

Perbandingan Metode Kontrol P, PD, PI, dan PID pada Robot *Self-Balancing* Beroda Dua

Kisna Pertiwi⁽¹⁾, Andry Agustian⁽²⁾, Jonathan Silitonga⁽³⁾, Hendra Sinaga⁽⁴⁾, Franky Siahaan⁽⁵⁾, Hotlas Silverius S⁽⁶⁾, Claudio Ghustav D.⁽⁷⁾, Sabar⁽⁸⁾, Joni⁽⁹⁾

^{1,8,9}Program Studi Rekayasa Instrumentasi dan Automasi
^{2,3,4,5,6,7}Program Studi Teknik Elektro, Institut Teknologi Sumatera
Jl. Terusan Ryacudu, Desa Way Hui, Kec. Jatiagung, Lampung Selatan, Indonesia

Email: ¹kisna.pertiwi@ia.itera.ac.id, ²andry.120130157@student.itera.ac.id,
³jonathan.120130149@student.itera.ac.id, ⁴hendra.120130151@student.itera.ac.id,
⁵franky.120130054@student.itera.ac.id, ⁶hotlas.120130152@student.itera.ac.id,
⁷claudio.120130142@student.itera.ac.id, ⁸sabar@ia.itera.ac.id, ⁹joni@ia.itera.ac.id

Tersedia Online di

<http://www.jurnal.unublitar.ac.id/index.php/briliant>

Sejarah Artikel

Diterima 3 Februari 2024
Direvisi 10 Februari 2024
Disetujui 13 Februari 2024
Dipublikasikan 23 Februari 2024

Keywords:

IMU sensors; PID controller;
Two-Wheeled Self-Balancing
Robot

Kata Kunci:

Pengontrol PID; Robot *Self-Balancing* Beroda Dua; Sensor IMU

Corresponding Author:

Name:
Kisna Pertiwi
Email:
kisna.pertiwi@ia.itera.ac.id

Abstract: Self-balancing robots are unstable and will fall without external control, so a very good control method is needed so that the robot can automatically maintain a balanced position. This research aims to design and control a two-wheeled self-balancing robot so that it is able to maintain proper balance perpendicular to a flat surface. In this research, tests were carried out by comparing the P, PD, PI, and PID control methods in order to obtain the best control method that will be applied to a two-wheeled self-balancing robot. PID controllers are proven to be the best compared to other control methods. The performance of the PID controller used in the research was measured by the length of time the robot was able to stand stably when stationary, moving and in the presence of external disturbances. This system uses an Inertial Measurement Unit (IMU) sensor to reduce the centrifugal force exerted on the robot, detect tilt angle and angular velocity and uses Arduino as a control center and data processor. The self-balancing robot that has been designed can balance itself stably on the PID parameters used, namely $K_p=11-13$, $K_d=0.8-1$, and $K_i=100-120$. With these PID parameters the robot is able to maintain an upright and balanced position with or without external disturbances and is able to be stable for 600 seconds (10 minutes) or even longer as long as battery power is still available.

Abstrak: Robot *self-balancing* bersifat tidak stabil dan akan jatuh tanpa kontrol eksternal sehingga diperlukan metode kontrol yang sangat baik agar robot dapat secara otomatis mempertahankan posisi seimbang. Penelitian ini bertujuan untuk merancang dan mengontrol robot *self-balancing* beroda dua agar mampu menjaga keseimbangan dengan baik secara tegak lurus terhadap permukaan datar. Dalam penelitian ini dilakukan pengujian dengan membandingkan metode kontrol P, PD, PI, dan PID agar memperoleh metode kontrol terbaik yang akan diterapkan pada robot *self-balancing* beroda dua. Pengontrol PID terbukti yang terbaik dibandingkan metode kontrol lainnya. Performa pengontrol PID yang dilakukan dalam penelitian diukur dari lamanya waktu robot tersebut mampu berdiri stabil saat diam, bergerak dan adanya gangguan eksternal. Sistem ini menggunakan sensor *Inertial Measurement Unit* (IMU) untuk mengu-

