melatih algoritma SARSA. Pelatihan ini dilakukan untuk membentuk tabel Q yang digunakan dalam implementasi. Pelatihan ini dilakukan dengan melakukan iterasi untuk membuat nilai Q dengan pembaruan pada setiap kondisi dan aksi yang mungkin terjadi dalam sistem. Nilai Q yang dibentuk merupakan penghargaan yang diterima agen pada setiap iterasi pelatihan. Penghargaan yang didapatkan dihitung berdasarkan kesalahan yang terjadi antara kecepatan aktual dan kecepatan target.

Proses terakhir pada pengembangan algoritma SARSA merupakan proses implementasi. Implementasi yang dilakukan merupakan tabel Q yang dimasukkan ke dalam pemrograman mikrokontroler dalam bahasa C++. Tabel Q yang diimplementasikan digunakan sebagai penentu aksi dengan keadaan yang terjadi secara kenyataan.

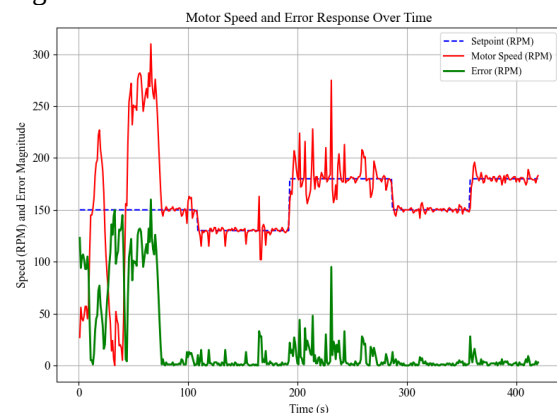
3.3. Metode Pengujian

Metode pengujian pada penelitian ini dilakukan untuk mengukur kestabilan sistem yang telah dirancang. Pengujian dilakukan dengan memanfaatkan variasi kondisi yang berbeda-beda, di mana nilai target diberikan secara berurutan dengan perubahan yang bertahap, yaitu dimulai dari 150 RPM, kemudian diubah menjadi 130 RPM, dilanjutkan kepada 180 RPM, Kembali ke 150 RPM, dan diakhiri pada 180 RPM. Proses ini dilakukan untuk memastikan bahwa sistem melakukan pengaturan parameter dengan baik secara langsung pada berbagai kondisi yang diujikan.

Setelah melakukan pengujian pada berbagai kondisi yang berbeda, sistem diperiksa akan perubahan parameter PID yang terjadi selama proses pengendalian. Pengujian ini dilakukan dengan memantau nilai parameter proporsional, integral, dan derivative secara berkala untuk melihat respons sistem terhadap perubahan kondisi. Selain itu, analisis dilakukan untuk memastikan bahwa terjadi perubahan parameter PID sesuai dengan kebutuhan pengendalian seperti mengurangi kesalahan terhadap nilai target serta meningkatkan kestabilan sistem.

4. HASIL DAN PEMBAHASAN

Dengan menggunakan implementasi di atas, sistem kontrol pengendali motor DC dapat diaplikasikan dengan algoritma SARSA. Pengujian dilakukan pada sistem untuk mengevaluasi kinerja sistem dalam mengubah parameter PID secara adaptif pada berbagai kondisi yang telah ditentukan. Hasil pengujian disajikan dalam bentuk grafik respons sistem. Grafik menunjukkan hasil respons kecepatan motor DC yang dikontrol menggunakan algoritma SARSA.



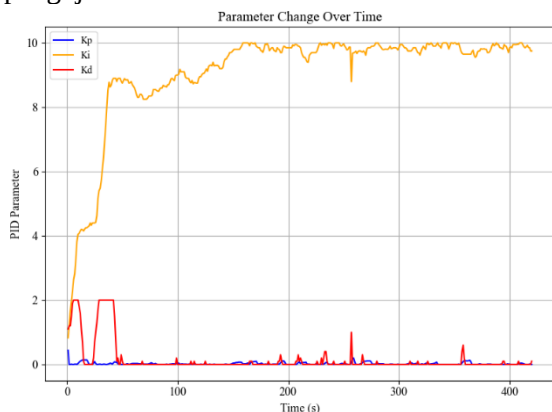
Gambar 3 Hasil Respons Kecepatan Motor dengan Kesalahan Terhadap Waktu

Hasil pengujian yang ditampilkan pada Gambar 3 menunjukkan performa sistem pengendalian kecepatan motor DC yang sudah diimplementasi dengan algoritma SARSA. Grafik di atas menunjukkan 3 elemen utama, yaitu nilai target (*setpoint*) yang ditampilkan dengan garis biru putus-putus, kecepatan motor (*motor speed*) yang ditampilkan dengan garis merah, dan kesalahan (*error*) yang ditampilkan dengan garis hijau.

