

Rancang Bangun *Auto Balancing Robot* Menggunakan Metode Kendali PID

Design and Implementation of A Self- Balancing Robot Based on PID Control Method

Asep Najmurokhman^{1*}, Kusnandar², Irfan Irfansyah³, Ahmad Daelami⁴

Jurusan Teknik Elektro, Universitas Jenderal Achmad Yani

Jl. Terusan Jend. Sudirman PO Box 148 Cimahi 40533

asep.najmurokhman@lecture.unjani.ac.id^{1*}, koes_kusnandar@yahoo.com², ivanirfan56@yahoo.com³,
ahmad.daelami@lecture.unjani.ac.id⁴

Abstrak – *Auto balancing pada robot adalah mekanisme penyeimbangan posisi robot secara tegak lurus terhadap permukaan bumi tanpa memerlukan gaya dari luar. Proses penyeimbangan dilakukan melalui pemberian sinyal kendali kepada dua buah roda. Makalah ini menguraikan tentang prototipe sebuah auto balancing robot dengan metode kendali PID (proportional integral derivative) untuk menjaga badan robot tetap seimbang selama geraknya. Komponen utama dalam sistem adalah sensor MPU GY-6050 dan Mikrokontroler Arduino Uno. Sensor MPU GY-6050 digunakan untuk mendeteksi kemiringan dan kecepatan sudut dari badan robot selama bergerak. Mikrokontroler Arduino Uno berfungsi sebagai pengendali proses penyeimbangan robot melalui teknik kendali PID. Proses penyeimbangan dilakukan oleh motor DC. Hasil pengujian prototipe menunjukkan proses penyeimbangan dapat dilakukan dengan baik. Torsi motor menghasilkan pergerakan robot sedemikian sehingga robot berada dalam posisi tegaknya. Hasil pengujian saat robot digerakkan miring ke kanan dari posisi tegaknya diperoleh persentase error sebesar 0,62 %, sedangkan saat digerakkan miring ke kiri persentase errornya sebesar 0,74 %.*

Kata Kunci: *balancing robot, kendali PID, mikrokontroler Arduino Uno, sensor MPU GY-6050.*

Abstract – *Autobalancing on robots is a balancing mechanism of robot position perpendicular to the surface of the Earth without an external force. The balancing process is rendered via actuating two wheels by a proper control signal. This paper describes the prototype of an autobalancing robot with the PID control method to keep the robot move steadily. The prototype comprises MPU GY-6050 sensor, Arduino Uno Microcontroller, and DC motor. MPU GY-6050 sensor is used to detect the slope and angular velocity of the robot body during movement. Arduino Uno microcontroller functions as a controller of balancing process through the PID control method. The balancing process is carried out by a pair of DC motor. The experimental results show that the balancing process can be done well. The motor torque produces robot movements such that the robot is in an upright position. The error percentage of the upright position deviation is around 0.62% when the robot's initial condition is tilted to the right and 0.74% when tilted to the left.*

Keywords: *balancing robot, PID control, Arduino Uno microcontroller, MPU GY-6050 sensor.*

1. Pendahuluan

Riset tentang robot penyeimbang (*balancing robot*) telah menarik perhatian praktisi bidang robotika lebih dari dua dekade [1]. Hal ini disebabkan dinamika ketidakstabilan yang muncul dalam sistem tersebut. Robot dikarakterisasi dengan kemampuan penyeimbangan oleh roda dan berayun dalam titik sekitar posisi keseimbangannya. Dewasa ini, produk terkait dengan robot penyeimbang sudah dijual di pasaran dengan *brand* tertentu misalnya Segway, Ninebot, Anybots QB, dan lain-lain [2]. Meskipun demikian, riset seputar robot penyeimbang terus dikembangkan oleh institusi perguruan tinggi untuk kepentingan pembelajaran dan pengujian teknik pengendalian dengan metode yang lebih canggih. Perkembangan teknologi elektronika telah menghasilkan komponen pemroses sinyal dengan dimensi semakin kecil dan kapabilitas semakin tinggi. Dengan demikian, aplikasi robotika memperoleh keuntungan dengan perkembangan tersebut sehingga ukuran robot dapat dibuat lebih kecil, kompak, dan berkinerja tinggi. Teknik pengendalian canggih seperti kendali cerdas dewasa ini sudah mulai diterapkan dalam menghasilkan kinerja robot yang lebih unggul [3].