rangi gaya sentrifugal yang diberikan pada robot, mendeteksi sudut kemiringan dan kecepatan sudut serta menggunakan arduino sebagai pusat kontrol serta proses data. *Robot self-balancing* yang telah dirancang dapat menyeimbangkan diri dengan stabil pada parameter PID yang digunakan, yaitu $K_p=11-13$, $K_d=0.8-1$, dan $K_i=100-120$. Dengan parameter PID tersebut robot mampu mempertahankan posisi tegak dan seimbang dengan atau tanpa gangguan eksternal serta mampu stabil selama 600 detik (10 menit) bahkan lebih lama sepanjang daya baterai masih tersedia.

PENDAHULUAN

Dewasa ini, perkembangan sains dan teknologi sangat pesat, khususnya bidang teknologi automasi robotik. Robot *self-balancing* merupakan konsep pendulum terbalik roda dua telah mendapatkan momentum dalam penelitian selama beberapa tahun terakhir (Bhowmik et al., 2017). Robot *self-balancing* merupakan pengembangan dari model pendulum terbalik (inverted pendulum) yang diletakkan di atas kereta beroda. Konsep robot ini telah digunakan sebagai alat transportasi yang bernama *segway* (Laksana et al., 2012). Robot *self-balancing* bersifat tidak stabil dan akan jatuh tanpa kontrol eksternal sehingga diperlukan metode kontrol yang sangat baik agar robot dapat secara otomatis mempertahankan posisinya secara tegak lurus terhadap permukaan datar (Suprpto et al., 2014). Robot *self-balancing* beroda dua merupakan robot yang mampu mempertahankan posisi seimbang secara otomatis berdasarkan perubahan dan pergeseran titik keseimbangannya (Kim et al., n.d.).

Dalam merancang robot *self-balancing* beroda dua, tantangan yang dihadapi adalah ketika robot mendapat gangguan eksternal, maka robot tetap mampu menjaga keseimbangan dan bergerak hanya dengan menggunakan dua roda saja. Metode kontrol yang dapat digunakan untuk menjawab tantangan tersebut adalah pengontrol *Proportional* (P), *Proportional Derivative* (PD), *Proportional Integral* (PI), dan *Proportional Integral Derivative* (PID). Analisis perbandingan pada pengontrol P, PD, PI, dan PID diharapkan dapat menentukan metode kontrol yang baik dan sesuai agar digunakan dalam mengontrol kecepatan dan arah putaran motor DC pada robot tersebut.

Beberapa penelitian terkait robot *self-balancing* telah dilakukan dengan menggunakan metode kontrol PID dengan menghitung *steady-state error*, *time settling*, dan *overshoot* (Taufiq & Aswardi, 2022). Penelitian lainnya juga pernah dilakukan dengan merancang robot *self-balancing* dengan memerhatikan diameter roda karena memengaruhi batas rentang sudut dan mengukur sudut kemiringan robot dengan menggunakan metode kontrol PID (Novandri et al., 2017). Selanjutnya, penelitian serupa juga dilakukan oleh Najmurokhman et al. (2019) mengenai rancang bangun robot *self-balancing* dan mengukur *error* sudut ketika robot miring ke kiri dan kanan terhadap posisi tegak. Namun, ketiga penelitian tersebut hanya fokus dalam menerapkan metode kontrol PID pada robot *self-balancing* tanpa membandingkan metode kontrol lain. Selain itu, Suprpto et al. (2014) melakukan penelitian tentang perbandingan metode kontrol PD, PI, dan PID pada robot *self-balancing* dan robot tersebut dapat menyeimbangkan diri pada posisi tegak selama 43 detik. Berdasarkan penelitian tersebut, maka penelitian ini menambahkan perbandingan menjadi empat metode kontrol, yaitu P, PD, PI, dan PID dan merancang robot *self-balancing* agar mampu menjaga keseimbangan dengan waktu yang lebih lama dan mampu bertahan meskipun terdapat gangguan eksternal.