Sistem pengendalian pada motor DC berhasil menyesuaikan kecepatan motor menuju nilai target yang diberikan, meskipun

terdapat fluktuasi yang cukup besar pada fase awal sistem. Hal yang serupa juga terlihat pada keadaan transien ketika nilai target diubah menjadi 180 RPM untuk yang pertama kalinya, menandakan respons adaptif algoritma SARSA terhadap perubahan kondisi. Secara keseluruhan, sistem menunjukkan respons yang kurang stabil dengan rata-rata kesalahan yang terjadi mencapai 12,96%. Nilai ini menunjukkan adanya ruang untuk peningkatan lebih lanjut, khususnya dalam meminimalkan kesalahan pada *steady state* serta osilasi selama fase transien.

Dari pengujian pada gambar sebelumnya, dapat diketahui bahwa algoritma mengubah parameter sesuai dengan keadaan. Analisis tambahan grafik parameter PID dilakukan untuk memahami bagaimana algoritma SARSA menyesuaikan parameter secara adaptif. Grafik parameter memberikan wawasan penting mengenai penyesuaian yang terjadi selama pengujian.



Gambar 4 Hasil Pengendalian Parameter Terhadap Waktu

Grafik menunjukkan perubahan parameter PID selama proses kontrol. Nilai K_p (garis biru) terlihat stabil dan memiliki perubahan kecil, menandakan pengaruh yang minim pada pengendalian. Nilai K_i (garis oranye) meningkat pesat pada awal pengujian untuk mengurangi kesalahan kumulatif dan kemudian stabil saat sistem mencapai *steady state*. Sementara itu, nilai K_d (garis merah) fluktuatif pada awal pengujian untuk merespons perubahan yang cepat, tetapi konstanta menurun setelah sistem stabil. Melalui grafik ini, algoritma SARSA mampu menyesuaikan parameter secara adaptif sesuai dengan kondisi, meskipun masih terdapat peluang untuk mengurangi fluktuasi pada saat awal dan meningkatkan kestabilan pada fase transien.

5. KESIMPULAN

Berdasarkan hasil yang didapatkan melalui pengujian yang telah dilakukan, dapat disimpulkan sebagai berikut:

- Algoritma SARSA berhasil diimplementasikan untuk mengendalikan kecepatan motor pada kondisi yang diberikan dengan menyesuaikan parameter PID secara adaptif.
- Sistem dapat mencapai nilai target kecepatan pada berbagai kondisi yang diuji. Namun, masih terdapat fluktuasi yang signifikan pada awal sistem serta kondisi transien, menunjukkan adanya ruang untuk memperbaiki respons adaptif algoritma SARSA.
- Rata-rata kesalahan sebesar 12,96% yang menunjukkan performa yang cukup baik namun masih terdapat ruang untuk perbaikan, terutama dalam meminimalkan kesalahan *steady state* dan osilasi selama transien.

UCAPAN TERIMA KASIH

Ucapan terima kasih diberikan kepada semua pihak yang telah memberikan dukungan dalam proses penelitian dan penulisan artikel ini. Terima kasih kepada keluarga, teman-teman dan kerabat yang terus membantu serta memberikan semangat. Penghargaan juga diberikan kepada Universitas Tarumanagara, khususnya kepada dosen-dosen yang telah memberikan tuntunan dan bantuan selama proses penelitian ini.

Ucapan terima kasih juga diberikan kepada penulis artikel sebelumnya yang karyanya menjadi referensi dalam penulisan ini. Dukungan dan kontribusi dari semua pihak tersebut sangat berarti dalam penyelesaian penulisan artikel ini.

