

Analisis Performa DMP MPU6050 dan Kendali PID pada Robot Self-Balancing Berbasis ESP32

¹ Arya Naradipa Oktaviano, ² Muhammad Izzata Falliant, ³ Dicky Nabila Hardiansyah, ⁴ Khusnul Hidayat

^{1,2,3,4} Program Studi Teknik Elektro, Universitas Muhammadiyah Malang, Malang

¹ aryaradipa04@webmail.umm.ac.id, ² zettafalliant77@gmail.com, ³ dickynabila13@webmail.umm.ac.id, ⁴ khusnulhidayat@umm.ac.id

Abstract - A two-wheeled self-balancing robot requires accurate tilt-angle estimation and a low-latency real-time control system to maintain balance. However, software-based sensor fusion algorithms generally increase the computational burden of the microcontroller, reducing control responsiveness. This study evaluates the effectiveness of the Digital Motion Processor (DMP) embedded in the MPU6050 sensor combined with a discrete Proportional-Integral-Derivative (PID) controller on an ESP32-based self-balancing robot. The proposed approach was evaluated by comparing three angle estimation methods, namely Raw Data, Complementary Filter, and Hardware DMP, in terms of sampling time, transient response, and balance stability. Experimental results show that the DMP achieved a sampling time of 8 ms, outperforming both the Complementary Filter (11 ms) and Raw Data (21 ms) approaches. The proposed system maintained the robot's upright position with a settling time of 1.5 s and a low steady-state angular deviation. Furthermore, offloading the sensor fusion process to the DMP significantly reduced the computational burden on the ESP32, as inferred from the reduced sampling time of 8 ms, resulting in more responsive and stable control performance. These results demonstrate that integrating the MPU6050 DMP with a discrete PID controller provides an effective solution for improving the responsiveness and stability of embedded two-wheeled self-balancing robots.

Keywords — *Self-Balancing Robot, MPU6050, Digital Motion Processor (DMP), PID Control, ESP32.*

Abstrak - Robot self-balancing dua roda memerlukan estimasi sudut kemiringan yang akurat dan sistem kendali real-time dengan latensi rendah untuk mempertahankan keseimbangan. Namun, metode sensor fusion berbasis perangkat lunak umumnya meningkatkan beban komputasi mikrokontroler sehingga dapat menurunkan respons sistem kendali. Penelitian ini bertujuan mengevaluasi pemanfaatan Digital Motion Processor (DMP) pada sensor MPU6050 yang dipadukan dengan pengendali Proportional-Integral-Derivative (PID) diskret berbasis ESP32 untuk meningkatkan performa robot self-balancing. Evaluasi dilakukan dengan membandingkan tiga metode pengolahan sudut, yaitu Raw Data, Complementary Filter, dan DMP, berdasarkan waktu sampling, respons transien, dan kestabilan robot. Hasil pengujian menunjukkan bahwa DMP menghasilkan waktu sampling sebesar 8 ms, lebih cepat dibandingkan Complementary Filter (11 ms) dan Raw Data (21 ms), serta mampu mempertahankan posisi tegak robot dengan settling time 1,5 detik dan simpangan sudut kondisi tunak yang rendah. Selain itu, pemrosesan sensor fusion oleh DMP mengurangi beban komputasi utama pada ESP32 sebagaimana terinferensi dari percepatan waktu sampling hingga 8 ms sehingga respons pengendali menjadi lebih cepat dan stabil. Hasil penelitian menunjukkan bahwa integrasi DMP MPU6050 dan pengendali PID diskret merupakan solusi yang efektif untuk meningkatkan responsivitas dan kestabilan robot self-balancing berbasis mikrokontroler.

Kata Kunci — *Robot Self-Balancing, MPU6050, Digital Motion Processor (DMP), Kendali PID, ESP32.*

I. PENDAHULUAN

Perkembangan robot penyeimbang dua roda, yang beroperasi berdasarkan prinsip pendulum terbalik, sangat penting dalam konteks teknologi transportasi personal otonom dan robotika logistik. Robot ini secara alami tidak stabil dan memerlukan sistem kontrol yang efektif untuk menjaga keseimbangan, terutama dalam menghadapi gangguan [1]. Berbagai metode kontrol telah diusulkan, termasuk kontrol PID yang terbukti efektif dalam menjaga posisi vertikal robot dengan menggunakan sensor MPU GY-6050 dan mikrokontroler Arduino Uno, maupun mikrokontroler lainnya [2], [3]. Selain itu, pendekatan hibrida yang mengurangi usaha kontrol hingga 64% juga telah dikembangkan, meningkatkan kemampuan robot dalam menanggulangi gangguan pada sudut *pitch* dan *yaw* [4]. Penelitian lebih lanjut menunjukkan bahwa penggunaan pengontrol berbasis pengamat adaptif dapat meningkatkan stabilitas robot meskipun terdapat variasi massa yang tidak diketahui [5]. Dengan demikian, penelitian ini

menekankan pentingnya pengembangan sistem kontrol yang responsif dan efisien untuk menjaga stabilitas dinamis robot penyeimbang dua roda dalam waktu nyata.