Robot *self-balancing* biasanya menggunakan jenis sensor seperti akselerometer dan giroskop untuk mengukur perubahan percepatan linier dan kecepatan sudut robot. Agar robot dapat berdiri tegak dapat menggunakan sensor *Inertial Measurement Unit* (IMU). Sensor ini digunakan untuk mengurangi gaya sentrifugal yang diberikan pada robot *self-balancing* (Alarfaj & Kantor, 2010).

Penelitian ini bertujuan untuk merancang dan mengontrol robot *self-balancing* beroda dua agar mampu menjaga keseimbangan dengan baik secara tegak lurus terhadap permukaan datar. Pengujian performa pada pengontrol P, PD, PI, dan PID yang dilakukan dalam penelitian ini

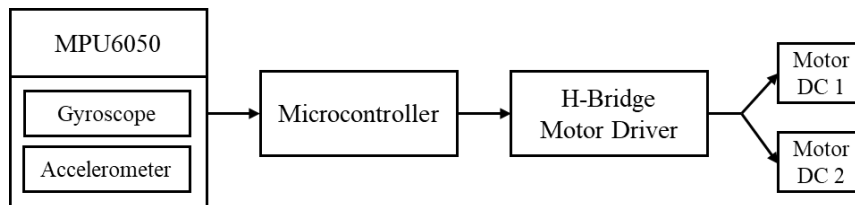
adalah mengukur waktu ketika robot diam maupun bergerak dan ketahanan terhadap gangguan eksternal sehingga robot tetap stabil.

METODE

Proses perancangan keseluruhan sistem robot *self-balancing* beroda dua memiliki tiga tahapan, yaitu perancangan perangkat keras, perancangan sistem kontrol, dan perancangan perangkat lunak.

Perancangan Perangkat Keras

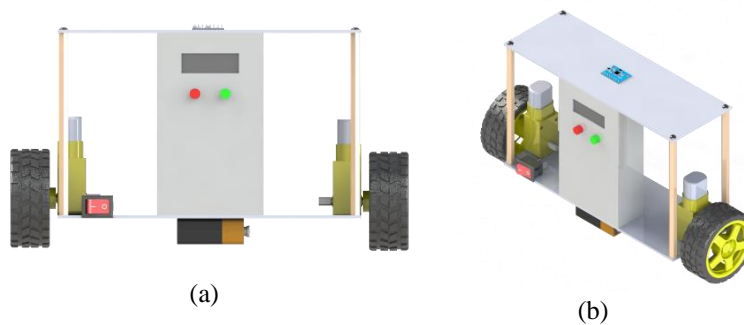
Perancangan perangkat keras meliputi perancangan sistem mekanik dan elektronik. Perancangan sistem pada robot *self-balancing* beroda dua dapat dilihat pada Gambar 1.



Gambar 1. Diagram sistem perancangan robot *self-balancing* beroda dua

1. Perancangan sistem mekanik

Perancangan sistem mekanik diperlukan untuk menentukan ukuran-ukuran bahan dan komponen yang dibutuhkan untuk mengurangi resiko kesalahan dalam pemasangan komponen pada robot yang akan digunakan. Pembuatan mekanik menggunakan bahan PCB karena bahan ini mudah diperoleh, ringan, dan murah. Bodi robot *self-balancing* terdiri atas tiga bagian utama, yaitu alas bawah, tiang penghubung, dan alas atas. Ukuran alas atas dan bawah adalah 70 mm x 200 mm dan tinggi tiang penghubung 120 mm. Selain itu, juga terdapat dua buah roda dengan diameter 68 mm (Gambar 2).



