

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian mengenai stabilitas sistem bandul terbalik (*inverted pendulum*) terus berkembang, terutama pada aspek akurasi pembacaan sensor dan optimasi algoritma kendali. Salah satu penelitian dilakukan oleh (Zahari, 2022b) dalam pengembangan *Two-Wheeled Self-Balancing* (TWSB). Penelitian tersebut mengintegrasikan simulasi MATLAB-Simulink dengan perangkat keras berbasis mikrokontroler untuk menguji parameter kendali PID melalui model Simscape sebelum diterapkan pada alat fisik. Meskipun terbukti mampu menjaga keseimbangan di berbagai kondisi permukaan, penerapan PID pada riset tersebut masih bersifat statis, yang menyebabkan sistem sulit merespons gangguan yang muncul secara tiba-tiba.

Selanjutnya, (Donatoni et al., 2024) melakukan studi pada *linier inverted pendulum* (LIP) dengan berfokus pada teknik *swing-up* menggunakan metode berbasis energi dan penstabilan posisi tegak melalui *Linear Quadratic Regulator* (LQR). Riset ini memanfaatkan platform *open-source* Arduino IDE untuk akuisisi variabel status, yang menunjukkan bahwa perangkat dengan biaya rendah tetap dapat digunakan pada sistem kendali kompleks. Namun, temuan mereka mencatat bahwa metode alternatif seperti *Fuzzy Logic* untuk tahap *swing-up* memiliki kelemahan pada aturan "*if-then*" yang terlalu rumit dan membebani memori kontroler.

Penggunaan sensor umpan balik posisi merupakan salah satu aspek penting dalam implementasi sistem kendali *inverted pendulum*. Salah satu sensor yang dapat digunakan untuk memperoleh informasi posisi sudut adalah *incremental rotary encoder*. Sensor ini menghasilkan pulsa digital berdasarkan perubahan posisi mekanis dan umumnya memiliki dua *channel* keluaran, yaitu *channel A* dan *channel B*. Kedua *channel* tersebut memiliki perbedaan fase yang dapat digunakan untuk menentukan arah pergerakan, sedangkan jumlah pulsa yang dihasilkan digunakan untuk menentukan perubahan posisi atau sudut. (Pop, 2022) membahas penggunaan *incremental encoder* dengan mikrokontroler untuk memperoleh

informasi gerak berdasarkan pulsa yang dihasilkan oleh *encoder*. Penggunaan *encoder* sebagai sensor umpan balik memungkinkan posisi diperoleh secara langsung dari perubahan mekanis sistem tanpa memerlukan proses estimasi sudut menggunakan sensor inersia.

Visualisasi data dan estimasi sudut secara real-time juga menjadi perhatian dalam penelitian ini. Merujuk pada (Das et al., 2025), penggunaan *serial plotter* pada Arduino IDE menjadi alternatif yang jauh lebih ringan dibandingkan perangkat lunak visualisasi eksternal yang berat. Temuan tersebut menekankan bahwa fitur bawaan ini memiliki efisiensi komputasi yang sangat ringan, sehingga sangat cocok untuk aplikasi pada mikrokontroler dengan daya pemroses terbatas seperti Arduino Uno. Dengan memanfaatkan *serial plotter*, data respons sistem seperti *overshoot* dan *settling time* dapat diamati secara instan tanpa risiko latensi komunikasi data yang tinggi. Hal ini membantu menjaga kestabilan data, mempermudah proses penalaan (*tuning*) parameter kontroler secara empiris, serta membantu memahami teori kendali secara lebih mudah tanpa bergantung pada lisensi perangkat lunak yang mahal.