Tantangan dalam pengembangan robot terkait akurasi pembacaan sensor sudut kemiringan dapat diatasi dengan penerapan metode penyaringan yang efektif. Penelitian menunjukkan bahwa penggunaan *complementary filter* dapat mengurangi derau dan akumulasi kesalahan pada data sensor *gyroscope* dan *accelerometer*, sehingga meningkatkan stabilitas sistem keseimbangan robot dua roda dengan kesalahan rata-rata $\pm 2^\circ$ dari sudut vertical [6]. Selain itu, filter Kalman terbukti optimal dalam memfilter data *accelerometer*, yang penting untuk pengukuran sudut *pitch* dan *roll*. Metode fusi sensor yang menggabungkan data dari *gyroscope* dan *accelerometer* juga dapat meningkatkan akurasi pengukuran sudut, terutama dalam kondisi dinamis. Dengan mengadopsi teknik penyaringan yang tepat, seperti *notch filter* dalam kerangka Kalman, dapat mengatasi gangguan frekuensi sempit yang dihasilkan oleh motor dan getaran mekanis [7]. Oleh karena itu, kombinasi teknik penyaringan ini sangat penting untuk memastikan sinyal kendali PID tetap stabil dan responsif.

Penelitian mengenai stabilisasi robot dua roda menunjukkan bahwa penggunaan Filter Kalman, meskipun memberikan estimasi sudut yang akurat, dapat menyebabkan beban komputasi yang tinggi pada mikrokontroler dengan sumber daya terbatas, yang berpotensi mengakibatkan fluktuasi waktu sampling dan penurunan performa kendali PID [8]. Alternatif seperti *Unscented Kalman Filter* (UKF) telah diusulkan untuk mengatasi masalah desain dan konvergensi yang dihadapi oleh *Extended Kalman Filter* (EKF), dengan hasil yang menunjukkan estimasi sikap yang lebih baik dan biaya komputasi yang sebanding. Selain itu, penggunaan algoritma genetika untuk estimasi parameter Filter Kalman dalam sistem robot dua roda juga menunjukkan efektivitas dalam mendapatkan *tuning* optimal untuk *noise covariance*, yang penting untuk menjaga stabilitas saat robot bergerak. Dengan demikian, pemilihan metode estimasi yang tepat sangat krusial untuk meningkatkan performa kendali robot dua roda [9].

Implementasi metode *sensor fusion* berbasis perangkat keras menggunakan Digital Motion Processor (DMP) pada sensor MPU6050 di mikrokontroler ESP32 menunjukkan keunggulan dalam mengatasi keterbatasan beban komputasi [10], [11]. DMP mampu melakukan ekstraksi sudut *pitch* secara mandiri, yang berbeda dari pendekatan konvensional yang mengandalkan CPU utama untuk pemrosesan filter, sehingga mengurangi beban komputasi minim pada CPU. Penelitian ini juga membuktikan bahwa integrasi DMP dengan kontrol PID dapat menjaga stabilitas dengan waktu sampling yang tepat pada 8 ms, serta mengatasi masalah getaran transien akibat tegangan baterai penuh melalui penskalaan daya motor sebesar 65% via *software*. Selain itu, teknik *sensor fusion* yang menggunakan filter Kalman dan metode estimasi sikap yang *robust* juga dapat meningkatkan akurasi pengukuran dan stabilitas sistem, yang relevan dalam aplikasi robotika dan sistem navigasi [12].

Kebaruan penelitian ini terletak pada penerapan pengujian komparatif dengan mempertahankan parameter kendali PID yang sama, yaitu ($K_p = 18,0$), ($K_i = 0,8$), dan ($K_d = 0,7$), pada tiga metode estimasi sudut, yaitu Raw Data, *Complementary Filter*, dan *Hardware DMP* MPU6050. Ketiga metode diuji pada platform robot fisik dan kondisi keseimbangan yang sama, sehingga perbedaan performa yang diamati dapat digunakan untuk mengevaluasi pengaruh metode estimasi sudut terhadap respons sistem kendali. Perbedaan nilai setpoint yang digunakan pada masing-masing metode disebabkan oleh perbedaan sistem referensi sudut yang dihasilkan oleh metode pengolahan sensor, bukan oleh perubahan posisi fisik robot. Selain itu, penelitian ini menerapkan penskalaan daya motor secara asimetris, yaitu 60% pada motor kiri dan 67% pada motor kanan, untuk mengompensasi perbedaan karakteristik mekanis kedua motor.

II. METODE PENELITIAN

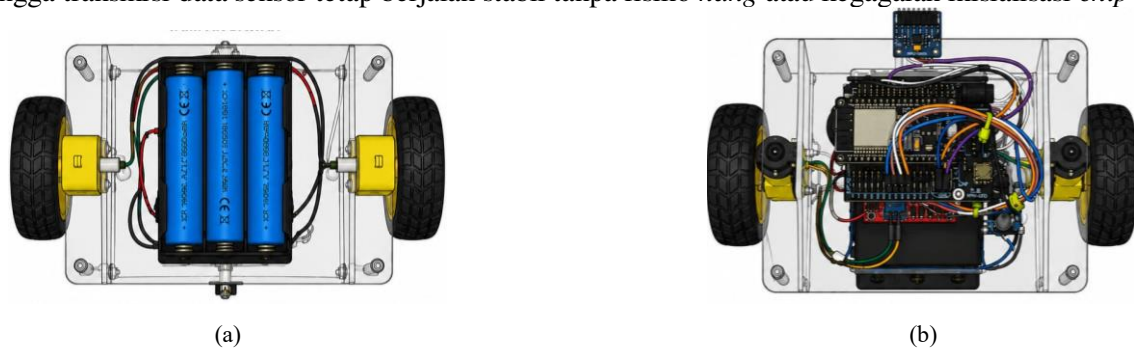
A. Desain Mekanis dan Tata Letak Komponen