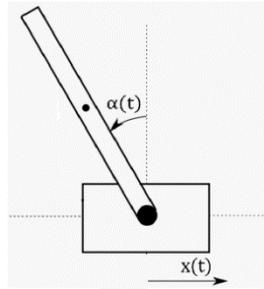
Tipe pengendali yang digunakan dalam sistem *auto balancing robot* ini adalah PID (singkatan dari *Proportional-Integral-Derivative*). Algoritma PID berbentuk sederhana sehingga mudah dipahami secara konseptual dan diterapkan secara praktis. Keuntungan ini membuat pengendali PID paling banyak digunakan di dunia industri sejak era 80-an sampai sekarang. Penggunaan yang cukup ekstensif dalam praktik pengendalian menjadikan pengendali PID dipandang sebagai “instrumen” dalam bidang sistem kendali dan komunikasi yang sangat penting di abad ke-20 setelah mikroprosesor [4]. Beberapa peneliti telah menerapkan teknik kendali PID ini dalam menghasilkan kinerja *auto balancing robot* yang diinginkan [5]-[7].

Makalah ini menguraikan tentang prototipe sebuah *auto balancing robot* beroda dua dengan metode kendali PID (*proportional integral derivative*) untuk menjaga badan robot tetap seimbang dalam posisi tegak lurus terhadap permukaan bumi selama gerakannya. Komponen utama dalam implementasi prototipe ini adalah sensor MPU GY-6050 dan mikrokontroler Arduino Uno. Sensor MPU GY-6050 berfungsi untuk mendeteksi kondisi kemiringan dan kecepatan sudut dari motor DC. Torsi yang dihasilkan motor DC digunakan untuk mempertahankan keseimbangan badan robot tersebut. Sementara itu, mikrokontroler Arduino Uno difungsikan sebagai pemroses utama sinyal-sinyal dalam sistem terutama untuk merealisasikan pengendali PID dalam prototipe tersebut. Dewasa ini, mikrokontroler Arduino menjadi komponen standar dalam realisasi sistem kendali dalam skala prototipe ataupun *pilot project* [7]-[10]. Dengan demikian, penelitian yang dilaporkan dalam makalah ini menerapkan pengendali PID yang direalisasikan dengan mikrokontroler Arduino untuk menyeimbangkan posisi badan robot lebih sempurna yang belum diperoleh dari penelitian sebelumnya.

Bagian makalah lain disusun dengan urutan sebagai berikut. Bagian ke-2 berisi tentang metode penelitian yang memuat deskripsi dari prinsip *auto balancing*, diagram blok sistem, dan komponen-komponen yang digunakan untuk merealisasikan prototipe. Realisasi sistem dan diskusi tentang hasil eksperimen yang diperoleh diuraikan dalam bagian ke-3. Selanjutnya, makalah diakhiri dengan bagian kesimpulan yang mengandung catatan penutup tentang realisasi prototipe dan saran pengembangan lebih lanjut untuk menyempurnakan hasil yang diperoleh.

2. Metode Penelitian

Prinsip dasar *auto balancing* dalam robot adalah pengendalian bandul terbalik. Ilustrasi bandul terbalik diberikan pada Gambar 1. Faktor utama penyeimbangan dalam robot adalah kestabilan. Karakteristik dasar dari bandul terbalik bersifat tidak stabil. .

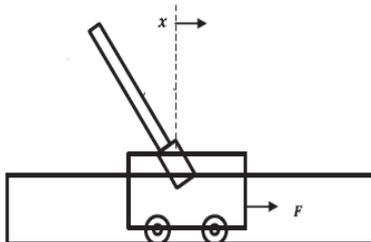


Gambar 1. Sketsa bandul terbalik [11].