DAFTAR PUSTAKA

- P. Mandal dan K. Dutt, "REVIEW ON SPEED CONTROL OF DC MOTOR USING PI & PI FUZZY LOGIC CONTROLLER," *International Journal of Creative Research Thoughts (IJCRT)*, 2021, [Daring]. Tersedia pada: www.ijcrt.org
- J. Barros, M. de Apráiz, dan R. I. Diego, "Power quality in DC distribution networks †," *Energies (Basel)*, vol. 12, no. 5, Mar 2019, doi: 10.3390/en12050848.
- A. Aripriharta dkk., "RANCANG BANGUN SENSORLESS (MINIMUM SENSOR)

- KONTROL MOTOR INDUKSI 1 FASA PADA MESIN PERONTOK PADI,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 2, Apr 2024, doi: 10.23960/jitet.v12i2.3986.
- [4] Y. Ambabunga, E. Bangapadang, dan Y. Pasuba, “TINJAUAN KARAKTERISTIK MOTOR MOTOR LISTRIK DC SEBAGAI MEDIA PEMBELAJARAN,” *Dynamic SainT*, vol. 08, no. 2, 2023, doi: <https://doi.org/10.47178/nvxs9173>.
- [5] A. Baciú dan C. Lazar, “Model-Free iPD Control Design for a Complex Nonlinear Automotive System,” dalam *2020 24th International Conference on System Theory, Control and Computing, ICSTCC 2020 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Okt 2020, hlm. 868–873. doi: 10.1109/ICSTCC50638.2020.9259724.
- [6] A. MA’ARIF, R. ISTIARNO, dan S. SUNARDI, “Kontrol Proporsional Integral Derivatif (PID) pada Kecepatan Sudut Motor DC dengan Pemodelan Identifikasi Sistem dan Tuning,” *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 9, no. 2, hlm. 374, Apr 2021, doi: 10.26760/elkomika.v9i2.374.
- [7] M. Nassim dan A. Abdelkader, “Speed Control of DC Motor Using Fuzzy PID Controller,” Agu 2021, [Daring]. Tersedia pada: <http://arxiv.org/abs/2108.05450>
- [8] A. J. Humaidi dan A. H. Hameed, “Design and comparative study of advanced adaptive control schemes for position control of electronic throttle valve,” *Information (Switzerland)*, vol. 10, no. 2, 2019, doi: 10.3390/info10020065.
- [9] I. H. Sarker, “Machine Learning: Algorithms, Real-World Applications and Research Directions,” 1 Mei 2021, *Springer*. doi: 10.1007/s42979-021-00592-x.
- [10] U. Alejandro-Sanjines, A. Maisincho-Jivaja, V. Asanza, L. L. Lorente-Leyva, dan D. H. Peluffo-Ordóñez, “Adaptive PI Controller Based on a Reinforcement Learning Algorithm for Speed Control of a DC Motor,” *Biomimetics*, vol. 8, no. 5, hlm. 434, Sep 2023, doi: 10.3390/biomimetics8050434.
- [11] Y. Tanimoto dan K. Fukumizu, “State-Separated SARSA: A Practical Sequential Decision-Making Algorithm with Recovering Rewards,” Mar 2024, [Daring]. Tersedia pada: <http://arxiv.org/abs/2403.11520>
- [12] R. Chen, “A Comprehensive Analysis of PID Control Applications in Automation Systems: Current Trends and Future Directions,” 2024.
- [13] N. Das, L. R. S. Paragond, dan B. H. Waghmare, “An Optimised Brushless DC Motor Control Scheme for Robotics Applications,” Apr 2024, [Daring]. Tersedia pada: <http://arxiv.org/abs/2404.17367>
- [14] I. M. R. A. Anantajaya, I. N. S. Kumara, dan Y. Divayana, “REVIEW APLIKASI SENSOR PADA SISTEM MONITORING DAN KONTROL BERBASIS MIKROKONTROLER ARDUINO,” *Jurnal SPEKTRUM*, vol. 8, no. 4, hlm. 171, Jan 2022, doi: 10.24843/spektrum.2021.v08.i04.p20.
- [15] S. Zhang, R. Tachet, dan R. Laroche, “On the Convergence of SARSA with Linear Function Approximation,” Feb 2022, [Daring]. Tersedia pada: <http://arxiv.org/abs/2202.06828>
- [16] A. Baradja dan T. Irianto Tjendrowasono, “PENGAPLIKASIAN DEEP REINFORCEMENT Q-LEARNING UNTUK PREDIKSI PERDAGANGAN VALAS OTOMATIS,” 2024.
- [17] I. Parinduri dan S. Nurhabibah Hutagalung, “PERANGKAIAN GERBANG LOGIKA DENGAN MENGGUNAKAN MATLAB (SIMULINK),” *JURTEKSI (Jurnal Teknologi dan Sistem Informasi)*, vol. 5, no. 1, hlm. 63–70, Jan 2019, doi: 10.33330/jurteksi.v5i1.300.
- [18] M. Fatwa, R. Ristu, S. Pandiangan, dan E. Supriyadi, “PENGAPLIKASIAN MATLAB PADA PERHITUNGAN MATRIKS,” 2022.