Sasis robot dirancang bertingkat (*multi-layer chassis*) menggunakan bahan akrilik transparan untuk mendapatkan kekuatan struktural yang kokoh namun tetap memiliki bobot yang relatif ringan. Desain bertingkat ini dirangkai menggunakan tiang penyangga logam (*spacer* kuning) untuk memisahkan kompartemen antar lantai secara presisi seperti yang ditunjukkan pada Gambar 1. Tata letak komponen diatur secara strategis dengan mempertimbangkan distribusi massa dan minimalisasi gangguan sinyal:

1. Lantai Bawah (Base Deck): Sepasang motor DC *TT gearbox* diletakkan pada sumbu horizontal terbawah dan terhubung langsung ke roda karet. Dudukan motor dirancang secara presisi agar sumbu putaran kedua roda sejajar secara horizontal. Di lantai bawah ini juga diletakkan *holder* baterai berisi baterai Li-ion 3-cell dengan

total tegangan nominal 11,1 V (mencapai 12,6 V saat terisi penuh). Penempatan baterai pada lantai bawah bertujuan menempatkan massa terbesar lebih dekat dengan sumbu roda untuk menurunkan pusat massa (*center of mass*) dan menjaga distribusi massa robot tetap terkonsentrasi pada bagian bawah sasis. Konfigurasi ini membantu menghasilkan karakteristik mekanis yang sesuai untuk sistem pendulum terbalik, sehingga torsi motor yang relatif terbatas dari motor DC gearbox mampu memberikan gaya pemulihan (*restoring force*) yang memadai untuk menegakkan robot. Selain baterai, modul driver motor L298N dan sakelar daya juga ditempatkan pada lantai bawah ini untuk menyederhanakan perkabelan daya arus tinggi.

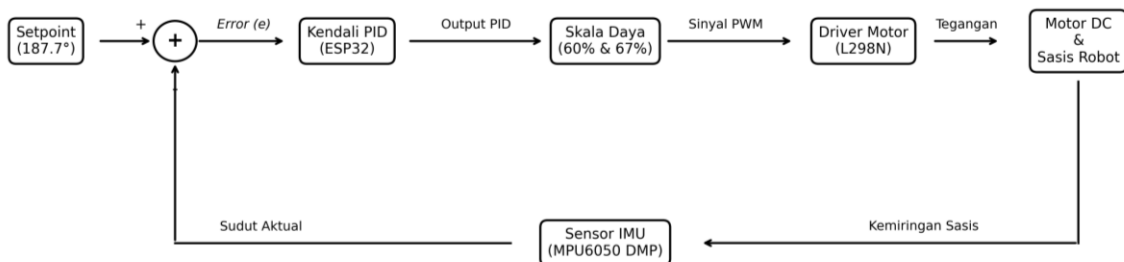
2. Lantai Atas (Upper Deck): Lantai atas menampung modul mikrokontroler ESP32 DevKit V1 yang terpasang di atas *expansion shield* serta modul sensor IMU MPU6050. Peletakan sensor MPU6050 dilakukan secara *rigid* di dekat sumbu vertikal pusat sasis dan sejajar dengan sumbu rotasi roda untuk meminimalisasi kesalahan pembacaan akibat akselerasi linear sekunder (*cross-axis coupling*) [13]. Pemisahan posisi sensor MPU6050 di lantai atas dari motor DC dan *driver* motor L298N dilakukan sebagai bagian dari pertimbangan desain tata letak sasis untuk mengurangi potensi gangguan mekanis dan interferensi elektromagnetik (EMI) pada jalur bus I2C, sehingga transmisi data sensor tetap berjalan stabil tanpa risiko *hang* atau kegagalan inisialisasi *chip* [14], [15].



Gambar 1. Self balancing robot: (a) tampak bawah dan (b) tampak atas

B. Diagram Blok Sistem Kendali

Metode kendali yang diterapkan adalah kontrol loop tertutup (*closed-loop control*) dengan diagram blok seperti yang ditunjukkan pada Gambar 2.



Gambar 2. Skema diagram blok sistem kendali robot self balancing

Penjelasan pada Gambar 2, yaitu :

1. Titik Komparator (*Summing Junction*): Berfungsi sebagai pembanding selisih antara nilai acuan target (Setpoint = $187,7^\circ$) dengan Sudut Aktual sasis saat itu. Output dari komparator ini adalah nilai selisih penyimpangan yang disebut Error ($e(t)$).
2. Kendali PID (ESP32): Komputasi algoritma PID pada ESP32 menerima masukan nilai Error dan menghasilkan Output PID ($u(t)$). Variabel *Output PID* ini berupa daya koreksi kontroler (nilai numerik digital mentah) dalam rentang limit $[-255, 255]$ Nilai mentah ini merupakan hasil penjumlahan matematis murni dari aksi kontrol Proportional, Integral, dan Derivative di memori CPU ESP32 (di mana nilai positif

merepresentasikan instruksi gerak maju, nilai negatif untuk gerak mundur, dan nilai mendekati nol menandakan sasis telah tegak seimbang).