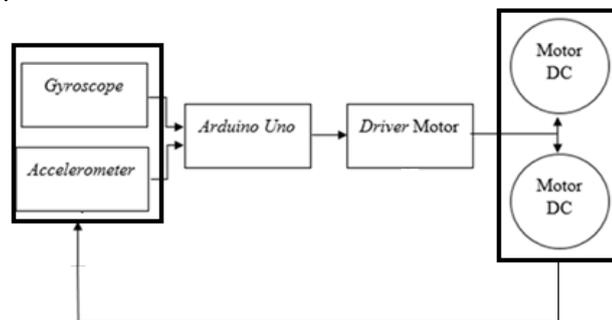
Salahsatu metode pengendalian yang dilakukan berupa penggunaan dua roda yang dikendalikan oleh sebuah kontroler sehingga robot dapat mempertahankan posisi tegaknya. Untuk menghasilkan kondisi seperti itu, sebuah pengendali dirancang dan diterapkan pada robot sedemikian sehingga robot selalu dapat menyeimbangkan bandul terbalik tersebut. Dengan demikian, tujuan pengendaliannya berupa mengatur posisi roda sehingga sudut kemiringan bandul tersebut berada dalam nilai yang ditentukan. Secara ilustratif, sebuah sistem *auto balancing* robot diperlihatkan pada Gambar 2. Pada saat *balancing robot* beroda dua condong ke belakang atau bandul miring ke kiri, sebuah mekanisme pergerakan motor memutar roda berlawanan arah dengan jarum jam diperlukan untuk mempertahankan bandul dalam posisi tegaknya sehingga *balancing robot* beroda dua berjalan mundur. Gaya yang digunakan untuk menyeimbangkan dihasilkan dari putaran roda. Putaran roda ini berasal dari torsi yang dihasilkan oleh motor listrik yang berfungsi sebagai aktuator dalam sistem yang dibangun.

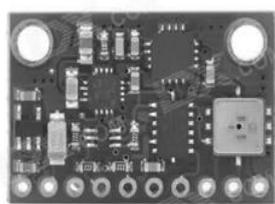
Diagram skematik dari sistem yang dirancang diberikan pada Gambar 3. Besaran yang dideteksi oleh sensor berupa sudut kemiringan dan kecepatan sudut. Dua besaran tersebut diperoleh menggunakan sensor MPU GY-6050 yang didalamnya mengandung *gyroscope* dan *accelerometer*. Data kemiringan diperoleh melalui bagian *accelerometer*, sedangkan bagian *gyroscopenya* mencatat data kecepatan sudut. Data tersebut kemudian diolah oleh mikrokontroler Arduino Uno yang sudah diinstalasi algoritma kendali PID untuk menghasilkan sinyal aktuasi tertentu. Sinyal aktuasi tersebut digunakan untuk memberikan torsi kepada motor DC agar bergerak sesuai dengan tujuan pengendaliannya, yaitu robot dapat mempertahankan posisi tegaknya.

Dalam sistem yang dirancang, sensor yang digunakan untuk mendeteksi besaran fisis berupa kemiringan dan kecepatan sudut adalah sensor MPU GY-6050. Sensor tersebut merupakan peralatan elektronika digital yang menggabungkan *gyroscope* tiga-sumbu, *accelerometer* tiga-sumbu, dan prosesor gerak dalam dimensi fisik 4x4x0,9 mm [13]. Sensor memiliki akurasi tinggi karena mengandung perangkat konverter analog ke digital dengan ukuran 16 bit. Bentuk sensor MPU GY-6050 diberikan pada Gambar 4.

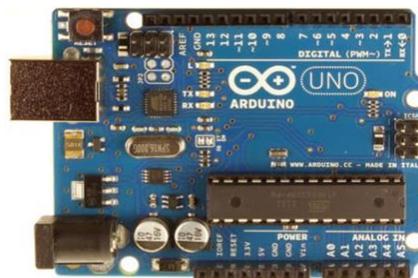


Gambar 2. Bandul terbalik pada robot beroda [12].

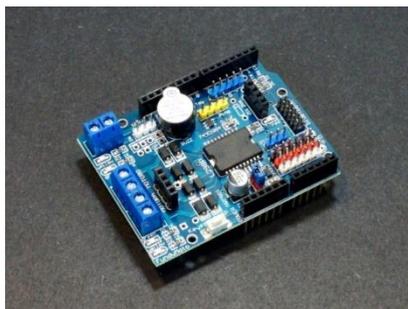
Gambar 3. Diagram skematik sistem *auto balancing robot*.



Gambar 4. Bentuk fisik sensor MPU GY-6050.



Gambar 5. Mikrokontroler Arduino Uno.



Gambar 6. IC L298P.