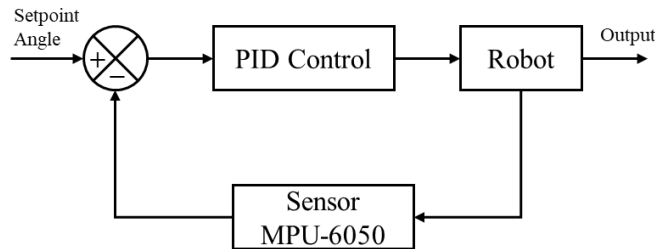
Gambar 2. Desain 3D bodi robot *self-balancing* beroda dua (a) tampak depan dan (b) tampak samping

2. Perancangan sistem elektronik

Perancangan sistem elektronik diperlukan untuk menentukan komponen-komponen elektronik yang akan digunakan dalam membuat robot *self-balancing* beroda dua seperti sensor, pengontrol, dan aktuator. Penelitian ini menggunakan mikrokontroler Arduino UNO R3, sensor MPU-6050, motor driver L298N, 2 baterai lithium ion voltase 6 Volt dan 7.4 Volt, 2 motor DC, dan jumper. Sensor IMU MPU-6050 merupakan sensor gerak yang terintegrasi yang terdiri dari tiga sumbu giroskop, tiga sumbu akselerometer, dan prosesor gerak digital (*MPU-6000 and MPU-6050 Product Specification Revision 3.3*, n.d.). Sensor ini memanfaatkan fitur I2C mikrokontroler (Destiarini & Kumara, 2019). Sensor MPU-6050 mendeteksi besaran berupa sudut kemiringan dan kecepatan sudut. Data kecepatan sudut diperoleh dari bagian giroskop dan data sudut kemiringan diperoleh dari bagian akselerometer (Najmurokhman et al., 2019). Selanjutnya, data-data tersebut diproses oleh mikrokontroler Arduino Uno yang telah terpasang algoritma kontrol

Kelemahannya adalah penguatan *Derivative* yang terlalu tinggi juga akan memperkuat derau, yang dapat menyebabkan sistem menjadi tidak stabil.

Kontrol PID menentukan nilai kecepatan dan arah putaran motor DC yang menjadi sumber penggerak, berdasarkan sudut kemiringan badan robot terhadap bidang datar. Sehingga, robot *self-balancing* ini dapat mempertahankan posisi secara tegak lurus dengan seimbang terhadap permukaan bidang datar. Pengontrol PID ini hanya memerlukan masukan data-data dari proses yang diukur sehingga perlu dilakukan penyesuaian parameter untuk menghasilkan respon sistem yang diinginkan (Gambar 4).



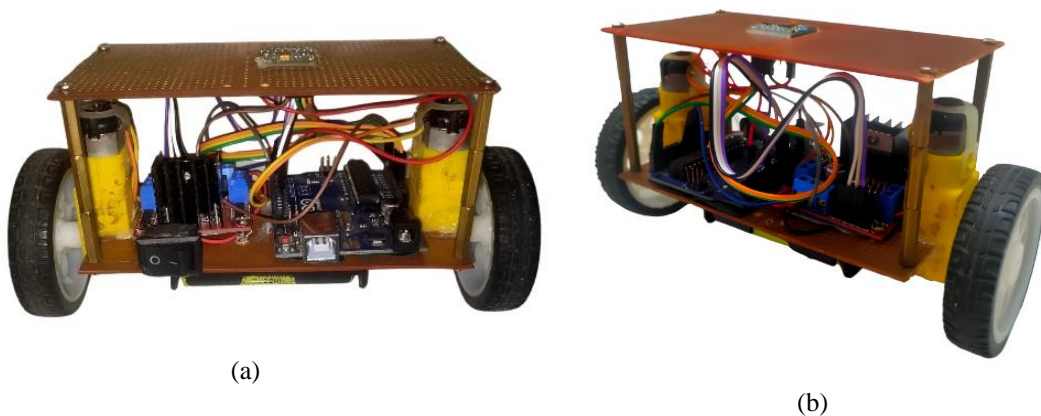
Gambar 4. Diagram blok robot *self-balancing* beroda dua

Perancangan Perangkat Lunak

Perancangan perangkat lunak merupakan pemrograman menggunakan perangkat lunak Arduino IDE. Tahapan dalam membuat pemrograman, yaitu menentukan algoritma yang akan digunakan untuk menjaga keseimbangan robot, seperti algoritma PID dengan memasukkan nilai K_p , K_i , dan K_d yang sesuai sehingga memperoleh kondisi robot yang stabil, membuat program Arduino untuk membaca data dari sensor kemiringan, melakukan perhitungan dan kontrol, dan mengendalikan aktuator, implementasikan logika kontrol yang sesuai untuk mempertahankan keseimbangan robot saat bergerak, kemudian uji program perangkat lunak dengan menyambungkan Arduino ke komputer dan mengunggah program ke dalam mikrokontroler.