3. Skala Daya (60% & 67%): Nilai numerik *Output PID* dikalikan dengan faktor skala asimetris secara *software* (skala $-0,60$ untuk motor kiri, $0,67$ untuk motor kanan). Konversi ini menghasilkan variabel Sinyal PWM akhir yang telah disesuaikan dengan karakteristik gesekan dinamo fisik.
4. Driver Motor (L298N): Berfungsi sebagai pengendali motor berbasis *dual H-bridge* yang menerima sinyal PWM dan sinyal arah dari ESP32. Driver meneruskan daya dari sumber baterai ke motor DC sesuai dengan sinyal kendali yang diberikan. Nilai PWM digunakan untuk mengatur besar daya rata-rata yang diterima motor dan dibatasi pada rentang -255 hingga 255 , sedangkan arah putaran motor ditentukan berdasarkan sinyal arah yang diberikan ESP32 melalui rangkaian *H-bridge*.
5. Motor DC & Sasis Robot (*Plant*): Tegangan DC memutar motor sehingga menghasilkan gaya mekanik translasi pada roda. Pergerakan roda ini secara dinamis mengubah Kemiringan Sasis robot secara fisik.
6. Sensor IMU MPU6050 DMP (*Feedback*): Mengukur perubahan sudut fisik sasis. Chip *Digital Motion Processor* (DMP) internal MPU6050 melakukan fusi sensor giroskop & akselerometer secara internal untuk menghasilkan variabel umpan balik Sudut Aktual yang bersih dari derau getaran, yang kemudian dikirim kembali ke titik komparator untuk siklus loop berikutnya.

C. Implementasi Algoritma Kontrol PID Diskrit dan Prosedur Kalibrasi

Metode kendali utama yang digunakan untuk menyeimbangkan robot adalah algoritma Proportional-Integral-Derivative (PID) tipe diskrit. Proses implementasi sistem kontrol ini dibagi menjadi dua tahapan perangkat lunak utama, yaitu tahap kalibrasi sensor dan tahap eksekusi kendali loop tertutup:

1. Prosedur Kalibrasi Sensor MPU6050 (Kode A): Sebelum algoritma kendali dijalankan, dilakukan kalibrasi sensor secara statis menggunakan program pencari offset otomatis (Calibration Sketch / Kode A). Robot diletakkan secara diam tegak lurus (disangga kokoh tanpa pegangan tangan) untuk mencegah derau getaran. Program Kode A akan membaca data mentah akselerometer dan giroskop secara terus-menerus, melakukan kompensasi bias gravitasi bumi, dan menghasilkan nilai offset register internal sensor. Nilai offset tersebut akan ditampilkan secara otomatis pada jendela Serial Monitor di Arduino IDE (misalnya perintah `mpu.setXGyroOffset(), dll`). Nilai ini kemudian disalin dan dimasukkan secara manual ke dalam fungsi `setup()` pada program kendali utama (Kode B) untuk mengunci pembacaan sudut referensi seimbang secara akurat.
2. Inisialisasi dan Setting Parameter PID (Kode B): Pada program utama (Kode B), pengontrol PID diinisialisasi menggunakan pustaka `PID_v1.h` dengan struktur deklarasi: `PID pid(&input, &output, &setpoint, Kp, Ki, Kd, DIRECT)`. Parameter masukan yang disetel meliputi:
 - Input: Sudut kemiringan aktual hasil filter DMP MPU6050 (dikonversi dari radian ke derajat).
 - Setpoint: Target sudut berdiri seimbang fisik robot (disetel pada $187,7^\circ$).
 - Kp (Proportional): Disetel pada $0,8$ untuk mengoreksi akumulasi error sehingga membantu mengurangi error keadaan tunak (steady-state error).
 - Ki (Integral): Disetel pada $0,8$ untuk mencegah penumpukan nilai error (*windup*) selama tahap penyetelan awal.
 - Kd (Derivative): Disetel pada $0,7$ untuk meredam laju kecepatan jatuh robot.
3. Matematika dan Eksekusi PID Diskrit: Sinyal kesalahan (*error*) dihitung berdasarkan selisih antara nilai *setpoint* dan sudut actual, yaitu $e(t) = \text{Setpoint} - \text{Input}$. Pada implementasi pengendali, waktu sampling PID ditetapkan sebesar ($\Delta t = 10$ ms) sehingga pengendali dikonfigurasi dengan periode sampling 100 Hz. Sinyal kontrol output PWM motor $u(t)$ kemudian dihitung secara periodik menggunakan persamaan PID diskrit berikut:

$$u(t) = K_p \cdot e(t) + K_i \cdot \sum_{i=0}^t e(i) \cdot \Delta t + K_d \cdot \frac{e(t) - e(t-1)}{\Delta t} \quad (1)$$

Aksi kendali $u(t)$ dibatasi oleh fungsi saturasi perangkat lunak pada rentang $[-255, 255]$ menggunakan fungsi `SetOutputLimits` $(-255, 255)$ untuk menyesuaikan dengan batas siklus kerja (*duty cycle*) PWM 8-bit dari ESP32 menuju motor L298N.

III. HASIL DAN PEMBAHASAN

Seluruh pengujian dilakukan menggunakan robot self-balancing pada permukaan lantai datar dengan konfigurasi parameter PID yang sama untuk setiap metode estimasi sudut. Tiga metode yang dibandingkan adalah Raw Data, *Complementary Filter*, dan Hardware DMP pada MPU6050. Parameter evaluasi meliputi waktu sampling, fluktuasi sudut, tingkat derau, settling time, serta kemampuan robot mempertahankan posisi tegak. Pengujian robot *self-balancing* dilakukan menggunakan konfigurasi parameter kendali optimal, yang dirangkum sesuai dengan Tabel 1.