Berdasarkan tinjauan berbagai penelitian terdahulu, proyek akhir ini difokuskan pada implementasi sistem kendali PID untuk menstabilkan *linier inverted pendulum*. Pemilihan kontroler PID didasarkan pada kemampuannya dalam memberikan respons terhadap eror posisi serta perubahan eror, dan kemudahan implementasinya pada mikrokontroler. Berbeda dengan penelitian yang menggunakan sensor inersia dan metode estimasi sudut, penelitian ini menggunakan *incremental rotary encoder* sebagai sensor umpan balik sudut. *Encoder* digunakan untuk memperoleh perubahan posisi sudut pendulum berdasarkan pulsa yang dihasilkan dari pergerakan mekanis sistem. Parameter kontroler diperoleh melalui proses *tuning* menggunakan MATLAB dan selanjutnya diterapkan pada sistem aktual berbasis Arduino Uno dan *stepper motor*. Pengujian dilakukan untuk menganalisis kemampuan kontroler dalam mempertahankan posisi vertikal pendulum ketika menerima gangguan eksternal.

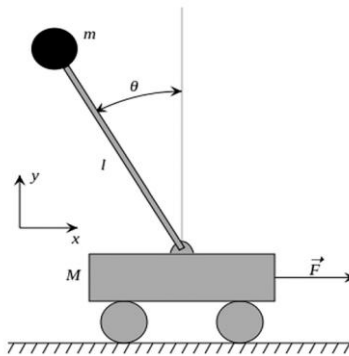
2.2 Landasan Teori

2.2.1 Linier Inverted pendulum

Linier inverted pendulum (LIP) merupakan sebuah mekanisme kontrol yang

terdiri dari batang bandul bebas yang diletakkan di atas kereta (*cart*), di mana kereta tersebut bergerak secara linier pada lintasan lurus. Sistem ini memiliki karakteristik yang tidak stabil karena pusat massa pendulum berada di atas titik tumpunya, sehingga gaya gravitasi akan selalu menarik batang untuk jatuh. Karakteristik ini menjadikan LIP sering digunakan sebagai standar pengujian (*benchmark*) dalam teknik kendali untuk menguji performa algoritma kontrol karena sistem ini bersifat *non-linear*, *underactuated*, dan sulit untuk distabilkan (Donatoni et al., 2024).

Untuk mempertahankan posisi vertikal pendulum, sistem memerlukan pergerakan horizontal kereta sebagai aksi koreksi terhadap sudut kemiringan yang terjadi. Pergerakan kereta harus dilakukan secara cepat dan presisi agar pendulum tetap berada pada titik keseimbangan.



Gambar 2.1 Parameter Fisik *Inverted pendulum*¹

Dalam melakukan pemodelan, lihat Gambar 2.1 mendefinisikan parameter-parameter fisik yang mempengaruhi dinamika gerakan sistem, parameter tersebut meliputi:

- a. M : massa kereta (*cart*) (kg)
- b. m : massa batang pendulum (kg)
- c. l : jarak dari titik poros ke pusat massa batang pendulum (m)
- d. I : momen inersia massa pendulum ($kg.m^2$)
- e. F : gaya luar (input) yang diberikan ada kereta (N)
- f. x : posisi horizontal kereta (m)
- g. θ : sudut kemiringan pendulum dari posisi vertical bawah (rad)
- h. L : panjang total kereta (m)
- i. B : koefisien gesek atau redaman pada kereta ($N/m/sec$)
- j. g : percepatan gravitasi ($9.81 m/s^2$)

¹ Irfan, S., et al. (2024). Control Strategies for Inverted Pendulum: A Comparative Analysis. PLoS ONE, 19(3), 1–19.

Dinamika sistem ini diformulasikan menggunakan pendekatan Newtonian. Berdasarkan hukum tersebut, gerakan sistem dapat dinyatakan melalui dua persamaan diferensial *non-linear* yang saling terhubung (Irfan et al., 2024):

1) Gaya pada kereta (*cart*):

$$\bullet \quad (M + m)\ddot{x} + ml\ddot{\theta} \cos\theta - ml\dot{\theta}^2 \sin\theta + B\dot{x} = F \quad (2.1)$$

2) Torsi pada pendulum:

$$\bullet \quad (I + ml^2)\ddot{\theta} + mgl \sin\theta = -ml\ddot{x} \cos\theta \quad (2.2)$$