HASIL DAN PEMBAHASAN

Hasil perancangan robot *self-balancing* beroda dua berupa produk setelah dirakit dapat dilihat pada Gambar 5. Hasil pengujian pada parameter kontrol P, PD, PI, dan PID digunakan agar robot dapat berdiri tegak pada permukaan datar dengan seimbang, maka perlu dilakukan penalaan untuk mendapatkan parameter terbaik. Tabel 1-4 berikut menunjukkan nilai-nilai konstanta K_p , K_d , dan K_i digunakan pada pengontrol P, PD, PI, dan PID beserta respon yang terjadi pada robot.



Gambar 5. Produk robot *self-balancing* beroda dua (a) tampak depan dan (b) tampak samping

Pengontrol *Proportional*

Pengujian pertama dilakukan dengan memvariasikan nilai K_p dengan mengatur nilai $K_d = K_i = 0$ guna memperoleh nilai K_p yang paling tepat. Parameter K_p berfungsi untuk mengontrol sejauh mana respon pengontrol terhadap selisih antara *set point* dan keluaran aktual sistem. Parameter K_p menentukan seberapa besar respon pengontrol terhadap kesalahan proporsional saat ini. Semakin tepat nilai K_p sesuai dengan sistem, maka akan semakin cepat respon pengontrol terhadap kesalahan dan semakin besar pula perubahan yang diberikan pada keluaran.

Berdasarkan Tabel 1, setelah dilakukan pengujian maka diperoleh nilai K_p yang tepat sehingga robot memiliki respon yang baik dan tidak terjadi osilasi pada rentang nilai $K_p = 11 - 13$. Robot yang memiliki respon baik adalah seberapa cepat respon motor DC dapat berganti arah roda ketika posisi robotnya berubah. Selain itu, pada kondisi ini robot telah mampu berdiri seimbang tetapi hanya sebentar saja kemudian robot terjatuh. Jika nilai $K_p < 11$, maka respon robot akan lambat dan tidak bisa seimbang sedangkan jika nilai $K_p > 13$, maka respon robot terjadi osilasi yang terlalu kuat dan robot tidak bisa seimbang.

Tabel 1. Hasil pengujian terhadap variasi nilai K_p

K_p	Keterangan
5	Respon robot lambat, tidak bisa seimbang
7	Respon robot agak lambat, tidak bisa seimbang
9	Respon robot agak lambat, tidak terjadi osilasi
11	Respon robot baik, tidak terjadi osilasi
13	Respon robot baik, terjadi sedikit osilasi
15	Respon robot baik, terjadi sedikit osilasi
17	Respon robot baik, terjadi osilasi
19	Osilasi sangat kuat, robot tidak bisa seimbang
21	Osilasi sangat kuat, robot tidak bisa seimbang
23	Osilasi sangat kuat, robot tidak bisa seimbang

Pengontrol *Proportional-Derivative (PD)*

Pengujian kedua dilakukan pada pengontrol PD dengan memvariasikan nilai K_d dan memilih nilai $K_p = 11$ yang telah diperoleh nilai yang paling tepat dari pengujian sebelumnya. Parameter K_d digunakan untuk mengontrol tingkat perubahan kesalahan pada sistem. Parameter K_d menghitung tingkat perubahan kesalahan saat ini dan memberikan respon pengontrol yang proporsional terhadap perubahan tersebut. Fungsi K_d untuk mengurangi *overshoot* dan memperbaiki waktu *settling* sistem. Nilai K_d yang sesuai dengan sistem akan memberikan respon pengontrol yang lebih cepat terhadap perubahan kesalahan.