Selanjutnya, data sensor akan diproses oleh mikrokontroler Arduino Uno. Arduino Uno adalah mikrokontroler berbasis ATmega328. Mikrokontroler tersebut memiliki 14 pin input / output digital dengan enam pin di antaranya dapat digunakan sebagai output PWM, enam pin input analog, osilator 16 MHz, koneksi USB, *header* ICSP, dan tombol *reset*. Bentuk fisik mikrokontroler Arduino Uno diberikan pada Gambar 5.

Komponen yang digunakan untuk menggerakkan motor DC dalam pengendalian keseimbangan adalah rangkaian driver motor L298P. Modul elektronika ini merupakan sebuah *chip* yang didalamnya memiliki dua buah *full-bridge driver* tegangan dan arus tinggi yang dapat diatur dengan menggunakan *level logic* TTL. Komponen tersebut dapat *drive* beban induktif seperti solenoid, motor DC, dan motor *stepper*. Selain itu, L298P mampu mengeluarkan output tegangan untuk motor DC dan motor *stepper* sebesar 50 volt. Keping ini terdiri dari transistor-transistor logic (TTL) dengan gerbang NAND yang memudahkan dalam menentukan arah putaran suatu motor DC maupun motor *stepper*. Rangkaian *driver* ini dapat mengendalikan sekaligus dua motor DC. Bentuk fisik komponen driver motor L298P diberikan pada Gambar 6.

Sementara itu, motor DC berfungsi sebagai penggerak roda yang memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada motor DC disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam pada medan magnet, maka akan timbul tegangan yang berubah-ubah arah pada setiap setengah putaran. Bentuk motor paling sederhana memiliki kumparan satu lilitan yang dapat berputar bebas di antara kutub kutub magnet permanen. Motor DC yang digunakan dalam sistem *auto balancing* ini berupa *mini metal gear* yang disusun oleh dua buah roda, dua buah *gear* motor DC, dan satu buah roda bebas. Spesifikasi teknis motor DC yang digunakan mencakup kecepatan putar sebesar 246 rpm, tegangan catu daya 5 Vdc, kebutuhan arus 1 A, torsi maksimum yang dapat dihasilkan sebesar 3 kg cm, dan bobotnya sekitar 90 gram.

Algoritma kendali yang digunakan untuk menjaga keseimbangan dalam sistem *auto balancing* dengan dua roda yang diuraikan dalam makalah ini adalah metode pengendali PID. Pengendali PID dikarakterisasi oleh tiga parameter yaitu K_p , K_i , dan K_d , yang masing-masing disebut sebagai konstanta proporsional, konstanta integral, dan konstanta *derivative*. Masukan ke dalam pengendali berupa sinyal *error*, yaitu selisih antara nilai referensi dengan nilai keluaran

aktual yang dihasilkan oleh sistem. Informasi *error* ini digunakan oleh pengendali PID melalui ketiga parameternya untuk menghasilkan sinyal kendali yang akan mereduksi *steady state error* dan menangani *overshoot* dalam sistem. Secara matematis, jika $e(t)$ menandai sinyal *error* dan $u(t)$ adalah sinyal kendali, maka keluaran pengendali PID dapat dituliskan menjadi

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt}. \quad (1)$$

Dalam bentuk diskrit, persamaan (1) dapat dituliskan menjadi

$$u(k) = K_p e(k) + K_i T_i \sum_{i=0}^k e_i + K_d \frac{e(k) - e(k-1)}{T_d}, \quad (2)$$

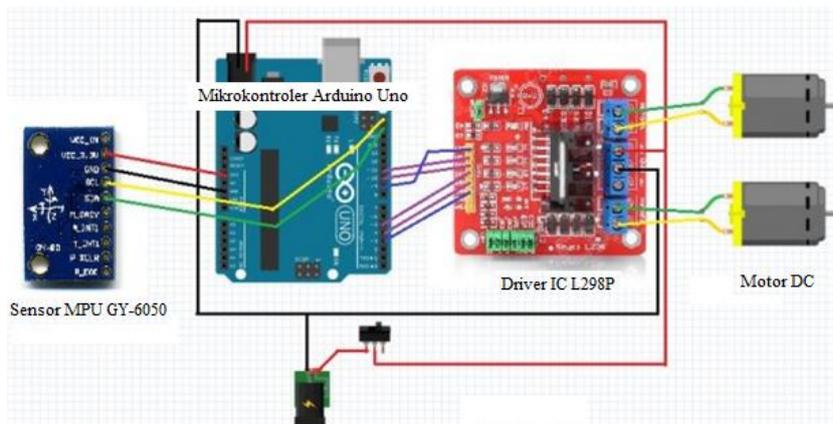
dengan T_i dan T_d masing-masing menyatakan waktu integrasi dan diferensial. Selanjutnya, persamaan (2) diterjemahkan dalam bentuk kode program sebagai berikut