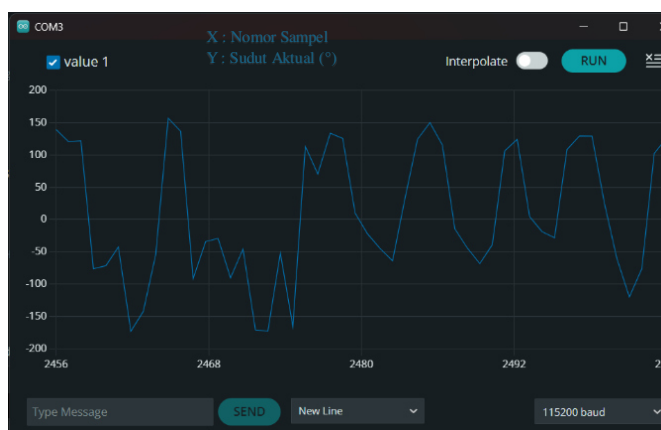
Tabel 1. Konfigurasi Parameter Sistem Kendali Robot

Parameter	Nilai
Kp	18
Ki	0,8
Kd	0,7
Setpoint	187,7°
Sample Time	10 ms
Output PWM	-255 sampai 255

A. Analisis Karakteristik Performa Tiga Metode Sensor

Pengujian performa dinamis sistem kendali dilakukan pada permukaan datar dengan menerapkan parameter PID konstan $K_p = 18$; $K_i = 0,8$; $K_d = 0,7$ dan target sudut setpoint $187,7^\circ$. Evaluasi membandingkan tiga metode pengolahan sudut: Raw Data (tanpa filter), Filter Komplementer, dan Hardware DMP MPU6050. Konfigurasi acuan sistem kendali dirangkum pada Tabel 1, sedangkan perbandingan hasil pengujian performa masing-masing metode disajikan pada Tabel 2.

Perbedaan nilai setpoint antara Raw Data (0°) dan metode lainnya ($187,7^\circ$) bukan disebabkan oleh perbedaan posisi fisik robot, melainkan perbedaan sistem referensi sudut. Metode Raw Data menggunakan kalkulasi trigonometri `atan2()` karena itulah satu-satunya cara mengkonversi data akselerometer mentah (berupa nilai ADC mentah dalam bilangan bulat integer) menjadi sudut kemiringan, yang secara matematis menghasilkan sudut relatif terhadap arah vertikal gravitasi. Sementara itu, DMP dan Filter Komplementer menggunakan sistem koordinat internal MPU6050 yang menghasilkan nilai referensi berbeda, sehingga posisi tegak robot terbaca pada sudut $187,7^\circ$. Meskipun nilai setpoint berbeda, posisi tegak robot secara fisik adalah sama pada ketiga metode, karena kendali PID bekerja berdasarkan nilai error (selisih setpoint terhadap sudut aktual), bukan nilai absolut sudut itu sendiri.



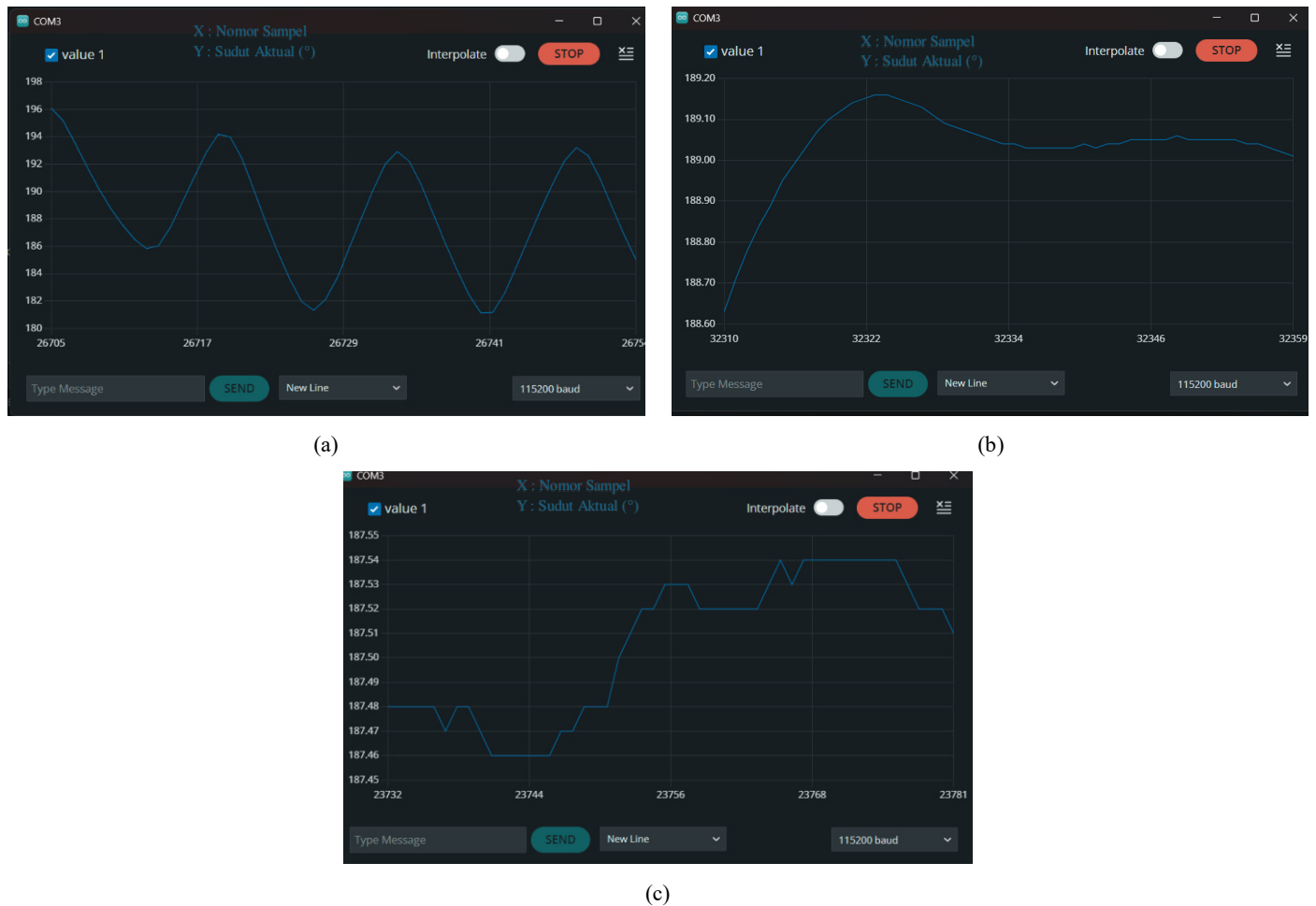
(a)