Berdasarkan Tabel 2, nilai $K_d < 0.8$ diperoleh respon robot sangat lambat dan tidak seimbang sedangkan nilai $K_d > 1$ diperoleh respon robot cepat, terjadi osilasi terlalu kuat dan robot tidak bisa seimbang. Selanjutnya, pada nilai $K_d = 0.8 - 1$, diperoleh nilai yang tepat dengan respon robot cepat dan mampu menyeimbangkan diri dengan baik hanya selama 2 detik kemudian robot tersebut terjatuh.

Tabel 2. Hasil pengujian terhadap variasi nilai K_d dan $K_p = 11$

K_d	Keterangan
0,2	Respon robot sangat lambat, tidak bisa seimbang
0,4	Respon robot lambat, tidak bisa seimbang
0,6	Respon robot agak lambat, robot bisa seimbang
0,8	Respon robot cepat, dapat seimbang dengan baik
1	Respon robot cepat, terjadi sedikit osilasi
1,2	Respon robot cepat, terjadi osilasi
1,4	Osilasi sangat kuat, robot tidak bisa seimbang
1,6	Osilasi sangat kuat, robot tidak bisa seimbang
1,8	Osilasi sangat kuat, robot tidak bisa seimbang
2	Osilasi sangat kuat, robot tidak bisa seimbang

Pengontrol *Proportional-Integral* (PI)

Pengujian ketiga dilakukan pada pengontrol PI dengan memvariasikan nilai K_i dan memilih nilai $K_p = 11$. Parameter K_i digunakan untuk mengontrol kesalahan integral pada sistem. Kesalahan integral muncul ketika ada selisih antara *set point* dan keluaran sistem yang terus-menerus terjadi seiring waktu. Parameter K_i menghitung akumulasi kesalahan sepanjang waktu dan memberikan perubahan keluaran untuk mengatasi kesalahan tersebut. Nilai K_i yang sesuai dengan sistem akan memberikan respon pengontrol yang lebih besar terhadap kesalahan integral.

Berdasarkan Tabel 3, pada nilai $K_i = 35 - 40$ merupakan nilai yang tepat karena respon robot mampu seimbang selama 600 detik. Selanjutnya, nilai $K_i < 35$ diperoleh respon robot tidak mampu seimbang, maju pada satu arah sedangkan nilai $K_i > 40$ diperoleh respon robot mampu seimbang tetapi hanya sebentar kemudian terjatuh, jika nilai K_i terus diperbesar, maka terjadi osilasi dan robot tidak mampu seimbang. Berbeda dengan pengontrol PD, pada pengontrol PI memiliki respon robot mampu seimbang dalam waktu yang lama sepanjang daya baterai masih tersedia. Namun, respon robot belum optimal, yaitu pergerakan robot masih belum halus.

Tabel 3. Hasil pengujian terhadap variasi nilai K_i dan $K_p = 11$

K_i	Keseimbangan (detik)	Keterangan
5	1.54	Robot tidak mampu seimbang
10	2.02	Robot tidak mampu seimbang
15	3.69	Robot tidak bisa seimbang, maju ke satu arah
20	6.66	Robot seimbang sebentar dan terjatuh, maju ke satu arah
25	8.9	Robot seimbang sebentar dan terjatuh, maju ke satu arah
30	13.42	Robot seimbang sebentar dan terjatuh, maju ke satu arah
35	600	Robot mampu seimbang tetapi pergerakannya tidak halus
40	600	Robot mampu seimbang tetapi pergerakannya tidak halus
45	21.76	Robot mampu seimbang sebentar, osilasi mulai terlihat
50	7.09	Robot seimbang sebentar dan terjatuh, maju ke satu arah
55	4.96	Robot tidak mampu seimbang
60	2.67	Robot tidak mampu seimbang