$$U = K_p \times error + K_i \times T_s \times (error + last_error) + \frac{K_d}{T_s} \times (error - last_error) \quad (3)$$

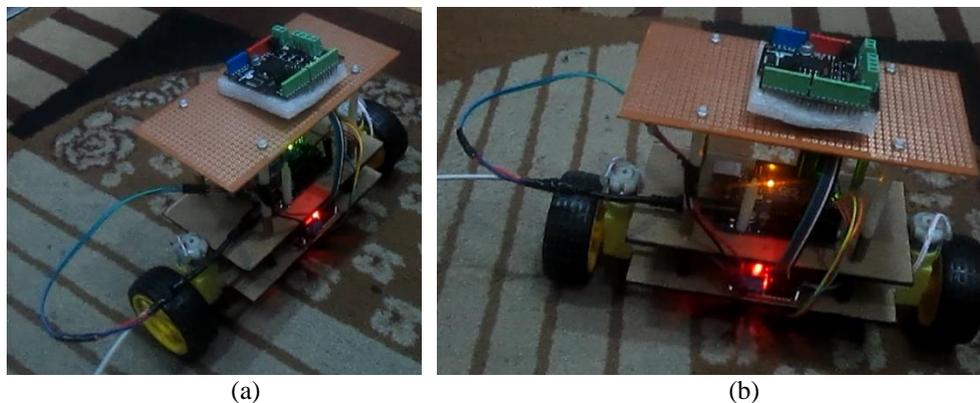
dengan U menyatakan sinyal kendali yang dihasilkan setiap saat, *error* adalah nilai selisih referensi dengan nilai output setiap waktu diskritnya, dan *last_error* menandai nilai *error* satu periode pencacahan sebelumnya. Sementara itu, T_s menyatakan periode pencacahan dari perangkat pengolah sinyal dalam mikrokontroler.

3. Hasil dan Pembahasan

Sistem interkoneksi antara komponen dalam prototipe diberikan dalam Gambar 7. Sementara itu, realisasi sistem *auto balancing* robot beroda dua diperlihatkan pada Gambar 8.



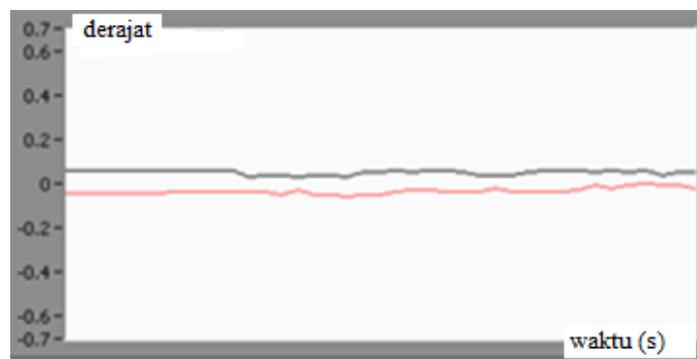
Gambar 7. Interkoneksi antar komponen dalam system.



Gambar 8. Realisasi *auto balancing* robot beroda dua tampak atas (a) dan depan (b).

Eksperimen dilakukan untuk memperoleh akurasi sensor dan pengujian sistem secara keseluruhan. Pengujian terhadap sensor dilakukan dengan cara mencatat hasil pengukuran bagian *accelerometer* dan *gyroscope* dari sensor MPU GY-6050. Karena ada dua motor DC yang digunakan, bagian *accelerometer* dan *gyroscope* dari sensor terhubung dengan dua motor DC tersebut. Saat posisi diam, nilai *accelerometer* pertama berkisar antara 0,07 sampai dengan 0,08 derajat, sedangkan *accelerometer* kedua bernilai antara -0,03 sampai dengan 0,1 derajat. Grafik hasil pengukurannya diperlihatkan pada Gambar 9.