(b)

Gambar 3. Grafik respon sudut aktual menggunakan (a) raw data dan (b) filter komplementer

Gambar 3(a) menunjukkan bahwa penggunaan data sensor mentah (Raw Data) menghasilkan estimasi sudut dengan fluktuasi yang sangat tinggi pada rentang $-172,35^\circ$ hingga $150,3^\circ$. Getaran mekanis yang ditransmisikan langsung dari motor ke sensor menyebabkan munculnya *erratic noise*, sehingga sinyal umpan balik tidak merepresentasikan sudut kemiringan yang sebenarnya. Kondisi ini menyebabkan pengendali PID tidak mampu menghasilkan aksi koreksi yang tepat dan robot kehilangan keseimbangan dalam waktu kurang dari 1 detik. Sementara itu, Gambar 3(b) menunjukkan bahwa *Complementary Filter* mampu mereduksi derau sehingga kurva sudut tampak lebih halus. Namun, proses penyaringan menimbulkan phase lag yang menyebabkan deviasi sudut meningkat secara bertahap (drift). Akumulasi keterlambatan tersebut mengurangi respons sistem terhadap perubahan sudut sehingga robot akhirnya roboh setelah sekitar 2 detik.



Gambar 4. Grafik aktual sudut DMP (a) kondisi osilasi batas, (b) respon redaman terkompensasi, dan (c) stasioner stabil

Gambar 4(a) menunjukkan munculnya osilasi batas (limit cycle) yang disebabkan oleh akumulasi getaran mekanis dari aktuator. Fenomena ini mengindikasikan bahwa sistem masih menerima gangguan periodik, namun pengendali tetap mampu menjaga robot berada di sekitar titik keseimbangan. Pada Gambar 4(b), respons transien mengalami peredaman (damped response), yang menunjukkan efektivitas aksi derivatif dalam mengurangi laju perubahan error sehingga amplitudo osilasi berkurang secara bertahap hingga sistem kembali mencapai setpoint. Selanjutnya, Gambar 4(c) menunjukkan bahwa sistem telah memasuki kondisi tunak (steady-state) dengan simpangan sudut yang sangat kecil, yaitu sekitar $\pm 0,05^\circ$ terhadap setpoint. Kondisi ini mengindikasikan bahwa kombinasi estimasi sudut menggunakan DMP dan pengendali PID mampu menghasilkan kestabilan yang baik dalam mempertahankan posisi tegak robot.

Tabel 2. Perbandingan tiga metode

Parameter Pengujian	Raw Data (No Filter)	Filter Komplementer	DMP MPU6050
Fluktuasi Sudut Aktual Robot	-172,35° s.d 150,3°, rentang $\pm 173^\circ$ (Setpoint 0°)	184,2 s.d 202,3, rentang $\pm 14,5^\circ$ (Setpoint 187,7°)	181,9° s.d 196,3°, rentang $\pm 9^\circ$ (Setpoint 187,7°)
Sampling Time	21 ms (0,021 detik)	11 ms (0,011 detik)	8 ms (0,008 detik)
Rentang Nilai Noise	3800 s.d 10000 (sangat tinggi)	908 s.d 2936 (tinggi)	2400 s.d 17970 (sangat tinggi)

Berdasarkan hasil visualisasi respon sudut pada Gambar 3 dan Gambar 4, serta rangkuman data pada Tabel 2, dilakukan analisis komparatif sebagai berikut:

1. Kecepatan Waktu Eksekusi dan Beban Komputasi: Implementasi Hardware DMP menghasilkan waktu eksekusi aktual tercepat, yaitu 8 ms, dibandingkan *Complementary Filter* sebesar 11 ms dan Raw Data sebesar 21 ms. Waktu 8 ms tersebut menunjukkan bahwa proses pembacaan dan pengolahan data pada konfigurasi DMP dapat dilakukan lebih cepat dibandingkan dua metode lainnya. Secara teoritis, interval 8 ms setara dengan frekuensi sekitar 125 Hz. Namun, nilai tersebut merupakan hasil pengukuran waktu eksekusi aktual dan tidak mengubah konfigurasi waktu sampling PID yang ditetapkan sebesar 10 ms.
2. Dampak Derau Mekanis (*Mechanical Noise*) pada Raw Data: Pada metode Raw Data, derau getaran mekanis akibat putaran motor DC (berada pada rentang 3.800–10.000) masuk secara langsung ke dalam algoritma PID tanpa penapisan. Hal ini memicu fluktuasi sudut aktual yang sangat ekstrem dalam rentang -172,35° hingga 150,30° (Gambar 3(a)). Derau tinggi ini menyebabkan sinyal keluaran PWM beresilasi secara kasar dan mendadak (saturasi PWM), sehingga robot mengalami kegagalan kestabilan (divergensi) dalam waktu < 1 detik.
3. Efisiensi Fusi Sensor dan Kestabilan Hardware DMP: Meskipun data pembacaan akselerometer mentah pada metode DMP menerima derau getaran fisik yang tinggi (2.400–17.970), unit prosesor internal DMP secara efektif memisahkan derau getaran frekuensi tinggi tersebut dari sinyal sudut kemiringan hakiki. Hasilnya, estimasi sudut aktual yang dihasilkan DMP tetap berada dalam rentang fluktuasi terkendali yaitu 181,90° hingga 196,30° (deviasi $\pm 9^\circ$ dari setpoint). Performa ini jauh lebih unggul dibandingkan Filter Komplementer yang beresilasi lebih lebar pada rentang 184,20° hingga 202,30° ($\pm 14,5^\circ$) dan mengalami hanyutan (drift) akibat akumulasi phase lag (Gambar (3b)). Stable-nya respon sudut pada DMP menjaga keluaran sinyal PWM motor tetap berada pada area linier yang halus (rentang -13 hingga -15), sehingga robot mampu mempertahankan posisi tegak secara berkelanjutan.

B. Analisis Kestabilan dengan DMP

Data detail keberhasilan robot saat mempertahankan keseimbangannya menggunakan metode DMP MPU6050 dirangkum pada Tabel 3.

Tabel 3. Hasil Pengujian Kestabilan Robot Menggunakan DMP MPU6050

Parameter Pengujian	DMP MPU6050
Sampling Time	8 ms (0,008 detik)
Rentang Fluktuasi Sudut	$\pm 9^\circ$
Rentang Output PWM Motor	-13 s.d -15
Settling Time (Waktu Pulih)	1,5 detik
Durasi Berdiri Maksimal	8-10 detik per siklus pengujian (stabil)
Keterangan Fisik	Bisa berdiri dan bergetar sedikit

Saat robot diberikan gangguan fisik berupa dorongan ringan, robot memiliki settling time yang sangat cepat yaitu 1,5 detik untuk meredam ayunan dan kembali diam di tempat. Hal ini membuktikan kombinasi PID ($K_p = 18,0$; $K_i = 0,8$; $K_d = 0,7$) memberikan gaya redam (damping force) yang sangat ideal. Batas minimum speed 38 dan skala motor kanan 0.67 serta kiri -0.60 berhasil mengatasi hambatan roda gigi (backlash) secara serempak sehingga robot tidak berbelok saat berakselerasi.



Gambar 5. Dokumentasi Fisik Kestabilan Berdiri Tegak Robot *Self-Balancing* Menggunakan Metode DMP MPU6050.

IV. KESIMPULAN

Penelitian ini berhasil mendesain, merealisasikan, dan mengevaluasi performa robot *self-balancing* dua roda berbasis mikrokontroler ESP32 menggunakan tiga metode pengolahan sensor. Berdasarkan hasil pengujian fisik dan analisis data yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Penerapan Digital Motion Processor (DMP) pada MPU6050 menunjukkan performa terbaik dibandingkan Raw Data dan *Complementary Filter* pada konfigurasi pengujian yang dilakukan. Metode DMP menghasilkan waktu eksekusi aktual sebesar 8 ms dan mampu mempertahankan robot pada kondisi keseimbangan selama durasi pengujian 8–10 detik per siklus, dengan rentang fluktuasi sudut sebesar $\pm 9^\circ$ terhadap setpoint $187,7^\circ$.
2. Metode Raw Data tidak mampu mempertahankan keseimbangan robot pada konfigurasi pengujian karena menghasilkan fluktuasi sudut yang sangat besar dan menyebabkan robot kehilangan keseimbangan dalam waktu kurang dari 1 detik. *Complementary Filter* menghasilkan respons sudut yang lebih halus dibandingkan Raw Data, tetapi robot mengalami drift dan kehilangan keseimbangan setelah sekitar 5 detik pada pengujian yang dilakukan.
3. Kombinasi parameter kendali optimal $K_p = 18,0$; $K_i = 0,8$; $K_d = 0,7$ dengan waktu sampling 10 ms (waktu sampling riil DMP 8 ms), batas kecepatan minimum 38 (nilai PWM), serta kalibrasi asimetri motor menggunakan skala pengali daya (skala – 0,60 atau 60% motor kiri dan skala 0,67 atau 67% motor kanan) terbukti mampu mengeliminasi getaran transien akibat baterai penuh (12,6 V) dan menyamakan respon roda kanan-kiri secara sinkron sehingga robot dapat berdiri tenang di atas lantai keras.
4. Kondisi kapasitas dan tegangan baterai terbukti sangat berpengaruh terhadap kestabilan kontrol robot. Tegangan baterai penuh (12,6 V) meningkatkan torsi motor secara drastis sehingga rentan memicu osilasi liar jika tidak diimbangi dengan pembatasan skala daya motor (skala 60% dan 67%) serta redaman parameter derivatif yang tepat. Sebaliknya, saat baterai lemah (20%), respon motor menurun secara alami dan membuat gerakan robot lebih tenang namun kurang responsif terhadap gangguan besar.
5. Desain tata letak sasis bertingkat (multi-layer) dengan menaruh baterai di lantai dasar terbukti efektif dalam menurunkan pusat massa (*center of mass*) robot sehingga memperkecil momen inersia pendulum. Selain itu, pemisahan sensor MPU6050 dan ESP32 di lantai atas berhasil mengisolasi gangguan elektromagnetik (EMI) dari dinamo motor, menjaga kestabilan komunikasi data I2C.