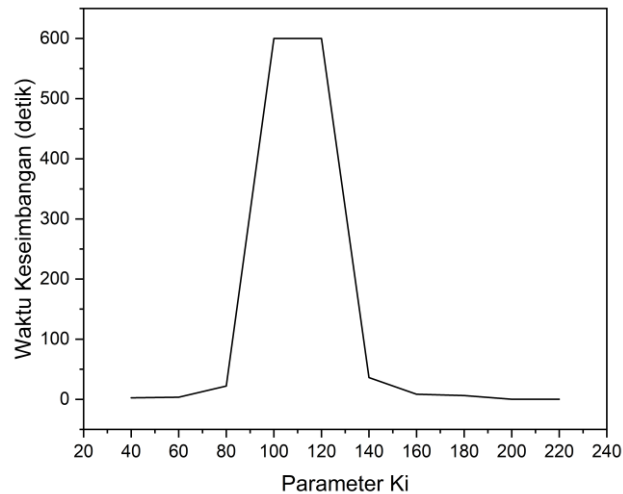
Pengontrol *Proportional Integral Derivative* (PID)

Pengujian keempat dilakukan pada pengontrol PID dengan memvariasikan nilai K_i dan memilih nilai yang paling stabil yakni $K_p = 11$ dan $K_d = 0.8$ yang diperoleh dari pengujian sebelumnya. Berdasarkan Tabel 4 dan Gambar 6, respon robot dapat stabil dan seimbang pada rentang nilai $K_i = 100 - 120$. Artinya, respon robot tersebut dapat berlangsung stabil dan mampu menyeimbangkan diri dalam kondisi diam maupun bergerak selama 600 detik bahkan dapat bertahan lebih lama sepanjang daya baterai masih tersedia. Selain itu, robot juga mampu menyeimbangkan diri meskipun terdapat gangguan eksternal seperti menumbuk sesuatu. Jika nilai $K_i = 80$, maka respon robot agak lambat sedangkan jika nilai $K_i = 140$, maka respon robot sangat cepat tetapi kedua nilai K_i tersebut kondisi robot tetap mampu menyeimbangkan diri. Selanjutnya, jika nilai $K_i < 80$, maka respon robot berjalan ke satu arah dan tidak bisa seimbang sedangkan jika nilai $K_i > 120$, maka respon robot sangat cepat dan seimbang tetapi sebentar saja kemudian robot terjatuh.

Tabel 4. Hasil pengujian terhadap variasi nilai K_i , $K_p = 11$, dan $K_d = 0.8$

K_i	Keseimbangan (detik)	Keterangan
40	2.51	Robot berjalan ke satu arah, tidak bisa seimbang
60	3.54	Robot berjalan ke satu arah, tidak bisa seimbang
80	22.16	Respon robot agak lambat, robot bisa seimbang
100	600	Respon robot cepat, dapat seimbang dengan baik
120	600	Respon robot cepat, dapat seimbang dengan baik
140	36.2	Respon robot sangat cepat, robot bisa seimbang

160	8.44	Respon robot sangat cepat, robot bisa seimbang sebentar lalu terjatuh
180	6.21	Respon robot sangat cepat, robot bisa seimbang sebentar lalu terjatuh
200	0.2	Respon robot sangat cepat, robot tidak bisa seimbang
220	0.2	Respon robot sangat cepat, robot tidak bisa seimbang



Gambar 6. Grafik hasil pengujian pada pengontrol PID

KESIMPULAN

Prinsip kerja dari sistem ini melibatkan penggunaan sensor akselerometer dan giroskop untuk mendeteksi kemiringan dan perubahan orientasi robot. Mikrokontroler berperan sebagai otak robot, membaca data dari sensor, dan mengendalikan motor untuk menjaga keseimbangan. Algoritma kontrol, seperti algoritma PID, digunakan untuk menghitung sinyal kendali yang diperlukan untuk menjaga keseimbangan. Algoritma ini membandingkan posisi aktual robot dengan posisi yang diinginkan dan mengatur kecepatan motor agar robot tetap seimbang. Analisis nilai pada pengontrol P, PD, PI, dan PID telah dilakukan guna mendapatkan metode kontrol yang baik dan tepat untuk digunakan dalam mengontrol kecepatan dan arah putaran motor DC pada robot *self-balancing* beroda dua.