Selanjutnya, pengujian sensor dilakukan saat robot digerakkan miring dari posisi tegaknya seperti diperlihatkan pada Gambar 10. Hasil pengukuran ditampilkan dalam Tabel 1. Ketika miring ke kanan, selisih pengukuran yang diperoleh sebesar 0,5 satuan atau persentase *error*nya sebesar 0,62 %, sedangkan saat miring ke kiri didapat selisih pengukuran sekitar 0,6 satuan atau persentase *error*nya sekitar 0,74 %. Ketika *accelerometer* pada posisi tegak lurus, nilai yang didapat adalah 0 (baik nilai sensor ataupun aktual). Selisih pengukuran yang diperlihatkan dalam Tabel 1 disebabkan proses komputasi saat pembulatan pecahan dalam pemrograman dan derau yang muncul dalam peralatan.



Gambar 9. Grafik nilai sudut yang diperoleh dari *accelerometer* saat posisi tegak.



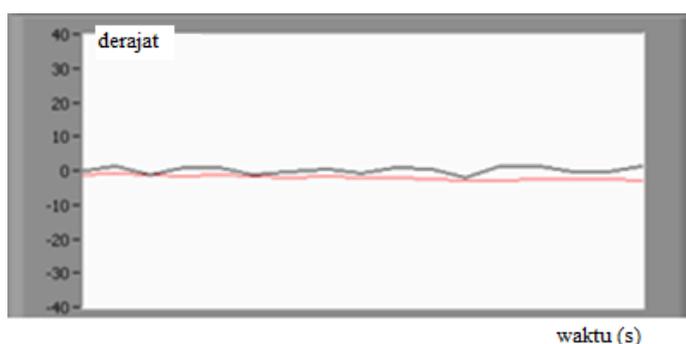
Gambar 10. Posisi miring robot.

Tabel 1. Data pengamatan pengukuran *accelerometer*.

Sudut <i>accelerometer</i> (derajat)	Sudut aktual (derajat)	Persentase <i>error</i> (%)	Arah robot
80	80,5	0,62	Miring Kanan
85	85	0	Miring Kanan
0	0	0	TegakLurus
-85	-85	0	Miring Kiri
-80	-80,6	0,74	Miring Kiri

Pengujian berikutnya dilakukan untuk mengamati akurasi *gyroscope* dalam membaca perubahan kecepatan sudut yang dinamis dari robot. Sensor *gyroscope* akan mempunyai nilai

keluaran dengan nilai tegangan yang mengecil jika sedang berotasi searah jarum jam terhadap sumbu Y, sebaliknya tegangan keluarannya akan membesar jika berotasi berlawanan arah jarum jam. Jika sensor *gyroscope* tidak berotasi (keadaan diam) maka keluaran tegangan *gyroscope* akan bernilai sama dengan nilai *offset*-nya. Gambar 11 menunjukkan grafik perubahan nilai *gyroscope* saat kondisi seimbang. Ketika dalam posisi seimbang, nilai *gyroscope* pertama berkisar antara -5,0 sampai dengan 5,0 derajat, sedangkan *gyroscope* kedua bernilai antara -0,3 sampai dengan 0,3 derajat. Idealnya, jika sensor *gyroscope* mendeteksi keadaan diam atau seimbang, maka keluarannya akan sama dengan nilai *offset*-nya, yaitu nol. Namun demikian, fluktuasi nilai di sekitar nol terjadi karena sensor *gyroscope* memiliki nilai bias. Meskipun terjadi fluktuasi nilai keluaran, namun variasinya sangat kecil sehingga sensor *gyroscope* yang digunakan memiliki akurasi yang relatif baik.



Gambar 11. Grafik nilai sudut yang diperoleh dari *gyroscope* saat posisi tegak.