V. DAFTAR PUSTAKA

- [1] T. Taufiq and Aswardi, "Self Balancing Robot Menggunakan Metode PID Berbasis Arduino," *JTEIN J. Tek. Elektro Indones.*, vol. 3, no. 1, pp. 15–24, 2022, doi: 10.24036/jtein.v3i1.192.
- [2] A. Najmurokhman, Kusnandar, I. Irfansyah, and A. Daelami, "Rancang Bangun Auto Balancing Robot Menggunakan Metode Kendali PID," *TELKA J. Telekomun. Elektron. Komputasi, dan Kontrol*, vol. 5, no. 1, pp. 15–23, 2019.
- [3] A. T. Ali, A. M. O. Mohamed, A. S. A. Salim, E. A. O. M. El-Amin, and O. M. K. Ahmed, "Design and Implementation of Two-Wheeled Self-Balancing Robot Using PID Controller," in *Proceedings of: 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering, ICCCEEE 2020*, 2021, pp. 9–13. doi: 10.1109/ICCCEEE49695.2021.9429579.
- [4] S. I. Khan, M. A. Choudhry, A. Ali, I. U. H. Shaikh, and F. Saleem, "A Hybrid Technique for Upward Stabilization and Control of Two Wheeled Self-Balancing Segway," *Mehran Univ. Res. J. Eng. Technol.*, vol. 41, no. 1, pp. 169–179, 2022, doi: 10.22581/muet1982.2201.17.
- [5] I. Jmel, H. Dimassi, S. H. Said, and F. M. Sahli, "Adaptive Observer-Based Output Feedback Control for Two-Wheeled Self-Balancing Robot," *Math. Probl. Eng.*, vol. 2020, 2020, doi: 10.1155/2020/5162172.
- [6] B. N. Cahyono, Sumardi, and B. Setiyono, "Self-Balancing Scooter Menggunakan Metode Kendali Proporsional Integral Derivatif," *Transmisi*, vol. 15, no. 4, pp. 164–169, 2013.
- [7] G. Lu and F. Zhang, "IMU Based Attitude Estimation in the Presence of Narrow-band Noise with Application to a Three-Axis Gimbal System," *IEEE/ASME Trans. Mechatronics*, pp. 841–852, 2019.
- [8] H. Zhang and N. Mohamad Nor, "Control Strategies for Two-Wheeled Self-Balancing Robotic Systems: A Comprehensive Review," *Robotics*, vol. 14, no. 8, 2025, doi: 10.3390/robotics14080101.
- [9] W. Fadlun and A. Sugiharto, "Pemodelan Dinamika Dan Perancangan Kendali Logika Fuzzy Untuk Robot Self-Balancing Roda Dua," *JITET J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 2, 2025, doi: 10.23960/jitet.v13i2.6402.
- [10] S. Gupta, K. Ray, and S. Kaiser, "Self-Balancing Mobile Robot with Bluetooth Control: Design, Implementation, and Performance Analysis," *Automation*, vol. 6, no. 3, pp. 1–26, 2025, doi: 10.3390/automation6030042.
- [11] C.-T. Hsieh, "Development of Proportional-Integral-Derivative Based Self-Balancing Robot Using ESP32 for STEM Education," in *Engineering Proceedings*, 2025. doi: 10.3390/engproc2025092024.
- [12] W. D. S. Oliveira and E. N. Gonçalves, "Implementação em c: filtro de kalman, fusão de sensores para determinação de ângulos," *ForScience*, vol. 5, no. 3, pp. 1–18, 2017, doi: 10.29069/forscience.2017v5n3.e287.
- [13] A. F. Albaghdadi and A. A. Ali, "An Optimized Complementary Filter For An Inertial Measurement Unit Contain MPU6050 Sensor," *Iraqi J. Electr. Electron. Eng.*, vol. 15, no. 2, pp. 71–77, 2019, doi: 10.37917/ijeee.15.2.8.
- [14] M. A. Rajitha, "Posture Correction Device Using ESP32 and MPU6050," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 13, no. 10, pp. 1615–1617, 2025, doi: 10.22214/ijraset.2025.74767.
- [15] H. Jian, "Design of Angle Detection System Based on MPU6050," in *7th International Conference on Education, Management, Information and Computer Science (ICEMC 2017)*, 2017, pp. 7–9. doi: 10.2991/icemc-17.2017.2.