Pada Persamaan (2.1) dan (2.2) cukup kompleks karena melibatkan fungsi trigonometri dan suku kuadratik kecepatan sudut ($\dot{\theta}^2$). Agar sistem dapat diintegrasikan dengan kontroler PID atau LQR, oleh karena itu dilakukan proses linierisasi di sekitar titik keseimbangan. Dengan asumsi nilai sudut ($\theta \approx 0$)), maka digunakan pendekatan sebagai berikut (Irfan et al., 2024):

$$\sin\theta \approx \theta \quad (2.3)$$

$$\cos\theta \approx 1 \quad (2.4)$$

$$\dot{\theta}^2 \approx 0 \text{ (mengabaikan suku orde tinggi)} \quad (2.5)$$

Dengan menggunakan pendekatan pada persamaan (2.3) dan (2.4) tersebut, maka diperoleh Persamaan Dinamika *Linear* yang digunakan sebagai dasar perancangan kendali (Irfan et al., 2024):

$$(M + m)\ddot{x} + B\dot{x} + ml\ddot{\theta} = F_a \quad (2.6)$$

$$(I + ml^2)\ddot{\theta} - mgl\theta = ml\ddot{x} \quad (2.7)$$

Persamaan *linear* ini merupakan dasar yang digunakan untuk menganalisis karakteristik sistem dan merancang algoritma kendali guna menjaga stabilitas posisi vertikal pendulum. Meskipun model matematis pada persamaan (2.6) dan (2.7) telah di*linearisasi* untuk mempermudah perancangan kendali, implementasi sistem *linier inverted pendulum* pada perangkat nyata masih memiliki beberapa kendala. Terdapat beberapa permasalahan fisik dan teknis yang mempengaruhi performa sistem, antara lain:

1) Ketidakpastian parameter fisis

Sistem *linier inverted pendulum* sangat sensitif terhadap ketidakpastian parameter seperti massa, panjang efektif, dan koefisien gesek, karena deviasi

kecil dari model matematis dapat mempengaruhi kestabilan sistem (Ogata, 2010).

2) Keterbatasan aktuator

Dalam sistem *inverted pendulum* berbasis mikrokontroler, aktuator seperti motor DC atau *stepper* memiliki keterbatasan torsi, kecepatan, dan respons dinamis. Ketika gangguan atau sudut awal terlalu besar, aktuator tidak mampu menghasilkan gaya horizontal yang cukup untuk mengembalikan pendulum ke posisi tegak. Studi eksperimental pada sistem *inverted pendulum* menunjukkan bahwa keterbatasan aktuator dapat menurunkan performa dan kestabilan sistem, terutama pada fase *swing-up* dan stabilisasi (Donatoni et al., 2024).

3) Nonlinearitas dan ketidakakuratan model linier

Model *linier inverted pendulum* diperoleh melalui proses *linearisasi* dengan pendekatan sudut kecil ($\sin \theta \approx \theta$), sehingga model hanya akurat di sekitar titik keseimbangan. Jika sudut penyimpangan membesar, model *linear* tidak lagi akurat dan dapat mempengaruhi kestabilan sistem (Ogata, 2010).

2.2.2 Kontrol PID

Kontrol PID (*Proportional-Integral-Derivative*) adalah salah satu algoritma kendali yang paling umum digunakan di industri. Sistem kendali ini merupakan algoritma kontrol yang digunakan untuk meningkatkan ketepatan respons pada sistem tertutup (*closed-loop*). Dalam tugas akhir ini, kontroler PID digunakan untuk mempertahankan keseimbangan *inverted pendulum* pada saat vertikal dengan menghitung selisih antara sudut setimbang (0°) dengan sudut aktual.