Pengamatan lainnya dilakukan terhadap hubungan antara sinyal aktuasi berupa sinyal kendali yang diatur *duty cycle*-nya terhadap gerak motor DC yang dihasilkan. Tabel 2 memperlihatkan korelasi besaran-besaran tersebut untuk dua buah motor DC yang digunakan dalam prototipe ini. Dari tabel pengujian tersebut, semakin besar *duty cycle* yang digunakan maka semakin besar kecepatan putaran yang dihasilkan oleh motor DC. Ketika dalam posisi diam, nilai PWM (*pulse width modulation*) dan RPM (*rotation per minute*) adalah 0, sedangkan jika robot bergerak ke kanan atau bergerak maju yang berarti ada perubahan sudut positif, maka nilai PWM akan terus membesar sesuai dengan besarnya kemiringan sudut aktual. Dengan demikian, nilai RPM motor juga ikut membesar. Jika sudut mengecil atau mendekati titik tegak lurus (posisi seimbang), maka nilai PWM juga mengecil. Hal tersebut juga berlaku saat motor berputar mundur atau robot miring ke kiri.

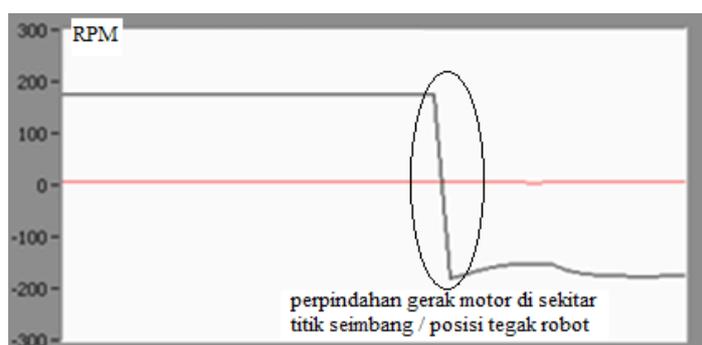
Tabel 2. Data pengukuran motor DC.

<i>Duty cycle</i>	PWM	RPM motor 1	RPM motor 2	Arah putaran motor
0	0	0	0	Diam
10	50	120	180	Kanan
15	101	150	200	Kanan
0	0	0	0	Diam
10	50	97	200	Kiri
15	101	100	210	Kiri

Selanjutnya, pengujian sistem keseluruhan dilakukan dengan cara mengamati respon robot terhadap adanya gaya luar yang mengganggu posisi tegak atau posisi seimbangnya. Pengujian tersebut dilakukan sebagai evaluasi terhadap kinerja pengendali PID yang telah diinstalasi ke dalam mikrokontroler.



Gambar 12. Grafik perubahan nilai RPM saat robot bergerak maju dan menuju keseimbangannya.



Gambar 13. Grafik perubahan nilai RPM saat robot bergerak mundur dan menuju keseimbangannya.

Hasil pengujian diperlihatkan dalam bentuk grafik nilai RPM terhadap waktu dalam Gambar 12 dan Gambar 13. Hasil observasi terhadap hasil pengujian tersebut memperlihatkan saat posisi robot cenderung ke kiri maka motor DC memberikan torsi berlawanan arah dengan jarum jam yang ditandai dengan nilai RPM negatif untuk mengembalikan ke posisi tegaknya. Sebaliknya, saat robot cenderung ke kanan dari posisi tegaknya maka motor DC memberikan torsi searah dengan jarum jam untuk mempertahankan posisi keseimbangannya. Mekanisme tersebut terus-menerus berlangsung di sekitar titik keseimbangannya sehingga teramati robot bergetar di sekitar posisi tegaknya. Observasi lain dari hasil pengujian menunjukkan proses penyeimbangan berlangsung dalam durasi yang relatif lama selama robot bergerak. Karena sifat non linier yang dimiliki oleh bentuk bandul terbalik, pendekatan kendali non linier dengan teknik kendali optimal seperti yang dilaporkan dalam [11] dan [14] atau kombinasi PID dengan teknik kendali cerdas seperti yang diuraikan dalam [15] menjadi bahan penelitian lanjutan untuk memperoleh kinerja proses *auto balancing* yang lebih baik.