Berdasarkan hasil pengujian pada pengontrol P, PD, PI, dan PID dapat disimpulkan bahwa pengontrol PID merupakan pengontrol yang paling baik dan tepat untuk diaplikasikan pada robot *self-balancing* beroda dua. Respon optimal robot diperoleh dengan parameter keseimbangan yang digunakan, yaitu $K_p = 11 - 13$, $K_d = 0.8 - 1$, dan $K_i = 100 - 120$. Parameter keseimbangan tersebut mampu mempertahankan robot dalam posisi stabil dan seimbang dengan atau tanpa gangguan eksternal. Keadaan ini mampu dipertahankan selama 600 detik (10 menit) bahkan lebih lama sepanjang daya baterai masih tersedia baik dalam kondisi diam maupun bergerak. Jika nilai K_i lebih kecil atau lebih besar dari rentang tersebut, maka robot akan sulit untuk menjaga keseimbangan.

Saran untuk penelitian selanjutnya agar dapat menentukan sudut kemiringan pada robot. Pengukuran sudut kemiringan menggunakan giroskop dapat menyebabkan pergeseran atau penyimpangan pada titik awal sudut kemiringan, sehingga dapat menggunakan filter komplementer untuk mendapatkan pembacaan sudut kemiringan yang ideal dan akurat.

DAFTAR RUJUKAN

- Alarfaj, M., & Kantor, G. (2010). Centrifugal Force Compensation of a Two-Wheeled Balancing Robot. *11th International Conference on Control, Automation, Robotics, and Vision*, 2333–2338.

- Bhowmik, S., Science, N., Bhawmick, B. K., & Das, N. K. (2017). *DESIGN AND IMPLEMENTATION OF A SELF-BALANCING ROBOT ICMERE2017-PI-294 DESIGN AND IMPLEMENTATION OF A SELF-BALANCING ROBOT*. February 2018.
- Destiarini, P. W. K., & Kumara, P. W. (2019). Robot Line Follower Berbasis Mikrokontroler Arduino Uno ATMEGA328. *Jurnal Informatika*, 5(1), 19–25.
- Johnson, M. A., & Moradi, M. H. (2005). *PID Control: New Identification and Design Methods*. Springer.
- Kim, S., Seo, J., & Kwon, S. (n.d.). Development of a Two-Wheeled Mobile Tilting & Balancing (MTB) Robot. *11th International Conference on Control, Automation and Systems*, 1–6.
- Laksana, A., Setiawan, I., & Sumardi. (2012). *Balancing Robot Beroda Dua Menggunakan Metode Kendali Proporsional Integral*. Univeristas Diponegoro.
- MPU-6000 and MPU-6050 Product Specification Revision 3.3. (n.d.). <http://www.invensense.com/mems/gyro/documents/PS-MPU-6000A.pdf>
- Najmurokhman, A., Irfansyah, I., & Daelami, A. (2019). Rancang Bangun Auto Balancing Robot Menggunakan Metode Kendali PID Design and Implementation of A Self-Balancing Robot Based on PID Control Method. *Telka*, 5(1), 15–23.
- Novandri, A., Roslidar, & Rahman, A. (2017). Rancang Bangun Robot Self Balancing Berbasis Mikrokontroler Atmega328P Dengan Kendali Pid. *Kitektro, Jurnal Online Teknik Elektro*, 2(2), 15–23.
- Suprpto, B. Y., Amri, D., & Dwijayanti, S. (2014). Comparison of control methods PD, PI, and PID on two wheeled self balancing robot. *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 1(August), 67–71. <https://doi.org/10.11591/eecsi.1.348>
- Taufiq, T., & Aswardi, A. (2022). Self Balancing Robot Menggunakan Metode PID Berbasis Arduino. *JTEIN: Jurnal Teknik Elektro Indonesia*, 3(1), 15–24. <https://doi.org/10.24036/jtein.v3i1.192>
- Yong, C., & Kwong, C. F. (n.d.). Wireless Controlled Two Wheel Balancing Robot. *International Journal of Network and Mobile Technologies*, 2(2), 88–109.