Secara matematis, keluaran PID merupakan gabungan dari tiga parameter utama yang bekerja secara bersamaan. Menurut (Paz, 2001) persamaan umum PID dapat dituliskan pada (2.8) sebagai berikut:

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (2.8)$$

dimana, $u(t)$ adalah sinyal kendali, $e(t)$ adalah *error* (selisih antara sinyal referensi dan keluaran sistem), K_p adalah penguatan proporsional, T_i adalah waktu *integral*, T_d adalah waktu derivatif, $\int_0^t e(\tau) d\tau$ menyatakan akumulasi *error*, $\frac{de(t)}{dt}$ menunjukkan perubahan *error* terhadap waktu, dan τ adalah variabel *dummy*

integrasi.

Agar sistem memiliki respons yang stabil dan *error steady-state* yang kecil, diperlukan proses *tuning* parameter K_p , K_i , dan K_d . Secara umum, terdapat tiga pendekatan yang digunakan:

- 1) Metode Heuristik (*trial and error*): Dilakukan langsung pada *hardware* melalui pengujian bertahap. Prosedur dimulai dengan menentukan K_p hingga sistem mulai memberikan respons, diikuti K_d untuk meredam osilasi, dan K_i untuk mengurangi *error steady-state*.
- 2) Metode Ziegler-Nichols (osilasi): Menentukan parameter berdasarkan nilai *ultimate gain* (K_u) dan *ultimate period* (P_u) yang diperoleh ketika sistem mengalami osilasi berkelanjutan (*sustained oscillation*). Nilai ini kemudian digunakan pada tabel standar untuk memperoleh respons sistem yang cepat dan stabil
- 3) Metode *Based Model* (berbasis model): Metode ini dilakukan dengan menurunkan persamaan matematis sistem menjadi fungsi transfer (TF). Proses *tuning* ini dapat dilakukan secara simulasi menggunakan MATLAB/PID tuner, sehingga mengurangi risiko kerusakan mekanik akibat respons awal yang berlebihan pada tahap awal pengujian. Pendekatan ini termasuk *model-based tuning* karena menggunakan parameter dan struktur internal sistem sebagai dasar pemodelan.

Dalam perancangan sistem kendali, model matematis diperlukan untuk menggambarkan perilaku sistem nyata dalam bentuk matematis yang dapat dianalisis. Secara umum, terdapat dua pendekatan utama dalam penentuan model sistem:

- 1) Metode *Black Box modeling*: Metode *Black Box* fokus pada pemodelan *input-output* tanpa mengenal struktur internal sistem, sering menggunakan data eksperimen untuk estimasi fungsi transfer. Pendekatan ini ideal saat parameter fisik sulit diukur, meski kurang memberikan pemahaman fisik yang lebih detail.
- 2) Metode *White Box modeling*: Metode *White Box* membangun model berdasarkan hukum fisika, seperti Hukum Newton II atau Euler-Lagrange untuk *inverted pendulum*. Prosesnya melibatkan persamaan diferensial *non-linear* yang dilinierkan di titik sudut 0° , menghasilkan fungsi transfer Laplace dengan

parameter seperti massa (M , m), panjang (L), gesek (b), dan inersia (I). Metode ini cocok digunakan untuk analisis kestabilan dan simulasi sistem di MATLAB, karena parameter sistem dapat dijelaskan secara fisik.

2.2.3 Arduino Uno

Board Arduino Uno, termasuk varian *Uno Plus*, merupakan salah satu mikrokontroler berbasis ATmega328P yang banyak digunakan untuk keperluan pendidikan, prototipe elektronik, dan proyek robotika. Board ini memiliki berbagai fitur *input/output* serta kemampuan pemrograman yang fleksibel, sehingga dapat digunakan untuk mengendalikan berbagai perangkat elektronik. Berdasarkan (Arduino, n.d.) tabel berikut menunjukkan spesifikasi utama Arduino Uno / Uno Plus.