4. Kesimpulan

Perancangan dan realisasi sebuah prototipe *auto balancing robot* dengan pengendali PID telah diuraikan. Prototipe tersebut dibangun dengan menggunakan sensor MPU GY-6050 untuk mendeteksi sudut kemiringan dan kecepatan sudut dari motor DC yang digunakan untuk menyeimbangkan badan robot. Selain itu, mikrokontroler Arduino Uno digunakan sebagai pemroses informasi dalam sistem serta komponen yang berfungsi sebagai pengendali PID. Hasil pengujian menunjukkan pengendali PID yang diinstalasi ke dalam mikrokontroler dapat menghasilkan proses penyeimbangan robot dengan baik. Persentase *error* saat robot digerakkan miring ke kanan dari posisi tegaknya sebesar 0,62 %, sedangkan saat digerakkan miring ke kiri diperoleh persentase *error* sebesar 0,74 %. Namun demikian, penelitian lanjutan diperlukan agar proses penyeimbangan dapat berlangsung dalam durasi yang lebih singkat saat robot bergerak. Karena sifat non linier yang dimiliki oleh bentuk bandul terbalik, pendekatan kendali non linier

dengan teknik kendali optimal atau kombinasi PID dengan teknik kendali cerdas menjadi bahan penelitian lanjutan untuk memperoleh kinerja proses *auto balancing* yang lebih baik.

Ucapan Terima Kasih

Ucapan terima kasih disampaikan kepada Lembaga Penelitian dan Pengabdian kepada Masyarakat Universitas Jenderal Achmad Yani (LPPM Unjani) yang telah mendukung sebagian dana penelitian dalam skema Penelitian Kompetitif Unjani Tahun 2018.

Referensi

- [1] Ooi RC, "Balancing a Two-Wheeled Autonomous Robot," Master Thesis. Crawley, WA. The University of Western Australia, 2003.
- [2] Park JH, Cho BK, "Development of a self-balancing robot with a control moment gyroscope," *Int. J. Adv. Robot. Syst.*, 15(2), 1–11, 2018.
- [3] Najmurokhman A, Kusnandar, Sofyan GI, Djamal EC, Munir A, Wibowo B, "Design and Implementation of Fuzzy Logic Controller for A Class of Hexapod Mobile Robot," *Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS)*, Batu, Indonesia: 2018.
- [4] Bista D, "Understanding and Design of an Arduino-based PID Controller," Master Thesis, Richmond VA. Virginia Commonwealth University, 2016.
- [5] Kim HW, An JW, Yoo HD, Lee JM, "Balancing Control of Bicycle Robot Using PID Control," *13th International Conference on Control, Automation and Systems (ICCAS)*. Gwangju, Korea: 145–147, 2013.
- [6] Sondhia S, Pillai R, Hegde SS, Chakole S, Vora V, "Development of Self Balancing Robot With PID Control," *Int. J. Robot. Res. Dev.*, 7(1), 1–6, 2017.
- [7] Bhagat S, Jogal R, Gharat J, Jamariya V, "Self balancing robot," *Int. J. Res. Appl. Sci. Eng. Technol.*, 6(3), 180–184, 2018.
- [8] Manikandan S, Reshma, Prasanna GV, Swetha C, Hariharan VS, "A new Design and Control of a Two-Wheel Self-Balancing Robot using the Arduino Microcontroller," *Int. J. Eng. Res. Electr. Electron. Eng.*, 4(2), 11–16, 2018.
- [9] Maung MM, Latt MM, Nwe CM, "DC Motor Angular Position Control using PID Controller with Friction Compensation," *Int. J. Sci. Res. Publ.*, 8(11), 149–155, 2018.
- [10] Gonzalez C, Alvarado I, La Peña DM, "Low cost two-wheels self-balancing robot for control education," *IFAC-PapersOnLine*, 50(1), 9174–9179, 2017.
- [11] Balcerzak M, "Optimal Control of an Inverted Pendulum Based on the New Method of Lyapunov Exponents Estimation," *Mech. Mech. Eng.*, 22(1), 25–32, 2018.
- [12] Kumar MA, Kanthalakshmi S, "H-infinity tracking control for an inverted pendulum," *J. Vib. Control*, 24(16), 3515–3524, 2018.
- [13] InvenSense Inc, "MPU-6000 and MPU-6050 Product Specification," California USA: InvenSense Inc. 2013.
- [14] Asali MO, Hadary F, Sanjaya BW, "Modeling, simulation, and optimal control for two-wheeled self-balancing robot," *Int. J. Electr. Comput. Eng.*, 7(4), 2008–2017, 2017.
- [15] Wu J, Zhang W, Wang S, "A Two-Wheeled Self-Balancing Robot with the Fuzzy PD Control Method," *Math. Probl. Eng.*, 2012, 1–13, 2012.