Tabel 2.1 Spesifikasi Arduino Uno

Parameter	Spesifikasi
Mikrokontroler	ATmega328P
Tegangan Operasi	5 V
Tegangan Input	6 – 12 V (disarankan)
Kecepatan Clock	16 MHz
Memori Flash	32 KB (± 0.5 KB digunakan <i>bootloader</i>)
SRAM	2 KB
EEPROM	1 KB
Pin Digital I/O	14 (6 pin PWM)
Pin Analog Input	6 (beberapa varian kompatibel memiliki A6/A7 tambahan)
LED Onboard	Pin 13
Antarmuka	UART (<i>serial</i>), I ² C, SPI
USB	USB type B / <i>micro</i> USB tergantung versi
Sumber Daya	USB atau input DC eksternal
Dimensi	68.6 × 53.4 mm
Bobot	± 25 g



Gambar 2.2 Arduino Uno²

² Arduino. (n.d.). Arduino Uno Rev3. Arduino. [https://doi.org/10.1016/S0140-6736\(72\)91649-2](https://doi.org/10.1016/S0140-6736(72)91649-2)

2.2.4 Driver A4988

Driver A4988 adalah modul penggerak *stepper motor* yang berfungsi mengatur kerja *stepper motor* berdasarkan sinyal kontrol dari mikrokontroler yang berasal dari algoritma PID. Sinyal kontrol atau *output* dari algoritma PID (u) merupakan nilai hasil perhitungan aksi *proportional*, *integral*, dan *derivative*. Agar sinyal ini dapat digunakan oleh *hardware*, mikrokontroler melakukan interpretasi sinyal sebagai berikut:

1) Penentuan arah gerak (*Direction*):

Tanda positif (+) atau negatif (-) dari nilai μ menentukan arah rotasi motor. Pada sistem ini, jika nilai sudut memiliki *error* positif (miring ke kanan), PID akan menghasilkan nilai positif yang membuat pin DIR pada *driver A4988* bernilai *HIGH* agar kereta bergerak ke kanan. Sebaliknya, *error* negatif akan memicu logika *LOW* untuk gerak ke kiri.

2) Konversi kecepatan (*Pulse Frequency*):

Besarnya nilai absolut dari *output* PID menentukan seberapa cepat motor harus bergerak. Mikrokontroler akan mengubah nilai tersebut menjadi frekuensi pulsa (PWM) yang dikirim ke pin STEP.

- a. Semakin besar nilai *output* PID (karena sudut miring terlalu jauh), maka frekuensi pulsa semakin tinggi, mengakibatkan *stepper motor* berputar lebih cepat untuk mengembalikan posisi pendulum.
- b. Semakin kecil nilai *output* PID (saat mendekati titik setimbang), maka frekuensi pulsa semakin rendah, sehingga pergerakan motor melambat untuk mencegah *overshoot* (ayunan berlebih).

3) Presisi melalui *microstepping*:

Pergerakan motor dibuat lebih halus oleh *driver A4988* menggunakan fitur *microstepping*. Jika PID memberikan perintah untuk bergerak sangat sedikit, *driver* akan membagi satu langkah penuh (1.8°) motor NEMA 17 menjadi bagian-bagian kecil (hingga $1/16$ *step*). Hal ini penting agar tidak terjadi getaran kasar yang dapat mengganggu pembacaan sensor sudut saat sistem menjaga posisi tetap tegak.

Fitur *microstepping* membantu motor bergerak lebih halus, meminimalkan getaran mekanik, serta menjaga kestabilan pembacaan sensor sudut. Berdasarkan (Allegro MicroSystems, 2022) tabel 2.2 merujuk kepada spesifikasi A4988 *Driver stepper motor*.

Tabel 2.2 Spesifikasi A4988 *Driver*

Parameter	Spesifikasi
Tipe <i>Driver</i>	DMOS <i>Microstepping</i> Motor <i>Driver</i> (bipolar)
Tegangan Motor (VMOT)	8 – 35 V DC (sumber motor)
Tegangan Logika	3 – 5.5 V (masukan kontrol)
Arus Maksimum	±2 A per fase (dengan pendinginan)
Arus Kontinu	±1 A per fase (tanpa heatsink)
Mode <i>Microstepping</i>	<i>Full</i> , 1/2, 1/4, 1/8, 1/16
Proteksi	<i>Thermal shutdown, under-voltage lockout, crossover-current protection</i>
Fitur	<i>Adjustable current limit</i> (potensiometer), <i>interface</i> STEP/DIR/ENABLE
Format Board	Modul <i>breakout</i> siap pakai untuk Arduino/CNC



Gambar 2.3 A4988 *Driver*³

2.2.5 *Stepper Motor*

Stepper motor merupakan aktuator elektromekanis yang mengubah pulsa digital menjadi gerakan sudut secara bertahap dan presisi. Berbeda dengan motor DC biasa, *stepper motor* bekerja dengan mengubah pulsa menjadi pergerakan posisi yang mengikuti sinyal kendali secara langsung. Dalam sistem *linier inverted pendulum*, *stepper motor* berfungsi menggerakkan kereta (*cart*) berdasarkan pulsa dari *motor driver*.

Motor ini memiliki torsi penahan (*holding torque*) yang cukup kuat, yang membantu kereta tetap berada pada posisi yang stabil saat sudut pendulum telah mencapai *setpoint* (0°). Performa motor dipengaruhi oleh kecepatan pulsa (*step*

³ HandsOn Tech. n.d. “A4988 Stepper Motor Driver Board – HandsOn Tech.” Accessed February 10, 2026. <https://handsontec.com/index.php/a4988-stepper-motor-driver-board/>.

rate) yang diberikan, jika sistem PID membutuhkan respons cepat, motor harus mampu berakselerasi tanpa kehilangan langkah (*missing steps*). Berdasarkan (DatasheetCafe, 2024) tabel berikut menunjukkan spesifikasi utama modul *stepper motor* NEMA 17 17HS4401.

Tabel 2.3 Spesifikasi *Stepper Motor* NEMA 17 17HS4401

Parameter	Spesifikasi
Jenis Motor	Bipolar 2-fase <i>stepper motor</i>
Flange / Ukuran	42 × 42 mm (1.7 × 1.7 inci)
Panjang Motor	40 mm
Sudut Langkah	1.8° (200 langkah/rev)
Torsi <i>Holding</i>	40 N·cm
Arus Per Fase	1.7 A (12 V)
Resistansi Fase	1.5 Ω
Diameter Poros	5 mm
Jumlah Kabel / Pin	4 kabel, 6 pin plug (tergantung modul)
Panjang Kabel	400 mm
<i>Step Angle Accuracy</i>	±5% (<i>full step, no load</i>)
<i>Temperature Rise</i>	80°C Max (<i>rated current, 2 phase on</i>)
Suhu Lingkungan	-20°C ~ +50°C
Insulasi	Resistansi ≥100 MΩ, Kekuatan ≥500 VAC



Gambar 2.4 *Stepper Motor*⁴

2.2.6 Rotary Encoder

Rotary encoder adalah komponen yang mengubah gerakan putar secara mekanis menjadi sinyal pulsa digital. Pada tipe optik, sensor ini memanfaatkan pemancar dan penerima cahaya untuk mendeteksi setiap gerakan poros secara presisi melalui mekanis berikut:

- 1) Piringan roda (*Disk*): Berupa piringan tipis dengan celah-celah kecil (kisi) transparan yang dipasang seporos dengan pendulum. Piringan ini akan ikut berputar saat pendulum mulai miring.

⁴ MotionKing. 2012. "2 Phase Hybrid Stepper Motor," 1106.

- 2) Komponen sensor cahaya: Terdiri dari LED inframerah sebagai pemancar cahaya dan sebuah *photodetector* sebagai penerimanya. Keduanya dipasang saling berhadapan dengan posisi piringan roda berada di tengah-tengahnya.
- 3) Proses menjadi pulsa digital: Saat poros berputar, celah pada piringan akan memotong cahaya dari LED secara bergantian. *Photodetector* akan menangkap kedipan cahaya tersebut dan mengubahnya menjadi sinyal pulsa kotak (*square wave*).
- 4) Sinyal *quadrature*: Agar bisa mendeteksi arah putaran, *encoder* memiliki dua jalur *output data*, yaitu *channel A* dan *channel B*. Posisi kedua jalur ini sengaja digeser sedikit sehingga menghasilkan perbedaan fase sebesar 90°.
- 5) Membaca arah putaran: Mikrokontroler menentukan arah gerakan pendulum dengan melihat sinyal mana yang masuk duluan. Jika *channel A* berubah naik (*rising*) mendahului *channel B*, artinya pendulum berputar searah jarum jam (*clockwise*). Sebaliknya, jika *channel B* naik lebih dulu, artinya pendulum berputar berlawanan arah jarum jam.



Gambar 2.5 *Rotary Encoder* ⁵

Untuk menghitung nilai sudut posisi pendulum secara kuantitatif, jumlah pulsa akumulasi dari *encoder* dikonversi menjadi satuan derajat (°). Penilaian sinyal dilakukan pada setiap transisi *rising* (perubahan dari kondisi *Low* ke *High*) dan *falling* (perubahan dari kondisi *High* ke *Low*) pada jalur sinyal *encoder*.

Hubungan dasar antara satu putaran penuh (360°) dengan resolusi piringan dasar *encoder* dinyatakan dalam persamaan:

$$1 \text{ Putaran Penuh} = 360^\circ = \text{PPR (Pulse Per Revolution)}$$

⁵ Precision Incremental Rotary encoder - Optical 360 PPR Quadrature | Buy in Australia | CE09809 | Core Electronics. (n.d.). Retrieved June 19, 2026, from <https://core-electronics.com.au/360-ppr-encoder.html>

Karena pembacaan dilakukan pada setiap kondisi *rising* dan *falling*, maka dalam satu siklus pulsa utuh terhitung sebanyak 2 kondisi perubahan (*edges*). Dengan demikian, jumlah total pulsa terdeteksi dalam satu putaran penuh menjadi $PPR \times 2$. Untuk *rotary encoder* dengan spesifikasi piringan dasar 400 PPR yang dinilai pada setiap kondisi *rising* dan *falling* (sehingga menghasilkan 800 pulsa terdeteksi per putaran).

2.2.7 MATLAB

MATLAB digunakan untuk pemodelan matematis dan simulasi sistem sebelum implementasi perangkat keras. Dalam perancangan kontroler PID, MATLAB digunakan untuk menganalisis fungsi alih (*transfer function*) sistem untuk memperoleh parameter PID yang sesuai berdasarkan hasil simulasi. Fitur *PID tuner* pada MATLAB mempermudah proses *tuning* dengan menyesuaikan parameter secara otomatis untuk agar sistem memiliki respons yang sesuai:

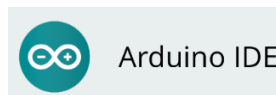
- 1) *Settling time*: Waktu yang dibutuhkan pendulum untuk berhenti berosilasi.
- 2) *Overshoot*: Batas simpangan maksimum agar pendulum tidak mengalami ayunan berlebih saat koreksi posisi.
- 3) *Stability margin*: Membantu menjaga kestabilan sistem meskipun terjadi sedikit perubahan beban atau gesekan.



Gambar 2.6 Logo MATLAB⁶

2.2.8 Arduino IDE

Arduino IDE digunakan untuk menulis *firmware* dan mengunggahnya ke mikrokontroler. Fitur *Serial Plotter* dan *Serial Monitor* dimanfaatkan sebagai sarana *data logging* secara *real-time* untuk mengamati pergerakan posisi sudut dan respons waktu dari kontroler PID.



Gambar 2.7 Serial Plotter Arduino IDE⁷

⁶ Irfan, S., et al. (2024). Control Strategies for Inverted Pendulum: A Comparative Analysis. PLoS ONE, 19(3), 1–19.

⁷ Board, A. N. Y. A. (2026). CHANGE CODE AND UPLOAD IT TO THE ARDUINO BOARD. THIS SOFTWARE WORKS WITH What is Arduino ? Arduino Main Features : Arduino Benefits : Options for Board Modules. 1–5.